28 PAGES OF EXPERT TUTORIALS

web designer™

FREE 195 MINS OF CSS VIDEO

HTML5 CSS3 jQuery WordPress

www.webdesignermag.co.uk

14 PAGE GUIDE
ADVANCED ANGULAR
INDUSTRY TIPS & TUTORIALS

BUILD WITH REACT NATIVE
A LOOK INSIDE FACEBOOK'S NEW FRAMEWORK

POWER UP CSS PREPROCESSORS
·SASS ·LESS ·STYLUS

EXPERT ADVICE TO MASTER PRO LEVEL TOOLS

WHAT IS WEB OF THINGS?
WHY IOT WILL ADOPT WEB STANDARDS

DYNAMIC CSS MODALS
ANIMATE POP-UP WINDOWS

67 DESIGNER RESOURCES
WEB TEMPLATES, FONTS, TEXTURES & ACTIONS

INTERACTIVE GRAPHICS
HTML 5
UNLEASH THE P5.JS LIBRARY FOR DRAWING IN THE BROWSER

ip IMAGINE PUBLISHING
DigitalEdition
GreatDigitalMags.com
ISSUE 236

+
=

27" iMAC

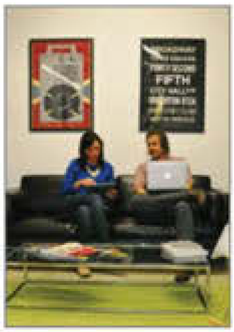
Welcome to the issue

THE WEB DESIGNER MISSION

“To be the most accessible and inspiring voice for the industry, offering cutting-edge features and techniques vital to building future-proof online content”

Steve Jenkins

Highlight



“We like to incorporate a discovery phase to define strategy”

Web Designer gets chatty with LA-based Bad Assembly and discovers thier passion to communicate. **Page 32**

Faster, more efficient CSS. What’s not to like?

Css is undoubtedly the king of styling, but it wears its crown unopposed. Plain CSS is not the simplest or smartest tool in a web designer’s toolkit, but it can be given extra IQ points and plenty of polish with the very able assistance of preprecessors.

Employing the services of the big three Sass, LESS or Stylus will ensure that you can be more creative, will find fewer bugs and have to indulge in less typing. What’s not to like?

Our eight-page feature (page 38) reveals the reasons why you need to be using CSS preprocessors before delving into and comparing the best tools for the job and how to use them to produce faster, more efficient code.

Another powerful tool that should sit in every web designer’s toolkit is AngularJS. This issue we dedicate 14 pages to the superheroic framework in the form of 21 tips, three tutorials, expert comments and a collection of essential resources. It all starts on page 72.

Getting a bit more esoteric we take a look at the current state of the Internet of Things and how the Web of Things is going to introduce web standards for a more flexible future for connected devices.

Plus, we have our staple selection of CSS and HTML tutorials to keep you busy. As always enjoy the issue and see you next time out.

“Preprocessors save you so much time when going from an idea to an implementation, by reducing the amount of code you have to write”

Follow us on Twitter for all the news & conversation @WebDesignerMag

Visit our blog for opinion, freebies & more www.webdesignermag.co.uk

FREE – exclusive with this issue

67 Designer resources

Video Tuition	- CSS3: 195 Minutes of expert video guides from CartoonSmart (www.cartoonsmart.com)
Assets	- 40 Brick Mosaics Actions from SparkleStock
Templates	- 3 Template packs from w3 layouts; Creative Pix photography template, Trail responsive and Umbrella mobile template
Fonts	- Mesmerize and Manbow from Typodermic

FileSilo

www.filesilo.co.uk/webdesigner

This issue's panel of experts

Welcome to that bit of the mag where we learn more about the featured writers and contributors...

"Preprocessors add some of the smarts that plain CSS never had. That means more creativity, fewer bugs and less typing"

Richard Wentk

Richard is a long-time producer of code and apps, writing numerous books on the subject. This issue takes an in-depth look at CSS and reveals why you need preprocessors and how to use them. He compares the top three tools to make sure you get the job done right. **Page 38**

Mark Shufflebottom

Mark Shufflebottom is professor of Interaction Design at Sheridan College, Canada. This month Mark is creating interactive HTML5 Canvas visuals with the p5.js library, a JavaScript port of the popular Processing library. **Page 46**

Tam Hanna

Tam is a veteran in the world of coding and has worked with a host of coding languages. This issue he delves into the world of the Internet of Things, reveals what's happening and how the Web of Things will add web standards. **Page 64**

Todd Motto

Todd is director of front-end engineering at startup, Mozio, a AngularJS developer and open source evangelist. This issue he unleashes his expansive knowledge and provides a host of expert AngularJS tips and tutorials. **Page 72**

Leon Brown

Leon is a freelance developer and trainer who assists web developers in creating efficient and secure code for their projects. This issue he shows us how to create a pop-up modal window using CSS. **Page 52**

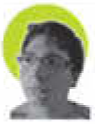
Sean Tracey

Sean is a creative technologist living and working in the south coast. He loves playing with the latest JS APIs he can get his hands on. This issue he gives the lowdown on how to start building with the React Native framework. **Page 58**

Mark Billen

Mark is a freelance technology journalist with a background in web design and development. This issue he tackles the Lightbox pages and reveals how to create a host of quick-fire techniques you need to know. **Page 14**

Tim Stone



Tim is a lead front-end developer at Redweb and has a first-class degree in Interactive Media Production. He loves coding and this issue demonstrates how to use Angular with Meteor to create a flexible, reactive UI. **Page 82**

David Howell

David is an experienced writer, author and journalist who runs his own publishing company Nexus Publishing. This issue he takes his interview skills to the office of the LA-based digital creatives Bad Assembly. **Page 32**

Got web skills?

We're always looking for the hottest web-design talent. Email **webdesigner@imagine-publishing.co.uk** with examples of your creative work

web designer

Imagine Publishing Ltd
Richmond House, 33 Richmond Hill
Bournemouth, Dorset, BH2 6EZ
☎ +44 (0)1202 586200
Web: www.imagine-publishing.co.uk
www.webdesignermag.co.uk
www.greatdigitalmags.com

Magazine team

Editor **Steve Jenkins**
steve.jenkins@imagine-publishing.co.uk
☎ 01202 586233
Senior Designer **Benjamin Stanley**
Production Editor **Carrie Mok**
Photographer **James Sheppard**
Senior Art Editor **Will Shum**
Editor In Chief **Dan Hutchinson**
Publishing Director **Aaron Asadi**
Head of Design **Ross Andrews**

Contributors
Stefan Mischook, Bart De Pelsmaeker, Mark Billen, David Howell, Richard Wentk, Mark Shufflebottom, Jayson Winters, Leon Brown, Sean Tracey, Tam Hanna, Brad Fry, Todd Motto, Tim Stone

Advertising
Digital or printed media packs are available on request.

Head of Sales Hang Deretz
☎ 01202 586442
hang.deretz@imagine-publishing.co.uk

Advertising Manager Alex Carnegie
☎ 01202 586430

Account Manager Luke Biddiscombe
☎ 01202 586431

FileSilo.co.uk
Assets and resource files for this magazine can be found on this website.
Register now to unlock thousands of useful files.
Support filesilohelp@imagine-publishing.co.uk

International
Web Designer is available for licensing. Contact the International department to discuss opportunities.

Head of International Licensing Cathy Blackman
☎ +44 (0) 1202 586401
licensing@imagine-publishing.co.uk

Subscriptions
subscriptions@imagine-publishing.co.uk
To order a subscription to **Web Designer:**
☎ 0844 848 8413 ☎ +44 1795 592 878
Email: webdesigner@servicehelpline.co.uk
13-issue subscription (UK) – £62.30
13-issue subscription (Europe) – £70
13-issue subscription (ROW) – £80

Circulation
Head of Circulation Darren Pearce
☎ 01202 586200

Production
Production Director Jane Hawkins
☎ 01202 586200

Founder
Group Managing Director Damian Butt

Finance
Finance Director Marco Peroni

Printing & Distribution
Printed by Southernprint Ltd, 17-21 Factory Road
Upton Industrial Estate, Poole, Dorset, BH16 5SN

Distributed in the UK, Eire & the Rest of the World by Marketforce, Blue Fin Building, 110 Southwark Street, London, SE1 0SU, ☎ 0203 148 3300, www.marketforce.co.uk

Distributed in Australia by Network Services (a division of Bauer Media Group), Level 21 Civic Tower, 66-68 Goulburn Street, Sydney, New South Wales 2000, Australia, ☎ +61 2 8667 5288

Disclaimer
The publisher cannot accept responsibility for any unsolicited material lost or damaged in the post. All text and layout is the copyright of Imagine Publishing Ltd. Nothing in this magazine may be reproduced in whole or part without the written permission of the publisher. All copyrights are recognised and used specifically for the purpose of criticism and review. Although the magazine has endeavoured to ensure all information is correct at time of print, prices and availability may change. This magazine is fully independent and not affiliated in any way with the companies mentioned herein.

If you submit material to Imagine Publishing via post, email, social network or any other means, you automatically grant Imagine Publishing an irrevocable, perpetual, royalty-free license to use the materials across its entire portfolio, in print, online and digital, and to deliver the images to existing and future clients, including but not limited to international licensees for reproduction in international, licensed editions of Imagine products. Any material you submit is sent at your risk and, although every care is taken, neither Imagine Publishing nor its employees, agents or subcontractors shall be liable for the loss or damage.

ISSN 1745-3534

IMAGINE PUBLISHING

recycle
When you have finished with this magazine please recycle it.

WORKS WITH MySQL
php
TYPO3
1&1
python
Windows
WORDPRESS
Joomla!
1&1

inside issue 236

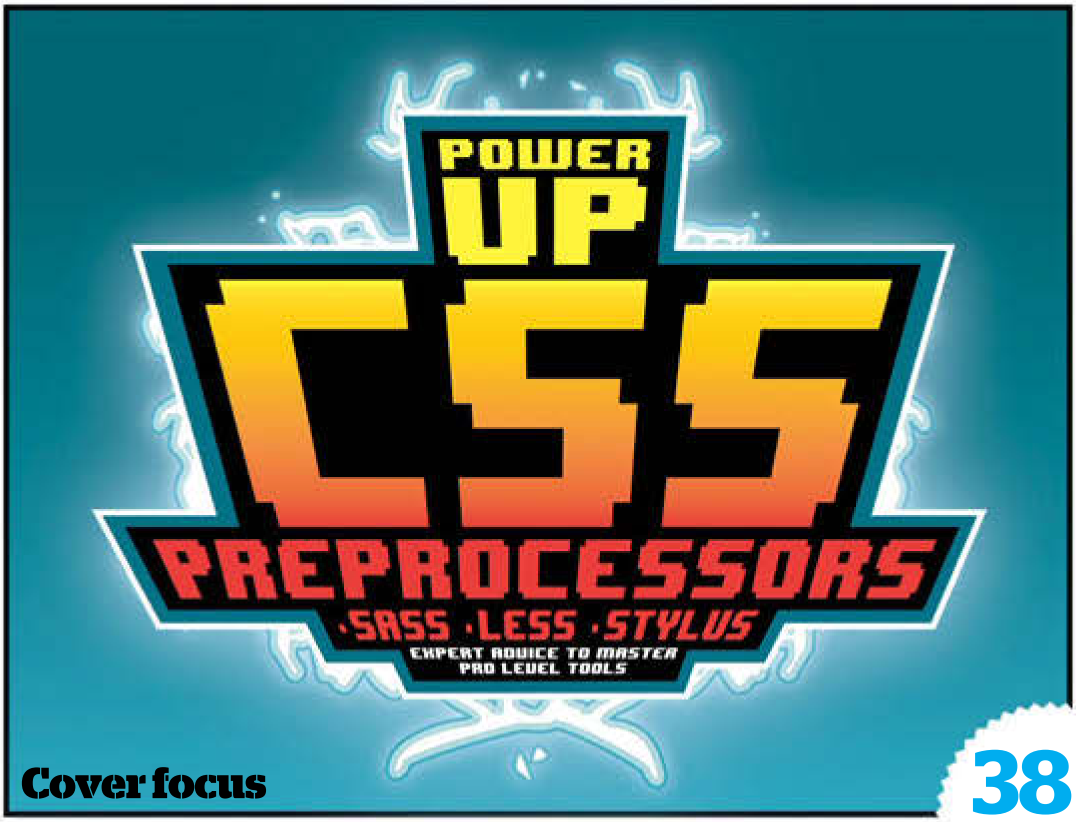
contents

Cutting-edge features, techniques and inspiration for web creatives

Chat with the team and other readers and discuss the latest tech, trends and techniques. Here's how to stay in touch...

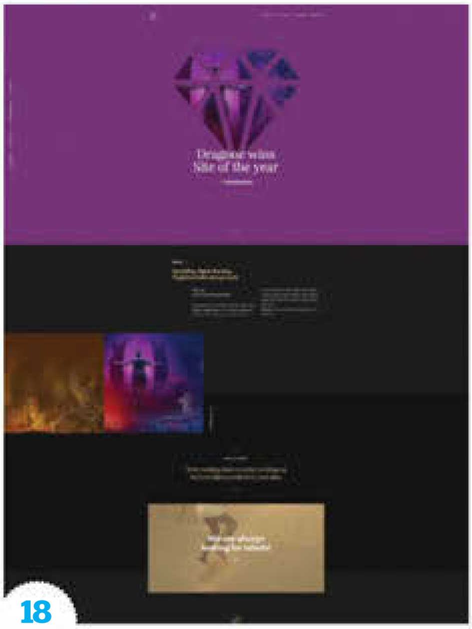
✉ webdesigner@imagine-publishing.co.uk 🐦 @WebDesignerMag 👆 www.webdesignermag.co.uk

Quick look...



Cover focus **38**

32 **ProFile:** Bad Assembly
A diverse portfolio reveals the talents of this LA agency



18 **Lightbox:** Dogstudio
A digital design that you will remember

FileSilo

A comprehensive collection of free designer resources!

- 195 Minutes of CSS3 video guides
- 3 Web templates from w3layouts
- 40 Brick Mosaics Actions
- Mesmerize and Manbow fonts

94

WEB OF THINGS

<header>

The tools and trends to inspire your web projects

> Don't fall in love with a technology, be ready to dump it when it's time
>
> **Stefan Mischook**

<tutorials>

Web gurus take you step-by-step through professional techniques

Never miss an issue **Subscribe** Turn to page 62 now USA readers turn to page 87 for an exclusive offer

web workshop

web developer

CSS4: a natural progression

It's only a matter of time before the next level of CSS arrives. Will it be called CSS4? Web Designer looks at what's happening now

Evolution is a wonderful thing and it's the perfect adjective to describe web technologies. HTML5, CSS3, ES6, jQuery 2.0 – all technologies evolve and the simplest way to present them to the masses is with the addition of a number. HTML5 will undoubtedly turn into HTML6, jQuery 2.0 will eventually become jQuery 3.0 and CSS3 will become CSS4.

The web's big boys HTML and CSS simply used to be recognised as just that, HTML and CSS. However, with the fifth incarnation and third incarnation respectively, numbers jumped into the equation. The addition of a number helps create a buzzword, but at the same time it indicates that a technology has evolved to a level where it finally needs recognition and it's easy enough for us mortals to understand. Admittedly support may be scarce for the next generation of CSS, but support will eventually catch up, it always does. That's when the magic starts.

While it only seems like yesterday that CSS3 breezed into our lives, it has now been around as a viable technology for over three years and inevitably the next version of the technology is being worked on.

The W3C, the international community that works to develop web standards, recently released an Editor's Draft for Selectors Level 4, detailing the latest updates and additions to the specification. An Editor's Draft is not an officially published paper, it is a work in progress as stated on their site "It is provided for discussion only and may change at any moment. Its publication here does not imply endorsement of its contents by W3C. Don't cite this document other than as work in progress".

However, an Editor's Draft gives a formative insight into what may well be appearing in the specification. Go to **dev.w3.org/csswg/selectors-4** for the full draft.

Another draft that will affect future CSS is CSS Cascading and Inheritance Level 4. This is a First Public Working Draft, which effectively has the same status as an Editor's Draft, but is further along in the process. To get a full look at the status of the draft head over to **bit.ly/1dZ7fUm**.

The drafts give us the latest status of a draft, but exactly who is putting together the specification? The CSS Working Group are the brains behind the spec. You can get a quick overview of what they've been up to at **w3.org/Style/CSS/current-work** and see who is working on the team at **w3.org/Style/CSS/members**.

The prospect of the next iteration of CSS, or CSS4, is an exciting one. If the specification introduces as many ground-breaking features as CSS3 then the future will undoubtedly be bright. What are we at Web Designer hoping for? Advanced animation techniques directly in the browser – now that would be exciting.

> "Admittedly support may be scarce for the next generation of CSS, but support will eventually catch up"

Essential CSS4 resources

Check out this collection to get the lowdown on the future of CSS

CSS Working Group Wiki

CSS Working Group Wiki

wiki.csswg.org

A wiki which shares CSS resolutions and issues with the public.

CSS4 Rocks

CSS - Level 4 Selectors

CSS4 Rocks

css4.rocks

This site looks at the latest modules to be added to the CSS specification.

CSS4-Selectors

Hey there! If you are interested in upcoming CSS4 levels you're completely right here!

CSS4-Selectors

css4-selectors.com

Find out about the latest spec and what's being said on social media.

Get Web Designer digital editions

Want to build HTML5 games? Check out Web Designer's latest digital special

Web Designer's latest digital additions to our burgeoning collection is **HTML5 Arcade** and **10 Amazing Websites and How to Build Them Vol 2**. Get an insight into the different games engines out there, plus discover how to start creating your very own.

Discover great-looking sites and the techniques used to build them.

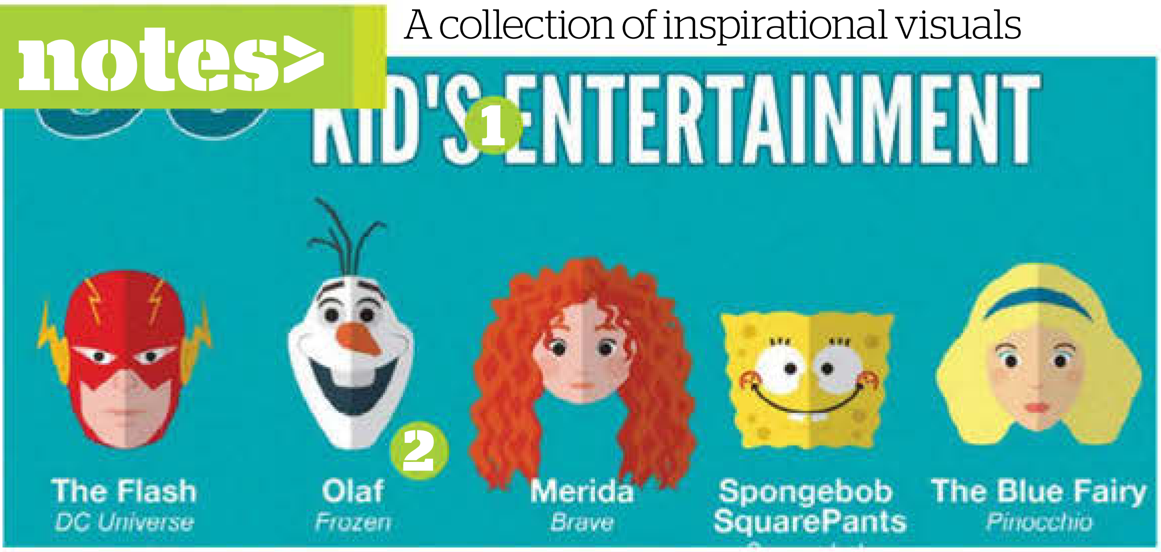
Plus, check out **Web Design Superstars**, **225 Best Web Tools and Resources** and **Amazing Websites and How to Build them Vol 01**. How can you get your hands on a **Web Designer** digital edition? Head to **bit.ly/1hsGYgl**, download the free **Web Designer** app and find them waiting for you as an in-app purchase.

<design notes>

A collection of inspirational visuals

Life advice from 50 beloved characters

bit.ly/1H9InDO



1 The title is big, bold and long, using only two colours and a thin navy outline, but it perfectly encapsulates exactly what the rest of the infographic is going to present.

2 To keep in-line with the kids theme the design uses cute, and instantly recognisable, illustrations of a host of great characters.

Colour picker

Hot hexidecimal codes

I am not your slave

bit.ly/1EpgmWV

#8B8892	#6AA5D2	#B0DBE7	#EDF2C0	#FFF6A9

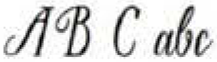
Fae forest

bit.ly/1DWs7k6

#313730	#525D55	#6C7E6D	#212A5C

Typesetter

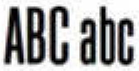
The best fonts you need



Saint Agnes

bit.ly/1DWrWFE

A handwritten font with an imperial Roman-like feel.



TT Bluescreens

bit.ly/1bDRkt7

A condensed font that will work perfectly with posters.

Graphics

Great visuals to inspire

THE
BALD
EAGLE

Animal Kingdom

on.be.net/1KWROa7

A self-initiated branding project from Jonathan Quintin that blends illustration, textures and design to create a great-looking end product.

Thememantic

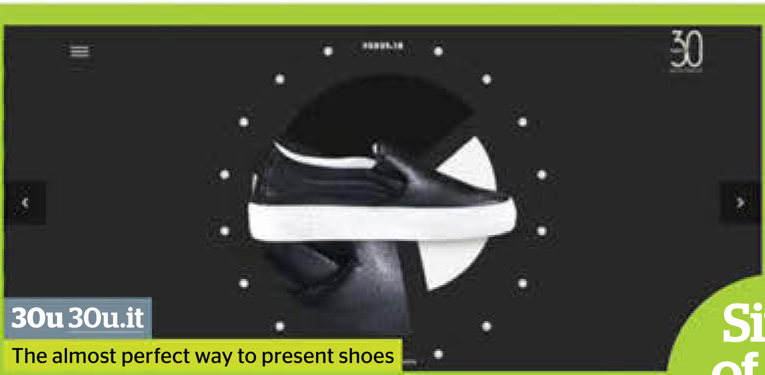
Style up your WordPress

Mode

bit.ly/1GYEpdu

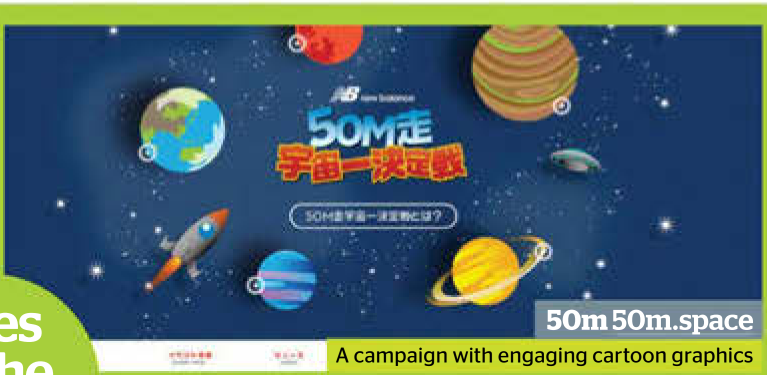
A classy monochrome theme that is the perfect option for fashion-conscious bloggers. Plenty of layout options to create a unique design.

Sites of the month



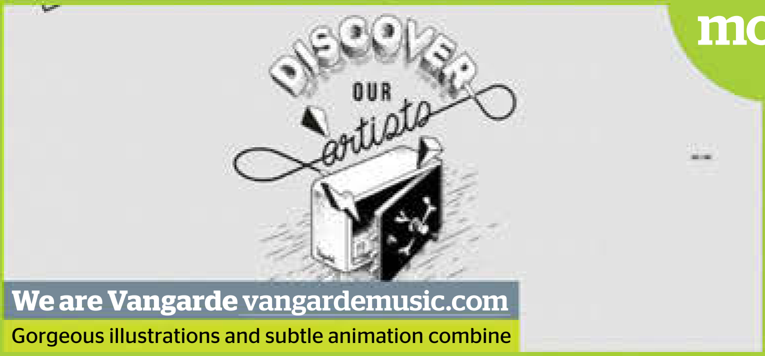
30u 30u.it

The almost perfect way to present shoes



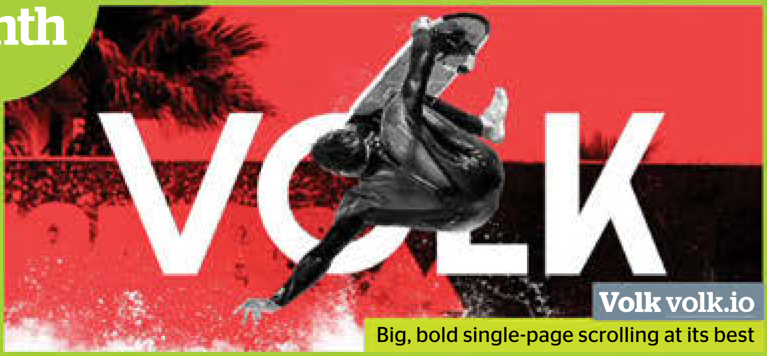
50m 50m.space

A campaign with engaging cartoon graphics



We are Vangarde vangardemusic.com

Gorgeous illustrations and subtle animation combine



Volk volk.io

Big, bold single-page scrolling at its best

<webkit>

Discover the must-try resources that will make your site a better place

CamanJS

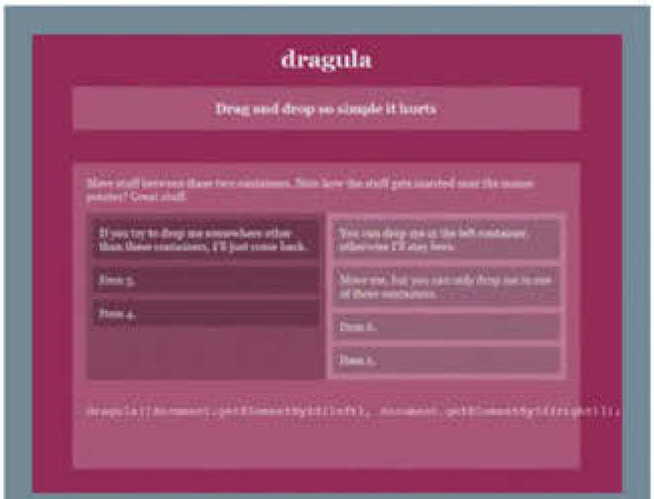
camanjs.com

CamanJS uses HTML5 Canvas to manipulate images with JavaScript. An easy-to-use interface lets users add filters and effects.

Strand

mediamath.github.io/strand

A library of common UI widgets built as Web Components that are built on top of Google's Polymer framework and are also designed to be interoperable.



dragula

Drag and drop so simple it hurts

Move stuff between these two containers. Note how the stuff gets inserted near the mouse pointer? Great stuff.

If you try to drop me somewhere other than these containers, I'll just come back.

Item 3.

Item 4.

You can drop me in the left container, otherwise I'll stay here.

Move me, but you can only drop me in one of these containers.

Item 6.

Item 2.

```
dragula([document.getElementById(left), document.getElementById(right)]);
```

Dragula

bevacqua.github.io/dragula

The premise of drag and drop is simple, but implementing the technique rarely is as simple as we think. The dragula library is here to help make the process simpler, as its tagline suggests "drag and drop so simple it hurts".

By default, dragula enables the user to drag an element in any of the containers and drop it in any other container in the list. If the element is dropped anywhere that's not one of the containers, the event will be gracefully cancelled according to the various available options. There are a host of examples that perfectly demonstrate the capabilities of the library.

jsblocks

jsblocks.com

Currently in beta, jsblock tags itself as a "Better MV-ish Framework". It is easy to learn, fast and covers UI and apps.

Sass Burger

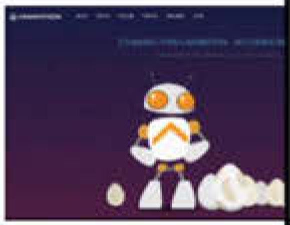
joren.co/sass-burger

An extremely simple option for creating a hamburger to cross animation for your navigation system.

TOP 5 HTML5 animation resources

Create browser-friendly animation without the need to code

01



Animatron

animatron.com

No coding required is a hook that works very well, and Animatron uses a friendly and intuitive timeline approach to create web-friendly animations

02

HTML5 Maker

html5maker.com

HTML5 Maker is a tool for creating relative simple animations such as banners and sliders that can be easily embedded into any site.

03

EaselJS

createjs.com/EaselJS

A JavaScript library that uses the HTML5 Canvas element to create its animations. Check out the Live Edit demos to see how the library works.

04

Mugeda

mugeda.com

A cloud-based pro authoring tool (with a free option) that uses a timeline interface to create animated and interactive content.

05

Tumult Hype 3

tumult.com/hype

A keyframe-based animation system that creates outputs in HTML, CSS and JavaScript. It can also create eBooks, infographics and presentations.

THE MUST-KNOW TECHNOLOGIES

New technology comes out all the time, new languages, new frameworks, but when is the time to drop the old for the new?

As a web designer or a web developer, you have to keep your eye on what's going on out there in the field. But that doesn't mean you need to learn every new thing that comes out. If you did, you would never have time to do any actual work!

The key is to look for bigger trends in the market and any major technology shifts. For example, when Steve Jobs blocked Flash from running on iOS devices, that was an event worth noting. Why? Because that meant the eventual death of Flash, given how popular iOS devices were at that time. So Flash developers had to take note, and people thinking of learning Flash should have put on the brakes.

In a short time, it was clear that Flash was on its way out, especially when Adobe (the makers of Flash) stopped producing a Flash player for Android.

The death of Flash meant a new technology had to take its place. HTML5 adoption was already growing fast, and Steve Jobs decision to kill Flash and back HTML5, pushing forward HTML5 adoption in the process. It was clear that it was finally time to dump Flash and learn HTML5, as the market tends to favour open standards over closed!

I've been in the web design and development game since the early Nineties, and the one thing that I've seen over and over again, is that open technology is typically favoured over closed. Open technology (often called open source) is free and not owned by any one company. The nerd community prefers this because they feel locked in otherwise – this feeling is all about being tied in to some corporation that could suddenly raise prices or changes features in closed technology.

Going back to my example of Flash vs HTML5, HTML5's open nature was often sighted as a reason why it should be adopted over Flash.

STEFAN MISCHOOK
President of Web Mentor
webmentor.org

Stefan has been building websites and web apps since 1994 and currently develops web training courses and apps.

Webmentor is **Killersites.com**'s new online video training site that features practical web design and development training videos.

HTML5 is just one of many examples of open technology winning the hearts and minds of the nerd community, much like PHP, MySQL, WordPress and JavaScript too, which are just a few examples of technologies that beat out competitors that were not open source. So when choosing a new language or technology, consider whether or not it is open first of all.

Don't jump into something new for only incremental improvement. Before I jump into some new technology (and dump another) the new technology had better offer some major advantages. Sometimes these advantages can be technical improvements, say for example it could run much faster then our previous tech. Or perhaps a new technology might do something the current one simply can't do at all – a new feature that is superuseful. One such example is that we are looking at using PostgreSQL for our database instead of MySQL, and this is because PostgreSQL does a few things that our project could really benefit from.

But it's important to understand that everything will come with its own set of unique problems. So a new technology may do something really well, that your current tech doesn't. But you might then discover that your current tech does other things, that the new technology has problems with. It is rare to actually find something that is an improvement over everything that you're already used to.

Look to where the jobs are going. Another factor to consider is the job market; will learning certain tech open you up to securing more, or better paid work?

With a few Google searches, you can find where the demand is and what the trend is. Today for example, JavaScript is growing fast, as is the demand for good UX developers. Be careful to not chase after lucrative jobs in an area where there is no growth. For example, there are high-paying jobs in older (dying) programming languages, simply because nobody wants to code in them anymore!

Experienced nerds know that they will code themselves into obscurity if they work with dead-end technology. So it's hard for employers to find the coders, and so the pay is high.

Finally, don't fall in love with a technology – be ready to dump it when it's time to move on!

STORYTELLING IN MODERN MARKETING

Readz CEO Bart De Pelsmaeker reveals how they pragmatically disrupt traditional content creation and design flows

Q. Content, strategy and marketing are all critical in engaging an audience. How does the Readz platform help ensure that these elements will combine for a successful conclusion?
Our value proposition in the larger context is to remove some of the bigger hurdles when executing a brand campaign. For brands and agencies, our platform disrupts the traditional flow when creating interactive content experiences that carry their content marketing strategy forward.

Readz accomplishes this feat by reducing upfront time and energy cost considerably, if not completely. For example, a lot of marketers come to us looking for design assistance. Our platform removes that hurdle by letting marketers tap into our roster of vetted designers and we add a layer of creative direction to ensure quality.

Alternatively, for people who don't need that design assistance, they can build content directly on our platform, which is probably the easiest platform on the market to create responsive content without being bound to pre-existing templates.

Q. Users are more sophisticated than they were five years ago, meaning that brand builders need to be more astute and aware than ever before. How important is it to tell the story of a brand?
Storylines and narratives are how we have been transferring information for centuries. It is almost like we are genetically programmed to better assimilate and remember information in the form of a story. So yes, it is a very effective method, and if your brand has an interesting or captivating story, your customers will only love you more after hearing about it.

BART DE PELSMAEKER
CEO
readz.com

Bart is the CEO of Readz, an LA-based startup active in digital content creation. A native from Belgium, beer, chocolate and cycling are in his genes.

What is very important is to tell stories about your brand. It is very often nothing short of amazing what incredible feats people have done when creating new products, or how unlikely the creation of some others was. While most of us know the story of Post-it Notes for example, many others stories remain as yet untold.

And truth is, even the most bland advertising message can be transformed into a story. I personally still enjoy it when an agency client comes with this beautiful storyline that tells in a clear and appealing manner what a product is and what it can do for you.

Q.What technologies does Readz provide to help tell a story? And, what examples does Readz have to demonstrate how well they work?
I believe strongly in the role of the designer in this story-telling process. Great content design requires that designers understand how their creativity and design can help a story come into its own.

Our role as a company in that process is to help the designer to make their idea come to life, whether the technology used is parallax scrolling, background videos, swiping and so on. We have seen successful projects produce over an 100-fold increase in engagement for some customers.

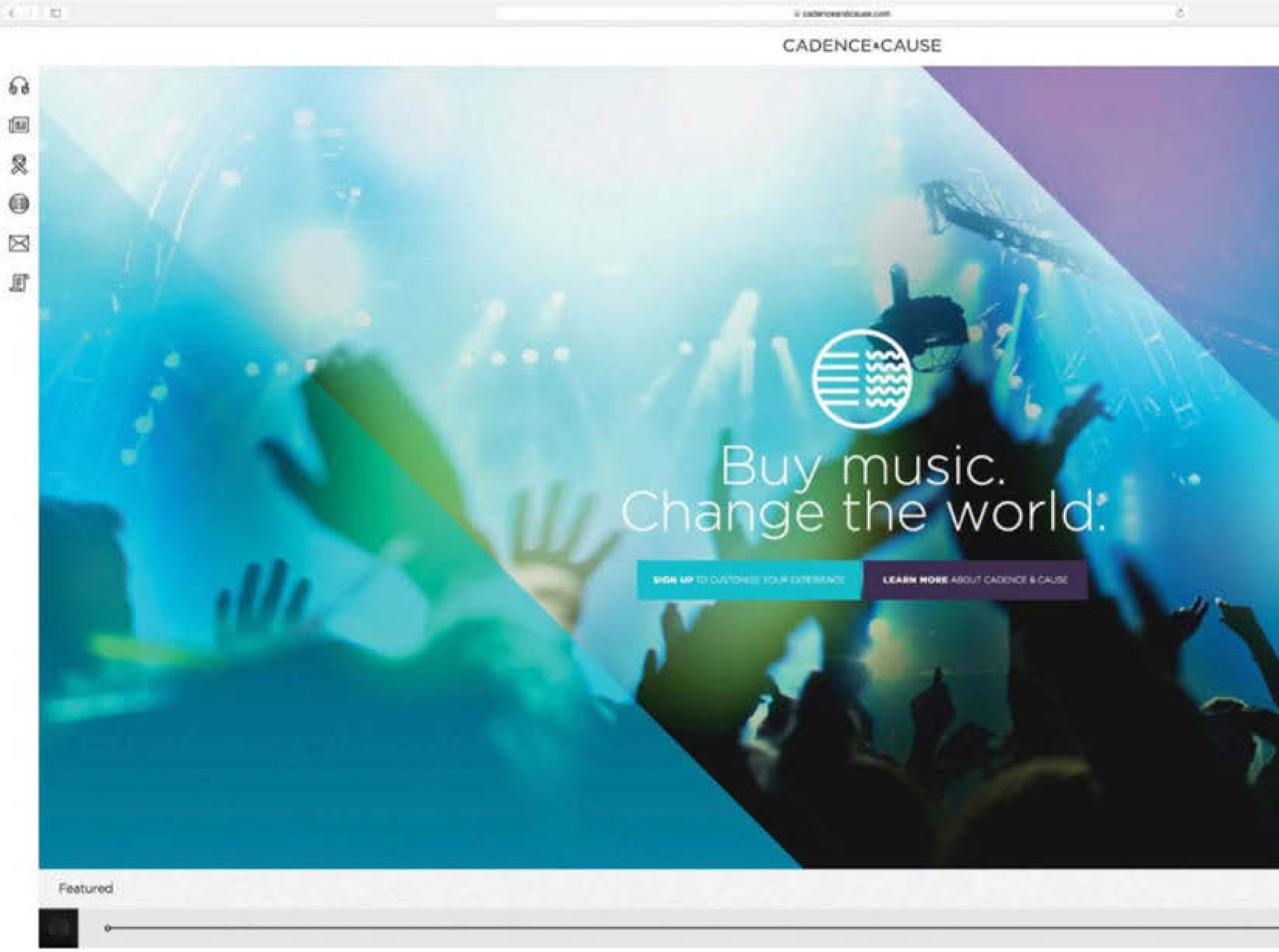
Q. Providing a beautiful frontend is only part of the creation equation. How does Readz ensure that its creations get seen? What techniques are used to drive traffic and get conversions going?
One of our common beliefs is that quality is essential and quality is only possible by making sure the basics are well mastered. We really tend to focus a lot on good readability, clear messaging, good images and illustrations, and if we combine all of that in a smooth technology wrapper, customers will end up being very appreciative of the content.

Q. Data analysis is how owners get to work out what works and what doesn't. What tools does the Readz platform provide for its customers?
While creating brand recognition and affinity is important, often our customers come to us with specific objectives and benchmarks in mind.

The breadth of what can be measured is enormous: from a general audience up to an individual's behaviour across all content and marketing pieces on Readz.

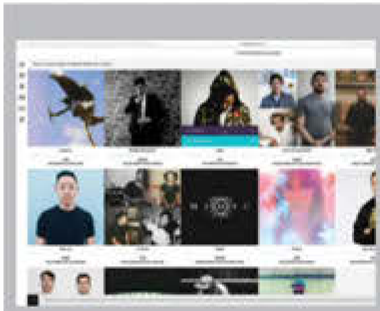
We can also connect with existing external systems, from Google Analytics all the way to Marketo, and we are releasing a dashboard with KPIs as well that works within our Readz application very soon.

Where ideas
take shape.


cadenceandcause.com
CADENCE&CAUSE
Buy music.
Change the world.
SIGN UP TO CUSTOMISE YOUR EXPERIENCE
LEARN MORE ABOUT CADENCE & CAUSE
Featured

<Above>
- The site feels more like a web app, featuring clean colours and sharp front-end work, opening on a big, bright splash screen

#3D2D51 #00BECC #F2F2F2 #1679BF #7F3498

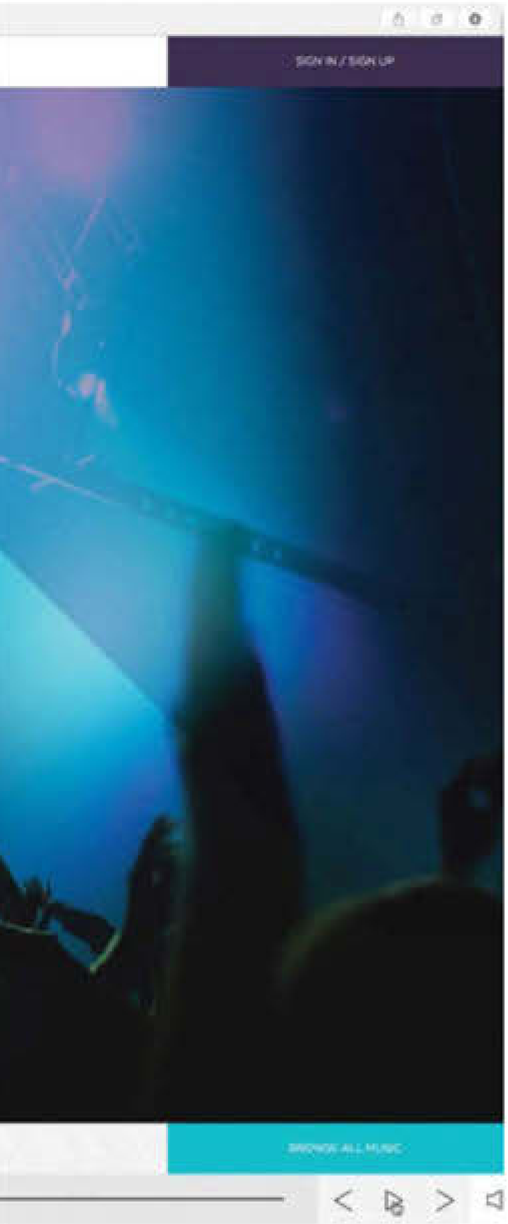


<Above>
- Available music is arranged in rollover rows itemising artists and offering instant previews, buying links and charitable achievements

<Above>
- Artist pages provide pictures, biographies and purchasing or donation offers while an Impact Timeline illustrates activity

<Above>
- A How It Works section breaks down the way Cadence & Cause works, using a long two-column layout with a zigzag theme

<Above>
- Refreshingly, the site pays as much attention to the featured good causes, but also lists charitable creative events within the News section



abcABC
1234567890
abcABC
1234567890
abcABC
123456789C

<Above>
• Gotham from H&Co via **typography.com** is used exclusively throughout the site in weights 200, 300 and 500

Cadence & Cause

cadenceandcause.com

Development technologies HTML5 Canvas, Web Workers, CSS3, flexbox, Modernizr, Require.js, WebGL

Designer **LRXD**

lrxd.com

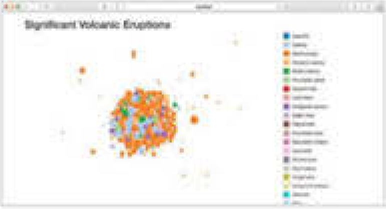
This innovative music commerce site invites fans to "Listen Today. Change Tomorrow" with a big, bold, brave approach

Think of some the biggest names in rock and pop – they seem a bit like superheroes don't they? Well imagine a world where our music heroes used their work to actually save people and help some genuine charitable causes. This is the concept behind Cadence & Cause – a service where music purchases from fans also contribute to responsible charities.

The site itself looks and behaves more like the kind of music purchase and streaming apps that users may come to expect. Core CSS colour palettes are clean, minimal and functional with highlight shades of blues and purples that add accents to the page. Big art imagery carries a distinctive angular, shard-like motif, while nicely shot promo photos of the featured artists provide faces and personality to the product.

Music preview and playback is thoughtfully and neatly integrated right into the latest versions of browsers like IE, Chrome and Safari, with service signup also available via social media. "Musicians have always believed they can change the world," explains John Gilbert, who is chief digital officer at Denver-based designers LRXD. "With Cadence & Cause, we're finally giving them a tangible way to see the positive effect their music is making. Fans will love it too, gaining access to exclusive music and experiences, a deeper connection with the artists they love and, of course, the opportunity to better the world – all by doing what they're already doing: buying and listening to great music. Everyone wins."

> "Musicians have always believed they can change the world"



Build animated bubble visualisations with D3.js

The developers at LRXD reveal the technologies, methods and code behind crafting the animated timeline elements found on cadenceandcause.com

01 The animated timeline

Cadence & Cause features animated timelines that use D3 (**d3js.org**) to animate campaign data over time. To get started with visualising you don't have to write a huge data collection or aggregation code. Here we'll use a public dataset to demonstrate how these timeline elements were created.

02 Set up the HTML

We'll be loading CSV data with AJAX, so make sure you have a local web server running. Let's use Bootstrap (**getbootstrap.com**) to get a CSS layout going, before dropping in some containers for the graph and the labels. Download the Significant Volcanic Eruptions dataset from Tableau Public (**tinyurl.com/pdzrnoc**) and save the XLSX data in CSV format.

03 Load the data

In our JavaScript begin by declaring some variables. The d3.scale.category20() function gives us some colours to work with. D3 has a CSV loader and parser, so we can use it to easily load our data. We'll also pass in a callback to run our visualizeData() function once the parsed data is available to us:

```
$(function () {
var data;
var nodes = [];
var throttledNodes = [];
var graph;
var width = 800;
var height = 600;
var throttleIndex = 0;
var throttleNum = 10;
var timer;
var labels = [];
var color = d3.scale.category20();
d3.csv('/csv/significantvolcanoeruptions.csv',
function (csvData) {
data = csvData;
visualizeData();
}
);
```

04 Prepare components

Now we need to 'massage' our data into a useable format. Iterate over each row of CSV data and stash what we need in the nodes array. Use D3 to add an SVG element that will work as our canvas. We'll use D3's layout to create some rules. The throttledNodes array is empty, but will populate with data points later.

```
var visualizeData = function () {
data.forEach(function (val, index, array) {
var node = {};
node.date = new Date(val['Year'], 1, 1, 0, 0, 0, 0);
node.type = val['Type'];
if ('' == node.type) node.type = 'Type N/A';
node.amount = val['Volcano Explosivity Index (VEI)'];
node.radius = node.amount * 2;
nodes.push(node);
if (labels.indexOf(node.type) == -1) labels.push(node.type);
});
nodes.sort(function(a, b) {return a.date  b.date;});
var svg = d3.select("#volcano-graph").append("svg")
.attr('width', width)
.attr('height', height);
```

05 Set up the animation rules

The D3 Force Layout function calculates where each bubble needs to go, triggering a tick event for rendering each animation frame. We then set up a listener to apply the calculated values to the page elements. We're using the d3.geom.quadtree tool to subdivide the graph into sections. The visit function will go through each section and run a collision handler (from **bl.ocks.org/mbostock/3231298**).

```
var force = d3.layout.force()
.gravity(0.04)
.charge(0)
.nodes(throttledNodes)
.size([width, height]);
force.on("tick", function(e) {
var q = d3.geom.quadtree(throttledNodes),
i = 0,
n = throttledNodes.length;
while (i < n) {
q.visit(collide(throttledNodes[i]));
i++;
}
svg.selectAll(".data-circle")
.attr("cx", function(d) { return d.x; })
.attr("cy", function(d) { return d.y; });
});
var collide = function (node) {
//handle collisions here
};
```

06 Show the data

The addCirces function copies nodes from the nodes array into the throttledNodes array in chunks. This lets us trickle in the circles. We also need to let the Force Layout object know to apply the rules to the new nodes. Finally, use a timer for adding circles over time:

```
var addCircles = function(){
var adds = nodes.slice(throttleIndex, throttleIndex += throttleNum)
throttledNodes = throttledNodes.concat(adds);
svg.selectAll('.data-circle')
.data(throttledNodes)
.enter()
.append("circle")
.attr("r", function(d) { return d.radius; })
.attr('class',function(d){ return 'data-circle'; })
.style("fill", function (d, i) { return color(labels.indexOf
(d.type));});
if (throttledNodes.length >= nodes.length) {
clearInterval(timer);
}
force.nodes(throttledNodes);
force.start();
}
timer = setInterval(addCircles, 100);
```

07 Label the graph

Now what does each colour represent? We'll fix any confusion by adding a key to our graph.

```
labels.forEach(function (val) {
$('#labels').append(
'<p style="border-left: 20px solid ' + color(labels.indexOf(val))
+ ';padding-left: 10px;">'
+ val + '</p>');
});
}});
```

THE BEST TOOLS FOR RESPONSIVE DESIGN*

Responsive LAYOUT MAKER

Prototype device-agnostic layouts from the content out with custom breakpoints. The result? Rock-solid layouts with code that is crisp, clean and production ready!

Responsive SITE DESIGNER

Create complete, one-of-a-kind, responsive websites without coding. All the power CSS offers is available through intuitive visual controls. 'Nuff said!

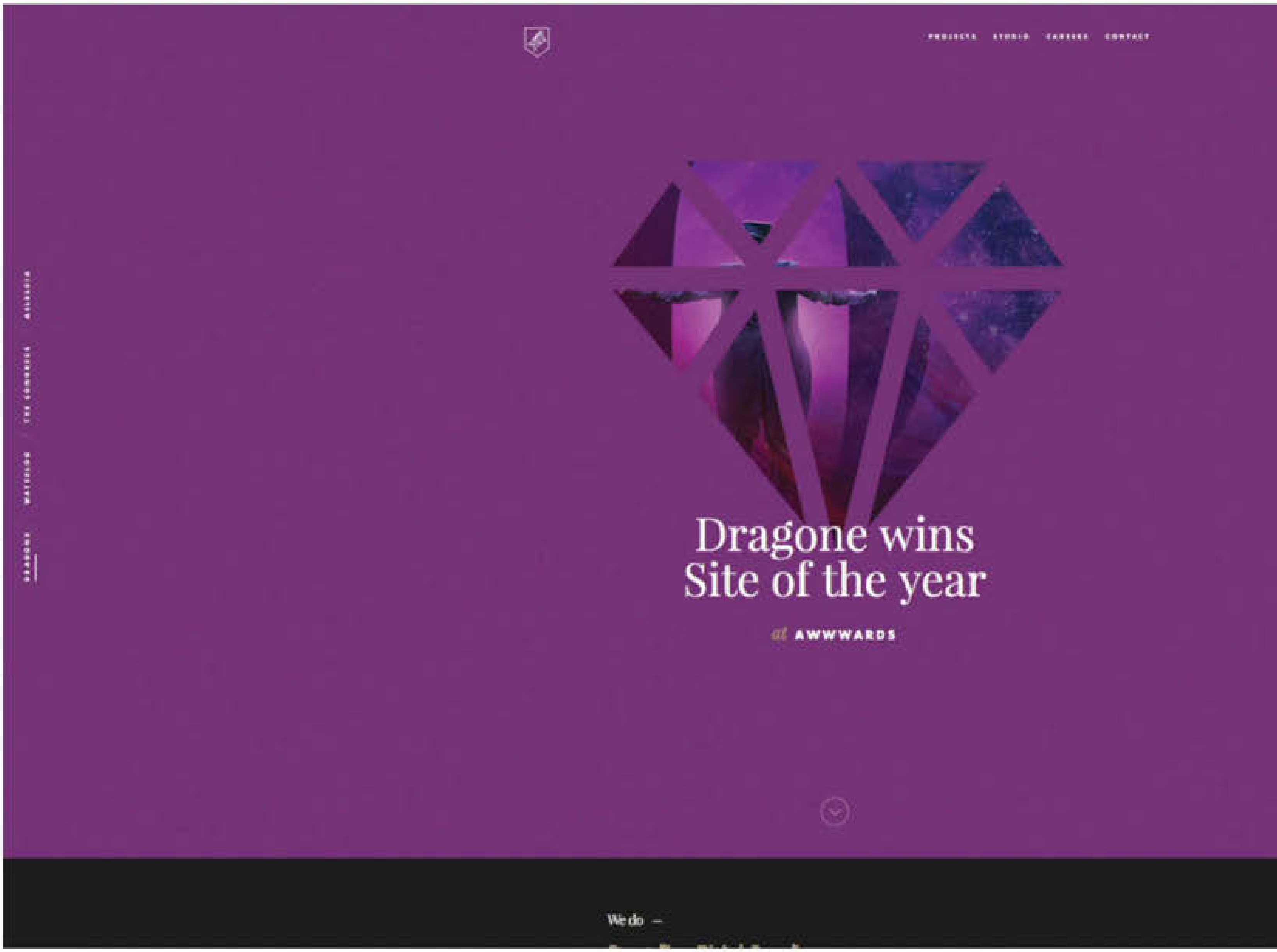
Responsive EMAIL DESIGNER

Drag-n-drop to create beautiful emails that work wonders on any device. Full story-driven design freedom, unrestricted by rigid templates. Need to tweak that font size for smaller devices? Do it!

*IT'S A CONTEST!!

tweet **#[insert:wordfoundabove]tools @coffeecup** for a chance to win one of the apps. More info at **www.coffeecup.com**.

by CoffeeCup



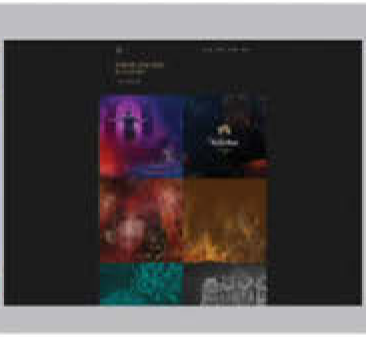
Dragone wins Site of the year

at AWWWARDS

We do –

<Above>
- The main homepage is fairly sparse, featuring a rolling header cycling through project case studies

🟪	🟧	🟦	🟥	🟨	🟦	🟫	⬛
#743475	#FF6600	#32567D	#923238	#B89A4E	#6363CE	#B0976D	#1C1C1C



<Above>
- A collection of rolling animated headers (shown here in sequence) uses Canvas to animate each of the logos

<Above>
- The main Projects page features a two-column grid of nicely animated rollover previews, each leading to themed studies

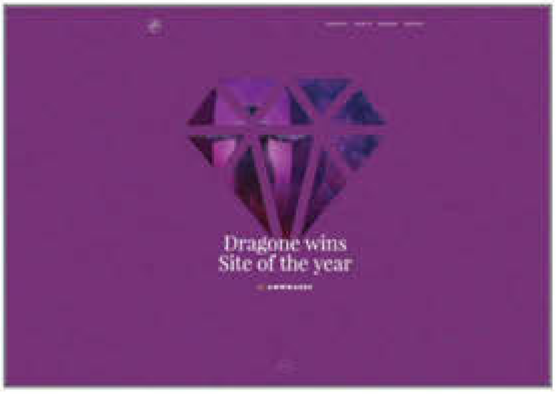
<Above>
- Featured project case studies are lengthy and richly populated with a variety of imagery, transitions and assets

<Above>
- The Studio section is more restrained in terms of colour but boasts a stylish layout, neat typography and genuine personality

Dogstudio

dogstudio.be

Development technologies HTML5, CSS3, SVG, jQuery



Dragone wins
Site of the year

Designer **Dogstudio**

dogstudio.be

This stylish Belgian agency showcase is by no means a dog, but boasts a design bearing real teeth to 'paw' over

It's not always that an agency website comes with a disclaimer and especially one warning of "stupid Belgian humour syndrome". But this new site for digital studio Dogstudio, based in the southern Belgian town of Namur, comes with precisely that – not to mention a 'cheeky' footer slogan. This multi-award-winning team offer services across branding, strategy, UX, design and development, spanning a range of celebrated work on show here. From a cycling morpher slide of flat colours and animating cut-outs, cascades sleek dark templates packed with eye-popping imagery. Click through to the Projects section and the content comes alive with big sumptuous headers, parallax scrolling, scroll-triggered animations, CSS, SVG and stylish typography. What's more, once qualified visitors feel suitably inspired they are encouraged to join Dogstudio's "crazy team" by applying for listed vacancies. "Dogstudio is a quality-driven digital agency from Belgium and we're hungry for the world," explains Henry Daubrez, co-owner and company director. "When people expected us to release a new website after more than two years without anything decent, it was the start of a journey we didn't expect to be this challenging. We knew we needed to come up with something to be remembered by, while showing what our DNA is all about. This is how we got to create immersive case studies, write some edgy and cheeky copywriting, and experiment with SVG animations and video-based CSS transitions."

> "We knew we needed to come up with something to be remembered by, while showing what our DNA is all about"

abcABC
1234567890

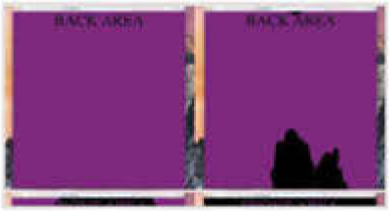
<Above>
• Futura PT by ParaType and available via Typekit is used on subsequent <h2> headings and page links

abcABC
1234567890

<Above>
• Playfair Display by Claus Eggers Sørensen styles header titles and <h3> headings in normal and italic

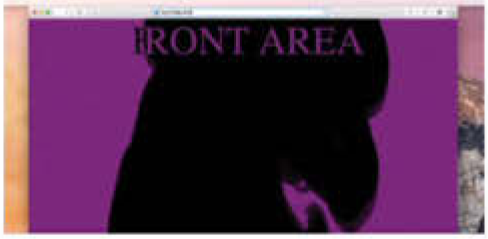
Add animated page transitions with spritesheets and CSS3 masks

The Dogstudio developers reveal the technique behind the site's eye-catching page transitions using CSS3 animation and WebKit



01 Observe the effect

On **Dogstudio.be**, front-end developers Larry Gérard, Anthony Dupont and Nicolas Masson used HTML5, jQuery, CanJS and CSS3 predominantly. Via a combination of these modern web standards, the team not only achieved a distinctive look, but also added ingenious effects. One of the most appreciated elements here is the page transition, which uses an animated mask. Over the following steps we'll examine this technique using CSS3 and jQuery.



02 The HTML structure

First of all, set up your stage with two stacked full-screen areas. In your HTML page this needs only to be two <section> elements labelled here as back and front. Each will have a container CSS class attached, along with a removed or added class used as a selector for toggling. We'll then use the id to differentiate any of our additional styles.

```
<body>
<section class=”container removed” id=”homepage”>
BACK AREA
</section>
<section class=”container added” id=”content-page”>
FRONT AREA
</section>
</body>
```

03 Set the page CSS

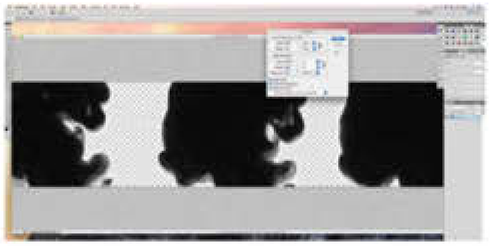
In the CSS we need to add the basic classes, starting with container and simply setting absolute positioning, zero margins and a z-index value. This gives us the full-screen areas we want for our section elements set in the previous step. We then use our element ids to set backgrounds and label colour to differentiate the two – before and after the transition.

```
CSS:
.container {
position: absolute;
top: 0;
left: 0;
right: 0;
bottom: 0;
z-index: 5;
font-size: 120px;
text-align: center;
}
#homepage {
background: purple;
}
#content-page {
background: black;
color: purple;
}

```

04 The spritesheet PNG

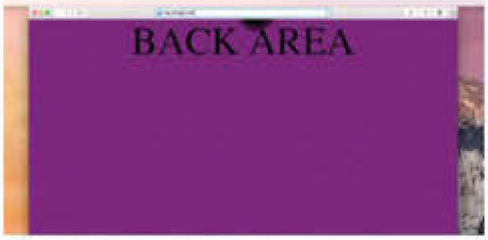
A PNG spritesheet (6474 x 280px) is then required to animate from a transparent frame. We suggest keeping it lightweight in order to improve performance. In this case, FFmpeg (**ffmpeg.org**) was used to extract 13 frames (498 pixels wide) from a video as PNG files we then stitched together. After the export, those files didn't have any transparency so we used Photoshop to manually remove the white area in each of the frames.



05 The CSS animation

The animation uses mask and animation CSS3 properties. Inside a container.added class we first pass in the spritesheet path with mask:url() before setting the mask size. Here the width is the number of frames (13) multiplied by 100% while the height is simply one frame. The mask position is initialised to the upper-left corner before we call the animation which shifts that position to the upper-right corner over 13 steps.

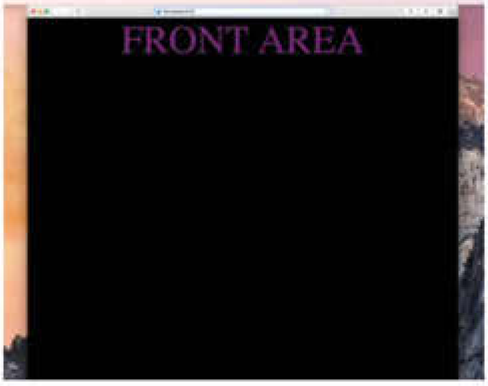
```
CSS:
.container.added {
-webkit-mask: url(”transition.png”);
-webkit-mask-size: 1300% 100%;
-webkit-mask-position: 0 0;
-webkit-animation: mask-play 0.8s steps(12) 0s both;
z-index: 10;
}
@-webkit-keyframes mask-play{
from{
-webkit-mask-position: 0 0;
}to{
-webkit-mask-position: 100% 0;
}
}
```

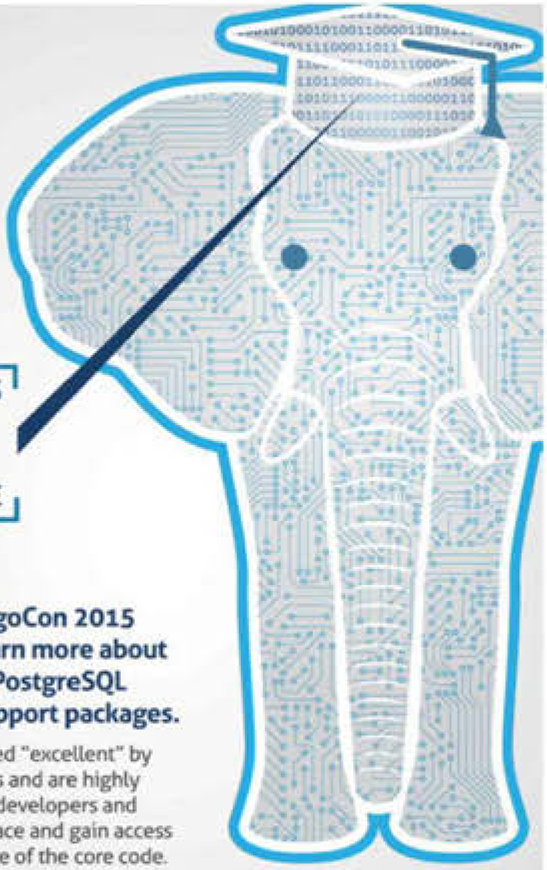


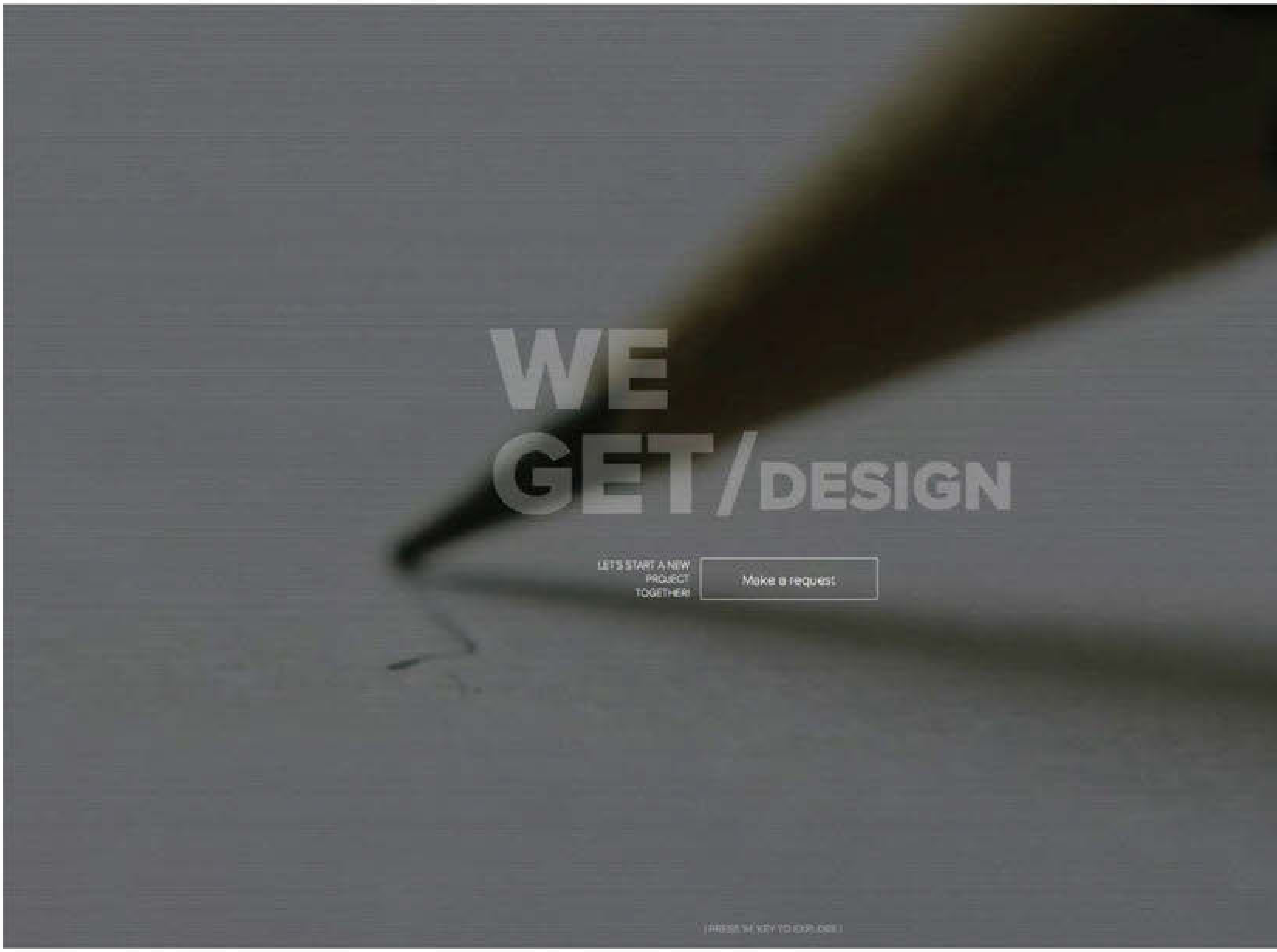
06 Complete the transition

Once your animation is ended, you will need to remove the older area with jQuery to keep your DOM clean and efficient. Then that's it! Please note however that this technique only works on WebKit-based browsers as WebKit-only CSS properties are used. IE and Firefox don't support those properties so Dogstudio uses a fallback to a simple fade-in or fade-out solution using Modernizr to detect support.

```
JQUERY:
setTimeout(function(){
$(”.container.removed”).remove();
$(”.container.added”).removeClass(”added”);
}, 800);
```






WE
GET/DESIGN
LET'S START A NEW PROJECT TOGETHER!
Make a request
[PRESS 'M' KEY TO EXPLORE]

<Above>
- Weget Enterprise's site is produced by Dan Ghimpu, Valentin Baranetchi (developers) and Alexandr Mocreac (art)

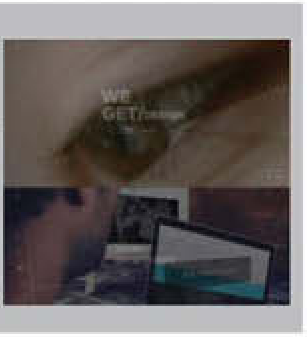
#42413B #285258 #34495E #280000 #006064

<Above>
- A pop-out side menu on the left provides links to key sections but also the option to toggle languages

<Above>
- Upon clicking the various button links in each section, page content glides into view as a semitransparent overlay

<Above>
- The backgrounds for each section use HTML5 video and looping clips in WEBM to add life and motion

<Above>
- A Portfolio section provides access to a range of featured projects, with a range of rollover previews

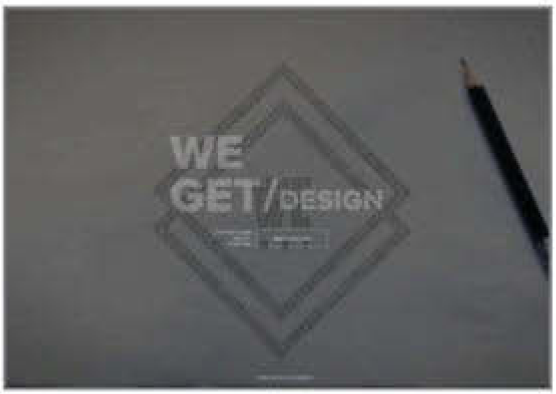


<Above>
- The site's eight main sections (shown stitched together here), uses mouse scrolls or arrow keys to shuffle through

Weget

weget.nl/en

Development technologies HTML5 video, CSS3, JavaScript, AngularJS, CodeJS, nanoScroller.js, CodeIgniter PHP framework (backend)



Designer **Weget Enterprise**

weget.nl/en

A cascading collection of HTML5 video clips forms a distinctive portfolio website for Weget, a Dutch design agency who pride themselves on digital

Weget is a digital and design studio based in the city of Groningen, Netherlands, offering a range of services. Whether you require essential web products for big or small business startups, eCommerce, web applications or hosting and domains – you'll find it here. Recognised for notable design work across numerous CSS award nominations, this unique portfolio experience details Weget's work in all languages. Content is arranged in a lengthy structure more akin to a header slider component than a traditional web page. These seven screen-width panels are defined by HTML5 <video> backgrounds, looping relevant clips (WEBM) of around 25 seconds long. Sparse snippets of text provide hints towards further information and a section of featured works. Most distinctive though is the navigational mode, prompting the use of the M key to pop out the main menu and the arrow keys to cycle down. Neat transitions then roll or flip the next panel into view, while on tablets and mobile a simple swipe gesture controls flow instead. "The goal for the website was to represent best our strongest points," explains Dan Ghimpu, CEO and project manager. "In order to achieve this, we chose to use video backgrounds for the most powerful visual impact, combined with a full-screen design, built [with] a block strategy. The gallery of the website had been designed and programmed separately, and then integrated. This was needed to complete the concept of the gallery, which didn't fit the main style of the website."

> “We chose to use video backgrounds for the most powerful visual impact”

abcABC
1234567890

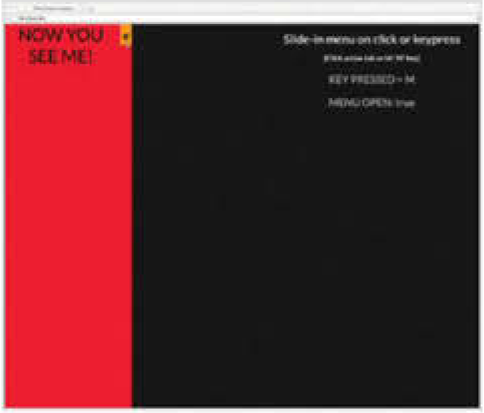
<Above>
• Lato by Łukasz Dziedzic, found on Google Fonts, is used on header links within the footer section

abcABC
1234567890

<Above>
• Proxima Nova by Mark Simonson Studio is available via Typekit and is used in Bold across page headings

Create slide-in menu panels with CSS3 and jQuery

Inspired by Weget.nl, we describe a solution for adding slide-in menu panels activated with both click and keypress events



NOW YOU SEE ME!

Slide-in menu on click or keypress

[Click arrow tab or hit 'M' key]

KEY PRESSED = M

MENU OPEN: true

01 Get started

In this example we will emulate Weget's slide-in navigation menu panel, using CSS3 transitions and jQuery to trigger open and closed states. The <div> panel will be activated both on click and also via pressing the M key. Start with a new HTML document and attach the latest jQuery library to the <head> of the page. In addition we'll be styling our page text with Lato via Google Fonts:

```
<script src="http://code.jquery.com/jquery-1.11.2.min.js"></script>
<link href='http://fonts.googleapis.com/css?family=Lato' rel='stylesheet' type='text/css'>
```

02 Define the HTML elements

For page elements set up a menu <div> containing a pair of nested divs – a main panel and a small tabbed edge. We'll style these in Step 4 with the subsequent <h1>, <h2> and <p> tags (these are labels for instruction and outputting event or variable states).

```
<body>
<div id="menu">
<div id="panel">NOW YOU<br>SEE ME!</div>
<div id="tab">&raquo;</div>
</div>
<h1>Slide-in menu on click or keypress</h1>
<h2>[Click arrow tab or hit 'M' key]</h1>
<p id="label1">KEY PRESSED: None</p>
<p id="label2">MENU OPEN: false</p>
</body>
```

03 Basic page styling

Moving to the CSS, start by styling optional body styles. Here we've opted for Weget-inspired background and text colours, along with the familiar use of the Lato font. Alongside this are sizing preferences for our various page labels. All of this is optional and included for illustration, as it has no bearing on the menu itself.

```
<style>
body {
font-family: 'Lato', sans-serif;
text-align: center;
background-color: #161616;
color: rgba(224, 224, 224, 0.8);
}
h1 {
font-size: 40px;
}
p {
font-size: 36px;
}

```

04 CSS menu element

The menu element contains two parts, the main panel which is hidden on close and the tabbed edge which remains visible. So in the CSS we define styles for those divs, crucially setting the left property negatively to hide it in the left page. By setting a transition we dictate how it will animate into view when we dynamically toggle on the open class and set left to zero:

```
#menu{ ______________________________
position: absolute;
top: 0;
bottom: 0;
left: -410px;
background-color:#F03;;
color: #000000;
font-size:60px;
-webkit-transition: all 0.4s ease;
transition: all 0.4s ease;
}
#menu.open{
left: 0;
}
#panel{
width: 410px;
float: left;
}
#tab{
width: 40px;
background-color: #F90;
cursor: pointer;
float: left;
}
</style>

```

05 Bind the key events

We'll add our script just before the </body> tag and start by defining a Boolean variable as an open or closed flag. By default the menu is closed so we set it false. Next we have a small function called on load which very simply binds keyup and click events to the page body and the menu's tab element. Upon either of these events we will then call an activated() function that is defined next.

```
<script>
var menuOpen=false;
$(function(){
$("body").bind("keyup", activated);
$("#tab").bind("click", activated);
});

```

06 Invoke the menu transitions

The main activated() function then performs our menu actions. Additionally, it outputs variable states to our page labels, which is purely cosmetic to track the events. However the main part checks the passed event object for an M press keyCode (77) or our bound click. If true we merely toggle the #menu.open class from our CSS before flipping our Boolean flag. This is then used to switch the tab arrow icon to in or out. All done!

```
function activated(event){
$("#label1").text("KEY PRESSED = " + String.fromCharCode(event.keyCode));
if(event.keyCode==77 || event.type=="click"){
$("#menu").toggleClass("open");
menuOpen=!menuOpen;
if(menuOpen){ $("#tab").html("&laquo;");} else {
$("#tab").html("&raquo;");
};
$("#label2").text("MENU OPEN: " + menuOpen.toString());
};
};
</script>

```

THE PLEASURE OF DISCOVERY

At a time when we have access to so much, and see so little, the Way to Go project will remind you of all that can still lie before you

Project	**Way to Go**
Web	**a-way-to-go.com**
Agency	**AATOAA**
Web	**aatoaa.com**
People involved	**10**
Total hours	**A full-time year on the project**

The web is awash with wonderful websites bringing new twists to design, but occasionally something that doesn't quite fit the norm makes its way onto the web. Little quirks, clever angles and the integration of new web technologies combine to bring a new style experience that has rarely, if ever, been seen before.

This is exactly how we would describe a Way to Go. It is not a website that drearily pumps out information, it is, as described by the creators "an interactive experience for human beings between five and 105 years old."

The simplicity of the Way to Go concept could easily lose a user quickly, but it is this simplicity that engages the user. It piques the user's curiosity just enough to make them want to go further. Just when you think you are going to lose interest a new point in the experience reignites the urge to push forward and investigate and explore. As a Way to Go mainstay Vincent Morisset points out, "Maybe it lasts six minutes; maybe it lasts forever". Of course it doesn't last forever, but the whole experience enables the user to submerge themselves and get lost in their own world.

The browser-based experience is compelling and intriguing, but there is an option to go one step further. If you're willing to go VR, if you're "Rift-y", Morisset compares the experience as to being like "a grey squirrel balancing on a branch, fearless".

A perfect description of the Way to Go experience lies in its associated preface:

YOU ARE ON YOUR WAY.

Yes, you are on your way.
It is not your first journey but Way to Go is the next journey before you. A walk through a strange country – strange, familiar, remembered, forgotten. It is a restless panorama, a disappearing path, a game

Vincent and the team showcase exactly how they've blurred reality with Way To Go's alien

and a feeling. Way to Go is a small experience that gets bigger as you uncover it. And the trees will change their shape, and the sky will widen.
And you will fly.
We go away every day. We plunge through the city, skate down roads, tunnelling toward a destination without remembering the quest we are on. A journey is a collection of moments – we are here, we are here, we are here, and yet we miss these moments. A journey is a collection of choices – turn here, stop here, choose here, and yet we surrender these choices.
What if we quit surrendering? What if we didn't miss?
Here is a world enclosed in a screen. Here is an adventure. A landscape of leaves and wildflowers, teeming with hidden life. A garden and a wilderness, a wistful blink of dream. You are Jean Painlevé, Marco Polo, Maria Merian. You are Alice, Sonic, Osvaldo Cavandoli. You are a visitor, a cartoon of face and limbs, and you are going on a walk. Using hand-made animation, music, 360-degree capture technology and WebGL sorcery, Way to Go imagines a dream world of journeys. Walk, run, fly; crouch in the grass and remember what's hidden all around. Slip like a rumour from one place into another; chase your shadow; listen to the slow pulse of the metronome, black-clad, following in your wake.
Are you alone? Are you not alone? Are you dreaming or awake? Can you ever reach the mountains?
Can you see what's here before you?
Set out through woods and fields, sunlight and aurora, grey and colours.
Set out, in deliberate lucid looking
and you'll find,
perhaps,
the present.
Are you ready for this luscious, sudden pleasure of discovery?

•

Inspired by his first personal project, Vincent Morisset reveals how the journey to Way to Go got started, "In 2012, Caroline Robert, Philippe Lambert, Édouard Lanctôt-Benoit and I created the film for *BLA BLA* (**blabla.nfb.ca**) with the support of the National Film Board of Canada – it was my first personal project. For more than a decade, I had been developing creative content for others, sometimes for brands or for other artists. One of my most prolific collaborations was with the band Arcade Fire, with whom I developed a multitude of projects (interactive videos, net art, a documentary, album artwork and more). But to do something that was coming from me was totally new. It was a revelation. As a director and an author, I had the opportunity to start from scratch!

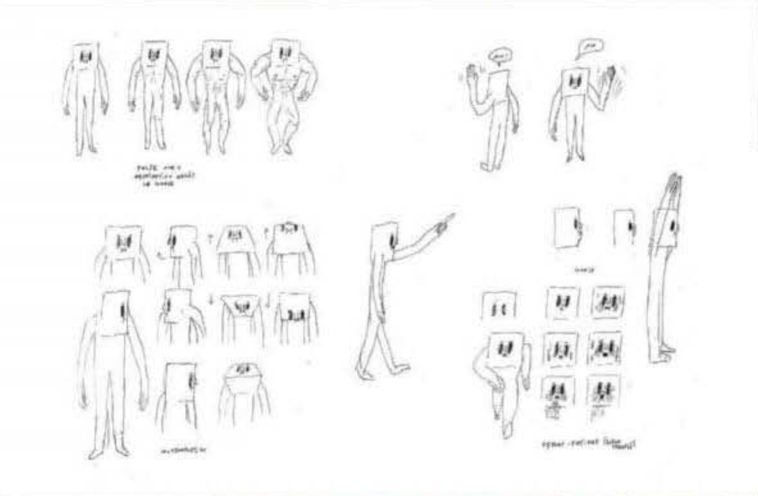
"After *BLA BLA*, I asked Hugues Sweeney, my producer and the head of the interactive studio at the NFB, if we could continue the journey through a new interactive web project. I wanted to explore the notions of space and time and combine the [world] of videogames and cinema. I wanted to put the spectator in a state of hyperlucidity where they can see their environment differently. [I was] inspired by a holiday in Granada, Andalusia, it's always fascinating how we look at things from another perspective when we are travelling. We were walking in the Albayzín neighbourhood and I was obsessed with all the details: the ceramics embedded in the walls, the plants, the light… That's the kind of sensation I wanted to translate into an interactive experience. An invitation to look around, to take your time and feel a connection."

A project is a two-way street that brings involvement between the creator and the client. We put the question of how much interaction was there between the two parties involved to Morisset: "This question is quite interesting because it illustrates how, historically, as creators of interactive projects, we have almost always been service providers. How we talk about the projects and how we produce them is often connected to the advertising world. At award shows, for instance, it's complicated to fill up the credits form for a personal project with no client and no agency. Way to Go was done in the tradition of independent film production.

"The National Film Board of Canada has been producing films since 1939, supporting documentaries, fictions and animations. They have been a laboratory for a very long time [with] IMAX, direct cinema, very early CGI films in the

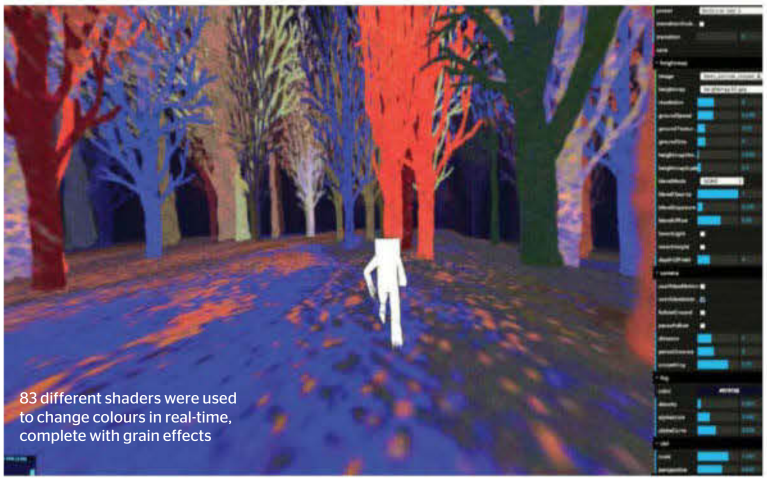
"I WANTED TO PUT THE SPECTATOR IN A STATE OF HYPERLUCIDITY WHERE THEY CAN SEE THEIR ENVIRONMENT DIFFERENTLY"

Caroline got to flex her creativity, from animating in 2.5D to making costumes

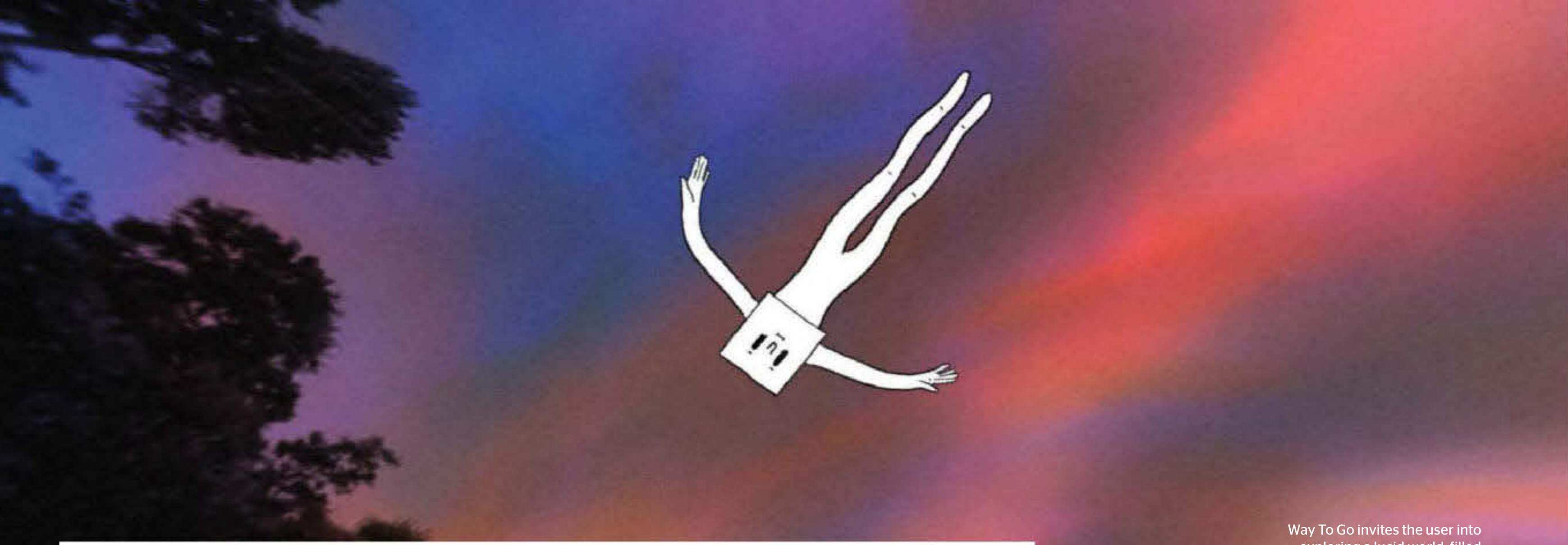


Philippe maximised the feel of nature by remixing sounds and never repeating them

Tree patterns are hand-illustrated before being turned into a digital landscapre



83 different shaders were used to change colours in real-time, complete with grain effects



Way To Go invites the user into exploring a lucid world, filled with endless possibility

To get a sense of realism, the AATOAA team filmed insects and other wildlife in macro

360-degree video capture proved to be an integral part of Way To Go

Way To Go employs video stitching to create a panorama-like sensation

The transition from photography to an imaginary world needed to be absolutely seamless

Seventies, and so on. Five years ago, they found the interactive studio to support original content. For both Way to Go and *BLA BLA*, I had carte blanche! I feel really fortunate about it. Hugues Sweeney, my producer, gave us a lot of creative room. He also transmitted and explained our creative process to the institution so we could move forward within its structure. We had the chance to dedicate ourselves full-time for a year. Once or twice a month, I would meet Hugues to share with him where we were. He would ask open questions to put our work in perspective. My projects are never really storyboarded. They need a lot of trust because, for a very long time, they are just a constellation of ideas. I start from an intuition and a desire to achieve a specific feeling. Then, as a team, with Édouard, Philippe and Caroline, we knit that first thread. It's highly experimental in the making. I want the end result to be accessible and enjoyable to everyone though. The web has this unique and amazing capacity to reach beyond generational and cultural boundaries. I envision my projects in this context."

Ideas make an experience and Way to Go is a concept that had a definite vision as Morisset revealed, "I wanted to explore how our relation to space and time affects our way to look." He went on to reveal more of the creative process. "The traditional first-person videogame mechanism came to mind. How could I translate gamepad controls (moving with left hand and making actions with right hand) compatible with cinematic assets? The use of 360-degree videos travelling forward, combined with control on the progression of the timeline (to give an illusion of movement) seemed to be a promising approach. Caroline, at an early stage, did a lot of sketches and research to create visual maps that would combine the key elements of the experience. How did we represent people walking in art history? How has panoramic representation been developed over the last centuries? Transhumance, path making, Street View, our relation to GPS and so on. My core team all worked from day one. Technology, sound and image all collide in an iterative process where ideas nourish each other. It is my role to distil and keep the project on track. As a team, we are becoming more and more effective with this way of working. We try to validate ideas as quickly as possible so we don't waste too much time on paths that are not relevant. But having time is really important. This is where we have the opportunity to truly experiment and tread on unknown territories. Always being a bit in the fog keeps us in a continuous state of creativity and excitement during the whole production!"

The visuals for Way to Go stray far from the usual static images, occasional animation and YouTube video. The experience is truly interactive as Morisset enthuses, "In most of my projects, the interface is invisible. It is important to me that the content doesn't have a layer between the spectator and the piece. By forgetting about technology, the spectator can put himself in an emotional mindset closer to cinema. The content is a mix of 360-degree videos, dynamic shaders and hand-made animations – all to give the illusion of a reactive world full of life. For Caroline, the visual artist, her role in this project was really diverse [and it included] designing and making costumes, imagining and animating a character in 2.5D, filming macro videos of insects and plants in the woods, hand painting patterns and trees with ink and chalks and more. The idea was to be able to bridge video-based content with imaginary elements. At the beginning of the experience, they are two separate worlds, but slowly they are merging into one. The spectator controls the pacing of the walk. He can also interact with his environment and change his perspective on it. We decided to set the experience in the woods because it felt like a universal premise that everyone can relate to. Philippe, the music composer and sound designer, worked in a similar way to Caroline – nature sounds were dissected and organised. Each time you try Way to Go, the birds and insects are placed randomly in space so it's never the same thing twice The music is generated by your walk. Using Euclidean rhythms, sequences of notes are triggered depending of your pace. It feels organic and hypnotic. Thousands of tiny notes were recorded acoustically in a studio from sounds generated by a synthesiser. The dynamic music score becomes the leitmotif of the journey."

It only takes a single glance at Way to Go to instantly know that there is a lot going on behind the scenes. Perfectly revealed by Morisset, "With Way to Go, Édouard really pushed open web technologies in unbeaten paths. First he imagined tools that Caroline and Philippe could use to sculpt in detailed images and sounds. Through an interface with sliders, we had the chance to see the effects and interactions in real time. This helped immensely to the way we all collaborated."

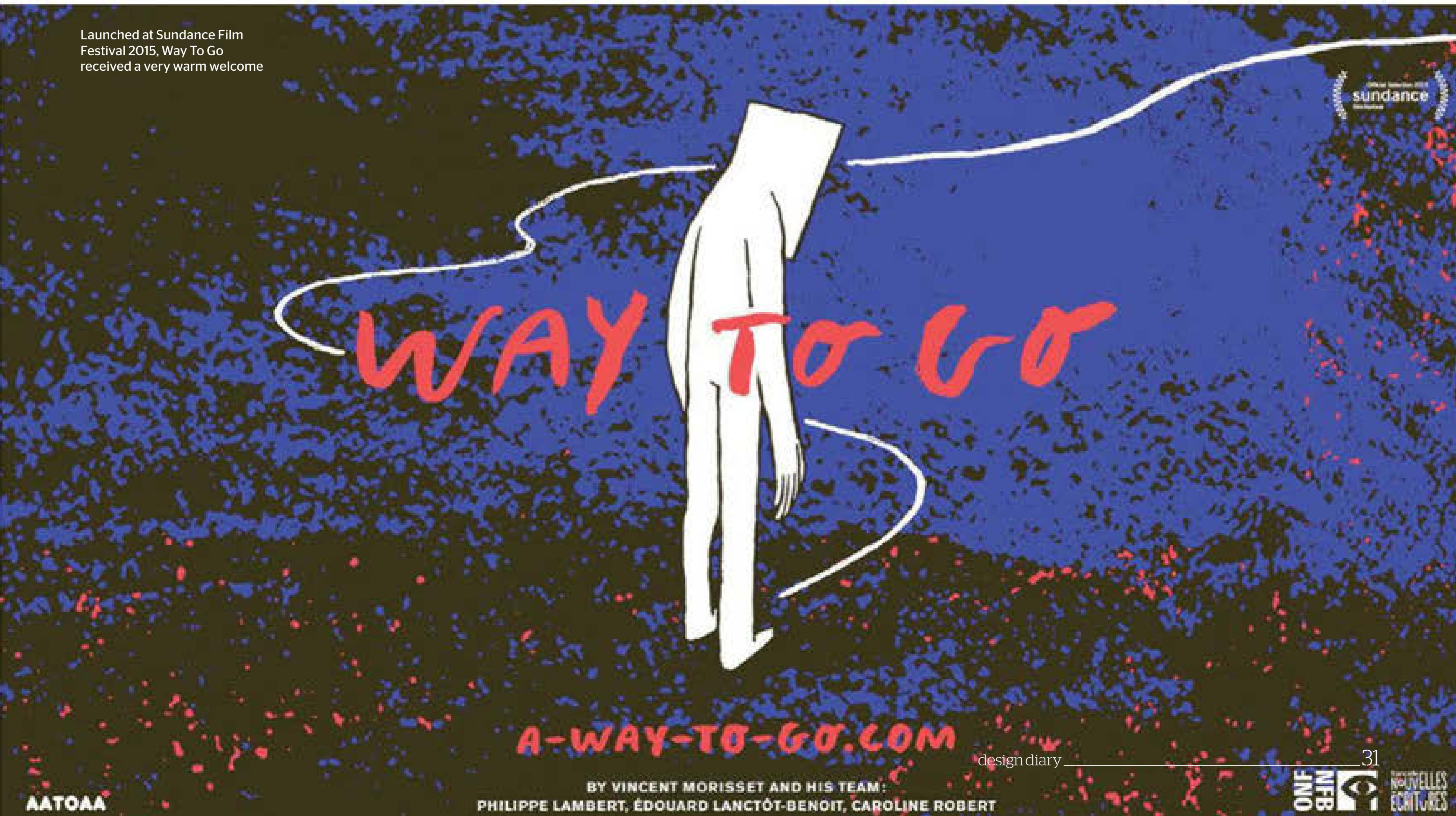
Exactly what web technologies built the project? Here is, in a nutshell, the tech behind the project:

- JavaScript, HTML5, WebGL, three.js, Web Audio
- Dynamic control of 360-degree videos
- 83 custom shaders to change lights, grain and visual effects in real time
- Infinite 3D worlds rendered from paper drawings
- Custom system for integration and synchronisation of 360-degrees videos, 3D environments and 2.5D elements
- Management and live mixing thousands of sounds
- Generative and interactive music score
- Envelopes, ADSR, filters and dynamic spacialisation
- Sound granulation
- Convolution and reverb
- Unique techniques for video stitching of 360-degree shootings in narrow environments
- Video analysis and tracking system for dynamic use of heightmaps and 3D scenes
- Live video stabilisation
- WebVR – adaptation for the Oculus Rift for use on a web browser
- Gamepad API – connection of game console controllers to web browser.

And this is a list that perfectly demonstrates the power and possibilities of the web.

EMBRACING ALL

The build scenario behind the Way to Go site was not your typical affair. This was reflected in the launch. It shunned the standard onslaught on social media and took on the sensibilities of a film launch and this is exactly where the project was first revealed to the public. The date: Thursday 22 January 2015. The place: the Sundance Film Festival. The occasion, Morisset explains: "The project was launched at the Sundance Film Festival in the New Frontier programme. We developed a VR version of the project that we presented there. It was fantastic. The venue was packed with visitors from morning to night and the reactions were amazing. For people doing web projects, this is a true chance to meet your public in person. We also released the project online. Having a hybrid project gives us the opportunity to reach everyone through the browser, but it also offers the possibility to present a unique experience in festivals and museums. We presented the Oculus Rift version at Phi, RVCQ and Excentris in Montréal, at GDC in San Francisco, Gaîté Lyrique in Paris, White Rabbit in London and at the Exit Festival in Maubeuge and Créteil. The project was really well received and has been shared by word of mouth in different communities. The NFB publicists also reached out with traditional and online publications. Way to Go has a foot in almost all genres. Every community talks about the project from its own perspective: video game, film, net art, animation, electronic music, web technologies, virtual reality… we like to be the UFO!"


Launched at Sundance Film Festival 2015, Way To Go received a very warm welcome

sundance
WAY TO GO
A-WAY-TO-GO.COM
BY VINCENT MORISSET AND HIS TEAM:
PHILIPPE LAMBERT, ÉDOUARD LANCTÔT-BENOIT, CAROLINE ROBERT
AATOAA
ONF NFB
NOUVELLES ÉCRITURES

Creativity assembled

With a wide and diverse portfolio, Bad Assembly has quickly established themselves as one of the leading digital design agencies in the world, blending a ceaseless passion to communicate with their masterful control of the digital landscape

who Bad Assembly
what Specialises in blending form seamlessly with function to create unique digital experiences
where 448. S Hill Street, STE 412, Los Angeles CA, 90013
web badassembly.com

Key clients

Amazon
A&E
Netflix
Toyota
72andSunny

In 2006, Bad Assembly's founding partners Jimmy Walker (creative director) and Nathan Holloway (technology director) were working at Big Spaceship's LA office. When Big Spaceship closed the LA office, Jimmy and Nathan decided now was the time to start one on their own and found Bad Assembly the very next day.

As Jimmy explains: "The name originated as the title of a collective which never formally took shape. The early sketches of our original logo were done at that time. Years later, when we started the company, we decided the name was a good fit for us. In terms of the URL it was easy to get since the combination of the words 'Bad' and 'Assembly' was unique. The name definitely came before the domain in our case."

Setting up their business of course meant developing a website. Nathan outlined their approach to getting their brand online: "An agency's website is extremely important and also one of the most difficult projects for a studio to work on. We fall prey to the common problem that we're often too busy to focus on self-promotion. Usually by the time we've updated our portfolio we are mentally ready for a complete redesign. Web trends change so fast that staying up to date is a constant process."

The method which Bad Assembly has used to attract its clients over the last few years has been evolving, as Jimmy explained: "There was a time when word of mouth was all that we needed to book work. For a long time we filled a vendor role for large ad agencies like Saatchi & Saatchi and Deutsch and were happy to do so.

"Over time, the competitive landscape grew exponentially while at the same time big agencies began building large internal development teams. In recent years we've placed a stronger focus on business development, specifically in terms of forging direct-to-client relationships. We still attract work based on reputation, but we actively pitch for new business all of the time."

Nathan also said: "We try to work with clients that are on the same page creatively with us. We've worked with clients big and small; what's important to us is the opportunity to create something great."

All agencies have their own particular approach to the work they do and clearly develop a set of design sensibilities that inform their approach. Is it clear to Bad Assembly what the DNA of their agency looks like to their clients? "We've worked on such a wide variety of projects: microsites, games, TV interfaces, social campaigns and digital billboards," explains Jimmy. "It's difficult for us to think of a single project that defines our ethos. The common thread through our work is that we put a

Rather than being static, Bad Assembly constantly evolves its project concepts and how it operates as a company

The team pride themselves on making engaging, cross-platform content

The agency uses a solid foundation to base its projects on, which in turn ensures cleans design

Bad Assembly finds it important to work on UX approach and design solutions as elegantly as possible

industry insight

Jimmy Walker, creative director

> “Every large project starts with working with the client to understand its needs to ensure proper scoping and planning of the project. Whenever possible we like to incorporate a discovery phase to define strategy and direction. From there we build a dedicated team that is custom tailored to the project’s needs. This team would consist of a combination of strategy/UX, creative, technical and project management”

premium on giving the best possible service to our clients by delivering for them.

Nathan continued: “One of our core tenets is to go beyond ‘good enough’ and strive to make every project better than we thought we could make it. We put a lot of effort into the fit and finish of our work; taking the time to dial in an animation or a user interaction, adding a subtle detail, or improving performance can go a long way towards making the overall experience better.”

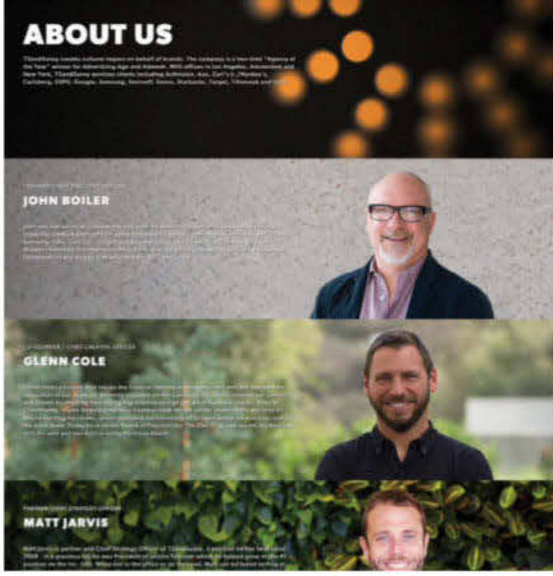
How agencies approach their workflow can often reveal quite different systems with many adopting the agile approach. Jimmy explained the working practices at Bad Assembly: “In our industry, a typical timeline is ‘ASAP’. It’s a luxury to work on a project with a timeline that matches the scope of the work. That said, a large-scale project typically comes with a six-month to one-year timeline.

“Every large project starts with working with the client to understand its needs to ensure proper scoping and planning of the project. Whenever possible we like to incorporate a discovery phase to define strategy and direction. From there we build a dedicated team that is custom tailored to the project’s needs. This team would consist of a combination of strategy/UX, creative, technical and project management.”

Jimmy also outlined the toolset that Bad Assembly typically use: “For project management we use Basecamp, for UX we use OmniGraffle, for visual design we use Photoshop and Illustrator. For motion design we use After Effects, and for 3D we use Cinema 4D. Our studio still creates a lot of standard and rich media ad units using either Flash or HTML5. We’ve been using a newer tool called Slack, which is great for internal communication.”

“We’re really excited by the rapid support for modern web application APIs on desktop and mobile devices,” Nathan continued. “We can create immersive 3D environments with WebGL. We can use Web Audio to build sequencers, filters and synthesisers in the browser. Web Sockets enable us to create second-screen and multiuser experiences. APIs, like notifications, gyroscope events and haptic feedback events, enables our apps to feel like native mobile applications. That’s barely scratching the surface.

“We are inspired by trends in the community to push CSS beyond its former limitations – people use pure CSS to create complex illustrations and animations. Retina displays have rekindled an interest in using SVG to create graphics that look good at any resolution. Web Components are the


ABOUT US

JOHN BOILER

GLENN COLE

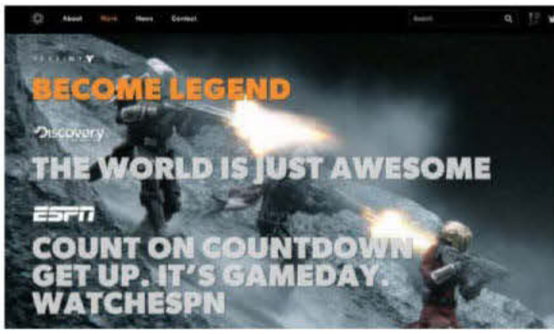
MATT JARVIS

72andSunny

72andsunny.com

After winning Adweek’s 2013 Agency of the Year award, 72andSunny teamed up with Bad Assembly. The goal was to work together to redesign the digital presence of the company and showcase its work to prospective clients, employees and industry press.

72andSunny wanted a remarkable online experience that would showcase groundbreaking work in a simple, highly visual interface with quick loading content and smooth transitions. It needed to strike the balance of encouraging site exploration and provide a way to easily access specific content.


About Work News Contact

BECOME LEGEND

Discovery

THE WORLD IS JUST AWESOME

ESPN

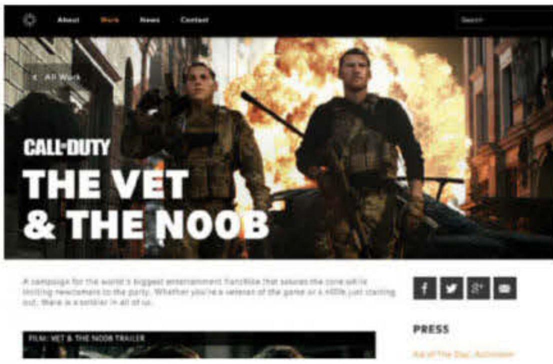
COUNT ON COUNTDOWN

GET UP. IT’S GAMEDAY.

WATCHESPN

The site needed to include a front-end framework that powered desktop and mobile sites seamlessly and a custom CMS capable of managing content and site updates at a moment’s notice.

Alongside 72andSunny’s team, we began to forge a clean, intuitive UI. This led to a full-bleed visual


About Work News Contact

CALL OF DUTY

THE VET & THE NOOB

PRESS

FILM: VET & THE NOOB TRAILER

experience that put the agency’s work front and centre. We closely collaborated with our client to create prototypes that illustrate the user experience so they could successfully sell internal ideas. In the end, we created a stunningly modern, content-rich site complemented by a powerful mobile version.

future, in the short term. Web Components are going to enable us to create reuseable modules with their own internal stylesheets and functionality. The ability to build web applications with encapsulated, modular components is going to change the way and speed in which we work.”

And how does Bad Assembly approach working on mobile platforms to ensure seamless design? “We try to be device-agnostic in our approach to design,” said Nathan. “We have more mobile devices at different resolutions and pixel densities than ever before, so we strive to make our designs look just right on all viewports using intentional, liquid layouts. So yes, responsive design is a facet of our approach. But every project has unique requirements, so we’ve had cases where the best solution was a dedicated mobile site with its own UX, design and front-end development.”

> “For anyone looking to get into this industry, you will likely succeed if you approach this work with a sense of joy and determination”

It’s essential for agencies to stay on top of the latest software and tool releases, so what new frameworks for technologies are exciting Bad Assembly at the moment? “It’s an exciting time to be a developer,” explained Nathan. “With the proliferation of new frameworks and technologies, we have more tools at our disposal than ever; it can be fatiguing to keep up, so we are selective about how we invest our time.

“Over the past two years, while Angular has reached critical mass, we’ve been moving away from monolithic frameworks with a strong bias towards custom code, using vendor libraries only for specific tasks. While we wait for Web Components to gain wide support, projects like Polymer are promising, but currently we’ve been using React for a lot of our front-end UI.”

Nathan also said: “We have used CoffeeScript for the last few years, but we are moving towards ES6, especially now that the latest version of React supports it! Babel transpiles ES6 to ES5 for wider browser support. We use Browserify to handle module dependencies and Gulp to manage build tasks (for example watching Sass changes).

“We recently launched a self-promotional holiday site (**holiday2014.badassembly.com**) built on three.js and we think it turned out great and had a lot of fun building it, so we will be incorporating 3D into more of our work. In the backend we have used Django, Rails and various PHP frameworks in production, but lately we are using Node and MongoDB to rapidly prototype lightweight servers and create real-time apps with Socket.io. We have also started to bring iOS development in-house – Swift has made it much more approachable [as it is a language that came] from web development. We are eager to see where that takes us.”

All businesses must now pay close attention to social media not only across their own business promotion, but also of course as an essential component of every new piece of work. Jimmy explained Bad Assembly’s approach: “Social media has changed the way brands reach and engage with consumers. In many ways, social media has replaced the traditional website. The purpose of sites like Twitter and Instagram in the context of digital advertising is to fuel consumer awareness and drive traffic to purchase. Social media also gives brands the ability to build communities of fans [organically].

“Another advantage that social media offers is earned engagement through UGC (user generated content). Nearly half of consumers trust words, pictures and videos from other users more than brand communications, and earned social engagement drives far better business results than paid social exposure.”

The quality of the work that Bad Assembly has been producing has meant that the agency has also had to grow alongside its portfolio. Jimmy outlined how Bad Assembly approach their recruitment: “We are extremely selective about who we choose to join our team, and we are lucky to have been able to work with some of the most talented people on the planet.

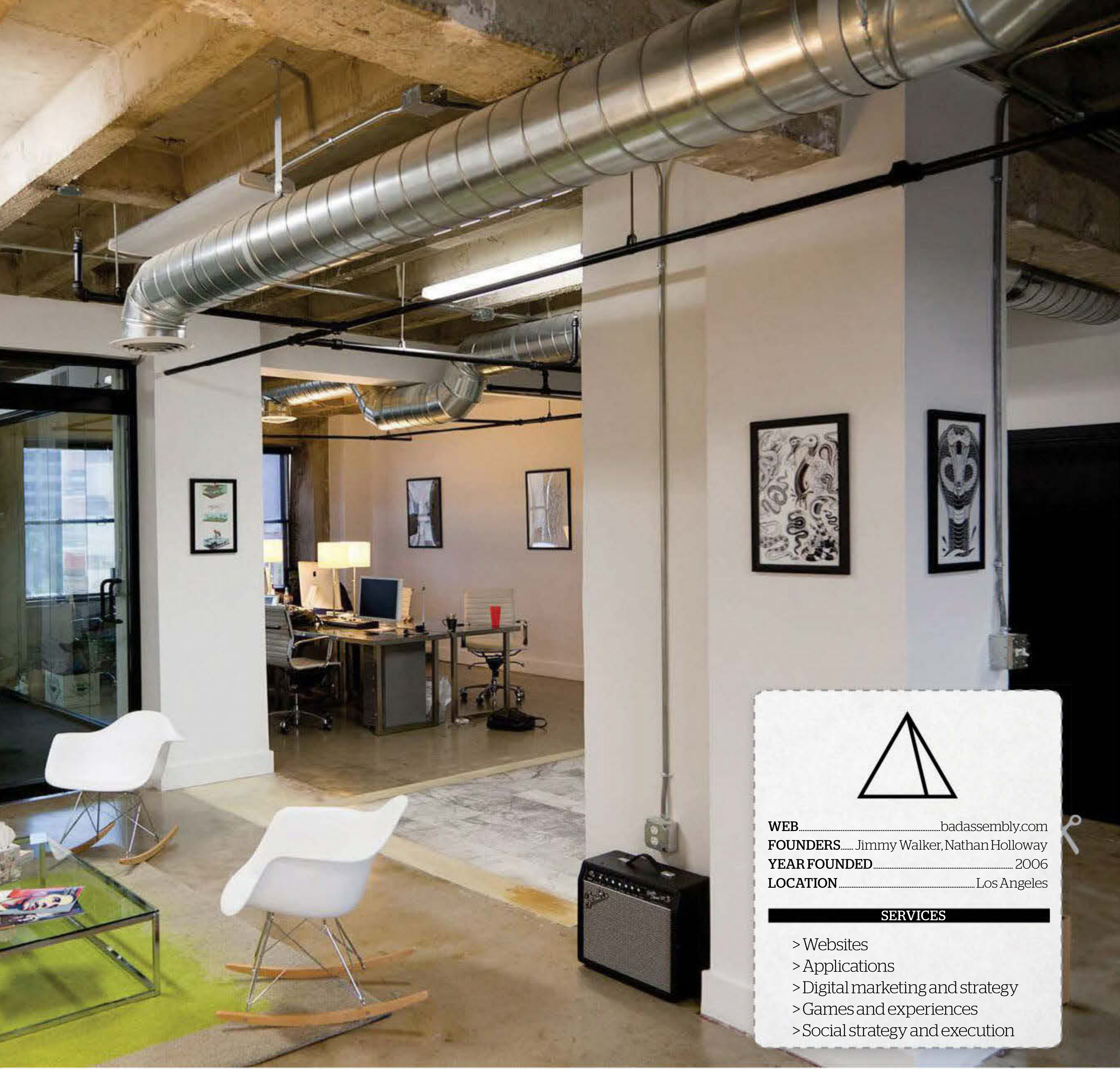
“We are averse to assholes; divas and bullies have no place in our studio. In the immortal words of Bill and Ted, ‘Be excellent to each other’. We look for people with raw talent; it’s okay if they don’t know everything as long as they exhibit the potential to learn and grow. If their attitude is positive and chill, and they show enthusiasm for contributing to the work, they are a fit.

“For anyone looking to get into this industry, you will likely succeed if you approach this work with a sense of joy and determination. Design and development trends are in a constant state of flux, so you will be perpetually learning something new. We are constantly inspired by all the brilliant work being created every day. That inspiration continues to ignite our drive to learn and create.”

And what does the future hold for Bad Assembly and what kind of projects are in the pipeline?

Jimmy concluded: “This is an exciting time to be a team member at Bad Assembly. Moving forward we aim to continue to expand our direct-to-client relationships. Working with clients directly enables us the opportunity to be a part of the conversation at the campaign’s inception.

“We recently forged a new relationship with Amazon and are excited about the future of that partnership. While we continue to design and build exceptional digital experiences for our clients, we’re also focusing on nontraditional ways to help them tell their stories with branded microcontent and second-screen experiences.”



WEB badassembly.com
FOUNDERS Jimmy Walker, Nathan Holloway
YEAR FOUNDED 2006
LOCATION Los Angeles

SERVICES

> Websites
> Applications
> Digital marketing and strategy
> Games and experiences
> Social strategy and execution

TIMELINE

2006

Projects completed 07

Bad Assembly is founded, beginning a long-lasting relationship with Toyota and Saatchi LA.

2007

Bad Assembly opens office in downtown LA and win an FWA.

Projects completed 28

2009

Bad Assembly begins working directly with A&E.

2012

Bad Assembly moves to a larger office in LA .

2013

FWA-winning **72andSunny.com** launches.

Projects completed 112

2015

Bad Assembly starts up a relationship with Amazon.

Projects completed 154

POWER UP CSS PREPROCESSORS

•SASS •LESS •STYLUS

EXPERT ADVICE TO MASTER PRO LEVEL TOOLS

Why you need preprocessors

VETOING THE VANILLA CSS

Web Components are a totally new way of building website parts, instead of using HTML, CSS and JavaScript. CSS is not the simplest and smartest tool in a web designer's toolkit. Between vendor prefixes, baked-in constants for colours and fonts, random requirements for Retina displays, and the lack of a good grid system, CSS is half-finished at best.

But that doesn't mean you have to keep hand-rolling your stylesheets and suffer in silence when you're asked to change the colour scheme of a project at 5pm on a Friday just before a bank holiday. Preprocessors add some of the smarts that plain CSS never had. That means more creativity, fewer bugs, less typing, and a deep sense of joy throughout the entire design team.

The catch is there are three preprocessors instead of one, and they all have different strengths and weaknesses. You have to unlearn vanilla CSS and get used to writing preprocessor CSS – it looks similar but is different enough to trip you up. There's more thinking time when you're starting out, but that leads to less thinking later as you get used to the styles.

The not-quite-a-catch is that you can take the process further by combining preprocessors with an automated workflow that handles code formatting, CSS preprocessing, minification and obscuration, and perhaps also final distribution – you don't need this stage but many preprocessor users do as it can save time.

But which preprocessor should you use? The right choice depends on the rest of your workflow. It also depends on which mixin features matter to you, and which tools and frameworks you use in the rest of your workflow. Choosing isn't hard, but you need all the facts. So that's exactly what you'll get over the next few pages…

> *"Preprocessors save you so much time when going from an idea to an implementation, by reducing the amount of code you have to write and providing some much-needed data-structures and logic to CSS."*
> **Max Glenister, front-end developer at Marmalade & Jam – @omgmog**

What does a preprocessor do?

THE MAIN TRIUMPHS OVER USING REGULAR CSS FOR YOUR DESIGNS

A preprocessor takes a file of something that looks like a variant of CSS with extra features, and then converts it into a file that really is vanilla CSS. Isn't this pointless extra work? Not at all. It's a total win because you get to use a version of CSS with superpowers that you secretly always dreamed about but didn't know actually existed in the world.

Technically, a preprocessor makes it easier to write DRY – Don't Repeat Yourself – code. Examples? Here's a really useful superpower: you can define a colour scheme in one location and change it by editing a couple of lines. This avoids the usual insane search and replace that happens when your client decides the hot flamingo pink menu he wanted last week should really be canary yellow with orange text.

Doesn't that mean you can experiment more? It does. Suddenly big design changes get simpler, and you can play with styles instead of trying to wrestle them into submission in a sweaty and not very creative way.

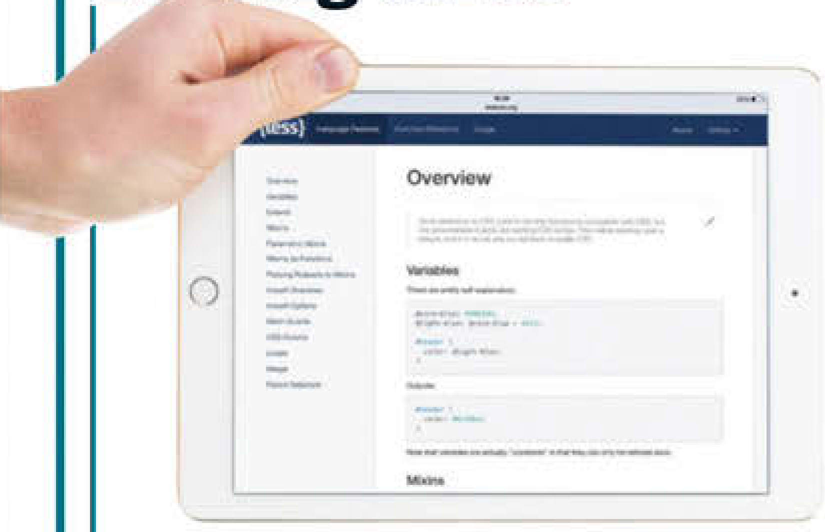
How about vendor prefixes? If you're tired of typing -moz- and -webkit- in front of every animation command, you can leave the preprocessor to add them for you. Again, this means you get to think more about the look and feel of your site and less about where in the file you left the styling for the fifty-eighth div on the sign-up page.

As there are three popular preprocessors, you can choose what kind of superCSS you want to write. You can start small and keep changes simple, or you can go for the most different option from vanilla CSS, which still does the same things – only with extra superpowers.

Top Tip

For maximum productivity, consider using an equivalent HTML preprocessor. Haml, for example, can provide condensed, shorter and clearer HTML.

The big three



Overview

Variables

Mixins

LESS lesscss.org

LESS was inspired by Sass, and arrived a couple of years later in 2009. It's considered mature and stable, but it still gets regular updates.

LESS includes variables, which enable you to set a value in one place and use it by reference everywhere: mixins, which are variables that drop all the content of a class into another with a single line; rule nesting, which clarifies inheritance; functions, which enable you to process numbers and strings; and plenty of colour management features. It includes basic search and replace and programmer-style text formatting.

LESS has good foreign language support, so it's worth considering if English CSS isn't your main interest. It also includes guarded mixins, which produce different CSS according to tests you can specify. It doesn't have fully general conditionals, which can be a limitation for some applications.

As it is used to define styles in the Bootstrap framework, you'll have to use LESS if you use Bootstrap. While developing, you can run LESS from the command line within Node.js or Mozilla's Rhino. You can also include it as a script in your pages and compile on the fly, but that's a bad idea for a production server unless your site gets a couple of hits a day.

CSS with superpowers

Sass is the most mature, stable, and powerful professional grade CSS extension language in the world.

Sass sass-lang.com

Sass (Stylistically Awesome Stylesheets) is a step up from LESS, with less (huh…) simple expansion and replacement, and more 'Let's pretend to be a developer' language features. Sass needs a language called Ruby, which needs some setting up on Macs, Linux, and Windows, although there's also a version that works on Node.js, with some differences.

Sass is more like three tools in one. You can write it as SCSS, which is an extension of CSS3 and looks like normal CSS with extra features, or you can write it as Sass, which uses indents instead of curly brackets. Both do the same things, but you can choose between familiarity and speed. The third tool is SassScript, which includes plenty of ready-made helper tools for manipulating numbers, strings, lists and so on. SassScript is quite clever. You can do things like define variables that work across a project or limit variables to the insides of a selector. It can even mix units, so you can calculate widths in a combination of ems and pixels, which is a neat trick.

Compared to LESS, Sass adds extra smarts, at the cost of extra learning time. It has more complicated and powerful conditionals and repeats, better selector management, and support for partials (or snippets of CSS you can pull in from a file).

stylus

CSS needs a hero

What if we could omit braces?

Stylus learnboost.github.io/stylus

Stylus isn't as popular as LESS or Sass, but it may be underrated, and is certainly worth a look. One obvious big difference is cruft-free syntax, with no useless curly brackets or pointless semicolons, all replaced by white space and indentation.

Stylus is really the closest thing to a full programming language for CSS, this side of JavaScript. In addition to all the usual preprocessor features, like variables and operators, you can define your own functions for smart CSS parameter management and expansion. Stylus includes all manner of conditionals, loops, as well as an @ import option for reusing CSS libraries.

Transparent mixins are a big win. You can define functions in the usual way, but you don't need to specify a list of parameters. Stylus can copy any parameter string into the CSS. It even supports introspection, which means code can see where it is and what's happening around it, so it can behave differently in different blocks.

Another unique feature is interpolation, which is the ability to step through a predefined list of strings of values with a few lines of code, and to drop the strings/values into the final CSS. It's a good way to write dense, efficient code with the downside that it can be harder to read.

PROS

Great for newcomers

LESS looks a lot like CSS, but removes some of the more obvious annoyances. It's not a whole new thing that will make your brain melt, so you get to be productive right away! And because it's used with Bootstrap, which is used on so many sites, it's some way to becoming a standard.

Excellent support

Straightforward descriptions and plenty of examples make the LESS documentation easy to understand. Even with complicated features, you can work out what's happening. LESS is also better for debugging as messages tell you exactly where problems are so you can find and fix them.

CONS

Limited conditionals and inheritance

LESS can't do some useful stuff. Conditionals are limited, so there's no way to make really complicated conditional mixins. Selector inheritance is also limited. The & selector adds some basic inheritance, but it doesn't work with variables and it's not very flexible. There's also no way to mix units when you're working with maths.

No debugging options

Sass has debug directives, which print useful messages. LESS doesn't, so when something goes wrong you're stuck with guesswork, prayer, wishful thinking or – as a last resort – going through the code line by line until you find the problem.

PROS

Still CSS, kind of

Sass still looks like CSS, so it's easy to read. It's not quite CSS as you know it, and some of the more complicated features may make you stop and think. But it's good enough that it won't confuse you when you come back to a project a year after finishing it.

SassScript

The SassScript mini-language is like CSS on illegal steroids. Once you get your head around the fact that everything can be scripted, it's incredibly powerful and expressive. If you're more of a designer than a developer then this can be a shock to the system, but it's well worth the effort.

CONS

Needs Ruby

If you're a Ruby on Rails dev this won't be a problem for you. However, if you're more of a Node or Apache kind of person, then installing yet another development framework just to run Sass is going to feel a bit wasteful of time and resources.

Extra setup

As a result of its Ruby nature, Sass includes tons of flags and set-up options. You can use Sass out of the box without getting distracted by them, but they include a lot of useful features, and you're going to have to spend some time working out what they do.

PROS

Autocompilation middleware

Stylus includes Connect middleware. In plain English this means that Node can work out when you edit a file and rebuild your project for you. If you combine it with a task runner or some custom JavaScript code, you can persuade Stylus to build, compress and upload your project as you edit.

Lists and pattern matching

You're not limited to simple variables or constants. If you use a templating language, you can pass lists to Stylus and then you can have it generate CSS for each item. There are also true or false tests that are available for string searches and pattern matching as well.

CONS

Complexity

Stylus is very powerful, but it's not ideal for beginners. If you're comfortable hacking together a Node project, you'll feel right at home with it. If you're not, it's hard work with a steep learning curve and some lost productivity until you get up to speed.

Momentum

...Or lack of it. Although technically Stylus may be the most powerful preprocessor, it has lost ground to Sass, which is used by so many designers it's on its way to becoming a standard. This may not be a good thing, but there's no doubt Sass is more popular and has better community support.

Mixins, addons and extra help

MORE TOOLS FOR A BETTER WORKFLOW

csspre

csspre.com

This is an essential overview site. It has a feature grid, a list of online and offline compilers and IDEs, and a further list of conversion tools. Short on details, but is a simple must-read.

Compass

compass-style.org

A complete CSS authoring framework, Compass takes Sass and runs with it. It adds features from Ruby and better and smarter everything – including advanced typography, do-everything resets, various helper functions and CSS sprites.

Bourbon

bourbon.io

Bourbon is a simple but powerful mixin library for Sass. It's lighter than Compass but easier to learn and just as useful for real projects. The docs are unusually comprehensive, and if you're using Sass you'll certainly want to look at it.

Lesshat

lesshat.madebysource.com

Lesshat is a worthy mixin, with stand-out support for keyframes, gradients, fonts, transitions and more. The documents available on GitHub are pretty good too.

nib for Stylus

nibstyl.us

A small library of ready-made CSS3 mixins for Stylus. Stylus has a lot of mixin goodness already, but adding nib makes it easier to write CSS3 one-liners for gradients, transparency, responsive images, border and size control and more.

Top Tip

If you have a standard workflow, use a popular IDE for compiling CSS. They'll do minification, gluing together, and other preflight jobs, without the hassle of custom code.

Client vs server, which is best?

THE OPTIMUM WAY TO COMPILE CODE AND SERVE THE FILES YOU NEED

Preprocessor code doesn't have to be precompiled. You can include a preprocessor as a script in your main site code. The usual trick is to put your preprocessor files into a special directory, add a stylesheet directive to tell the browser which preprocessor to use, and include the preprocessor itself as a JavaScript file.

Will this work? Yes. Should you do it? No. Or at least, not for production code. It doesn't take a lot of thought to understand that you're serving files you don't need, weighing your server down with extra work and slowing down page loads.

Compiling CSS can take a while. If you force the compilation, you're forcing users to wait for every page that uses a preprocessor script.

Worst case is this can add whole seconds to the load times. That's very bad, and certainly something you want to avoid.

There's also the CORS – Cross Origin Resource Sharing – problem. On a simple site, all the resources will be local. On a more complex site they may be served from different locations, and site loads can go horribly wrong.

As a rule, it's much smarter to precompile your code to vanilla CSS and minify it too to save download times. You get snappier page loads and happier visitors. The only extra cost is a compile or upload stage when you make changes. But if you have an automated workflow set up, that shouldn't cost you much extra time or effort.

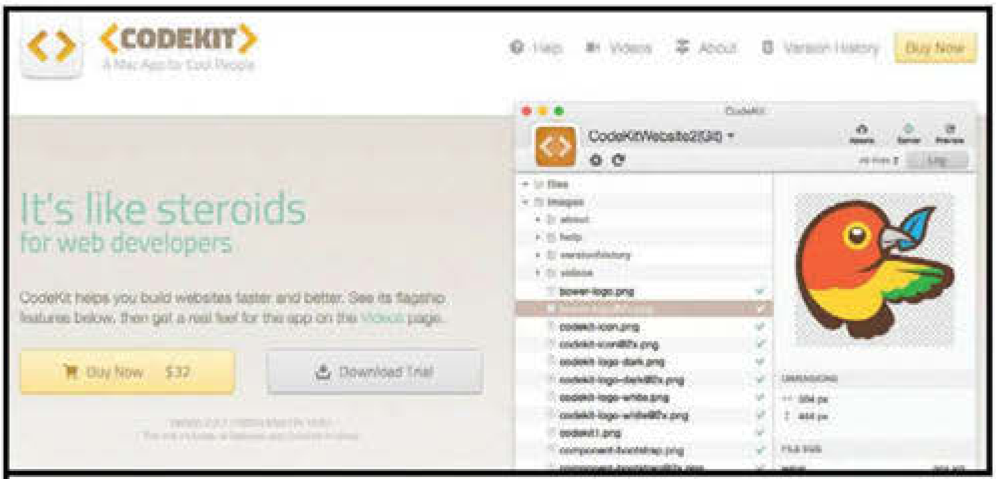
It's fine to use preprocessor scripts for testing and development. It saves you time while you're making changes because you can skip the compilation stage and leave it to the browser. But production code should be precompiled. Always.

Top Tip

Remember – your goal is to create superb, silky smooth and stylish websites. A preprocessor will save time, but you'll get a bigger benefit with the right framework.

Best compilers and IDEs

PREPROCESS WITHOUT PAIN WITH THESE COMMAND LINE-AVOIDING IDES AND COMPILERS



It's like steroids for web developers

CodeKit helps you build websites faster and better. See its flagship features below, then get a real feel for the app on the Videos page.

CodeKit incident57.com/codekit

For the not-so-epic price of $32 – just over £21 – CodeKit gives you a tool that compiles everything from an editor window. LESS, Sass and Stylus are just the start. It also handles CoffeeScript, TypeScript, JavaScript, Slim, various HTML preprocessors including Haml and Jade, oh – and Markdown.

And there's more! The Foundation web framework, Bourbon, and Compass are built in, as is Autoprefixer. So you get to use all of that mixin goodness with almost no effort.

And the Sass compiler is the superfast libSass, which means you won't have to wait around for vanilla CSS to come out the other end.

If you want a good JavaScript editor, CodeKit automatically checks your JS for errors, so you won't spend hours hunting down missing semicolons. If you have no idea what most of these do, you can ignore them. If you do, you'll love the fact that you get everything in one place, with no need to drop down to the command line or set up your own task runner. Minification and file joining are also available, so you can check distribution sizes before you ship. Lossless image optimisation is also built in for JPGs and PNGs.

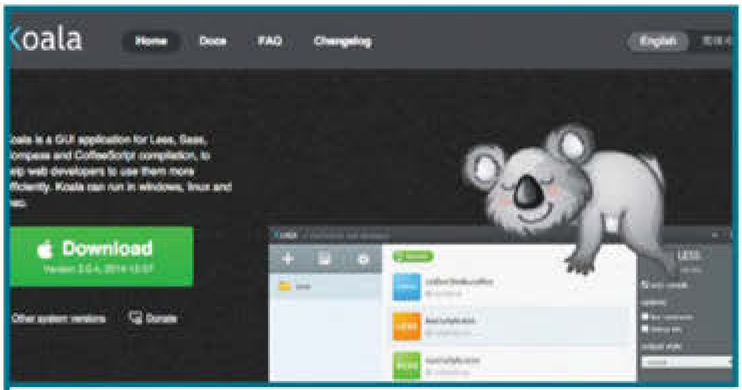
It even includes a browser update feature, so changes automatically update in your development browser windows. The Bower component library is also included so you can download and install useful frameworks with a few mouse clicks.

The downside? It's Mac only. Windows and Linux people are out of luck.

Run tasks with Gulp and Grunt

GETTING THE MOST OUT OF AUTOMATION

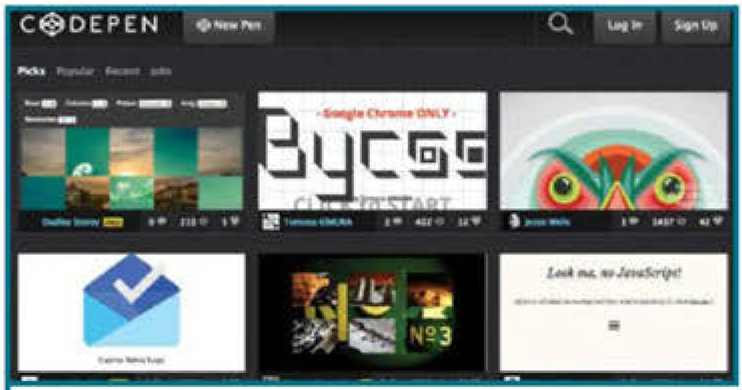
The hard way to use a preprocessor is to do everything from the command line, with hand-typed commands. The smarter way is to use some form of automation to precompile your preprocessed styles into vanilla CSS. Unfortunately automation is more complicated than it should be. If you're using Ruby on Rails, you can roll your own code. For Node.js users, the two tools of choice are Grunt (**gruntjs.com**) and



oala is a GUI application for Less, Sass, ompass and CoffeeScript compilation, to elp web developers to use them more fficiently. Koala can run in windows, linux and ac.

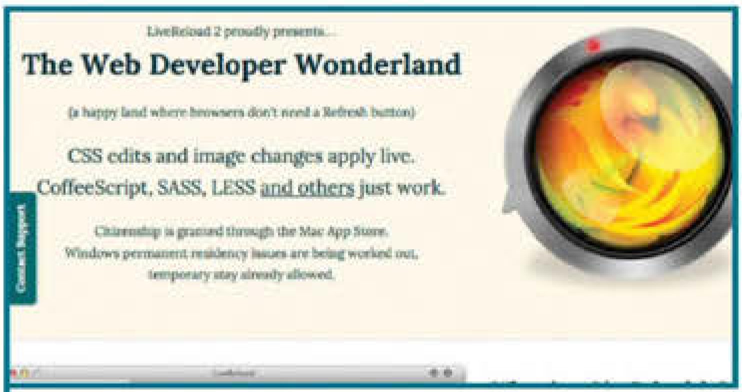
Koala koala-app.com

A freebie take on CodeKit, Koala works on Linux and Windows as well as on Mac. It can handle LESS and Sass (with Bourbon) but there's no Stylus support. So it's not as sophisticated as a commercial IDE, but it is cross-platform. And did we mention that it's also free?



CodePen codepen.io

For a different take on preprocessors, try Codepen's live code compiler. It's an easy to use but full-featured web designer's playpen. Choose between Sass, SCSS, LESS and Stylus, and select various mixins and addons using menus. Experiment with prefix-free syntax or add Autoprefixer support.



LiveReload 2 proudly presents...

The Web Developer Wonderland

(a happy land where browsers don't need a Refresh button)

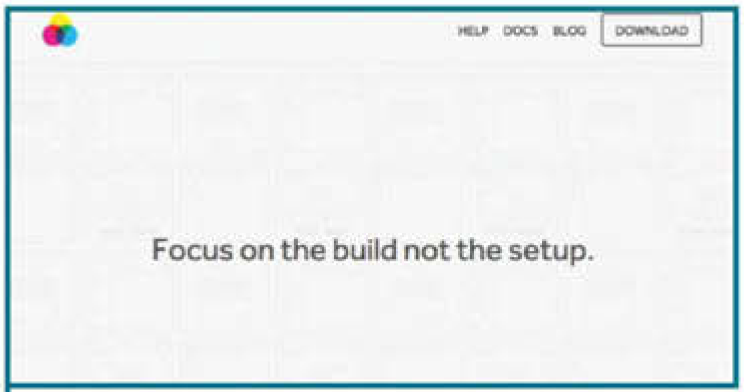
CSS edits and image changes apply live.
CoffeeScript, SASS, LESS and others just work.

Citizenship is granted through the Mac App Store.
Windows permanent residency issues are being worked out,
temporary stay already allowed.

LiveReload livereload.com

Another compiler-of-compilers, LiveReload offers Sass, LESS, Compass, Stylus, and a slew of JS and HTML preprocessors – all-in-one handly package. The price? It's $9.99 (£7.99) in the app store for OS X. But the source is free on GitHub, so if you have a copy of Xcode and building a project from scratch holds no fears for you, you can make your own version – although if you do you should still pay for a licence anyway, just because.

Should you use it? It's maybe a touch busier and less straightforward than the competition, but if you're a code geek then you'll find it easier to customise (if that's what you need). It's certainly worth a look, anyway.



Focus on the build not the setup.

Mixture mixture.io

Mixture used to be a commercial product, but now it's been released for free. It's a capable editor, template tool and preprocessor for Mac and Windows. It covers all your preprocessing needs with Sass, Compass, LESS and Stylus. You can even combine code written for different preprocessors in a single project by using a special Mix mode.

There's also support for Autoprefixing, script linting, debugging, minification and concatenation. This is not at all bad for a permafree download, and certainly well worth a look so you can compare to it the paid-for competition and maybe save yourself a little cash.

Gulp (**gulpjs.com**). Both are task runners which can package up a collection of regular tasks you need to do into a mini-app you can run with a single command.

That means you can lint/hint, preprocess, glue together, minify, and maybe upload all your code with a single command. The catch is that getting to the point where the command works for you will take a while – a day or so if you know how to code, or a week or so if you're completely new to coding. But from then on, you can then use automation on all future projects – which means real timesaving.

Is Grunt better than Gulp, or vice versa? They're both useable, but they work a little differently. Some people get on better with Gulp's syntax (instruction code), other people prefer Grunt's.

There isn't room here for a complete tutorial, but there are plenty of beginner resources online – like **24ways.org/2013/grunt-is-not-weird-and-hard** for Grunt and **travismaynard.com/writing/getting-started-with-gulp** for Gulp.

And what about IDEs? There are a million and one – well, ten or so – online and offline editors and compilation tools that can take preprocessor code and turn it into plain CSS. They don't usually include full automation but they'll skip that time-wasting command-line stage, which is always a good thing.

Preprocessing in action

WANT TO SEE HOW PREPROCESSING WORKS IN THE REAL WORLD? TAKE A LOOK AT THESE EXAMPLES

Nesting code with Sass

SASS

```
a.myAnchor {
 color: blue;
 &:hover {
 text-decoration: underline;
 }
 &:visited {
 color: purple;
 }
}
```

With Sass you can nest items to group them together and make them easier to read. The magic & directive references the parent style

CSS

```
a.myAnchor {
 color: blue;
}
a.myAnchor:hover {
 text-decoration: underline;
}
a.myAnchor:visited {
 color: purple;
}
```

Here's the compiled CSS. See how Sass expands the nest and fills in the full references to the parent, but the code keeps all related elements in one place?

SASS

```
.first-component {
 $font-base: 1.4rem;
 .text { font-size: $font-base; }
 .button { font-size: $font-
base+0.3rem; }
 @at-root .second-component {
 .text { font-size: $font-base
0.2rem; }
 .button { font-size: $font-base;
}
 }
}
```

But what if you want to pull a style out of a nest so you can use it elsewhere? Here's some code to define some related text sizes

CSS

```
.first-component .text {
 font-size: 1.4rem;
}
.first-component .button {
 font-size: 1.7rem;
}
.second-component .text {
 font-size: 1.2rem;
}
.second-component .button {
 font-size: 1.4rem;
}
```

In the compiled code the @at-root directive flattens the nest so you can use . second-component elsewhere as it's not a subclass of first-component

CSS Tricks

@Real_CSS_Tricks

This is mostly autoposted content, but it's all design-relevant, and there are occasional gems of CSS wisdom to be found here.

CSS Preprocessors

@csspre

The clue is in the name – essential information and news for anyone who uses or is interested in CSS preprocessors.

Mark Otto

@mdo

CSS and preprocessors tweets from one of the creators of Bootstrap, with regular posts and news on future Bootstrap updates.

Harry Roberts

@csswizardy

Slightly technical industry tips and occasional tricks from this respected consultant CSS expert, author, conference speaker and top designer.

Sass

@SassCSS

Official Twitter feed of the Sass team – this is absolutely essential reading for regular Sass users who want to keep on top of Sass news.

Using inheritance with Sass

SASS

```
.message
  border: 1px solid #ccc
  padding: 20px
  color: #444
```

First, make a generic style. Note how with Sass you don't have to include curly brackets or semicolons – although you do need to get your indentations right

SASS

```
.ok-message
 @extend .message
 border-color: green
.error-message
 @extend .message
 border-color: red
```

Now you can use @extend to change some of the properties for special cases. These examples keep most of the original style, but override the colour

CSS

```
.message, .ok-message,
.error-message {
 border: 1px solid
#cccccc;
 padding: 20px;
 color: #444;
}
.ok-message {
 border-color: green;
}
.error-message {
 border-color: red;
}
```

Here's the compiled CSS. Sass puts back the curly brackets and semicolons, and it groups related items together, while still pulling out the override styles

Custom mixins with Stylus

STYLUS

```
border-radius(val)
  -webkit-border-radius:
val
  -moz-border-radius: val
  border-radius: val
  button
  border-radius(5px);
```

The first part of the code defines a custom mixin and autoprefixer. Val just means 'plug something in here'. The second part uses the mixin to define a button

CSS

```
button {
  -webkit-border-radius:
5px;
  -moz-border-radius: 5px;
  border-radius: 5px;
}
```

Here's the compiled CSS. The "5px" is plugged into the mixin. You can reuse the mixin over and over, with whatever values or symbols you want to plug into it

PostCSS and the future of preprocessing

MAKE YOUR OWN WITH PLUGINS

PostCSS (github.com/postcss) is a processor that can do what you want it to. Its a combination of custom JavaScript and ready-made plug-in code for each feature you want to add is great in theory. It's also part of the Autoprefixer autoprefix tool (**github.com/postcss/autoprefixer**). Writing a custom preprocessor can be hard, but as plugins increase in number, PostCSS will get more useful.

Resources

THREE EXTRA TOOLS AND PREPROCESSING ALTERNATIVES

Chrome DevTools

developer.chrome.com/devtools/docs/css-preprocessors

It can be hard to debug preprocessor code, but DevTools makes it easy by including links to the original preprocessor source file and not the generated CSS file.

Myth

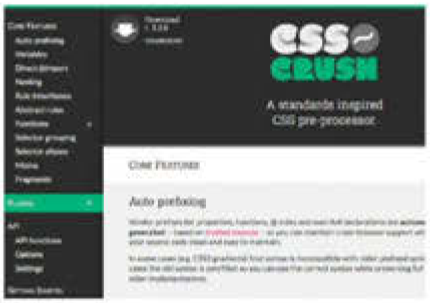
myth.io

A simple but interesting alternative preprocessor with built-in future-proofing. Myth is simpler than the big three but worth a look anyway if you want simple, fast preprocessing.

CSS Crush

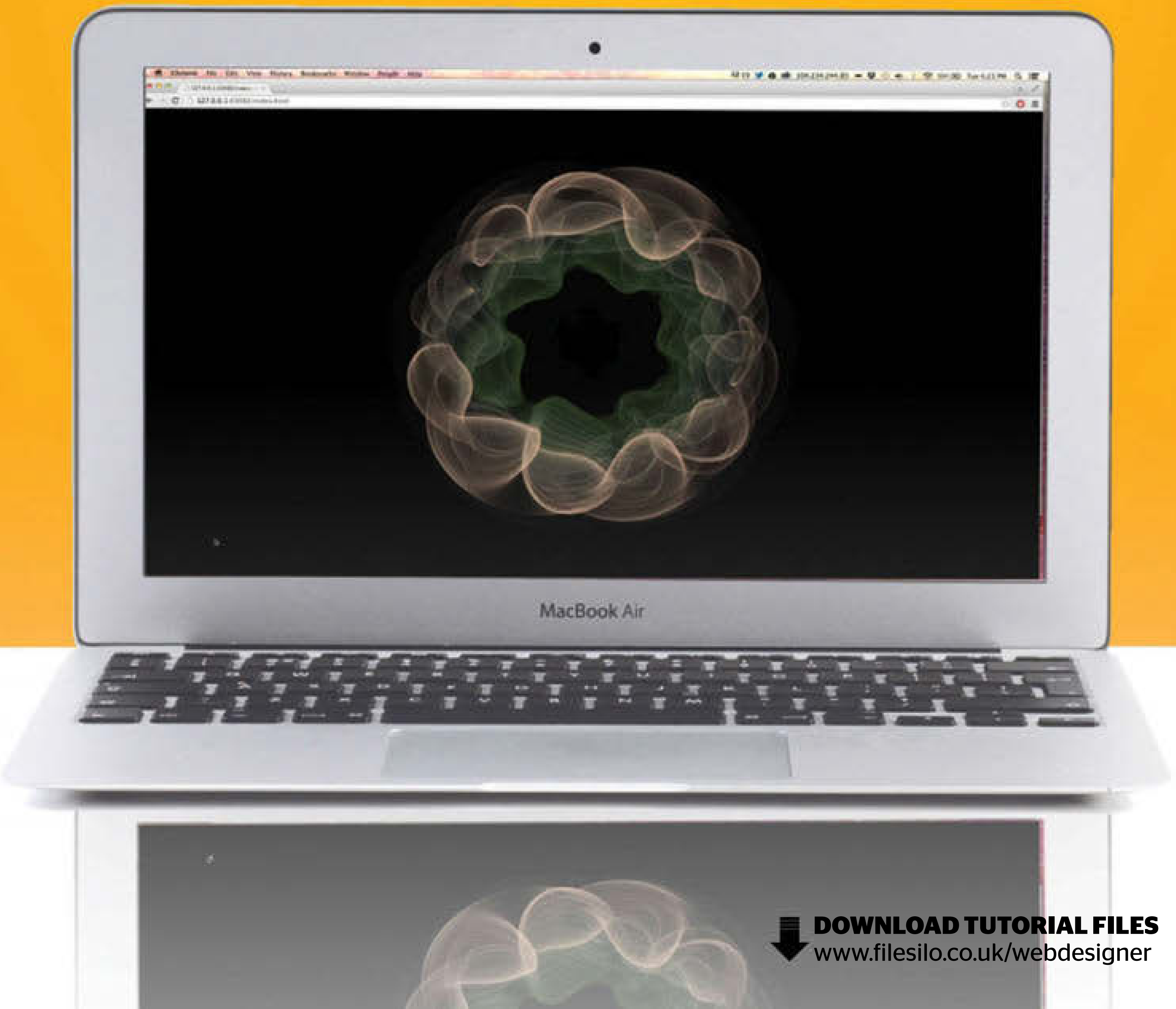
the-echoplex.net/csscrush

A standards-inspired CSS preprocessor written in PHP. You'll want this if you're a PHP shop, and Node or Ruby are of no interest. Also does minification, which is cool.



Make dynamic graphics with the p5.js library

Create interactive drawings with HTML5 Canvas and p5.js, a JavaScript port of the popular open source Processing library



DOWNLOAD TUTORIAL FILES
www.filesilo.co.uk/webdesigner

Since the early 2000s, the Processing library has been used by designers and artists as a creative coding environment to create stunning visuals with very few lines of code. The Processing library is actually a Java library and this is useful for creating interactive installations and prints. The p5.js project is a new initiative to bring the ease of Processing to the web by enabling designers and developers the same easy controls to draw to the Canvas element in HTML5. This means that live, interactive visuals can easily be created for just about any HTML5 capable browser, which is almost any browser from the last three or four years.

The screenshots for this project do not do it justice as the project has to be interacted with in order to see it work properly. We are going to be creating an organic shape that is drawn pretty similar to an old spirograph, but it's going to morph, change size, shape and colour based on the user's mouse input. As the user moves their mouse, the shape ripples and changes to create a stunning interactive shape-shifting visual. With a little manipulation this could easily become a music visualiser!

1. Start the project

Open up your web browser, type in **p5js.org,** click Download and then choose Complete Library. Once it has finished downloading, open the folder 'empty-example' and copy the file p5.min.js into it. Open index.html in your code editor and change the script file to use the minified version.

```
<script language="javascript" type="text/
javascript" src="p5.min.js"></script>
```

2. Open sketch.js

After installation, the app is usually extracted. The main code for our project will be found in sketch.js – this is a bare-bones template ready for us to start with the coding process. Open this up in your code editor and you will see that there are two functions here. The first function, setup, is used to set all the things at the start of the project like background colour and size. The draw function is called every frame.

```
function setup() {
// put setup code here
}
function draw() {
// put drawing code here
}
```

3. Edit the setup function

In the setup function add the code that is presented below. This creates a new Canvas element to draw into and set this up to be the current width and height of the browser window. The background of the Canvas is set to black, and the Canvas itself is set to smooth, or antialias. This will ensure that the lines are crisp but not jagged at the edges.

```
createCanvas(windowWidth, windowHeight);
background(0);
smooth();
```

4. A new function

In between the setup and draw function, add a new function as we have shown in this step. This code is called to draw an ellipse on the screen and we will be drawing a large number of ellipses later using this. The fill colour is turned off and the stroke is white but with a low opacity that is just over 10 per cent visible. The ellipse is drawn from 0, 0 pixels to 120, and 80 pixels on the x and y axis respectively.

```
function drawEllipse() {
noFill();
```

```
stroke(255, 255, 255, 28);
ellipse(0, 0, 120, 80);
}
```

5. Start the draw function

Now inside the draw function, add the first line in the code below, which is the same as in the setup, for making the background a black colour. The difference here is that it is called every frame and therefore removes any drawing from the previous frame so that we end up with a new drawing every frame. The translate line moves the drawing point to the middle of the browser window.

```
background(0);
translate(windowWidth/2, windowHeight/2);
```

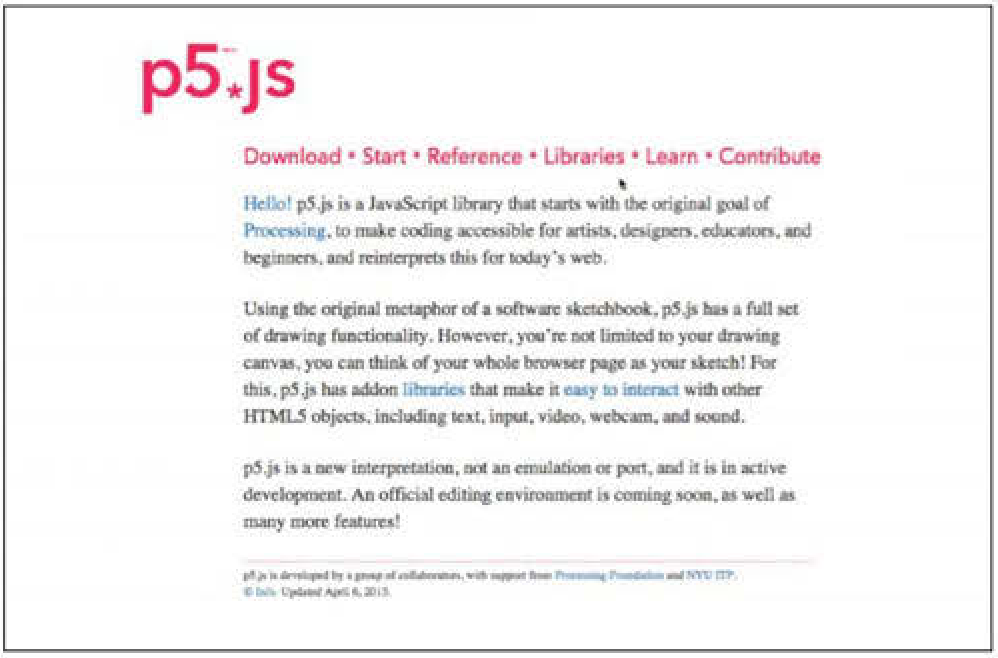
6. Draw around a circle

A 'for' loop is added here to create 720 ellipses. The 'for' loop increases by 0.5 every loop so it will draw 720 instances. The push line enables the drawing to rotate around the middle of the screen. The drawing position is then pushed out 200 pixels on the y axis.

```
for (var i=0; i<360; i += 0.5) {
push();
rotate(radians(i));
translate(0, 200);
```

Adding colour in p5.js

Colour in p5.js follows a pattern of 0-255 to give 256 colours for each colour channel and this in turn will give around 16 million different colours! To get random colours, a random number in that scale can be created.



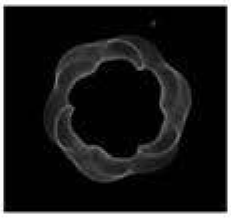
p5.js
Download • Start • Reference • Libraries • Learn • Contribute

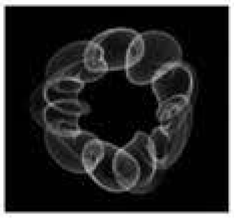
Hello! p5.js is a JavaScript library that starts with the original goal of Processing, to make coding accessible for artists, designers, educators, and beginners, and reinterprets this for today's web.

Using the original metaphor of a software sketchbook, p5.js has a full set of drawing functionality. However, you're not limited to your drawing canvas, you can think of your whole browser page as your sketch! For this, p5.js has addon libraries that make it easy to interact with other HTML5 objects, including text, input, video, webcam, and sound.

p5.js is a new interpretation, not an emulation or port, and it is in active development. An official editing environment is coming soon, as well as many more features!

p5.js is developed by a group of collaborators, with support from Processing Foundation and NYU ITP.
© Info. Updated April 6, 2015.





Left
The p5.js library can be downloaded from **p5js.org** where all documentation can also be found as well as some very useful example files to introduce you to some concepts. Also look out for the p5.js editor, available on Mac OS X and coming soon to Windows and Linux

Top left
Once the core code is in place it can be run and you will see a circular pattern appear on the screen. Next we will make this modulate and change based on mouse position so that it is interactive with the user

Top right
By adding some minimum and maximum values to the waves undulating around the shape, we can start to affect the interaction of the graphic in much more detail than we were able to previously

7. Rotate again

Now as the drawing point is 200 pixels out from the centre of the screen, p5 will draw an orbit around that point but first it will rotate again here. The whole drawing will be scaled but it will also be based on a sine wave that modulates between 1 and – 1 so that the result is a very organic shape.

```
rotate(radians(i*3));
scale(map(sin(radians(i*6)),   1, 1, 0.5,
1), map(sin(radians(i*3)),   1, 1, 0.5,
1));
```

8. Draw the ellipse

Now as the drawing point is at the right place, the ellipse is drawn and the pop command places the drawing point back to the centre of the document. Save the file and view the index.html page in the browser to see the drawing of the organic shape in action. Now we are going to change the shape interactively based on the mouse movement of the user.

```
drawEllipse();
pop();
}
```

9. Add some variables

In order to add interactivity to the screen, we will need some variables to dynamically control some of the numbers that draw the shape. Here the variables are added just above the setup function so that we can call them from any function that we want. The names are fairly self-explanatory, and you will also see them in action in the coming steps.

```
var scaleX;
var scaleY;
var min;
var max;
var radius;
var r, g, b;
```

10. Scale based on position

In the draw function add the following code at the top, and this code will set the values of scaleX to take the mouse's x position. As it moves 0 pixels to the right-hand side of the screen, those values will be mapped onto a range of values from 1.5 to 11.5. The same thing will then happen with the y position of the mouse and the window height as well.

```
scaleX = map(mouseX, 0, windowWidth, 1.5,
11.5);
scaleY = map(mouseY, 0, windowHeight, 1.5,
11.5);
```

11. Amend the 'for' scale

Look inside the 'for' loop, further down in the draw function. Here you will see a line of code beginning with 'scale'. Change the numbers from 6 and 3 to scaleX and scaleY as we have shown here. This means that the code will now take the mouse position for how the shape will be drawn. Save this now and try it out in your browser to see it in action.

```
scale(map(sin(radians(i*scaleX)),   1, 1,
0.5, 1), map(sin(radians(i*scaleY)),   1, 1,
0.5, 1));
```

12. Map other values

Just below the code that was added in Step 10, add the code that we have included below. It looks very similar to our other pieces of code, but what's different is that this time the position of the mouse on the x axis is mapped onto a range of numbers from just 0.1 to 0.5 and the y mouse is mapped onto 0.8 to 1.8. These positions will provide some subtle differences in the drawing when they are applied in the next step.

```
min = map(mouseX, 0, windowWidth, 0.1,
0.5);
max = map(mouseY, 0, windowHeight, 0.8,
1.8);
```

13. Scale again

Just like we did in Step 12, locate the 'scale' line of code inside the 'for' loop. Here the min and max variables are added. Save this and view it in the browser to see how it adjusts the drawing on the screen. Remember that these numbers will be adjusted as the mouse moves around on the screen.

```
scale(map(sin(radians(i*scaleX)),   1, 1,
min, max), map(sin(radians(i*scaleY)),   1,
1, min, max));
```

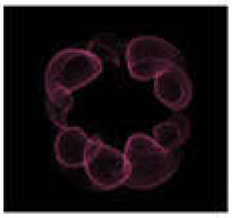
14. Change colours

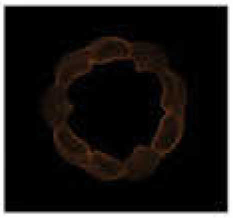
In much the same way as the shape of our graphic can be altered based on where the mouse is, as we have shown in the previous steps, it is also possible to dynamically change the colour. In the setup function add these random numbers for the red, green and blue values. The colour values are measured from 0 (for darkest) to 255 (the brightest).

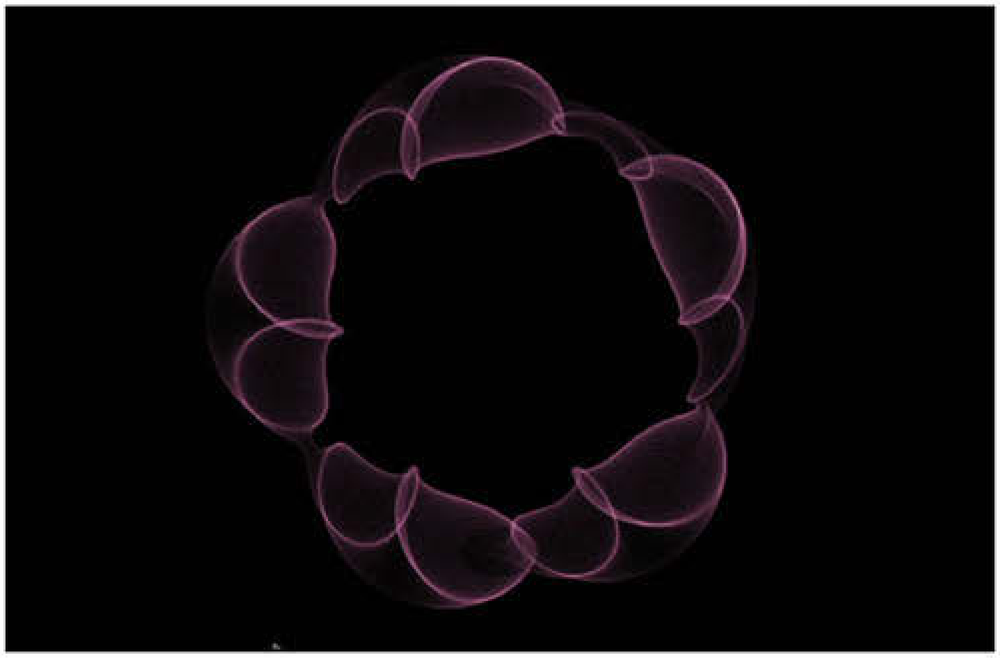
```
r = random(255);
g = random(255);
b = random(255);
```

The nature of code

Because p5.js is a port of Processing, most Processing code can quickly be adapted. Check out Dan Shiffman's eBook *The Nature of Code* (**natureofcode.com/book**) for more information.



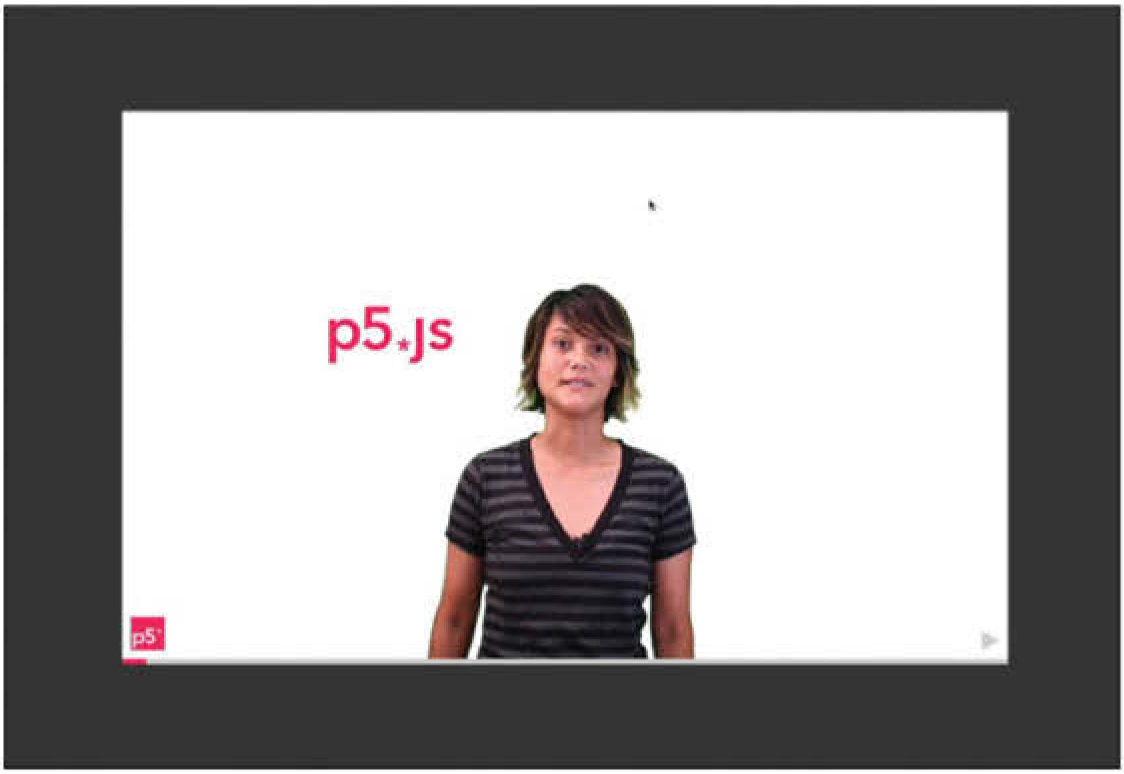




Top left
A random colour is generated each time the browser is refreshed, this works well at setting an initial colour but what we really want to do is give the user a little more control over the colour that appears rather than giving them sporadic choices

Top right
The colour is now adjusted so the height of the mouse in each third of the screen controls either red, green or blue and this provides a way of adding slightly more control over the colour on the screen

Right
Now the radius of the overall shape has been changed so that it is affected by the mouse position on the y axis. The top of the screen makes the object smaller while the bottom of the screen makes it larger



Getting to know hello p5.js

Starting with a new library can take a little getting used to and a lot of programming languages seem to start with a simple 'hello world'. Taking this idea further, p5.js created a site called 'hello p5.js' (**hello.p5js.org**), which introduces the basic ideas in a very simple interactive video format. Live code can be seen running alongside the video as the presenters introduce the video. The video itself even has its green screen around the presenters removed live by code in the browser so that the power of what is available through p5.js can be clearly demonstrated. In the video you will see how a particle system is created, a flocking behaviour is added to it and then live windspeed data from New York affects the flocking!

15. Use the colour

In order to get those random colours working add the letters r, g and b to the stroke line which is located in the drawEllipse function. This means that when it draws the ellipse it will also make use of the random values. Save this now and test it in the browser to see how it works and looks. Each time you refresh the page a new colour will be used.

```
stroke(r, g, b, 28);
```

16. Colourful mouse moves

Having to refresh the screen to change the mouse colour is okay, but we may also want to update the colour based on the screen position of the mouse. This colour effect is possible via an 'if' statement as we have shown in the code below, which divides the screen into three horizontal strips for RGB colour. In this step we have started off with the first third and mapped it to the red, giving it a random colour.

```
function mouseMoved() {
if (mouseX < (windowWidth/3)){
r = random(255);
//r = map(mouseY, 0, windowHeight, 0, 255);
}
```

17. Green and blue changes

Continuing on from what we've learned in the previous step, we can use the 'if' statement to take the last third of the screen and map that to a random blue colour. Then, we can also do the same for the green in the middle. Save all of this now and test it in the browser to see all of the random colours as the mouse moves around, all over the screen. Keep in mind though that each third will affect a different colour.

```
else if (mouseX > ((windowWidth/3)-
windowWidth)){
b = random(255);
//b = map(mouseY, 0, windowHeight, 0, 255);
} else {
g = random(255);
//g = map(mouseY, 0, windowHeight, 0, 255);
}
}
```

18. Slight change to the colour

The random colour changes have worked well so far, but now we can specify that the colour now changes specifically based on how far down the page the mouse is. To do that we will need to revisit the 'if' statements that we have already used in Steps 16 and 17. First, comment out the random line and uncomment the line afterwards. Then, save this and test this change in the browser to see it all working.

```
//r = random(255);
r = map(mouseY, 0, windowHeight, 0, 255);
```

19. Change the drawing radius

Now we go back to the shape itself, it is possible to change how large the circle is drawn around the centre point. In the setup function add the radius value as shown here. This started off the project with an initial value of 150 pixels as the radius from the centre of the screen, but this can now be updated.

```
radius = 150;
```

20. Update the radius

Now in the mouseMoved function, add the following line just before the closing bracket of the function. The y position of the mouse will decide how large the radius of the circle will be. Note that this will never be smaller than 100 pixels and never larger than 350 pixels, but you can change these values if you want them to be slightly larger or smaller.

```
radius = map(mouseY, 0, windowHeight, 100,
350);
```

21. Set the radius

In order for these radius values to actually be used, look inside the draw function and inside the 'for' loop is a translate line. Update the value of 200 to the radius variable. Now you can save the file and then refresh in the browser to see this in action. There will be a lot of changes now to the shape, size and colour.

```
translate(0, radius);
```

22. Fade out

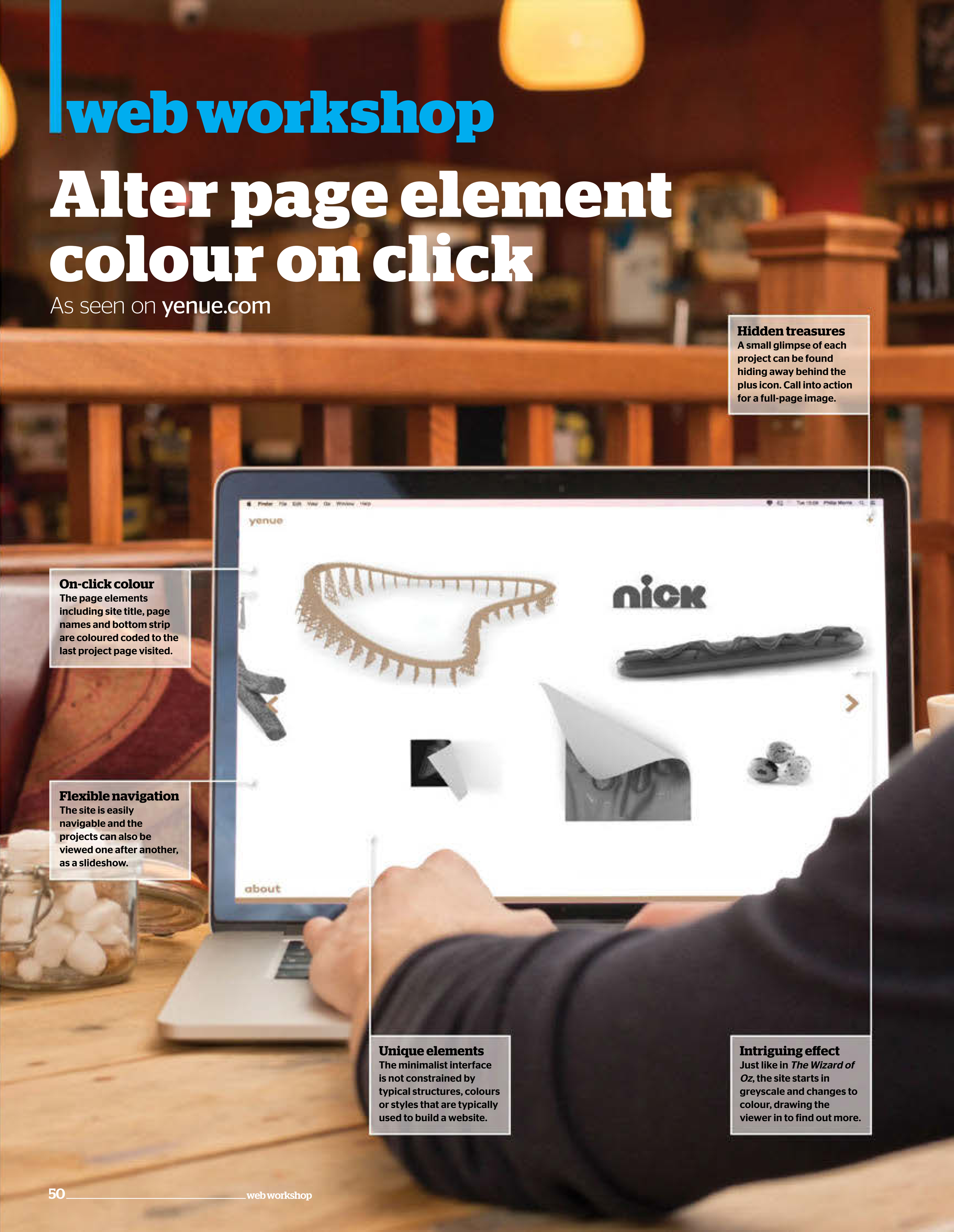
Finally it is possible to fade out the drawing from all of the previous frames. In the 'draw' function, comment out the background colour. So now we want to set the fill colour to black with a low opacity of around 10 per cent. Then, a rectangle is drawn over previous frames with the low opacity on top and this will cause the previous frames to fade out. Save and test this now to see the final effects in all their glory.

```
background(0);
fill(0, 25);
rect(0, 0, windowWidth, windowHeight);
```

web workshop

Alter page element colour on click

As seen on **yenue.com**



Hidden treasures
A small glimpse of each project can be found hiding away behind the plus icon. Call into action for a full-page image.

On-click colour
The page elements including site title, page names and bottom strip are coloured coded to the last project page visited.

Flexible navigation
The site is easily navigable and the projects can also be viewed one after another, as a slideshow.

Unique elements
The minimalist interface is not constrained by typical structures, colours or styles that are typically used to build a website.

Intriguing effect
Just like in *The Wizard of Oz*, the site starts in greyscale and changes to colour, drawing the viewer in to find out more.

Alter page element colour on click

DOWNLOAD TUTORIAL FILES www.filesilo.co.uk/webdesigner

This workshop explores just a small detail of London based-artist Yenue's portfolio site that may even have been overlooked by the casual visitor. It's a subtle technique that changes the hover colour of the elements on the homepage to the background colour of the previously visited page. It neatly speaks to an important artistic concept, the memory and experience of viewing a piece of art. Yenue's website of surreal and sometimes Dali-esque work inspires contemplation on the meaning and presentation of art, and, through this small detail of colour change, we have an opportunity to learn a range of design techniques that actually have much broader applications. Yenue says: "I tailored my website to enhance the connections between the index and the individual projects. Each project is unique, so each background is customised to suit each project. I continued to find ways to connect different sections with small design features, such as the index border overlapping with the border of the project as you switch between them; the overlapping border is still visible. The colours of the index page are impacted by the design you choose to view. I felt that all of these features enabled me to connect my projects and overall website."

<comment>
What our experts think of the site

Mi casa es su casa

"My portfolio website is another design project for me, where I show all the projects in which I have invested a large portion of my time. It is like my home on the internet and so I wanted to create something very special and visual for anyone visiting."

Yenue

Technique

1. Head first

SVGs are popular for their flexibility but to take full advantage of this, it needs to look inline. You could add graphics to HTML but Drew Baker's (**bit.ly/1ciz3Cc**) SVG inliner is a great solution. Dem Pilafian's 'Two Line Style Switcher' (**bit.ly/1CxTh1O**) provides a neat and effective method for changing a colour scheme as required.

```
<html>
<head>
<title>Change colour on hover</title>
<script src="http://ajax.googleapis.com/ajax/
libs/jquery/1.6.2/jquery.min.js"></script>
<script type="text/javascript" src="scripts/
svginliner.js"></script>
<script>
var styleFile = "style" + document.cookie.
charAt(6) + ".css";
document.writeln('<link rel="stylesheet"
type="text/css" href="styles/' + styleFile +
'">');
</script>
```

2. CSS in the HTML

To keep things simple, the styling that is specific to each page that does not want to be switched has just been added direct to the HTML. Centring elements in CSS can be tricky but if you follow through each divw and experiment with changing the settings that should help make things clearer. If you haven't come across the 'helper' class before then that is worth exploring.

```
<style>
.helper {
display: inline-block;
height: 100%;
vertical-align: middle;
}
```

3. Set the stylesheet

Immediately before this, our helper will enable vertical centring. A JavaScript call sets a cookie that the JavaScript at the top uses to set the stylesheet. Note the attributes of the image that are used by the SVG inliner.

```
<span class="helper"></span><a
href="javascript: document.cookie='style=2';
window.open('stpauls.html','_self');"><img
id="stpauls-icon" class="svg stpauls"
src="images/stpauls.svg"></a>
```

4. Style switch

In this solution the style is switched on when exiting the homepage, and it could be argued that it's only necessary to include the switcher code in one file here. The portfolio pages are similar to the homepage but the background colour is changed to the new theme colour.

```
<script>
#royalalberthall-col {
float:left;
width:100%;
position:relative;
left:0%;
overflow:hidden;
text-align:center;
background-color:#baa5a0;
}
```

5. Theme styles

It's necessary to follow the naming system of style.css, style2.css and so on. In this solution each element is coloured the same but the technique provides plenty of flexibility to switch the styling of any page element.

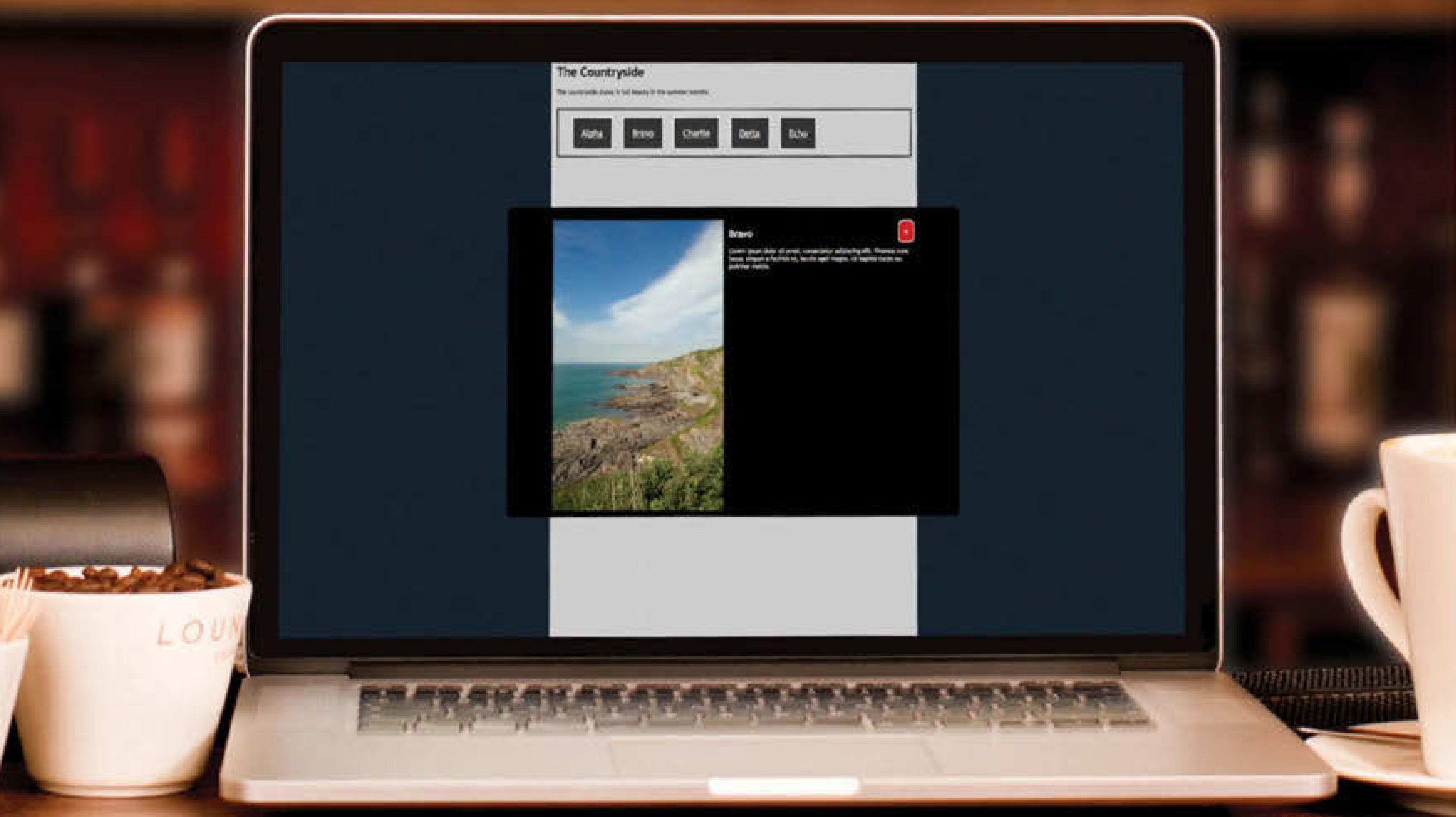
```
#stpauls-icon:hover path{fill:#9d7554}
#housesofparliament-icon:hover
path{fill:#9d7554}
#royalalberthall-icon:hover path{fill:#9d7554}
#toweroflondon-icon:hover path{fill:#9d7554}
```

EXPERT ADVICE
Vision and technique

Make sure you also check out **facebook.com/yenue.studio**, where Yenue showcases other ongoing work. The typographical plasticine creations are especially interesting and his projects demonstrate original thinking and first-class execution – something that designers from any field will appreciate.

Make pop-up modal boxes with pure CSS

Show new information in a modal box along with animation effects without loading a new page



DOWNLOAD TUTORIAL FILES
www.filesilo.co.uk/webdesigner

Good design is always about presenting information in a way that is simple to understand. And yet, sometimes the amount of information causes a conflict with the need to present information in a way that is easy to comprehend and where the information is still complete. One option would be to separate content onto different pages, but this causes scope for unnecessary page loading and poor usability, especially where users are accessing the webpage through a slow internet connection such as via mobile data with poor reception.

Modal boxes offer a perfect solution to the problem by enabling additional information to appear on the page when they are clicked without the need to load a new page. Furthermore, modal boxes can be closed again in order to return the user to the original content without the need to reload.

This tutorial looks at several options for how modal boxes can be developed with CSS to provide highly useable designs for websites and web apps that are easy to adapt for different types of content you may want to present. The functionality of these modal boxes will focus on easy access in a way that enhances the user experience of the webpage.

1. Get started

First, declare the main HTML page structure, including the head and body section within the HTML page. We will need a content container made from a div element with an ID called container to control page content flow.

2. Content overview

With the main layout elements in place, we can insert the main content inside the container element. The page will have a descriptive title and introduction text to provide the main overview of the information contained on the page. This will provide information needed by the reader to identify whether they want to read further.

3. Option navigation

Next insert the available options that will provide the user with access to additional information. These options will be regular <a> links contained inside a <nav> container. Unlike regular <a> tags that use their href to link to external pages, these <a> links will link to ID components on this page using a # symbol in their href attribute preceding the ID name being linked to.

```
<nav>
<a href="#alpha">Alpha</a>
<a href="#bravo">Bravo</a>
<a href="#charlie">Charlie</a>
<a href="#delta">Delta</a>
<a href="#echo">Echo</a>
</nav>
```

4. Add modal boxes

The modal boxes will all use <article> as their container element and will have an optional attribute of data-transition to specify the type of transition they will use to appear and disappear, providing a high degree of flexibility for individual sections.

```
<article id="alpha">
</article>
<article id="bravo" data-transition="left">
</article>
<article id="charlie" data-
transition="right">
</article>
<article id="delta" data-transition="zoom">
</article>
<article id="echo" data-transition="fade">
</article>
```

5. Modal box content

Insert content into each of the <article> containers. Each modal box will need the ability to be closed, which will be triggered by an <a> tag linking to a blank ID using just # for its href. We also want to give this closing button a 'data-button="close' attribute so we can style it later.

```
<article id="bravo" data-transition="left">
<a href="#" data-button="close">x</a>
<img src="img/photo1.jpg" alt="Coastline" />
<h2>Bravo</h2>
<p>Lorem ipsum dolor sit amet.</p>
</article>
```

6. Initiate CSS styling

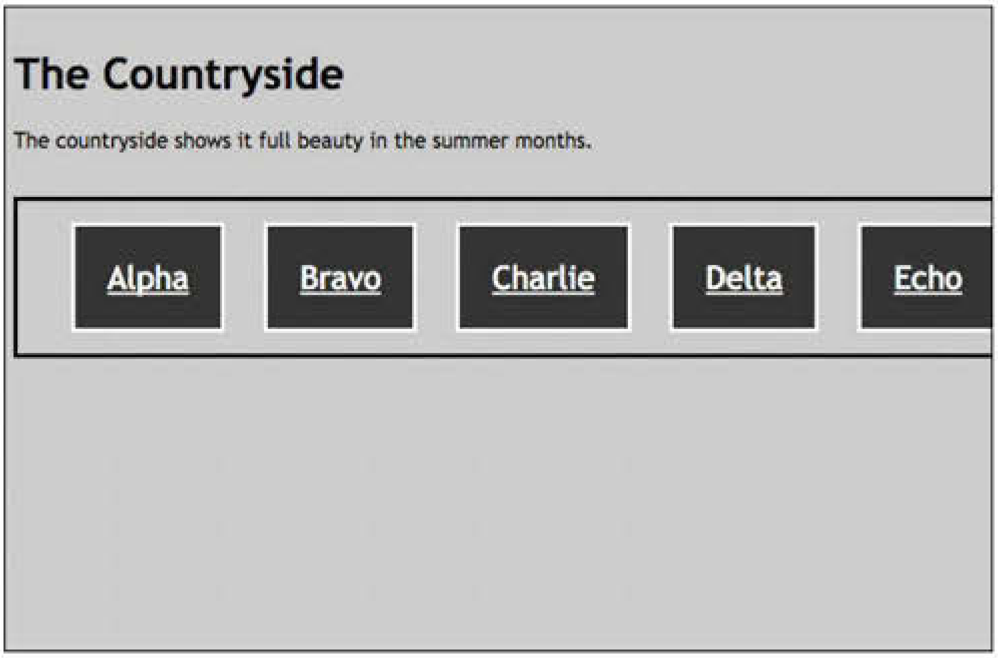
The main HTML template content is now complete, but the additional CSS stylesheet resource file is still required to add the styling. Create a text file called 'styles.css', making sure that your text editor does not add .txt as a file extension. These also need attaching from the <head> section of HTML document.

7. Style HTML body

Don't forget that the <html> and <body> elements on a webpage are set to a height of just one line by default, hence causing problems if we want to create a container that has a full webpage height. Solve this by setting the <html> and <body> elements to be full height in the CSS.

Multiple transitions

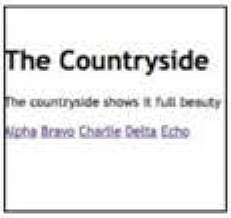
Use data attributes to enable multiple transitions to be defined that can be easily added to individual modal boxes without having to overcomplicate your CSS.



The Countryside

The countryside shows it full beauty in the summer months.

Alpha Bravo Charlie Delta Echo



The Countryside

The countryside shows it full beauty

Alpha Bravo Charlie Delta Echo

Left
The navigation container and its options are now styled to appear separate from the main content and identifiable as buttons

Top left
HTML elements in place with content added to article sections, but no styling yet

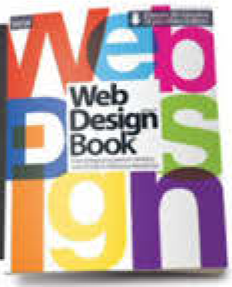
Top right
Article sections acting as the modal box containers look like this without their styling

```
html,body{
display: block;
width: 100%;
height: 100%;
margin: 0;
padding: 0;
}
```

8. Style the container

The container needs to be styled in a way so that it will stand out. This can be done by changing the main page background colour and positioning the container in the centre of the page. To do this, we need to update the <head>, <body> and container sections.

```
html,body{
font-family: "Trebuchet MS", Helvetica,
sans-serif;
background: rgb(21, 34, 47);
}
#container{
display: block;
width: 1000px;
height: 100%;
background: #ccc;
margin: 0 auto 0 auto;
padding: 1em;
}
```



9. Options container

We want the options to stand out from any standard content on the page. You can do by styling the options container that was created with the <nav> tag. This will be styled with a distinctive border and margin spacing to ensure it appears separate from surrounding content.

```
nav{
display: block;
padding: 1em;
border: 3px solid #000;
margin: 2em 0 2em 0;
}
```

10. Option styles

The option navigation elements need to appear like buttons, even though they are just <a> link tags. This can be done through CSS by making these elements display as an inline-block, which enables us to apply sizing and padding to make them appear like buttons.

```
nav   a {
display: inline-block;
padding: 1em;
margin-left: 1em;
font-size: 1.5em;
border: 3px solid #fff;
background: #333;
color: #fff;
}
```

11. Option hover

Good usability will make it easy for users to know which option they will select if they are to press on their trackpad or mouse button. We can apply the :hover property to the options so that they change colour when the user's mouse pointer is placed over them, indicating which option is about to be selected.

```
nav   a:hover{
border-color: #333;
background: #fff;
color: #333;
}
```

12. Define modal boxes

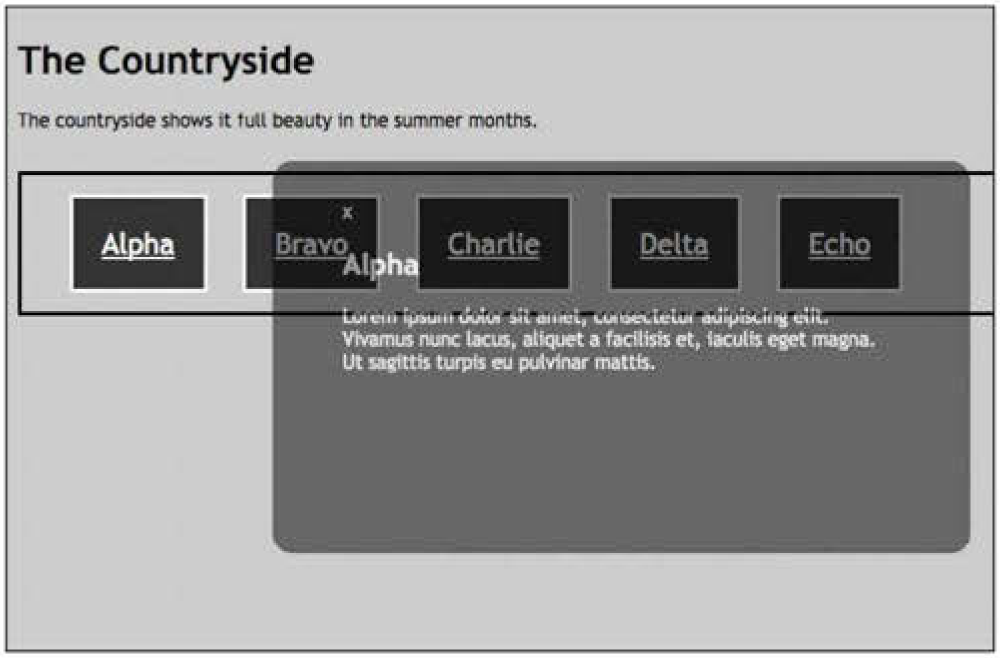
The modal boxes all use <article> elements as their containers, hence we can use CSS to define their default styles for width, height and positing. We want these elements to appear at pixel-specific locations on the screen so we use absolute positioning to allow attributes for top, left and z-index.

```
article{
display: block;
position: fixed;
z-index: 9999;
left: 25%;
top: 100%;
width: 40%;
height: 50%;
padding: 2em 5% 0 5%;
border-radius: 1em;
background: #000;
color: #fff;
overflow: auto;
}
```

13. Set transitions

Modal boxes need to be able to show transition from one CSS property state to another for the animation to be visible. To keep our CSS simple, set 'all' properties to be eligible for transition animations. You can add other properties later to trigger their animation sequence.

```
article{
padding-bottom: 1em;
```



The Countryside

The countryside shows it full beauty in the summer months.

Alpha

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus nunc lacus, aliquet a facilisis et, iaculis eget magna. Ut sagittis turpis eu pulvinar mattis.

Top left
Define the style of the modal box to make it clearly stand out from the main page content

Top right
Modal boxes can also use transitions to slide in from the side when links to their associated ID are clicked

Right
Alternatively, the modal boxes will now fade into view using transition animations

```
-webkit-transition: all 2s;
transition: all 2s;
}
```

14. Target selector

Now apply the 'target' selector to define the CSS states for elements that are selected via their ID. Elements will transition to this state when they are clicked on – ie this is the state that all modal boxes will appear as. The !important element ensures that these styles overwrite the previously defined styles for <article> elements.

```
:target {
visibility: visible !important;
left: 25% !important;
top: 25% !important;
width: 40% !important;
height: 50% !important;
opacity: 1 !important;
}
```

15. Transition: fade

Now that the modal boxes have the ability to transition from the default <article> settings to the :target settings, we can apply rules that will overwrite the default <article> settings through the data-transition attribute. Our first rule will let the modal box fade in and out of view.

```
[data-transition="fade"]{
left: 25%;
top: 25%;
opacity: 0;
visibility: hidden;
}
```

16. Transition: left

The next transition we want to use is to let the modal box slide in from the left by placing the default position of the modal box as fully off the screen to the left. This means that it will move from this position to the centred position set in the :target when this element is selected for display.

```
[data-transition="left"]{
left: -100%;
top: 25%;
}
```

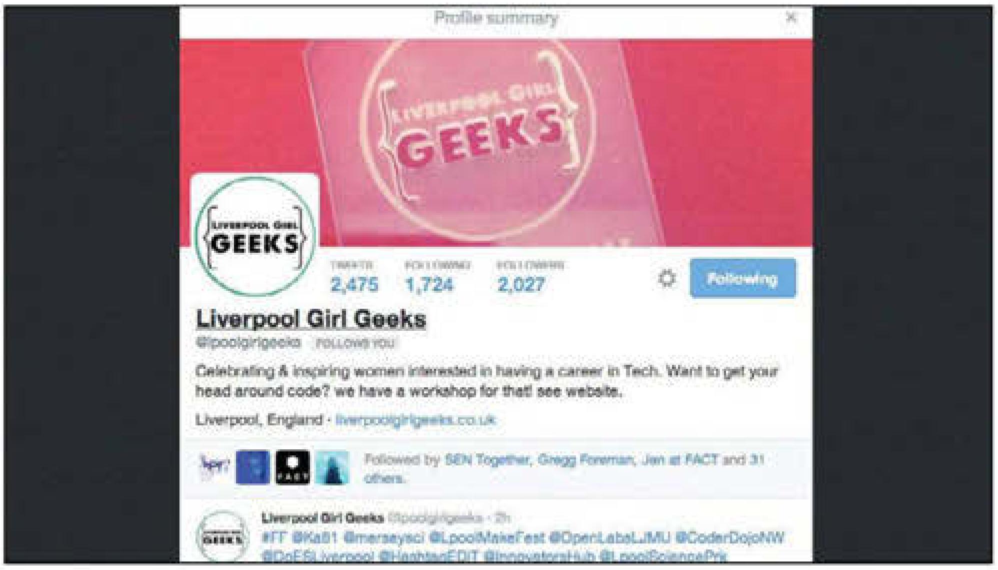
17. Transition: right

We also want a transition effect to make the modal box slide in from the right by placing the default position of the modal box as fully off the screen to the right – ie 100 per cent of the screen width from the left.

```
[data-transition="right"]{
left: 100%;
top: 25%;
}
```

18. Transition: zoom

The final transition to add is the ability for a modal box to zoom in and out from the screen centre. This transition requires the default style to be of zero width and height. We'll also add an opacity of zero so that the modal box will fade in and out of view as it sizes up or down.



Profile summary

TWEETS 2,475 FOLLOWING 1,724 FOLLOWERS 2,027

Following

Liverpool Girl Geeks
@lpoolgirlgeeks FOLLOWS YOU

Celebrating & inspiring women interested in having a career in Tech. Want to get your head around code? we have a workshop for that! see website.

Liverpool, England · liverpoolgirlgeeks.co.uk

Followed by SEN Together, Gregg Foreman, Jen at FACT and 31 others.

Liverpool Girl Geeks @lpoolgirlgeeks · 2h
#FF @Ka81 @merseysci @LpoolMakeFest @OpenLabsLJMU @CoderDojoNW @DoES Liverpool @HashtagEDIT @InnovatorsHub @LpoolSciencePrk

Modal box usage in UX design

Visual information often needs to take up a large amount of space in order to be fully useful. Check out Twitter, which uses modal boxes to display basic information about a person's profile that are triggered when you click on a Twitter handle. Using text or thumbnails to trigger the presentation of the full image or video when clicked provides a way to make full use of visual information without compromising the presentation of the main page content. Forms can also provide users with complementary functionality such as search, registration and account logins. Or, modal boxes can be used with alerts and notices – automated appearances of information can be provided to show important information that the user must see. Similarly, modal boxes for tips are presented as a clickable options for requesting additional information – useful for enabling users to access more information when they want it.

```
[data-transition="zoom"]{
visibility: hidden;
left: 50%;
top: 50%;
width: 0;
height: 0;
opacity: 0;
}
```

19. Style close buttons

The modal boxes have an an element with an attribute of 'data-button="close"' used to close the modal box. We will style this to appear in the top-right corner as a red circle – achieved by setting the border radius to be curved and for the element to float to the right.

```
article [data-button="close"]{
display: block;
float: right;
background: #c00;
color: #fff;
border: 3px solid #fff;
border-radius: 1em;
padding: 1em;
text-decoration: none;
}
```

20. Style images

Now we change the colour of the close button by applying styles through the :hover selector that will trigger the change when the user hovers their mouse pointer over the close button.

```
article [data-button="close"]:hover{
background: #fff;
color: #c00;
border: 3px solid #c00;
}
```

21. Image spacing

Our modal box design makes use of an image to be displayed at the left of the text content. This image should be the height of the modal box – we can use CSS to set the image as 100 per cent of the modal box article

```
img{
display: block;
float: left;
height: 100%;
}
```

22. Place a margin

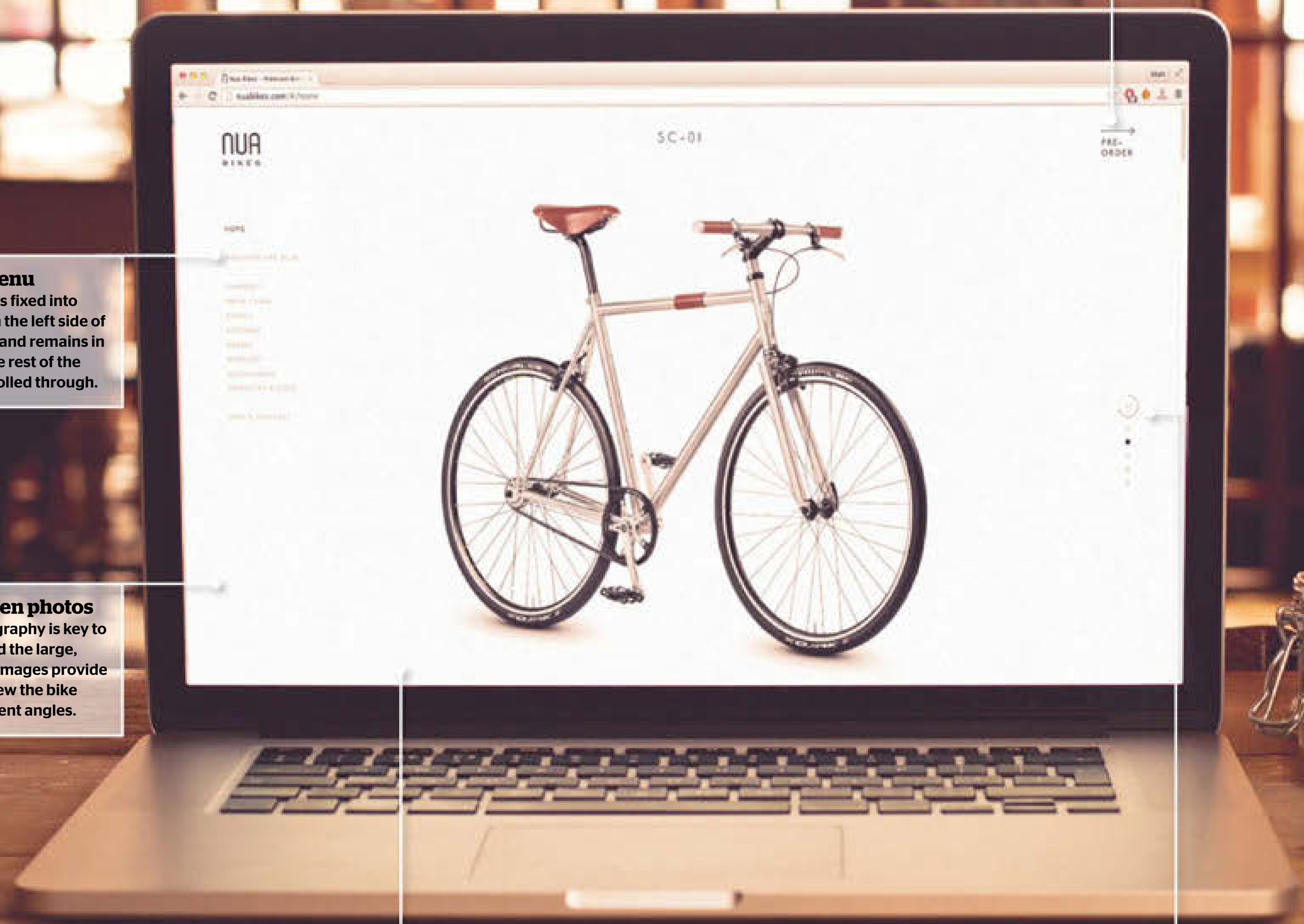
Make sure that there is a margin of one text character to the right of the image so the text isn't too close to the image. Place a margin to the right of the image, so that the positioning of all elements in the modal box will take this into account.

```
article img{
margin-right: 1em;
}
article img{
display: block;
float: left;
height: 100%;
}
```

web workshop

Create a rotating product viewer

As seen on **nuabikes.com**



Signposted URL
Getting sales for the bike is key so a large pre-order label in the top-right of the screen will always provide access to it.

Fixed menu
The menu is fixed into position on the left side of the screen and remains in place as the rest of the page is scrolled through.

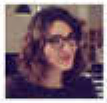
Fullscreen photos
The photography is key to this site and the large, fullscreen images provide a way to view the bike from different angles.

Fluid sections
The page is a one-page site that enables the user to scroll down each section or use the menu on the left-hand side.

Pausable scroll
The right-hand navigation controls the images and the user can click on the image, pause it or watch it from different sides.

Nua Bikes produce a very elegant and clean urban bike that has a minimal impact on the environment due to the simplicity in its design. The bikes are handcrafted and reflect the conscious commuter who needs to get around urban environments with ease.

The website needs to reflect the styling of the bike and this is always easy to do with photography when there is a strong product on display. The homepage features a large, fullscreen image of the bike that has been photographed from different angles so that the user can see the bike rotating on screen. The images change automatically and gives the user a chance to see the design of the bike in detail. When the product you are designing for has such clean, elegant lines and is made from quality materials such as titanium, carbon fibre and aluminium, then it is important that the styling of the site reflects the construction aesthetic. It is so easy for designers to get carried away by adding more and more complex content to sites but sometimes all that is needed is good typography, navigation, simple colour and the product itself. Organising the content can be tricky, but the Nua Bikes site has managed to do this without overstating the simplicity of the design.



<comment>
What our experts think of the site

Innovative but intuitive

"The website uses the minimum number of elements to do its function. Innovative but intuitive at the same time, the site shows the product and enhances its qualities. A single guided scroll shows all components while the user can contemplate how beautiful the bike is – a website to be enjoyed on different devices." **Alicia Gomez Garcia, freelance digital art director and graphic designer**

Technique

1. Add the images

To create a rotating image of a bike, we need to add five images to the body section of the page. Using jQuery a simple image rotator can be created which moves the image on every few seconds.

```
<div id="fader">
<img src="img/bike1.png"/>
<img src="img/bike2.png"/>
<img src="img/bike3.png"/>
<img src="img/bike4.png"/>
<img src="img/bike5.png"/>
</div>
```

2. Style the images

Now move to the head section of your page and add the style tags for the div tag that holds the images. This simply positions them relatively so that the JavaScript can change the images later on.

```
<style>
html, body{
height: 100%;
background-color: #f0f0f0;
padding: 20px;
}
#fader {
position: relative;
width: 100%;
height: auto;
}
</style>
```

3. Library link

The images have to be hidden and positioned on top of each other, so a link to the jQuery library is needed. This will aid the adding and removing of CSS to make each of the images appear at the appropriate time.

```
<script src="https://code.jquery.com/
jquery-2.1.3.min.js”></script>
```

4. Move through the images

After the link to the jQuery library the following code can be added. This hides all the images except the first one. They are positioned within the div tag so that they will be in the right position when they are called to fade in with the code in the next step.

```
<script>
$(function() {
$('#fader img:not(:first)').hide();
$('#fader img').css('position', 'absolute');
$('#fader img').css('top', '0px');
$('#fader img').css('left', '50%');
$('#fader img').each(function() {
var img = $(this);
$('<img>').attr('src', $(this).attr('src')).
load(function() {
img.css('margin-left', -this.width / 2 +
'px');
});
});
```

5. Switch images

The 'fadeNext' function does the hard work of fading out the old image while fading in the next. This is called every three seconds by the set interval, which in turn calls the fadeNext function. Save this now and test it in your browser to see the images appear on the screen.

```
function fadeNext() {
$('#fader img').first().fadeOut().
appendTo($('#fader'));
$('#fader img').first().fadeIn();
}
var rotate = setInterval(fadeNext, 3000);
});
</script>
```

EXPERT ADVICE
Keep navigation simple
On the homepage the navigation is hidden away in an offscreen menu, giving space on the page to create more of a showcase for the site. As the user moves through onto content pages, the menu is on hand to provide easier browsing and there are even previous and next buttons to move through the content.

Build apps with Facebook's React Native framework

Create a JavaScript-powered native cataloguing image app with the React library

DOWNLOAD TUTORIAL FILES
www.filesilo.co.uk/webdesigner

So, you've been writing JavaScript for a little while now, you'd say you've gotten quite good at it, you've made websites, a couple of small libraries and even a few web apps. Thing is, no matter how awesome the web is (and it is fully awesome) the call of the native platform still beckons. You decide to give it a crack, but you've never written anything with Objective-C before – sure, there's this new Swift thing that looks familiar, but the process of building an app is still foreign to you. We could try some transpilers, that is, write some apps in JavaScript and then have something like Titanium figure out what your code is trying to do, but that's a painful process to get right. We could try out PhoneGap – that's nice and familiar and you can get great results after some tweaking, but JavaScript optimisation is hard – especially when it's inside a native wrapper. What are we to do? Wouldn't it be great if we could write apps in JavaScript with our favourite IDE but have everything in our views be native – wouldn't that be ideal? Well, that's React Native – a supercool framework from Facebook that lets you write your app logic in familiar React.js syntax and then renders all of the view using native iOS/Android elements. It's not transpiling or reinterpreting your JavaScript code, it's running it separately in the background. In this tutorial, we're going to be making a catalogue app, that is, an app that shows us cats (see what we did there?), so let's get to it!

1. Get React installed

React Native is a complicated piece of kit, so there are quite a few dependancies but thanks to homebrew, NPM and Node.js setting up React Native is simple. To install just run these commands in a terminal window:

```
brew install watchman
npm install -g react-native-cli
```

2. Create a new project

Now we have the React Native CLI tools installed, CD to a directory that you want to create your React Native project in and run 'react-native init catalogue'. This will create a bare-bones project with everything we need.

3. Fix Xcode project/iOS simulator

For some, the Xcode project might not work out of the box, Xcode can fix that for us. Open the Cat-alogue.xcodeproj file inside the folder you just created. Xcode will open. Click on the project tab on the left and select 8.0 from the Deployment Target drop-down menu.

4. Select device

Look at the play button in the top left of Xcode, just to the right of it there should be a small icon with iPhone 4s or something similar, click it and select iPhone 5s or iPhone 6 (depending on your version of Xcode) from the dropdown that appears and hit play.

5. Fix errors

If you get errors, the fix is simple. Click on the tabs showing the error and select the first fix that's suggested. Hit play and the app should run. The iOS simulator might throw an error so select 'iOS simulator' in the menubar and click 'reset content and settings' and hit play again.

6. Our bare-bones app

In our iOS simulator, our app will now be open. Go back to the catalogue folder we created and open index.ios.js in your favourite IDE. This is the main entry point for React Native to run our app from. We're using the virtual DOM to create elements, but instead of DOM elements, we're using equivalent iOS elements.

7. Clear the house

This React Native project has one view, but we need two views for this tutorial: one for listing our categories, and another for viewing the photos in that category. Delete everything in index.ios.js and insert the following in its place to import the required modules for our project.

```
'use strict';
var React = require('react-native');
var {
AppRegistry,
StyleSheet,
Text,
Image,
ListView,
TouchableHighlight,
NavigatorIOS,
View,
} = React;
var CatViewing = require('./catViewing');
var List = require("./list")
```

8. Create a starting point

We now have everything we need except for a view. Add the following code to index.ios.js, it uses the NavigatorIOS module to let us create views, this means we get a lot of things like back buttons and animations between views for free. The initialRoute property determines where our app should find the code for our first view.

```
var catalogue = React.createClass({
render: function() {
return (
<NavigatorIOS
style={styles.container}
initialRoute={{
title: 'Cat-Alogue',
component: List,
}}
/>
);
}
});
```

9. Stylings

Let's create a simple stylesheet. React doesn't use CSS, it uses a JavaScript polyfill to emulate a limited subset CSS with native components, and as such we define styles like so. AppRegistery exposes our React class to our native code, it's our true entry point to the application.

```
var styles = StyleSheet.create({
container: {
flex: 1,
flexDirection: 'row',
justifyContent: 'center',
// alignItems: 'center',
backgroundColor: 'white',
},
});
AppRegistry.registerComponent('catalogue',
() => catalogue);
```

10. Create list.js

In index.ios.js we defined the List component as our first view. Create the file 'list.js' in our project folder and a file called 'catViewing.js'. Import the following modules:

```
'use strict';
var React = require('react-native');
var {
AppRegistry,
StyleSheet,
Text,
Image,
ListView,
TouchableHighlight,
View,
} = React;
var cats = require("./cats"),
thumbCats = [];
var CatViewing = require('./catViewing');
```

just after that, we're going to create the List class and all of the functions that it will use to build our view:

```
var list = React.createClass({
getInitialState : function(){
},
renderCatThumb : function(cat){
},
loadCat_egory : function(category){
},
render: function() {
},
});
```

11. Find the cats module

If we run our code, we get an error, this is because it can't find the cats module we defined earlier. You can grab cats.js from FileSilo and add it to our project folder, it's a

Learn once, write everywhere

We've all heard of 'write once, deploy everywhere' but there's rarely an instance where that *actually* works out. Facebook's suggested philosophy is "learn once, write everywhere". Once you've learned React, you can write it for any platform, rather than maintaining one codebase that has to accomodate everything.

really simple JavaScript module that contains URLs for the cat images we'll be using.

12. Set our initial state

We need to help React make sense of our data for iOS so we use a ListView to list categories and a ListView will only accept a DataSource object as an argument. Inside of getInitialState, add the following code to create an object for each of our categories for making a tab:

```
..
for(var key in cats){
thumbCats.push({
image : cats[key][0],
title : key
});
}
var ds = new ListView.
DataSource({rowHasChanged: (r1, r2) => r1
!== r2});
return { dataSource: ds.
cloneWithRows(thumbCats), };
...
```

13. Render the thumb cats

Now that we have a data source our ListView can use to render items, we're going to need a function to define what each list item should look and how it works. We can do this in the renderThumbCat function. Here, we're using React Native's iOS analogous virtual DOM to express how the list elements should be laid out.

```
renderCatThumb : function(cat){
return (
<TouchableHighlight onPress={() => {this.
loadCat_egory(cat.title)}} underlayColor="rg
ba(0,0,0,0.1)">
<View style={styles.cell} >
<Image source={{uri : cat.image}}
style={styles.thumbnail} />
<View style={styles.rightContainer}>
<Text style={styles.title}>{cat.title}
Cats</Text>
</View>
</View>
</TouchableHighlight>
);
},
```

Is it honestly native?

Yes! Facebook have yanked out the JavaScript engine from the WebViews and run it in a seperate thread. Your app logic runs in JS but everything you see on the screen is completely native.

14. The first render

We've not had much to look at so far, but that's about to change. We now have almost enough code to render our first view. Once every React Module has been loaded, the render function is the last function called and it will render the view. Now change your render function to the following:

```
render: function() {
return (
<View style={styles.container}>
<ListView dataSource={this.state.dataSource}
renderRow={this.renderCatThumb}
style={styles.listView} />
</View>
);
},
```

15. JsCSS stylings

Add some stylings or everything will be clumped together. We can do this with Stylesheet.create(). This is not CSS, but what Facebook calls JsCSS, a subset of CSS expressed as objects. Each object in styles can be thought of as a class name and each property is a styling. A key difference is that we separate stylings with a comma, not a semicolon.

16. Rendering

Head over to the iOS simulator and hit Cmd/Ctr+R on your keyboard to render. We don't need to recompile the app, because we're using JavaScript to determine our views, we can rerender on-the-fly. If your keyboard did nothing you need to go to Hardware>Keyboard>Connect Hardware Keyboard in your iOS Simulator Menu bar.

17. Push a view to the stack

Now we have a scrollable list of categories, but when we tap them, they highlight, but nothing happens. If we want to see the images in the category, we need to push a new view to our stack and pass the relevant images to it for loading. Add the following code to loadCat_egory() and then create the file catViewing.js in our project folder:

```
..
var catsToView = cats[category];
this.props.navigator.push({
title: category + " Cats",
component: CatViewing,
passProps: {catsToView},
});
...
```

18. catViewing.js

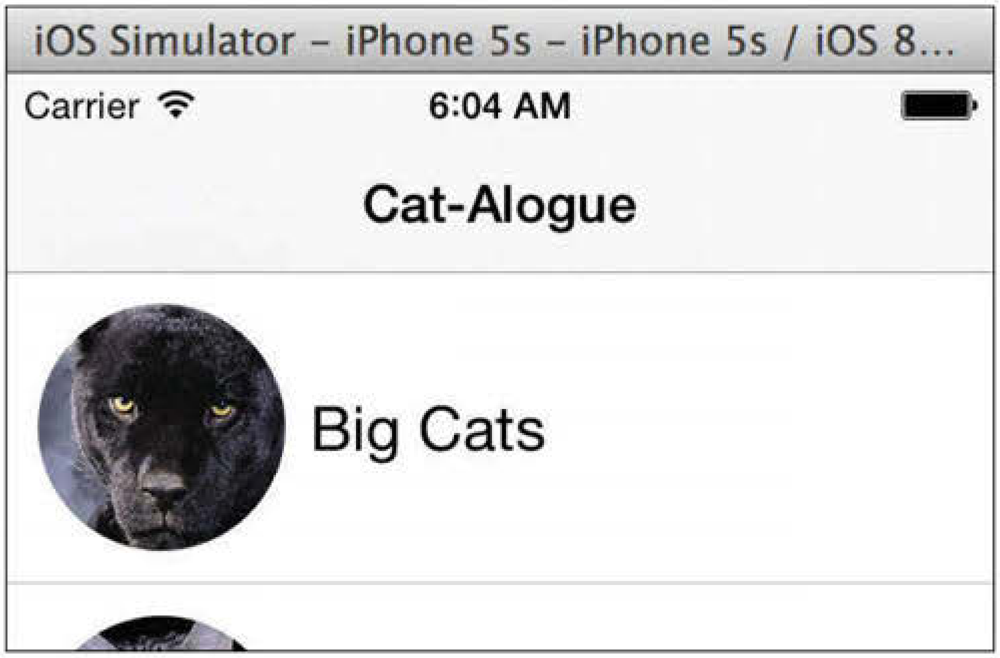
With loadCategory() we've passed everything we need to render our images, but where did we pass it to? What

Requiring unknown module "Joats". If you are sure the module is there, try restarting the packager.

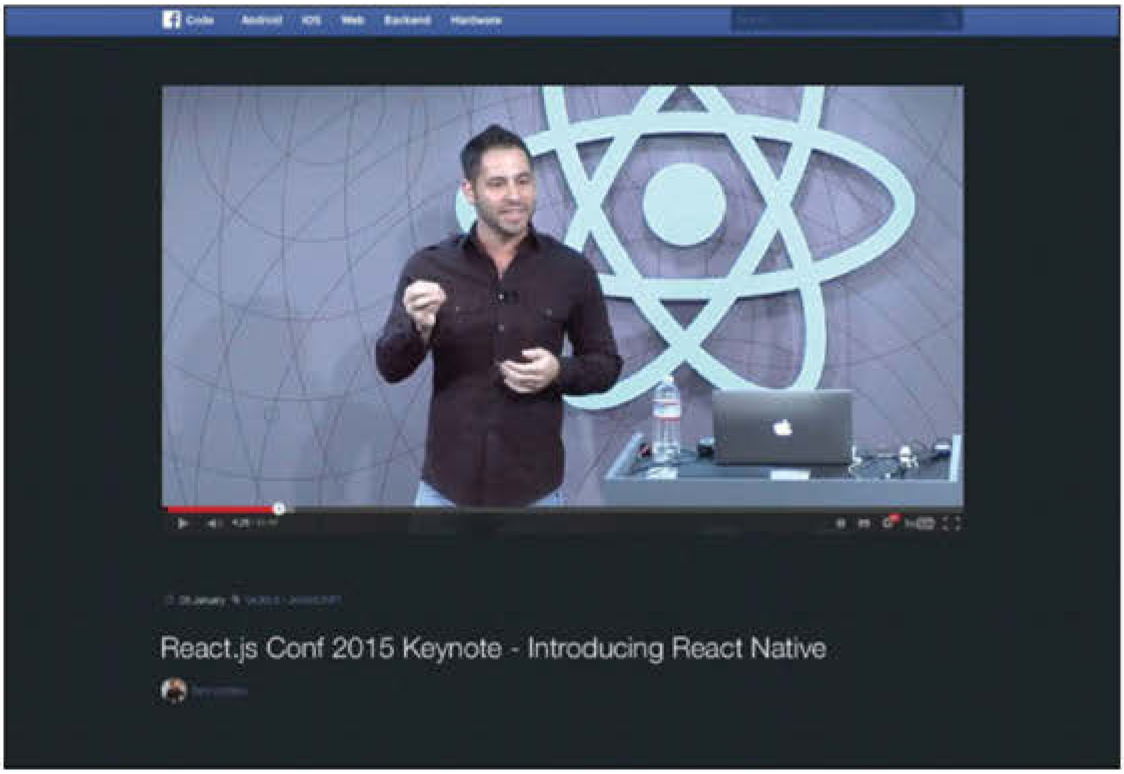
Top left
Historically, error messages have been obfuscated noise that are rarely any help. Fortunately, that's not so much the case these days. The fix Xcode prescribes is spot on in this case

Top right
Facebook doesn't want your code to fail silently. It wants to get up in your face so you write better code, so here is what they've dubbed the 'Red Screen of Death'

Right
This is how our app should look so far. Our categories for our ListView are being dynamically generated in JavaScript and then rendered with iOS native components



iOS Simulator – iPhone 5s – iPhone 5s / iOS 8...



React.js Conf 2015 Keynote - Introducing React Native

JsCSS - WAT?

Aligning objects on a screen is a tough business, just ask any iOS developer who's done a lot of manual positioning, it can be a nightmare. Facebook looked around for a solution to aligning things intuitively and it found one, CSS! Problem is, implementing CSS for iOS was not a prospect they fancied much, instead, they opted to create a JavaScript subset of CSS styles that could be used to position native iOS elements in and around native iOS views so it has all the goodness of CSS and JavaScript and full speed of a native. One of the coolest things about JsCSS? It has Flexbox, not the full Flexbox, but something very close and easy to use. Check out the Facebook Developers YouTube channel for more information on React Native **bit.ly/1AK9zmZ**.

about the CatViewing class we're about to create in catViewing.js? CatViewing works just the same as our list class worked, except instead of being a ListView, we're using a ScrollView which gives us a little more flexibility when it comes to how we lay content out. Add the following code to catViewing.js to get started:

```
'use strict';
var React = require('react-native');
var {
Image,
PixelRatio,
ScrollView,
StyleSheet,
Text,
View,
} = React;
var CatViewing = React.createClass({
render: function() {
},
});
module.exports = CatViewing;
```

19. Get our list of cats

The first thing we want to do is get a reference for our list of cats that we passed to the view from our List class. Each React class has a props object which we can access by using 'this'. We don't need many functions here for our CatViewing class, in fact we only really need one and that is a render(). So ammend render() to look like the following:

```
render: function() {
var catsToShow = this.props.catsToView;
return (
<ScrollView contentContainerStyle={styles.
contentContainer}>
{catsToShow.map(createThumbRow)}
</ScrollView>
);
},
```

20. Map our cats

In our scrollView, we have mapped catsToShow to createThumbRow, but we haven't written that yet! So the actual making of the createThumbRow is a very simple process – for each image we pass to it, it will create an instance of the Picture class we're about to create. Picture is a very simple class, so it's very similar to CatViewing in that sense, it only has a render function and all it does is take a wrap for an image in a view and then it will grab that image's source. Just think of it as if you are wrapping an <img> in a <div> and then setting the src attribute over in HTML.

```
var createThumbRow = (uri, i) => <Picture
key={i} uri={uri} />;
var Picture = React.createClass({
render: function() {
return (
<View style={styles.centering}>
<Image style={styles.imageHolder}
source={{uri:this.props.uri}} />
</View>
);
}
});
```

21. Finish up

We're almost ready to view all of our cats. If you refresh the iOS simulator, we get a nice big error (we might even crash Xcode!) because we haven't added styles for the Picture or CatViewing classes. But, that's simple, if we just add the following after our Picture class closes, we're all done. Now we can take our Cat-alogue, and put it wherever we want it to be.

```
var styles = StyleSheet.create({
contentContainer: {
padding: 10,
},
centering : {
alignItems : "center"
},
imageHolder : {
width : 280,
height : 280,
marginBottom : 5
}
});
```

22. Fade out

Finally it is possible to fade out the drawing from previous frames. In the 'draw' function, comment out the background colour. Set the fill colour to black with a low opacity of around ten per cent. A rectangle is now drawn over previous frames with the low opacity, causing them to fade out. Save and test to see this final effect.

```
background(0);
fill(0, 25);
rect(0, 0, windowWidth, windowHeight);
```

28 PAGES OF EXPERT TUTORIALS
web designer
FREE 195 MINS OF CSS VIDEO
www.webdesignermag.co.uk
HTML5 CSS3 jQuery WordPress
14 PAGE GUIDE ADVANCED ANGULAR INDUSTRY TIPS & TUTORIALS
BUILD WITH REACT NATIVE
A LOOK INSIDE FACEBOOK'S NEW FRAMEWORK
POWER UP CSS PREPROCESSORS
·SASS ·LESS ·STYLUS
EXPERT ADVICE TO MASTER PRO LEVEL TOOLS
DYNAMIC CSS MODALS
ANIMATE POP-UP WINDOWS
WHAT IS WEB OF THINGS?
WHY IOT WILL ADOPT WEB STANDARDS
INTERACTIVE GRAPHICS
UNLEASH THE P5.JS LIBRARY FOR DRAWING IN THE BROWSER
67 DESIGNER RESOURCES
WEB TEMPLATES, FONTS, TEXTURES & ACTIONS
ISSUE 236
£5.99

POWER UP CSS PREPROCESSORS
What does a preprocessor do?

Creativity assembled

Every issue packed with...

- Expert tutorials on HTML, CSS, jQuery and leading libraries and frameworks
- In-depth interviews with the world's leading creative agencies
- A close-up look at the latest web development technologies
- Designer resources and pro video guides

Why you should subscribe...

- **Save 30%** off the cover price – **just £12.50 every 3 issues** on Direct Debit
- **FREE delivery** direct to your door
- **Never miss an issue**

Order by

31ST JULY
Use code **PQ15** for this offer.

BY POST

Send your completed form to:
Web Designer Subscriptions, 800 Guillat Avenue,
Kent Science Park, Sittingbourne, Kent ME9 8GU

YOUR DETAILS

Title________________First name____________________________________

Surname____

Address____

____

____

Postcode______________________Country_________________________

Telephone number____

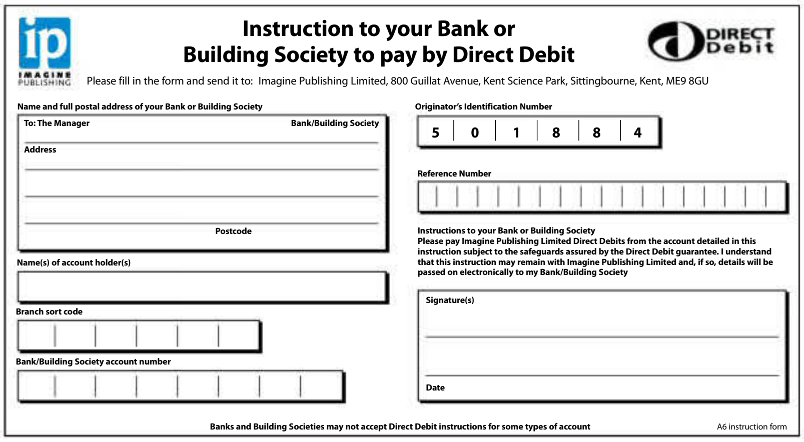
Mobile number____

Email address____

DIRECT DEBIT PAYMENT

□ **UK Direct Debit Payment**

I will pay just £12.50 every 3 issues (Save 30%)



Instruction to your Bank or Building Society to pay by Direct Debit

Please fill in the form and send it to: Imagine Publishing Limited, 800 Guillat Avenue, Kent Science Park, Sittingbourne, Kent, ME9 8GU

Name and full postal address of your Bank or Building Society

To: The Manager **Bank/Building Society**

Address

Postcode

Name(s) of account holder(s)

Branch sort code

Bank/Building Society account number

Originator's Identification Number

5	0	1	8	8	4

Reference Number

Instructions to your Bank or Building Society
Please pay Imagine Publishing Limited Direct Debits from the account detailed in this instruction subject to the safeguards assured by the Direct Debit guarantee. I understand that this instruction may remain with Imagine Publishing Limited and, if so, details will be passed on electronically to my Bank/Building Society

Signature(s)

Date

Banks and Building Societies may not accept Direct Debit instructions for some types of account

A6 instruction form

PAYMENT DETAILS

YOUR EXCLUSIVE READER PRICE, 1 YEAR (13 ISSUES)

□ UK £62.30 (Save 20%) □ Europe – £70 □ World – £80 □ USA – £80

Cheque

□ I enclose a cheque for £_________________________________
(made payable to Imagine Publishing Ltd)

Credit/Debit Card

□ Visa □ Mastercard □ Amex □ Maestro

Card number Expiry date

Issue number □□ (if Maestro)

Signed ____

Date ____

Please tick if you do not wish to receive any promotional material from Imagine Publishing Ltd by post □ by telephone □ via email □

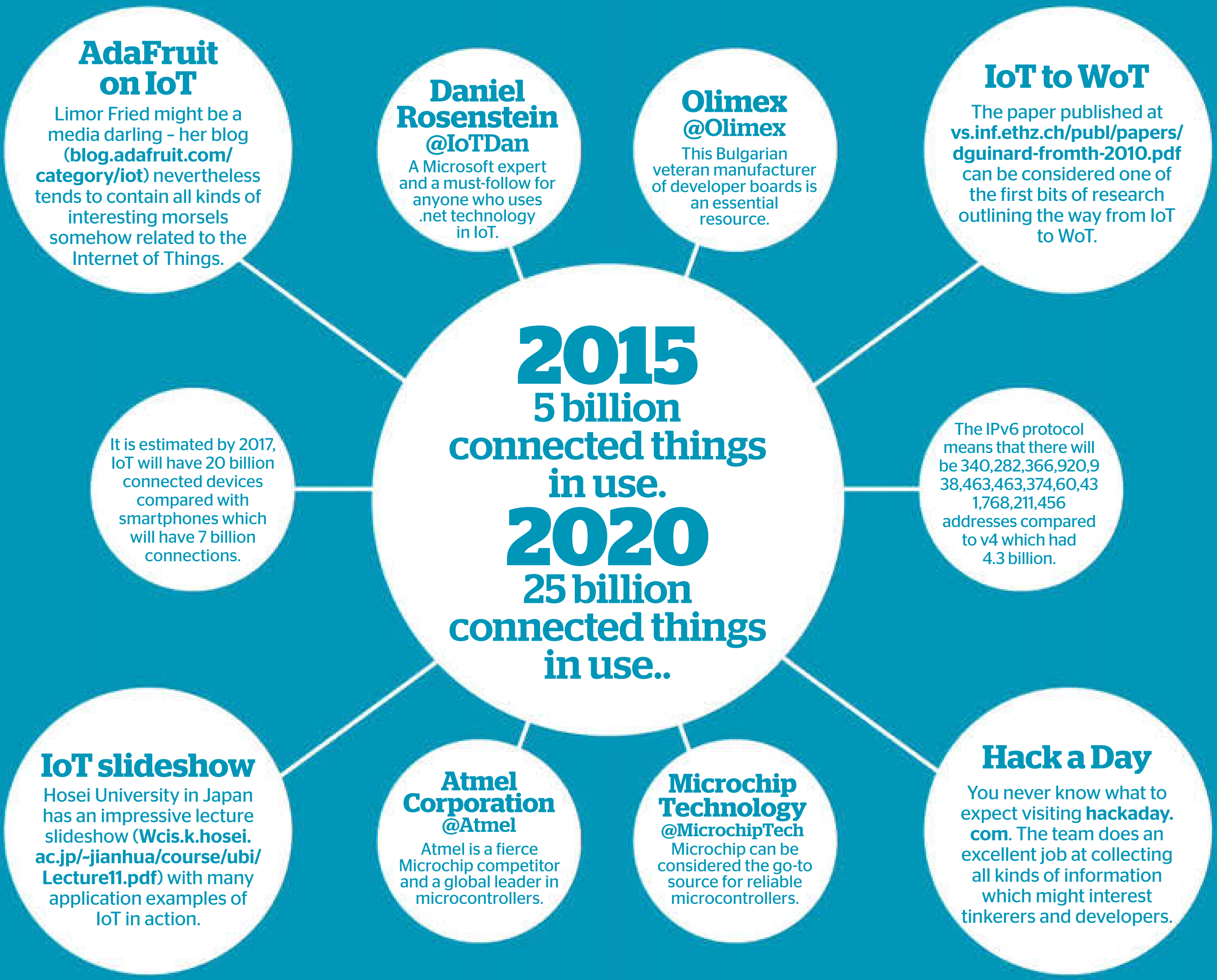
Please tick if you do not wish to receive any promotional material from other companies by post □ by telephone □ Please tick if you DO wish to receive such information via email □

TERMS & CONDITIONS

Terms and Conditions: This offer entitles new UK direct debit subscribers to pay only £12.50 every 3 issues. New subscriptions will start from the next available issue. Offer code PQ15 must be quoted to receive this special subscription price. Details of the Direct Debit Guarantee are available on request. This offer expires 31st July 2015 . Imagine Publishing reserves the right to limit this type of offer to one per household. Subscribers can cancel this subscription at any time.

WEB OF THINGS

The IoT revolution has only just begun. The Web of Things is here to brush aside proprietary protocols and help bring web standards to the party



AdaFruit on IoT

Limor Fried might be a media darling – her blog (**blog.adafruit.com/category/iot**) nevertheless tends to contain all kinds of interesting morsels somehow related to the Internet of Things.

Daniel Rosenstein @IoTDan

A Microsoft expert and a must-follow for anyone who uses .net technology in IoT.

Olimex @Olimex

This Bulgarian veteran manufacturer of developer boards is an essential resource.

IoT to WoT

The paper published at **vs.inf.ethz.ch/publ/papers/dguinard-fromth-2010.pdf** can be considered one of the first bits of research outlining the way from IoT to WoT.

It is estimated by 2017, IoT will have 20 billion connected devices compared with smartphones which will have 7 billion connections.

2015
5 billion connected things in use.
2020
25 billion connected things in use..

The IPv6 protocol means that there will be 340,282,366,920,938,463,463,374,60,431,768,211,456 addresses compared to v4 which had 4.3 billion.

IoT slideshow

Hosei University in Japan has an impressive lecture slideshow (**Wcis.k.hosei.ac.jp/~jianhua/course/ubi/Lecture11.pdf**) with many application examples of IoT in action.

Atmel Corporation @Atmel

Atmel is a fierce Microchip competitor and a global leader in microcontrollers.

Microchip Technology @MicrochipTech

Microchip can be considered the go-to source for reliable microcontrollers.

Hack a Day

You never know what to expect visiting **hackaday.com**. The team does an excellent job at collecting all kinds of information which might interest tinkerers and developers.

> *"The IoT is the next frontier of distributed computing and ambient intelligence and as such meant to change the way we live, work and play. However, IoT is still perceived by most people as 'just' a framework of things connected and controlled by a smart device. A more comprehensive view sees IoT as the bridge between the digital and physical world, the space where the natural boundary between the two becomes blurred and new things happen. The Web of Things is the additional application layer which implements a broader view, allowing links with the web and web data, content and services. Dizmo fully embraces the Web of Things, its concept and programming technologies. It provides the infinite whitespace where the objective of the Web of Things, to extend the reach of the IoT and to simplify its full implementation, can be achieved"*

Luigi Mantellassi, CMO at Dizmo

When Kevin Ashton was working at Procter & Gamble's, he recognised that RFID technology could simplify the ever-confusing factory floors. It takes but one look at the vast factory halls of an aeronautic company – be it Tupolev, Bisnovat or Ilyushin – to see, as Howard Moon would put it, a microcosm of creative chaos.

Kevin Ashton used the term Internet of Things to tie his idea into the dot-com bubble. His suggestion was simple: to provide the individual containers with some degree of intelligence to enable process optimisation.

For example a bakery could know that a particular type of product, ie a chocolate cake, is running low; an email to waiters would be ardent in order to inform them that pushing sales of chocolate cake is not sensible due to the lack of stock.

Miniaturisation has enabled chip vendors to produce-ever smaller microcontrollers which can subsist on amounts of power which would have been obscene ten years ago. Back then, a MCU draining ten milliamps was considered frugal. Today, this would be excessive.

For individuals, the Internet of Things can bring both benefits and problems. For example, insurance companies now deploy health-measuring devices which are to be worn around your wrist at all times. The data collected by these systems is transmitted to a central server, where your GP can analyse it in order to determine whether you are behaving in an healthy and sane way. Thus, a binge-drinking party might be cut short by a call from your friendly GP reminding you that you already have hypertension – whether this is a good or bad reminder will need to be decided by each individual of course.

This kind of scenario is now more common though and we will look at a selection of topics connected to our everyday lives with this upcoming field of technology.

Turning IoT into WoT

HOW THE WEB OF THINGS NETWORK MAKES USE OF WEB STANDARDS

Ever since the OSI model was specified, networked systems had a tendency to be broken down into individual layers.

The Web of Things can best be described as the application layer for the Internet of Things. What the IoT does is it looks at the technological side of things that may not have looked techy before. A coffee machine which communicates via a proprietary protocol is a perfectly valid IoT application for example.

In the WoT the situation is a bit more complex. Normally, Web of Things solutions are based on standardised web technologies. This means that the individual devices tend to be addressable via a URL.

Moving our aforementioned coffee machine to the WoT would require the implementation of a standardised communication protocol: be it low-level or a more general and developer-friendly interface such as JSON via REST.

One interesting extension to the topic involves real-time-systems: HTTP is badly suited for this and that's because of its relatively complex handshake architecture. Developers can solve this problem by abusing streams of media protocols such as RTP/RTSP. XPP or WebSockets may also see some use here.

Microsoft on IoT

Open **blogs.microsoft.com/iot** in order to treat yourself to a selection of all things IoT and Redmond. You might not believe it, but Microsoft is a significant IoT player.

Arduino

When working with Arduino planars, creating a system for Internet of Things can be as easy as hacking together a web interface and a basic hardware driver

IoT hardware

HOW SINGLE BOARDS HAVE TRANSFORMED THE INTERNET OF THINGS

When looking for IoT solutions, always keep in mind that your customers' brains are firmly tuned to the good old radio station going by the name WIFM. In case any of you are not aware of the meaning of this acronym: it stands for 'What's In it For Me?'. Successful IoT projects tend to be the ones which provide monetary or comfort value to their customers: humans are, by nature, not masochists and want a bang for their buck.

Arduino and friends

Picking a microcontroller is a daunting task. Manufacturers provide an insane pile of different models: large vendors such as Microchip sell thousands of different controllers. Purchasing a chip is but a small part of the solution: all kinds of additional hardware is required in order to get the MCU up and running. This might not be a problem for experienced electrical engineers – if you don't know the difference between an easistor, a transistor and a thyristor then taking this approach is not sensible.

A variety of third-party vendors provide so-called evaluation, or development boards. Initially intended for electrical engineers evaluating a new chip type, eval boards provided simple access to various peripherals. An Italian researcher called Massimo Banzi was among the first to figure out that eval boards could also be used as single-board computers. His Arduino series has since established itself as the dominant player in amateur embedded development.

The biggest strength of the Arduino Platform is its standardised expansion slot. Known by the name of Shield, it has two rows of sockets. Expansions are plugged into these, the compiler architecture ensures that driver deployment isn't an issue in most instances.

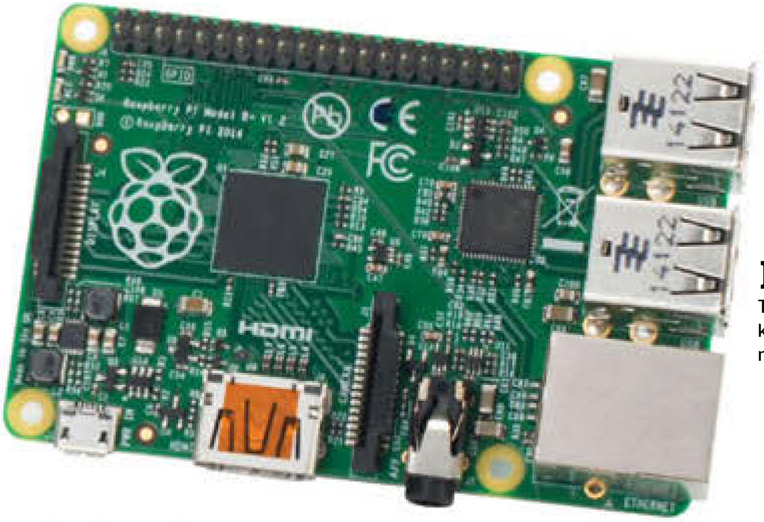
Cheap as chips

When purchasing electronal components, local retailers tend to be ideally suited for prototypes. Once large-scale manufacturing starts, importing your components from China provides significant savings. One good website to start out is AliExpress: be aware that shipping can take up to a month and that American Express credit cards are not accepted.

Due to the ubiquity of the Italian process computer family, a large ecosystem of Shields have cropped up. Your Arduino can be connected to wired ethernet, Wi-Fi or Bluetooth LE – it is but a question of selecting the correct Shield and plugging it into the planar.

In addition to that, a large group of more or less efficient combination planars has seen the light of day.

Apple watch
One of the latest, and most popular smart devices to enter the consumer market



RasPi
This single-board computer can be used to connect to all kinds of items. What about a doorbell that sends a text, makes a call and captures video of who's at your door?

What's WoT?

THE WORK THAT DIFFERS THE TWO NETWORKS

Having started out with the use of RFID, the Internet of Things can best be considered the lower levels of the Web of Things. IoT-related applications are confined to state information contained in real-world objects. They must be made available to software actors and other devices, which then act accordingly.

For example, a shipping container could be equipped with an RFID tag containing its ID information and a microcontroller measuring the weight of the contained iron ore. An intelligent steel plant could then parse this data in order to determine whether a second ore smelter needs to be fired up.

Even though the distinction between IoT and WoT is blurry at best, at the current state of the procedures it is safe to consider lower-level work the pinnacle of IoT development. In practice, developers should not be too concerned with the two terms: just choose the one which sounds best.

Why it matters

A GREATER CONNECTIVITY IS JUST ONE BENEFIT

Internet of Things can be considered the pinnacle of the convergence trend, started by smart TVs, smartwatches and similar devices.

Human life would be made easier if the things surrounding us were provided with intelligence of their own. Imagine a washing machine which completes work on sensitive shirts only when its owner is on their way home. There's also the old adage of the fridge which automatically orders more Red Bull as stock dwindles.

For developers, unlimited opportunities await in this field. Take a look around you and we're sure you will see an endless deluge of dumb devices begging to be made a little smarter. Programming their embedded firmware, is of course, one way to profit from this trend. This, however, requires significant amounts of highly technical knowledge. Fortunately, the frontend of most devices needs to look great: an area where web designers creative experience will be of unlimited value to the manufactures and the developers alike.

Cutting a long story short: IoT and WoT are buzzwords which will make many people really, really rich. Web developers can profit from it in two ways: first of all, endless opportunity awaits in the creation of the user interfaces for devices. Secondarily, accessing IoT and WoT peripherals adds a new layer of context sensitivity to your systems.

The superpricey Arduino Yún combines the AVR-based Arduino with a Linaro-based minicomputer which has a Wi-Fi transmitter. A circuit connected to an Arduino Yún can link into a local Wi-Fi with minimal effort. Third parties provide similar boards with embedded RFID or Bluetooth low-energy radios.

Arduino is but one possible system. The Raspberry Pi has recently seen quite a bit of use in the IoT space. Boards from the Bulgarian consulting company Olimex have also made significant inroads.

Once your solution has reached production readiness, bundling it with a process computer can be expensive due to place and material cost constraints. Many single-board computers have an open hardware design, letting you embed their circuits into a planar of your own. This is especially true of the Arduinos except for the Yún: they are based on stock AVR processors that run on simple breadboards.

When working on small-scale solutions, the added cost of an evaluation board tends to be offset by the cost of getting boards and sensors manufactured. In this case Microsoft's Gadgeteer can be an attractive alternative – it provides a set of prespecified components ranging from displays to cameras and a wide array of other sensors. Their most significant benefit is that they can be connected using a standardised interface: building up your IoT system is as easy as connecting the elements to one another.

The IP address problem

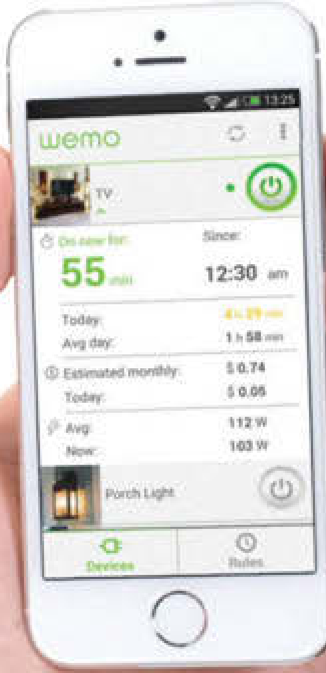
Radical proponents of the Internet of Things propose, that at some point in time, every device in the world should be intelligent. Sadly this does not work well with the existing IP protocol and that's because it was developed to provide about 4.3 billion addresses. This was more than enough for the tiny network at the time – the explosive growth of the internet has since led to an phenomenon known as IPv4 address exhaustion. The price of individual IP addresses has been on a permanent climb ever since.

In the media, Network Address Translation (NAT) is considered one way to mitigate the problem. In practice, however, it does not really work – routing multiple devices of the same type through a router can become a nightmare to set up and furthermore makes the use of a unique port number difficult.

The solution is a new generation of the IP protocol which goes by the name of IPv6. It has been around for ages, but sadly, adoption has been rather slow so far. The biggest barrier to a widespread adoption of IPv6 is that IPv4 hosts cannot contact their more technologically advanced brethren directly.

In the case of the Internet of Things, this should not be too much of a problem. If all sensors and their

> *“Many single-board computers have an open hardware design, letting you embed their circuits into a planar of your own”*



In the real world

THE IOT OUTSIDE OF ITS DIGITAL ARCHITECTURE

Smart LED bulbs

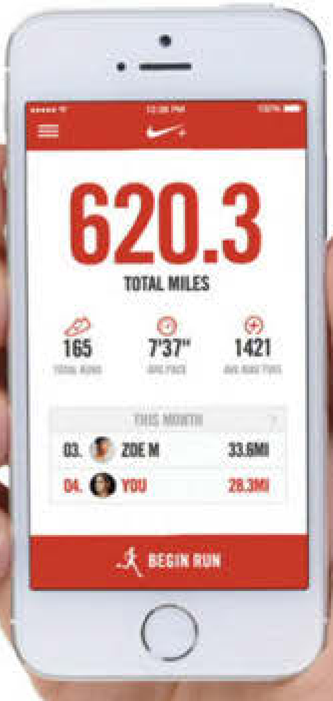
belkin.com/us/Products/home-automation/c/wemo-home-automation

Turning lights on requires you to walk to a nearby switch in the dark. Belkin's WeMo system wants to break this never-ending cycle by providing a smart adapter which is plugged into a mains outlet. A companion application connects to the adapter, permitting you to turn the attached peripheral on and off as needed. In addition to that, smart LED bulbs are sold – their colour hue can be adjusted dynamically. Finally, third parties are permitted to license the standard for use in their own products.

Run tracking

apple.com/ipod/nike/run.html

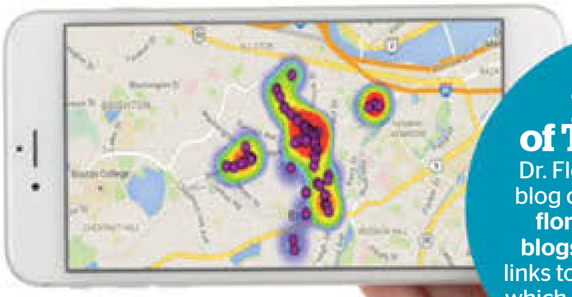
Music can transform a boring jog into a meditative experience. Apple promoted its iPhone 3GS via cooperation with Nike which has since been expanded across the entire product portfolio. The idea is relatively simple: a small sensor is embedded into a space below the sole of your pair of sneakers. While running, data is transmitted to the iPhone – it uses this data for profiling. More advanced systems could perform real-time exercise intensity tracking: overtraining is the bane of amateur and professional joggers.



Fulfilling garbage potential

bigbelly.com/solutions/stations

Overfilling trash cans are an eyesore. Sadly, emptying them every day puts an unnecessary drain on community resources. Bigbelly solves this problem by providing intelligent garbage cans. Each Bigbelly can contains a small solar cell. It provides energy for a small wireless transmitter which returns fullness information to a cloud service. Municipal administrators use this information tc optimise garbage collection: the waste collection service no longer needs to check which cans are empty. Furthermore, valuable information on citizen habits can be collected by analysing the amount of garbage produced.



Mobile Internet of Things blog

Dr. Florian Michahelles's blog can be accessed via **florian-michahelles.blogspot.sk**. It contains links to all kinds of material which might be interesting to IoT devs.

controllers are based on IPv6, there are no legacy systems which need to be supported in the first place. As of writing, most providers are able to handle IPv6 traffic. In practice, developers rarely need to worry about this – keep it in the back of your head though in case something goes really, really wrong or some pesky VC asks you about it.

Harness the power

We name-checked Bluetooth LE earlier. The technology – called Bluetooth Smart when trading in the consumer space – shares the name and the frequency hopping with its famous ancestor. Other than that, it can be described as complete redesign of the original shortwave wireless service.

Bluetooth LE fixes many previous problems. The discovery section, which once took more than ten seconds can now be accomplished in less than a second. The availability of the GATT profile ensures that developers and device manufactures don't need to bother themselves with creating, specifying and certifying Bluetooth profiles.

> *"Due to widespread adoption... [Bluetooth LE] is a winner – it could even thrive without the IoT itself"*

Bluetooth LE works on a relatively simple software model based on characteristics. A characteristic is an attribute of a device: it could be the colour of an LED or the amount of energy left in the battery. Further information on each characteristic can be contained in the descriptors, both of which can be wrapped up in services. The GATT profile acts as transport layer around these concepts: you use it to read and write characteristics and descriptors. Implementing Bluetooth LE by hand is a painful and futile task: a variety of companies provide ready-made modules which can be attached to a process computer of choice via SPI or L2C. LE is far superior to proprietary protocols like the ones used in the once-popular LPRS easyRadio family. Due to the widespread adoption of Bluetooth LE and the highly cooperative way Special Interest Group handles licensing, this technology is a winner – it could even thrive without the IoT itself.

Sensors...

By and large, single-board computers are a relatively stupid bunch. They can read digital signals, a much slower A/D converter samples analogue voltages with moderate accuracy. When performing measurements of real-world dimensions, the first step usually involves analysing the type of the quantity to be measured. If you are dealing with a voltage, the situation is easy – you need to check if the A/D converter can sample it directly or if amplification or dampening is needed.

If the dimension at hand is not of an electronic type, a transducer is needed. Theses can be grouped up into two groups: dumb and smart ones. The classic

WoT Interest Group

W3.ORG/WOT/IG

After the success of the Web of Things Workshop held in Berlin in 2014, the W3C started a Web of Things Interest Group. The group will be responsible for defining standards: the topics discussed will range from simple things such as scripting language subtypes to more complex issues such as data encoding, metadata formats and the protocols used for communication. Being a member of this group permits the providing of input to these standard drafts. It can also serve as a first-class opportunity for networking as the current chairs are from Siemens and Intel and it wouldn't hurt to have these contacts.

Sadly joining the Web of Things Interest Group is not cheap for an amateur developer. To join the W3C you will need to pay the minimum annual fee of €1,950 (£1,395) for two years. After that, the fees will quadruple and will stay in the range of about €8,000 (£5,725) a year.

WoT community

W3.ORG/COMMUNITY/WOT

If the official Web of Things Interest Group is too expensive, the WOT community might be an alternative. It was first organised in 2013, and was responsible for the highly successful workshop in Berlin which spawned its aforementioned larger brother.

Sadly, its activities have since started to slow down – no updates have been posted since June 2014. Nevertheless, joining the remaining 160 members of the WoT Community Group might be an interesting way to start connecting yourself to the W3C. Linking up is as easy as visiting their website (**w3.org/community**). Fill out the form, and you are all set to go – if there still is anything taking place, that is.

You can also look for local events. In most big cities, one or more Internet of Things meetups take place – joining one is a cheap way to find out more.

example of a dumb transducer is an LDR or an NTC: they are resistors, which change their value in relationship to the measured dimension. Using them requires the amplification or dampening of the voltage, which must then be digitised. Due to the nonlinear characteristic of most transducers, a complex linearisation is required in order to yield useful values. This problem can be bypassed by using smart transducers. They communicate with your single-board computer via a defined protocol: some simply provide parallel data, while others can be addressed via SPI or I2C. Using these industry-standard bus protocols simplifies sensor integration: in the case of an Arduino, accessing the peripherals is as easy as invoking the correct library functions. The datasheets of smart transducers tend to contain one or more sample circuits, which explain how to condition the incoming signal for maximum accuracy.

...and actors

Reading values is but part of the solution. Acting on collected information usually requires quite a bit of power: the GPIO ports of the average MCU are not strong enough to provide the kind of current needed to move motors, switch relays or do other funny things.

Well, the transistor comes to the rescue here: this component can amplify small currents, thereby increasing the current drive capability. Connecting components via a transistor is not always the ideal solution though because when you are switching mains or even higher voltages, galvanic separation is often desired and this can be achieved via the use of a relay.

Relays are an electromechanic component which combines a coil and a magnetically driven switch. Chasing a current through the coil creates a magnetic field, thereby moving the switch to the closed position. This will load currrent flows over the switch, which is not connected electrically, to the relays coil. Sadly, relays can grace multiple issues. First of all, the coil can create magnetic fields, which might not be entirely acceptable in some circumstances. Secondarily, the actual switching process can cause sparks which can ignite combustible atmospheres. Finally, the coil needs quite a bit of current – use a transistor to amplify, and don't forget the kickback protection diode.

Relays are limited to switching on and off. In many cases, a more granular level of control is required. This can be achived via D/A converters: hey take a digital value and convert it into a variable voltage or a variable current.

Diana Kupfer
@dianakupfer
Veteran media manager who recently was assigned to manage a large IoT focus group. Retweets loads of valuable messages.

Run an engine

One very common real-world task involves motor driving: be it a big, beefy cable winch lifting a crate of arms, or a tiny little motor moving a print head across a rail.

Motors come in incredible variety of types: entire books have been written, cataloguing each of the subtypes. When

Enriching lives

EXPERT INSIGHT

Brad Fry

Director of Strategy & Insights at Folk
bradfry.com

By the end of 2015 it is predicted that there will be 4.9 billion connected things in use. By 2020 it estimated to reach 25 billion. Where do you think the IoT's greatest potential lies?

At the micro level the greatest potential for us as end users is in convenience. There are so many ways our technology enriches and simplifies our lives, but we need to take too many manual actions to benefit from this. When everything is connected things will just work, no buttons to tap or things to log.

Moving to the macro scale, the volume of the data collected by billions of connected devices can help profoundly. It will help improve weather models, redesign traffic systems and accurately diagnose illness earlier.

The Web of Things is an attempt to use existing and well-known web standards rather than specific manufacturer standards to help simplify the creation of IoT applications. How quickly and how effective do you think they will be?

We've been through similar standards wars in the past: with browsers, video codecs and even network protocols (anyone remember Novell? I actually liked IPX). They have all eventually converged into a standard (although Bluetooth still has a way to go).

Modern web standards now rely on compatible endpoints – dumb pipes and smart hubs – so the lack of standardisation may be less of an issue. Apple has made a big play towards acting as a intermediary in connected devices (HealthKit and HomeKit). As long as we have compatible endpoints, we can probably relax about the transport protocols uses by the devices themselves.

With more and more connected devices joining the IoT party, what will designers and developers need to consider when creating UIs for applications?

Design has been moving from complex to simple for the last decade and that will continue. We're going from the traditional engineering-led approach of making something that does everything to the user-centric things that do a few jobs brilliantly.

The challenge for developers is to make smarter AI to figure out what I want before I know it myself, like Google Now. All the big strides being made in technology today are in this area just look at Timeful (recently purchased by Google), a calendar application with artificial intelligence, or Transit (just imagine what they can do with all this Internet of Things data).

For my life with connected devices, voice is the interface of choice. As if being able to turn my heating on from bed wasn't lazy enough, I'm looking forward to the day I can declare "Hey Siri, turn the heating on" – just need to wait on HomeKit! ■

> *"As long as we have compatible endpoints, we can probably relax about the transport protocols uses by the devices themselves"*

The future

THE COMMON PROTOCOL TRENDS

Ever-rising marketing costs ensure that devices will get smarter over time: selling customers a service, rather than a gadget, increases loyalty and provides a continuous revenue stream.

LG sold a combination of smart TV and mobile phones in the hope that if clients have a TV set which works only with LG phones, they are likely to opt for the handset too. As LG is a giant in the whiteware business, we can assume this trend will continue. The IoT will standardise itself: be it by the manufacturers establishing a common protocol, or by a third party selling a cross-platform framework.

dealing with the IoT, you usually have to use one of three different types. Stepper motors can be considered fine motor skills wizards: they are extraordinarily accurate and move by a fixed amount of degrees when provided with an impulse. When combined with external gear shift, amazing levels of accuracy can be achieved.

DC motors are on the other side of the equation. Their rotation speed depends on the load and the voltage: the only way to ensure a reliable RPM involves measuring it continuously and adjusting drive voltage accordingly.

Exerting very large forces requires the use of AC motors. They come in the forms of the synchronous and the asynchronous motors – sadly, discussing them in further detail would exhaust more space. The asynchronous one is especially critical as it has still not been modelled fully: when working on an application with an asynchronous engine, you should not be ashamed to seek help from professionals.

The power bank

Power supply remains the Achilles heel of the Internet of Things: when a sensor runs out of energy, it cannot transmit any further data. Users have shown themselves to be extraordinarily hesitant to changing the battery of their gadgetry frequently. If you want to keep your clients happy, try to aim for battery lifetimes that range from six month to almost infinity.

This obviously disqualifies some battery types – NiCD and NiMH types suffer from a relatively fast rate of self-discharge. This means that the internal resistance of the battery slowly destroys the energy stored inside: if the cell sits on a shelf for a few months, it might end up empty. Modern MCUs provide low-power standby modes: When no data is to be collected, hibernation occurs. Sadly, this is but part of the solution. In practice a field of technologies, which goes by the name of energy harvesting, promises some solutions for this very problem.

Energy harvesting can take more classic or more modern forms. The most traditional form, by far, is a solar cell: it can collect a small amount of energy from artificial light, and can collect much more power once provided with sunlight. This however, is but a small part of the available options: Peltier elements can be used to generate energy from differences in ambient temperature, while more exotic systems use bloodflow or high-frequency radiation emissions to collect small amounts of energy. These will then let the attach system eke out its lifespan.

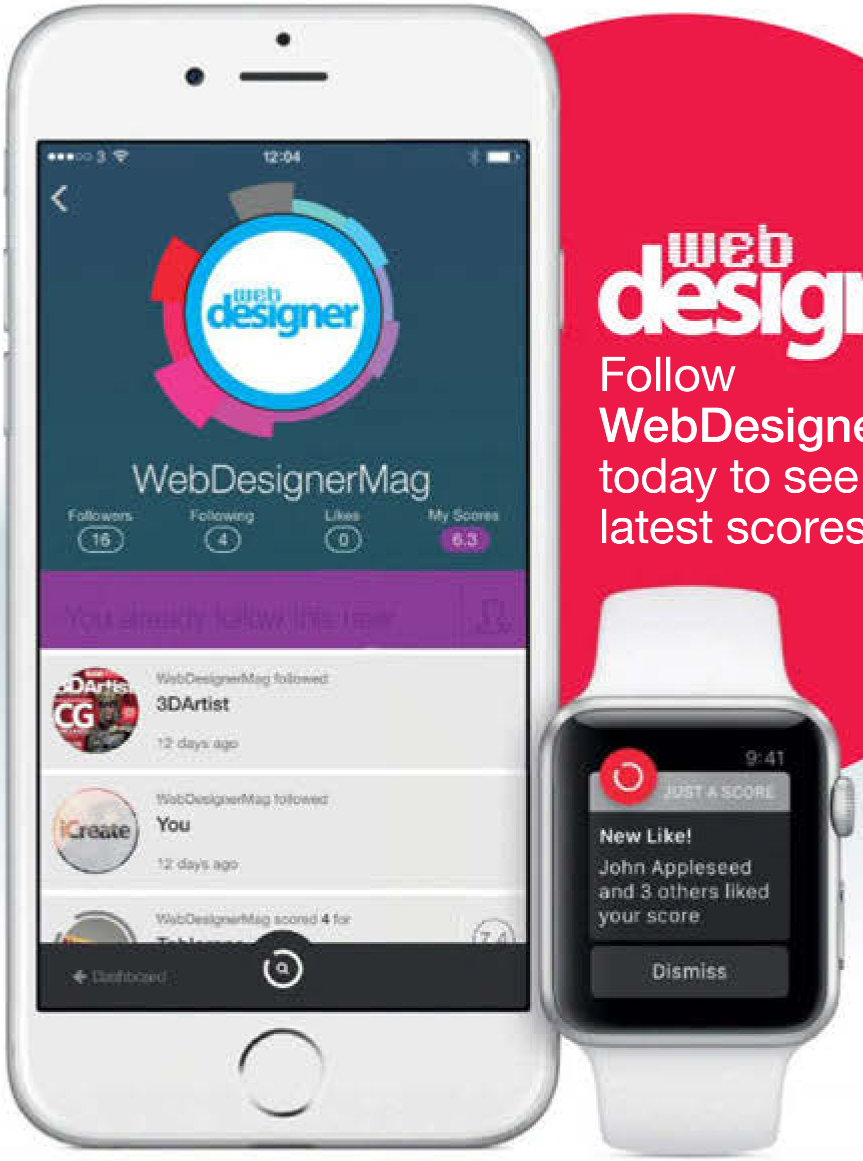
Designing energy harvest in circuits is a subject which requires a high level of knowledge in the areas of electronics. Fortunately various manufactures provide ready-made intergraded circuits, which require you to link them up with an energy source and a temporary storage medium: The rest of the work is complied inside of the intergraded circuit.

Martin Woolley
@bluetooth_mdw
Extraordinarily active developer advocate working for the Bluetooth SIG. A must-follow due to his diverse range of topics.

Follow **WebDesignerMag** on

JUST A SCORE

THE FREE NEW SCORING NETWORK



12:04

WebDesignerMag

Followers	Following	Likes	My Scores
16	4	0	6.3

You already follow this user

WebDesignerMag followed
3DArtist
12 days ago

WebDesignerMag followed
You
12 days ago

WebDesignerMag scored 4 for

9:41
JUST A SCORE
New Like!
John Appleseed and 3 others liked your score

web designer

Follow **WebDesignerMag** today to see our latest scores

- ○ It's completely free
- ○ **Get the latest review scores**
- ○ Get involved with the team
- ○ **Give your opinion instantly**
- ○ Like and share scores
- ○ **Score anything in the world**
- ○ Follow a range of scorers for your favourite topics

SCORE ANYTHING JUST A SCORE

Download on the App Store

web designer

PICK OF THE SCORES…

Love or Hate, you choose…

You can score absolutely anything on Just a Score. We celebrate our favourite agencies and sites, as well as our sporting heroes and our disdain for a certain celebrity

10 Web Designer scored **10** for
Eddie Howe

9.0 Web Designer scored **9** for
HTML5

8.0 Web Designer scored **8** for
Rally Interactive

7.0 Web Designer scored **7** for
Jeff Bridges Sleeping

2.0 Web Designer scored **2** for
Russell Brand

GET THIS FEATURE'S FULL CODE ON FILESILO •
filesilo.co.uk/webdesigner

ADVANCED ANGULAR

Get your hands on an unmissable collection of tips and techniques to take your Angular code to the next level

BE AN ANGULAR EXPERT

AngularJS is a client-side MVW framework written in JavaScript, aimed at writing single-page applications (SPAs). Initially released in 2009, Angular has surfaced as one of the most popular choices for developers. It's maintained by Google as an open source project and boasts a lot of functionality built with modern web standards in mind. MVW stands for Model-View-Whatever, which gives us flexibility over design patterns when developing applications. We might choose an MVC (Model-View-Controller) or MVVM (Model-View-ViewModel) approach with it.

Angular focuses on our data, enabling our HTML to become the dynamic language that it was certainly not built for. It brings powerful concepts and ideas from various programming languages and has server-side influences that make it a top choice for web application development at present. Angular parses our declared HTML, binds it to our models that consist of plain-old JavaScript objects and keeps everything in sync with the DOM to spring our application to life. This synchronisation of DOM and JavaScript is called two-way data-binding – it is one of Angular's most powerful features amongst others such as automated DOM creation or destruction, powerful conditional logic, automated event listeners, template encapsulation, class-like components, dependency injection (DI) and promises.

Angular focuses on a fully data-driven approach to developing applications without needing to refresh models, update the Document Object Model (DOM) and other time-consuming tasks such as ironing out browser quirks and structuring our code consistently. model data can be set manually or fetched via XMLHttpRequest (XHR), which is usually a JSON resource, making it extremely easy to integrate within Controllers.

In this feature, we've packed tons of tips that originate from expert industry experience with AngularJS in the last few years, from simple time-saving to high-end performance enhancing. We're also going to create an custom Angular filter for enabling us to filter arrays of content at speed and reflect them in the UI. directives are a huge part of Angular. We'll also be looking at creating our own simple directive, focusing on structure and good practices, built-in features and more, to take away and apply to your development techniques.

REASONS FOR DIRECTIVES AND ENCAPSULATION

Modern web standards are rapidly approaching, more specifically the Web Components standard. Web Components are an exciting new dimension of how we'll build web documents in the future, with the intention to bring a component-based approach to developing web applications. These components will comprise of many small, reuseable and possibly interlinking parts.

Encapsulate in a Web Components fashion from day one and your applications will have the ability to scale with ease, be well tested, and offer flexibility and reusability. Angular lets us pretty much do all this already (and more) through its directive API, opening up this new paradigm to developers right now.

Understanding Web Components and knowing how to implement Web Components concepts in Angular today will help you succeed in writing better components overall.

So let's see how each of these specifications is defined in the Web Components standard and how Angular lets you benefit from similar features today.

Custom Elements:

What Web Components standard says:

Lets you define your own HTML elements.

What Angular lets you do already:

Angular offers the concept of directives to let you define your own custom HTML elements. You can create new HTML elements or augment existing ones. You can also add behaviour to HTML attributes.

Shadow DOM:

What Web Components standard says:

Gives you the ability to scope markup and styles in a separate DOM tree.

What Angular lets you do already:

Shadow DOM requires browser support so it's technically not possible for Angular to implement it if your browser does not support it. However Angular already provides similar benefits from a developer standpoint. Angular enables you to encapsulate markup using a HTML template, logic using a directive controller and content using an isolated scope. It also offers transclusion, which can be thought of as a single insertion point from the Shadow DOM specification.

Used for encapsulating and scoping DOM, Angular mimics this behaviour by compiling HTML and injecting it into our DOM.

The idea behind Shadow DOM is to separate content from presentation, creating a clear distinction between where our templates contain placeholders require content, and the content itself to be parsed and injected. Shadow DOM is technically a 'DOM within DOM', it's entirely scoped and you can create as many instances of each component as you need.

HTML Imports:

What Web Components standard says:

Provides a way to include and reuse HTML documents in other HTML documents.

What Angular lets you do already:

Our components need dependencies, so we need to import them using HTML Imports. Angular actually does all this for us, and we tie our imports much closer to the component definition itself with Angular, as the HTML Imports standard actually requires us to import in the <head> of our document.

HTML Templates:

What Web Components standard says:

Enables you to define blocks of markup with the ability to inject dynamic content into.

What Angular lets you do already:

Angular enables us to use string templates or reference HTML files, enabling us to use a method best for our team, us or the project. Our templates contain no content, Angular uses {{ handlebar }} syntax for adding placeholders for where our content will be injected, compiled and finally inserted in the DOM.

BUILD A REUSEABLE FILTER

Filters are really powerful ways to manipulate data. A filter could parse data into a different output, such as converting millisecond dates into a human readable format, or filter content (like lists) by returning items that match certain criteria. Let's look at the Angular .filter() API, create a dummy email list and make an ng-repeat filter.

1. Change the drawing radius

Let's hook into the Angular module that we've made called 'app' and extend it further. Inside the filter's callback, return another function – this function is called with filter arguments and will return a filtered response.

```
angular
.module('app', [])
.filter('important', function important() {
return function () {
//
};
});
```

2. Hook up the DOM

Create an expected ng-repeat with our filter intentions in place. There's an email in vm.emails and a pipe | which is where we declare the filter name 'important'.

```
<ul>
<li ng-repeat="email in vm.emails |
important" ng-class="{important: email.
important}">
{{ email.label }}
</li>
</ul>
```

Ng-class here adds 'important' class for emails with an 'important: true' property to help style them differently.

3. Pass in the array to be filtered

Next allow for some arguments, and name the first one 'items'. Let's 'return' the items to let the filter work, as we don't return anything, Angular won't get an array back to rerender the DOM with our updated filter.

```
angular
.module('app', [])
.filter('important', function important() {
return function (items) {
return items;
};
});
```

4. Define important properties

Assuming we have data coming from our backend, we can make it available in our Controller, and pass it to our filter. Before it hits the filter, we need to make sure we have an 'important' property containing a Boolean value.

```
angular
.module('app', [])
.controller('MainCtrl', function MainCtrl()
{
this.emails = [{
label: 'Item dispatched',
important: true
},{
label: 'Reset your password',
important: false
},{
label: 'Renew your home insurance',
important: true
},{
label: 'Welcome to Amazon!',
important: false
},{
label: 'Angels & Airwaves ticket
confirmation #402881',
important: true
},{
label: 'TravisCI test passing',
important: false
}];
});
```

5. Return the important objects

Our filter function now only looks at returning the important emails. Create a new empty array called 'filtered', loop through the 'items' (which Angular will pass in the bound ng-repeat for us) and check if the important property is 'true'. If 'true', we use array.prototype.push to add that object into the filtered array, and finally return it once iteration is complete, giving us all important objects.

```
.filter('important', function important() {
return function (items, important) {
var filtered = [];
for (var i = 0; i < items.length; i++) {
if (items[i].important) {
filtered.push(items[i]);
}
}
return filtered;
};
})
```

6. Toggle capability

We could make the filter better by adding in an optional parameter to enable us to toggle between 'All' emails and 'Important' emails in the same list. We can pass in a second argument to the DOM filter using a colon to separate arguments. Let's add a checkbox to enable us to reflect the showImportant model Boolean value. Once checked, showImportant becomes 'true', and our filter will run again with the second argument available as 'true', enabling us to decipher between whether the UI is requesting 'all' or 'important' emails.

DEPENDENCY INJECTION ANNOTATION SYNTAX

Dependency Injection (DI) in Angular is fairly simple, we just pass the dependency names that we want to the function as arguments, and Angular injects them in for us to use. Behind the scenes Angular converts our functions to a string and then it reads any dependencies that we're requesting. Note that at this point the arguments cannot be renamed.

When we minify our code (and we should), the argument names will be minified and Angular won't be able to interpret them correctly, so we need to explicitly tell Angular what dependencies we're requesting, and in what order we need them.

There are several ways to inject our dependencies to keep them safe when minified, for example by making use of the inline $inject syntax or array syntax. With the array syntax, we will need to pass the dependencies into our callback as an array, which will then hold the function as a final argument.

```
angular
.module('app', [])
.controller('MainCtrl', ['$scope',
'$rootScope', function MainCtrl($scope,
$rootScope) {
//
}]);
```

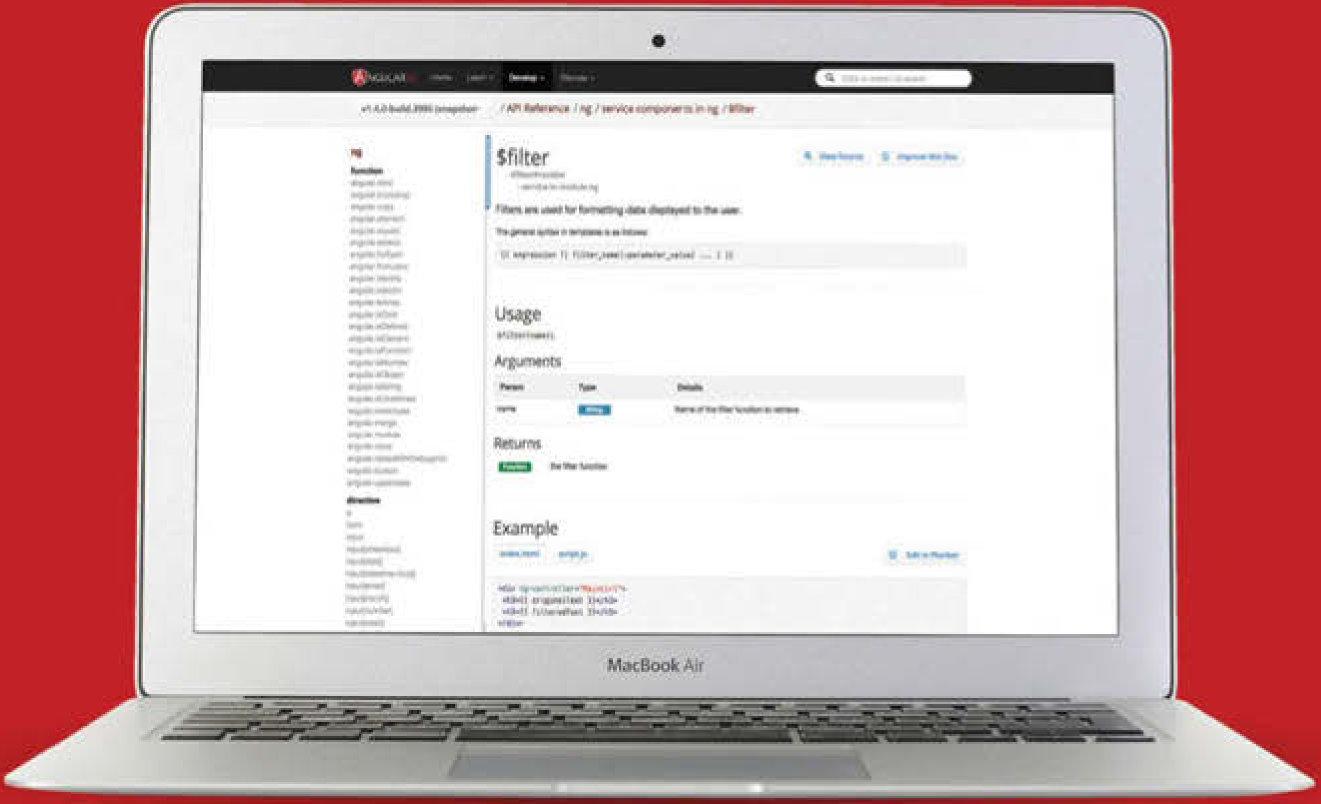
With the $inject syntax, we can create a property on the function itself, which will offer us more flexibility if we want to make the code slightly less callback-looking and enhance readability.

```
function MainCtrl($scope, $rootScope) {
//
}
MainCtrl.$inject = ['$scope', '$rootScope'];
angular
.module('app', [])
.controller('MainCtrl', MainCtrl);//
}]);
```

We can alternatively use a mix of both if preferred (using the array syntax but also passing a function in).

```
function MainCtrl($scope, $rootScope) {
//
}
angular
.module('app', [])
.controller('MainCtrl', ['$scope',
'$rootScope', MainCtrl]);
```

Use ng-annotate to automate both of these syntaxes so we can save a lot of time typing, it's clever enough to know whether to use $inject or the array syntax. You can install ng-annotate via NPM, point it to your project files, let it compile and automatically add the dependency injection annotations for you.



```
<input type="checkbox" ng-
model="showImportant">
<ul>
<li ng-repeat="email in vm.emails |
important:showImportant" ng-
class="{important: email.important}">
{{ email.label }}
</li>
</ul>
```

7. Augment arguments

As we've declared another argument in the DOM, we can get another argument available in the JavaScript filter code, aliased as 'important'. Add an if statement to check if 'important' is passed in, otherwise return all array items.

```
.filter('important', function important() {
return function (items, important) {
if (important) {
// return only important emails
}
// if important is false, we'll return all
items
return items;
};
})
```

8. Refactor our filtering

Now we have an 'important' flag being passed in, we can move our earlier code into the 'if' statement, and run our filter when we want all our important Objects back, otherwise we'll just assume the user requests 'all' and send the entire collection back.

```
.filter('important', function important() {
return function (items, important) {
if (important) {
var filtered = [];
for (var i = 0; i < items.length; i++) {
if (items[i].important) {
filtered.push(items[i]);
}
}
return filtered;
}
return items;
}; })
```

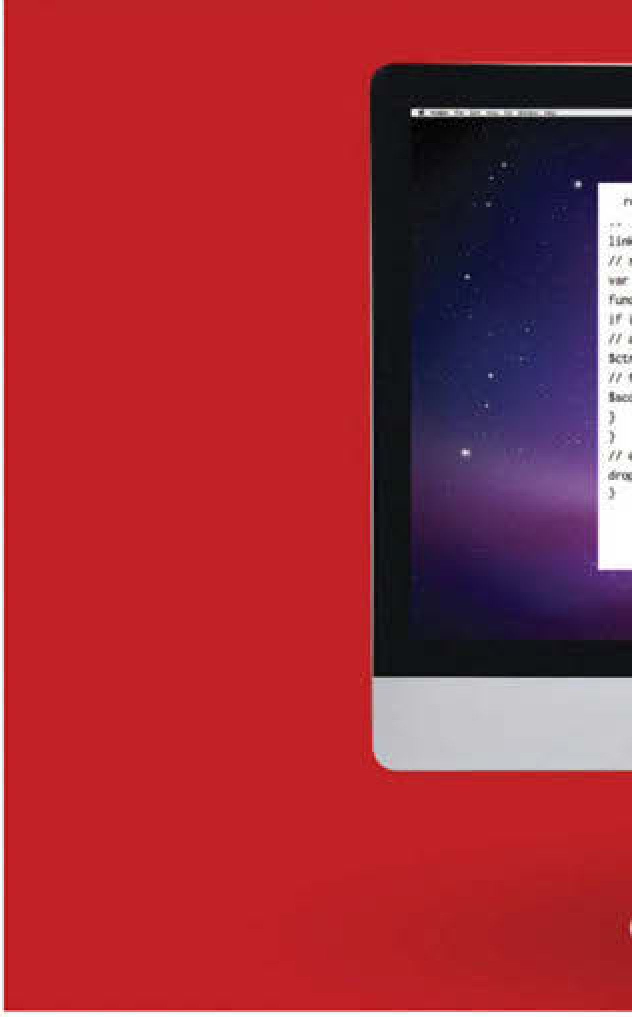
ONE-TIME BIND WHERE IT IS NEEDED

Angular 1.3 introduced a brilliant feature called one-time binding. Using one-time binding is good as it removes $$watchers from the $digest cycle after they've been parsed and populated. If the data is a one-time static render then it can be bound once and removed from further $digest loops, this will help to keep future loops much lighter and therefore faster for the JavaScript to look up any change. This feature will also enable us to remove the value from Angular $digest cycle (which contains all our data to check states when any model values change) as soon as it becomes anything other than 'undefined'.

The $digest cycle can become heavy and slow down our app, which isn't very helpful if we only need to render our data once, for example with a dynamically populated navigation, static list content or view titles. To specify that an expression only needs to be bound once, we can prefix the expression with '::', like '{{ ::name }}'.

MATT GIFFORD
monkehworks.com
Consultant developer,
Monkeh Works Ltd.
@coldfumonkeh

> "AngularJS is not just for single-page web applications or browser-based sites. The implementation of AngularJS within development stack of frameworks, for example, as Ionic enhances the effectiveness of PhoneGap or Cordova mobile development."



LINK FUNCTIONS FOR DOM

It may be extremely tempting to litter your Controllers with DOM manipulation, especially when integrating things such as plugins or third-party scripts and you need to set or get values. Directives are a perfect way to encapsulate any necessary presentational logic, as they include a Controller, and offer a gateway to the DOM through the 'link' function. This 'link' function gives us the root element for the directive, so that we can actually access any element inside it and bind things such as ontouchstart listeners for example, which currently aren't part of the Angular core.

When using a Controller alongside a directive, Angular passes it to the 'link' function as a fourth argument, so we can run any presentational logic from callbacks to raw DOM manipulation.

Remember that inside raw DOM listeners we will need to run $scope.$apply to tell Angular to look at any new values. As an example (excluding most directive properties) it is possible to pass in a Controller, aliased $ctrl to access presentational methods inside our raw DOM event listeners.

```
return {
..
link: function postLink($scope, $element,
$attrs, $ctrl) {
// some code to fetch files from a made up
drag/drop uploader
var drop = $element.find('.drop-zone')[0];
function onDrop(e) {
if (e.dataTransfer && e.dataTransfer.files)
{
// assumes “uploadFiles” method handles the
upload!
$ctrl.uploadFiles(e.dataTransfer.files);
// force a $digest cycle
$scope.$apply();
}
}
// events
drop.addEventListener('drop', onDrop,
false);
} .. }
```

UNBIND $ROOTSCOPE.$ON LISTENERS

You'll likely find the need at some stage during your Angular career to use the internal events system, $emit, $broadcast and $on. These event methods are available in both the $rootScope and $scope.

If we change views in our application, Angular will destroy our Controller and thus it will destroy the $scope object associated with it. This is great, however, all $scope objects are child objects that will inherit from the $rootScope, and this will then persist throughout the entire application.

We will need to manually unbind $rootScope listeners by calling the returned closure function when the $scope is destroyed by listening to the $destroy event, otherwise when we revisit a view where the same Controller is used, the $rootScope.$on will be bound again. This can then cause duplicated events and also some issues with the data syncing.

```
var unbind = $rootScope.$on(‘fooEvent’,
function () {});
$scope.$on(‘$destroy’, unbind);
```

With multiple events, we tend to use an array and loop through to automatically unbind.

```
[
$rootScope.$on(‘fooEvent’ function () {}),
$rootScope.$on('barEvent, function () {})
].forEach(function (unbind) {
$scope.$on('$destroy', unbind);
});
```

ABSTRACT BUSINESS LOGIC INTO SERVICES

If you've ever been tempted to use things like $http (Angular's XHR wrapper) inside a Controller, then it's time to understand the patterns you're using. A Controller is a glorified ViewModel, a $scope object, which consists of maintaining and presenting (you guessed it) the presentational layer. Controllers should only withhold presentational logic.

Angular provides us with service, factory and provider APIs, to delegate business logic into. This makes sure our presentational layer is separated clearly from our business logic. Business logic would usually hit an API endpoint, fetch data and a Controller will make a copy of that data, pass it to the $scope for Angular to express what our model currently looks like.

AVOID DOM FILTERING

Using a filter in the HTML of Angular is great, but can impact performance. Filters can be easily added using a pipe inside an expression like so:

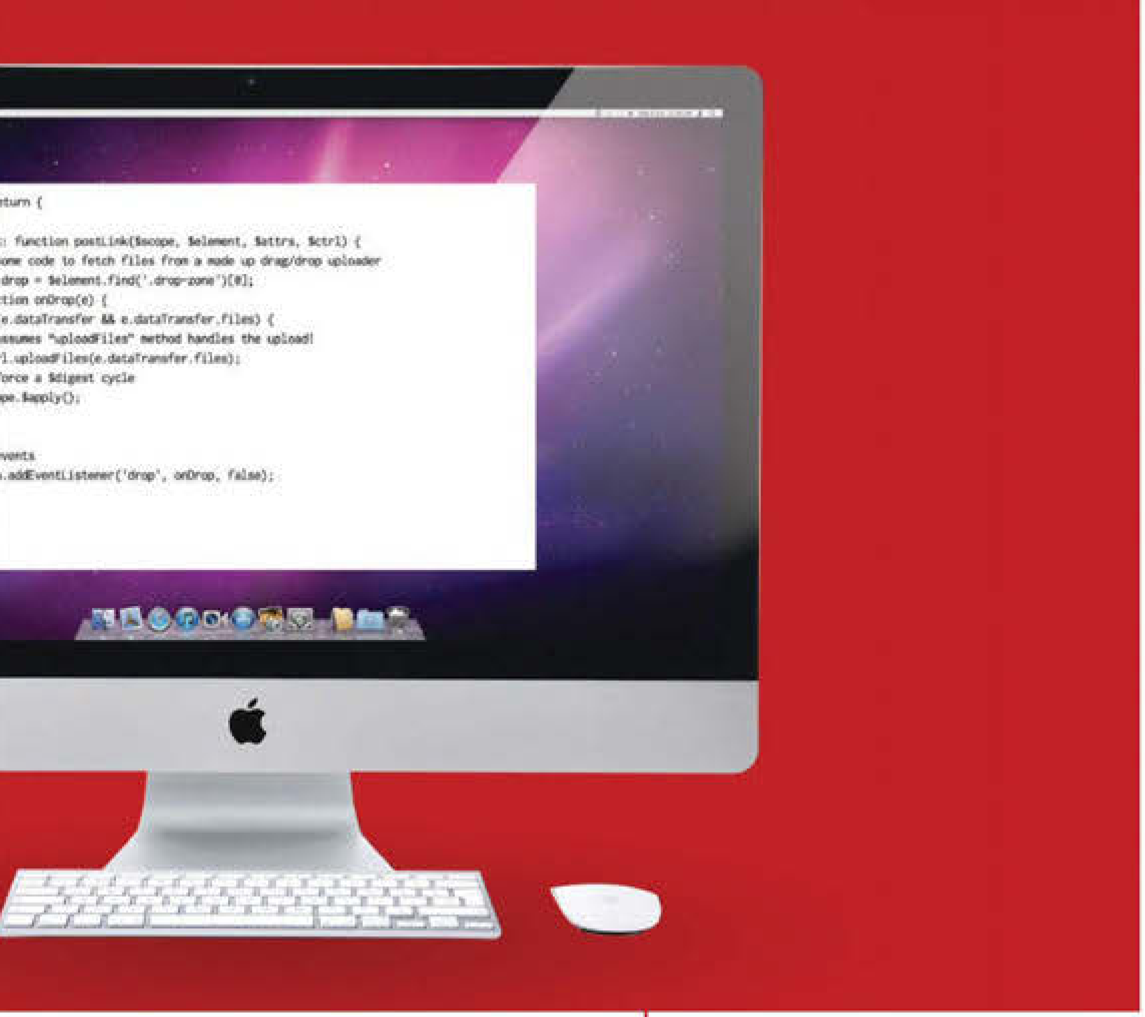
```
{{ someDateValue | filter:date }}
```

That's an example of using Angular's built-in date filter, we can however write our own filters. Filters are fairly common on ng-repeat declarations however, and can be huge performance killers.

Filters run on every $digest loop and will create a new array every time that it runs. Watch for changes to a specific model (such as a user typing) or a user interaction (such as a user clicking a button) and then run a filter inside the Controller's JavaScript using the $filter method, which will manually run your own filter upon detection of any change.

TIM STONE
fetimo.com
Lead front-end developer, Redweb.
@timofetimo

“Updating your UI to react to changes can be a messy, hard to maintain mess. Angular abstracts this by dealing with the DOM for you. You then start to think of your applications state, which is a fundamental shift.”



```
turn {

: function postLink($scope, $element, $attrs, $ctrl) {
ome code to fetch files from a made up drag/drop uploader
drop = $element.find('.drop-zone')[0];
tion onDrop(e) {
e.dataTransfer && e.dataTransfer.files) {
ssumes "uploadFiles" method handles the upload!
l.uploadFiles(e.dataTransfer.files);
orce a $digest cycle
pe.$apply();


vents
.addEventListener('drop', onDrop, false);
```

$ROOTSCOPE. $APPLYASYNC FOR BATCHING XHR

The Angular 1.3 release saw another quietly brilliant feature, $rootScope.$applyAsync for intelligent XHR batching. Angular uses 'dirty-checking' to make sure all values are up to date, it does this by running what it calls a $digest cycle. The $digest can have a lot of data in, making UI response quite slow at times due to high watcher count. If we have multiple XHR calls happening around the same time, Angular is going to run a $digest after each of these to make sure all of its values are up to date. That's a lot of potential $digest loops happening around the same time. Using $applyAsync mitigates this issue and can result in significant performance improvements for apps making many concurrent HTTP requests. It's easy to use, so why not just drop it in to your config to see how it works?

```
angular
.module('app')
.config(function ($httpProvider) {
$httpProvider.useApplyAsync(true);
});
```

UNIT TEST ANGULAR MOCKS

Karma can run your unit tests whether in a physical browser (such as Chrome) or headless (PhantomJS). Angular mocks (from the Angular core team) extends Angular with extra functionality that you may need inside your unit tests. For example, Angular mocks makes it easy to test Controllers, directives and services (and more) by letting you mock HTTP requests, test async methods and simulate responses.

WHEN TO USE NG-BIND OVER HANDLEBAR EXPRESSIONS

Angular provides us two ways to insert text into our HTML, using the ng-bind or expression {{ }} syntax. An issue you will face if you have HTML content already on the page whilst the browser is parsing the HTML, before Angular has even loaded, is you'll see these curly expression braces, which then populate with data as Angular springs into life. To get around this we can use ng-bind, which will inject the value once Angular has loaded, and we'll just see an empty element which is a better experience than expression tags everywhere.

```
<!-- expressions -->
<p>{{ foo }}</p>
<!-- ng-bind -->
<p ng-bind="foo"></p>
```

Any views that are loaded with Angular routers, such as ui-router or ngRouter will compile all HTML and data before injecting the view, so you only have to cater for this issue on initial load.

FAVOUR NG-HREF ATTRIBUTES OVER HREF

Avoid 404 errors whilst the browser downloads content with expressions in the HTML. The browser will begin downloading <img src="images/{{ profile }}.jpg"> rather than the parsed value (ie app/images/niels.jpg). Using ng-href will overcome this, as Angular will populate and set href attributes for us after the expressions have been evaluated.

STANDARDISE HTML BINDING SYNTAX

Angular promotes the use of their directives like ng-click. However ng-click can also be written as x-ng-*, ng:*, ng_* and data-ng-*. If you need to be standards-compliant with your attributes, use data-ng-*, though this is most commonly seen as ng-* prefixed.

NAME ANONYMOUS FUNCTIONS

If you're passing in anonymous functions into Angular's API, such as .controller() then name the function you are passing in. This allows '. controller('MainCtrl', function() {...});' to become '. controller('MainCtrl', function MainCtrl() {...});', which really aids in stack-trace debugging in the dev tools.

BE RESPONSIBLE AND AVOID LONG API CHAINS

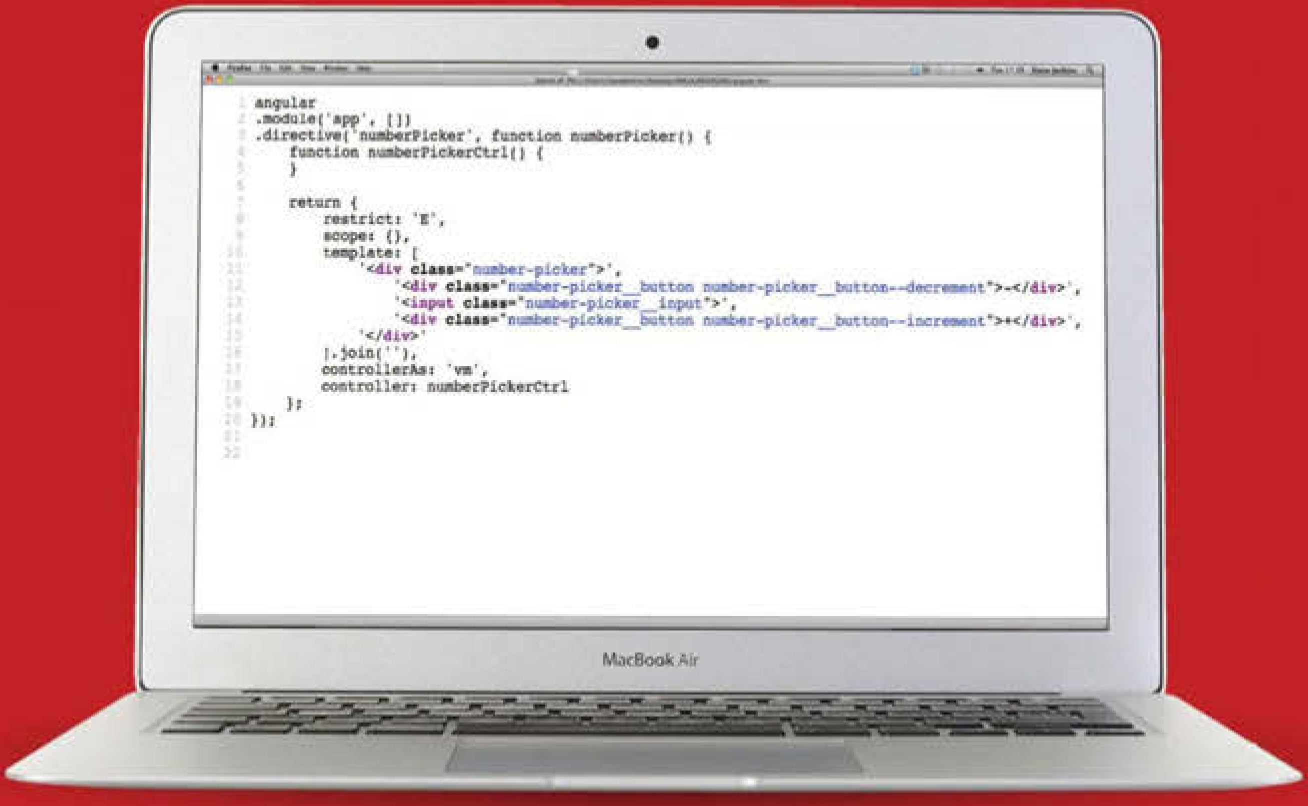
Aim to keep one component per file, a single responsibility. Only chain what is necessary, for instance a MainCtrl.js file will contain:

```
angular.module('app')
.controller('MainCtrl', function MainCtrl()
{...});
```

DOWNLOAD AND USE BATARANG

Brian Ford (Angular core team member) launched Batarang for AngularJS developers. Batarang is very useful as it "extends the Chrome Developer Tools, adding tools for debugging and profiling AngularJS applications."

> *"The $digest can have a lot of data in, making UI response quite slow at times due to high watcher count. If we have multiple XHR calls happening around the same time, Angular is going to run a $digest after each of these to make sure all its values are up to date. That's a lot of potential $digest loops happening"*



```
angular
.module('app', [])
.directive('numberPicker', function numberPicker() {
    function numberPickerCtrl() {
    }

    return {
        restrict: 'E',
        scope: {},
        template: [
            '<div class="number-picker">',
                '<div class="number-picker__button number-picker__button--decrement">-</div>',
                '<input class="number-picker__input">',
                '<div class="number-picker__button number-picker__button--increment">+</div>',
            '</div>'
        ].join(''),
        controllerAs: 'vm',
        controller: numberPickerCtrl
    };
});
```

REUSEABLE DIRECTIVE FOR INCREMENTING OR DECREMENTING NUMBERS

This directive will use two-way data binding to take care of updating a parent model from any $scope, through the use of isolate scope. Encapsulating logic enables us to reuse it, and isolating $scope enables us to use the same directive in many different places.

1. Custom elements

Before we begin, let's create a custom element named 'number-picker', which will have an attribute called 'value' (this can be named something different). The 'expression' is just a placeholder for a model value we're passing later.

```
<number-picker value=”expression”></
number-picker>
```

2. Directive declaration and HTML

Next we'll create a basic directive definition, and add a template property. Directives can use templates from a string or from an external file using templateUrl (pointing to a relative URL). Let's set up the base directive properties and add our HTML template in.

```
angular
.module('app', [])
.directive('numberPicker', function
numberPicker() {
function numberPickerCtrl() {
}
return {
restrict: 'E',
scope: {},
template: [
'<div class="number-picker">',
'<div class="number-picker__button number-
picker__button--decrement">-</div>',
'<input class="number-picker__input">',
'<div class="number-picker__button number-
picker__button--increment">+</div>',
'</div>'
].join(''),
controllerAs: 'vm',
controller: numberPickerCtrl
};
});
```

3. Isolate scope binding

We need to pass the updated value in the directive back to where it was initialised after changes have taken place, this lets us keep other data models in the parent Controller updated. Pass the value in via the scope property, creating a new object to initiate 'isolate scope', and creating a property name called 'value'. We can use = as the value, which tells Angular to bind the value two ways, back up the parent Controller.

```
.directive('numberPicker', function
numberPicker() {
 ...
return {
...
scope: {
value: '='
},
...
};
});
```

4. Controller methods

We need to implement methods to increase or decrease our number inside the Controller. As we're using the controllerAs syntax, we need to attach these to the function instance using the 'this' keyword rather than injecting $scope.

```
angular
.module('app', [])
.directive('numberPicker', function
numberPicker() {
function numberPickerCtrl($scope) {
this.increment = function () {
};
this.decrement = function () {
};
}
...
});
});
```

5. Model manipulation

As our value is passed through $scope, we access it using $scope.value to pass back up to the parent model and not with this.value.

```
angular
.module('app', [])
.directive('numberPicker', function
numberPicker() {
function numberPickerCtrl($scope) {
this.increment = function () {
++$scope.value;
};
this.decrement = function () {
--$scope.value;
};
}
...
});
```

6. Controller methods

Now we've got Controller methods to modify our value, we need to make sure our HTML template can talk to those methods through Angular's built-in directives, we need ng-click. As we're using controllerAs, the Controller is instantiated as a reference object in the DOM as vm (ViewModel) so we can access our public increment or decrement methods through the vm object.

```
angular
.module('app', [])
.directive('numberPicker', function
numberPicker() {
...
return {
...
template: [
'<div class="number-picker">',
'<div class="number-picker__button number-
picker__button--decrement" ng-click="vm.
decrement()">-</div>',
'<input class="number-picker__input"
ng-model="value">',
'<div class="number-picker__button number-
picker__button--increment" ng-click="vm.
increment()">+</div>',
'</div>'
].join(''),
...
};
});
```

7. Add $watch to validate input

As we enable the user to type into the input, we need to ensure that they only enter a number. We can use $watch to watch changes to the value, and reset it back to its old value if it's anything other than a number. We can use a simple Regular Expression to test if anything isn't a number, and revert it back to the previous value. This is just basic validation, a more comprehensive solution may fit your needs in a production application. We could use ng-pattern, but it still enables users to type anything, and Angular will refuse to set the model unless the element matches the pattern.

```
angular
.module('app', [])
.directive('numberPicker', function
numberPicker() {
function numberPickerCtrl($scope) {
$scope.$watch(function () {
return $scope.value;
}, function (newVal, oldVal) {
if (!/^[0-9]+$/.test(newVal)) {
$scope.value = oldVal;
}
});
...
}
...
});
```

8. Link function keyboard input

We're also going to let the user press up or down to increment or decrement the number. We can do this via the 'link' property on the directive, which gives us access to the root node of the directives instance, as well as the $scope and any attributes it has. Set up placeholders for handling up and down arrow keys through the keyCode property on the events passed to us.

```
angular
.module('app', [])
.directive('numberPicker', function
numberPicker() {
function numberPickerCtrl($scope) {
$scope.$watch(function () {
return $scope.value;
}, function (newVal, oldVal) {
if (!/^[0-9]+$/.test(newVal)) {
$scope.value = oldVal;
}
});
this.increment = function () {
++$scope.value;
};
this.decrement = function () {
--$scope.value;
};
}
function numberPickerLink($scope, $element,
$attrs) {
function handleKeyUp(e) {
if (e.keyCode === 38) {
}
if (e.keyCode === 40) {
}
}
$element.find('input').on('keyup',
handleKeyUp);
}
});
```

9. Pass in Controller

You can also pass through the Controller into the 'link' function as the fourth function argument (aliased as $ctrl) to make a clear distinction between where we hold presentational logic and DOM logic. We should ideally only use 'link' for raw DOM communication. This means we call our existing methods to manipulate the value, without polluting our Controller with DOM logic – a big antipattern in Angular. We need to run $scope.$apply() after raw DOM APIs to inform Angular of changes and to update any values that we updated ourselves.

10. Final touches

Let's see what everything looks like together and check FileSilo for the full code. Using the expression "{{ vm.value }}" we can get updates as the user types which are instantly reflected back up to the parent Controller, as it's the property delegated to the directive, manipulated and passed back up.

TODD MOTTO
@toddmotto
The Google developer expert and director of front-end engineering at Mozio gives us five Angular tips.

USE DIRECTIVECTRL STRING METHODS

Instead of assigning a Controller and using an anonymous/assigned function, use a string-based approach. This references a Controller created using the .controller() method and makes the Controller reuseable elsewhere in the application and testing much easier.

APPLY NG-CLOAK CLASS NAME

Elements with conditional statements such as ng-if, ng-show or ng-hide are sometimes visible until Angular has evaluated the expression. We can use ng-cloak attribute, which is essentially styled as display: none;. After Angular has evaluated expressions, it removes all ng-cloak attributes leaving the element in its correct conditional state.

ROUTING 'RESOLVE' PROPERTY

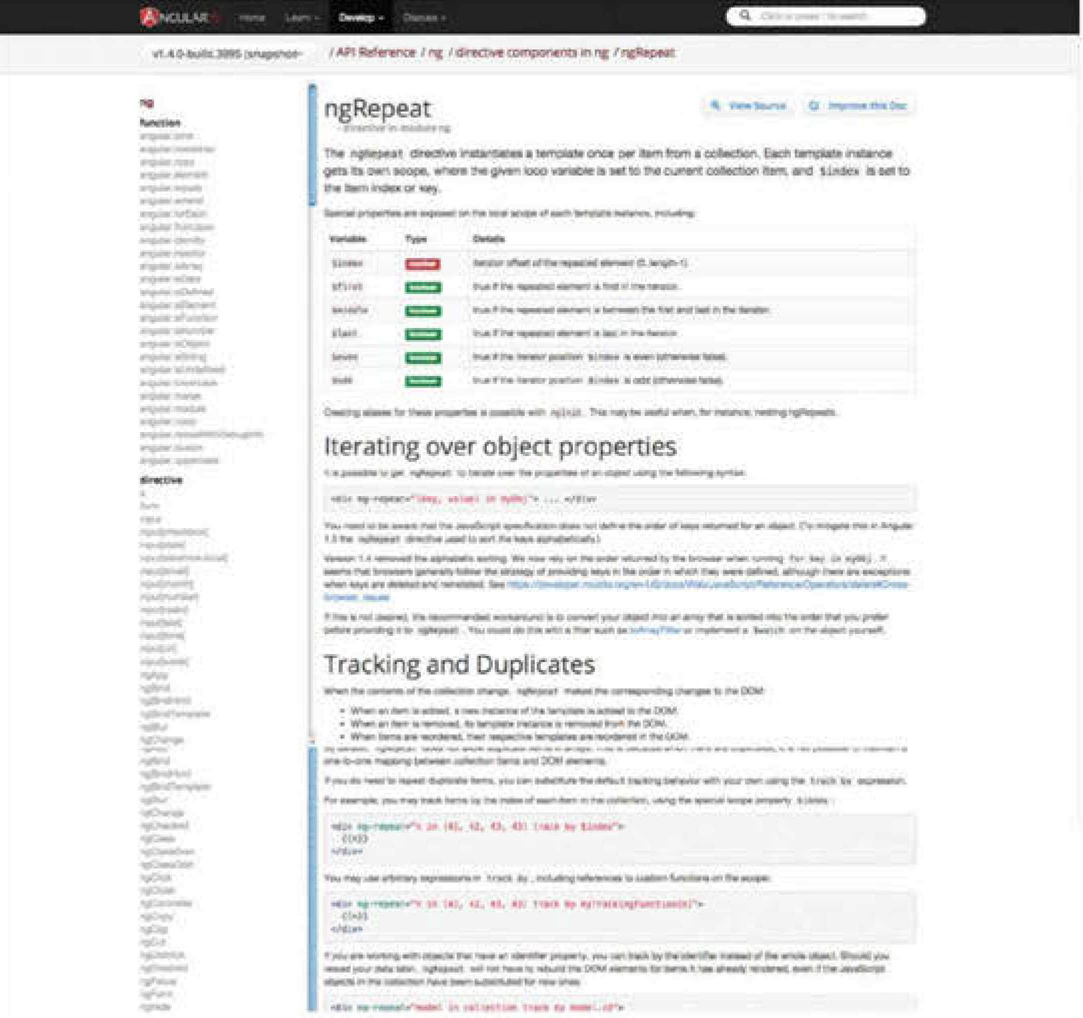
Ensure asynchronous operations are complete by the time Angular switches views, this ensures all data is fetched prior to rendering the view. The resolve property sits on both the Angular router (ngRouter) and ui-router. We can pass in services that resolve a promise, these resolved promises are then given to us in the Controller through dependency injection.

$SCOPE IS JUST A VIEWMODEL

Treat your $scope as a ViewModel at all times, never use DOM logic or even think to include jQuery! All a Controller should do is make a copy of data that's provided to it from a service, and express that model through Angular's ng-* directives. Angular Controllers are our presentational layer, and shouldn't be mixed with DOM logic or business logic.

TESTABLE DIRECTIVES THROUGH ABSTRACTION

Lessen the impact of testing directives by abstracting as much as you can into Controllers and services (providing the architectural pattern is correct for your code). This enables us to bullet-proof our workflow by writing tests for the Controller and service independently, and it also enables our directive test to be smaller and less data focused.


ANGULARJS
Home
Learn
Develop
Discuss
Click or press / to search
v1.4.0-build.3995 (snapshot)
/ API Reference / ng / directive components in ng / ngRepeat
ng
function
directive
View Source
Improve this Doc

ngRepeat

- directive in module ng

The ngRepeat directive instantiates a template once per item from a collection. Each template instance gets its own scope, where the given loop variable is set to the current collection item, and $index is set to the item index or key.

Special properties are exposed on the local scope of each template instance, including:

Variable	Type	Details
$index	number	iterator offset of the repeated element (0..length-1)
$first	boolean	true if the repeated element is first in the iterator.
$middle	boolean	true if the repeated element is between the first and last in the iterator.
$last	boolean	true if the repeated element is last in the iterator.
$even	boolean	true if the iterator position $index is even (otherwise false).
$odd	boolean	true if the iterator position $index is odd (otherwise false).

Creating aliases for these properties is possible with ngInit. This may be useful when, for instance, nesting ngRepeats.

Iterating over object properties

It is possible to get ngRepeat to iterate over the properties of an object using the following syntax:

```
<div ng-repeat="(key, value) in myObj"> ... </div>
```

You need to be aware that the JavaScript specification does not define the order of keys returned for an object. (To mitigate this in Angular 1.3 the ngRepeat directive used to sort the keys alphabetically.)

Version 1.4 removed the alphabetic sorting. We now rely on the order returned by the browser when running for key in myObj. It seems that browsers generally follow the strategy of providing keys in the order in which they were defined, although there are exceptions when keys are deleted and reinstated. See https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Operators/delete#Cross-browser_issues

If this is not desired, the recommended workaround is to convert your object into an array that is sorted into the order that you prefer before providing it to ngRepeat. You could do this with a filter such as toArrayFilter or implement a $watch on the object yourself.

Tracking and Duplicates

When the contents of the collection change, ngRepeat makes the corresponding changes to the DOM:

- When an item is added, a new instance of the template is added to the DOM.
- When an item is removed, its template instance is removed from the DOM.
- When items are reordered, their respective templates are reordered in the DOM.

By default, ngRepeat does not allow duplicate items in arrays. This is because when there are duplicates, it is not possible to maintain a one-to-one mapping between collection items and DOM elements.

If you do need to repeat duplicate items, you can substitute the default tracking behavior with your own using the track by expression.

For example, you may track items by the index of each item in the collection, using the special scope property $index:

```
<div ng-repeat="n in [42, 42, 43, 43] track by $index">
  {{n}}
</div>
```

You may use arbitrary expressions in track by, including references to custom functions on the scope:

```
<div ng-repeat="n in [42, 42, 43, 43] track by myTrackingFunction(n)">
  {{n}}
</div>
```

If you are working with objects that have an identifier property, you can track by the identifier instead of the whole object. Should you reload your data later, ngRepeat will not have to rebuild the DOM elements for items it has already rendered, even if the JavaScript objects in the collection have been substituted for new ones:

```
<div ng-repeat="model in collection track by model.id">
```

REFERENCE FILTERED OBJECTS/ ARRAYS OUTSIDE NG-REPEAT

Using an ng-repeat is one of the most powerful features in Angular, enabling us to clone template portions (such as <li> elements) and populate them dynamically from model data. These repeats can be used in conjunction with a filter, for instance if the user types to search for a user in a set of users.

Angular will rerender the DOM as the user types to gradually show less results until that user is found (like a fuzzy search for instance). This filtered list becomes scoped to the ng-repeat, which is fantastic, but we might want to use this filtered result elsewhere on the page, for instance we may want to show how many users were found in the results.

Here's what our initial code might look like:

```
<div ng-app="app">
<div ng-controller="MainCtrl as vm">
<input type="text" ng-model="search"
placeholder="Search emails">
<ul>
<li ng-repeat="email in vm.emails |
filter:search">
{{ email.label }}
</li>
</ul>
Emails found: {{ vm.emails.length || 0 }}
</div>
</div>
```

Let's create fake data to populate the "vm.emails" array:

```
angular
.module('app', [])
.controller('MainCtrl', function MainCtrl()
{
this.emails = [{
label: 'Item despatched'
},{
label: 'Reset your password'
}];
});
```

In the DOM, our Controller is instantiated as vm, which means that the this.emails inside of our Controller becomes vm.emails. We also have a section 'Emails found' which will contain vm.emails.length. If a user begins typing, the vm.emails.length never changes when we search, even though the DOM is updated to reflect any emails found. To get access to this, we will need to create a new $scope variable on the ng-repeat declaration, we call it 'filterEmail' and then assign the filter value to it:

```
<li ng-repeat="email in filterEmails = (vm.
emails | filter:search)">
...
</li>
```

As this scope is created locally by the DOM, we have access to it inside our scope, which means we can change our HTML that's trying to output the search result length to this:

```
Emails found: {{ filterEmails.length || 0 }}
```

This now works perfectly, and the filterEmails property gets updated with our filtered results as the user types. This is just one example of how to use the data, but we have access to the entire filtered array if we need it.

ANGULAR RESOURCES

NG-BOOK

NG-BOOK.COM

Mastering Angular is no easy task, but this book goes a long way in helping the reader become an expert. Touted as "The Complete book on AngularJS", and with over 500 pages, this book covers the framework from nearly every angle. Get your hands on this to take your skills to the next level.

ANGULARJS STYLEGUIDE

GITHUB.COM/TODDMOTTO/ANGULARJS-STYLEGUIDE

Need some assistance with AngularJS and style? Then check out this Github repository. It provides a standardised approach for developing AngularJS applications in teams. The styleguide touches on concepts, syntax, conventions and is put together by an expert Google developer.

EGGHEAD.IO

EGGHEAD.IO

Videos add another dimension to learning, it presents what is happening in real-time with any narrative being explained. This is how Egghead.io approaches learning. It provides bite-sized training videos like 'angular-formly: Extending Types', 'Using ng-messages with ng-animate' and 'Introduction to ng-messages', all typically under five minutes long.

TO FOLLOW

John Papa
@john_papa
Angular evangelist, Pluralsight author and conference speaker

Jurgen Van de Moere
@jvandemo
Front-end architect, creator of Angular Express

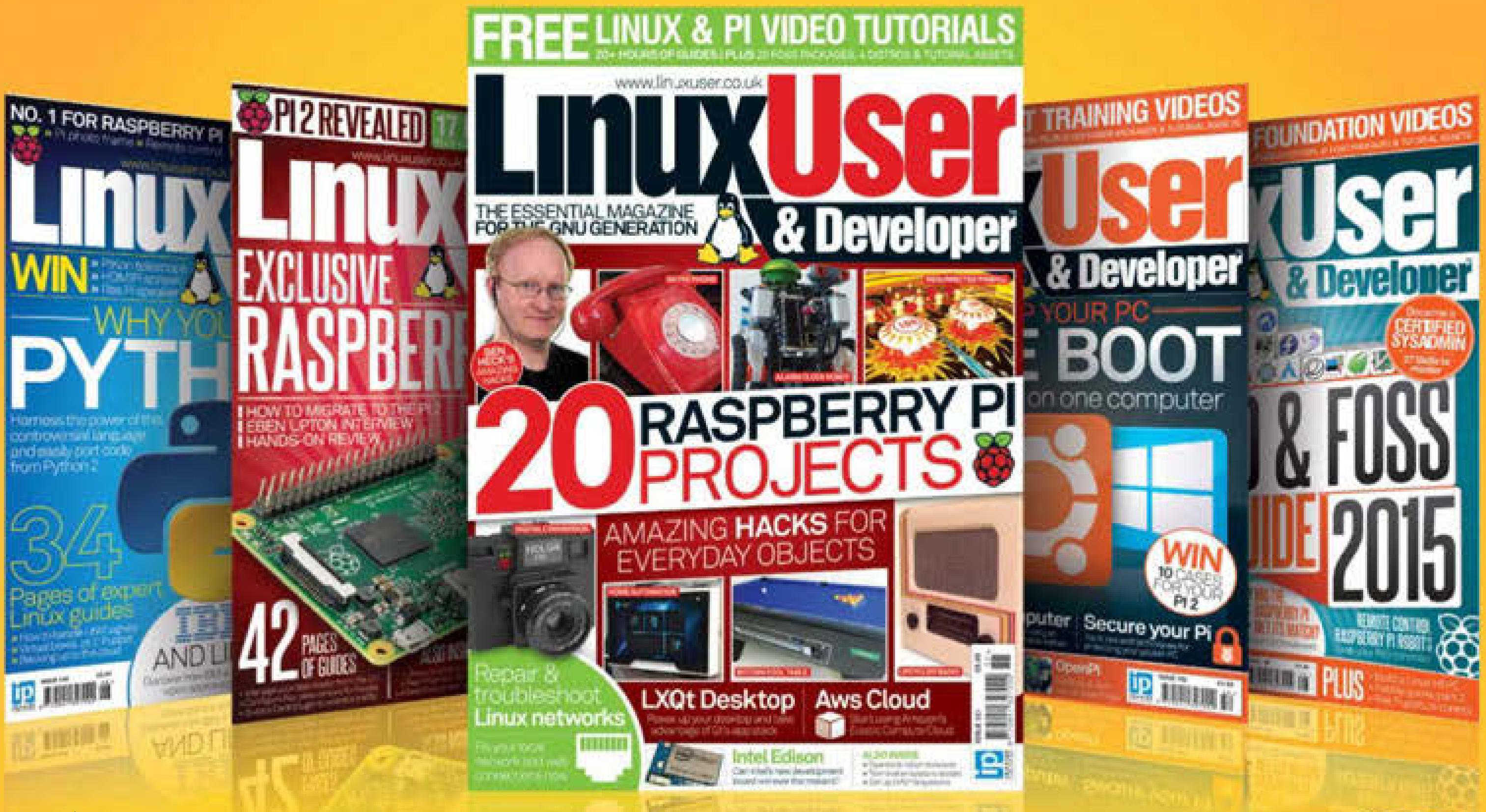
John Lindquist
@johnlindquist
Egghead.io creator

Pete Bacon Darwin
@petebd
Angular 1.x lead developer and the organiser of AngularConnect

Brad Green
@bradlygreen
Google engineering director, manages AngularJS

FOR THE GNU GENERATION

www.linuxuser.co.uk



EVERY ISSUE • FREE RESOURCE DOWNLOAD

LinuxUser & Developer

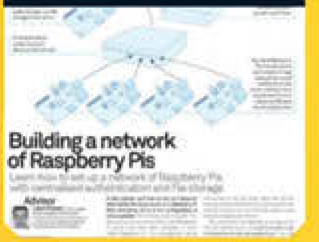
Available from all good newsagents & supermarkets today

ON SALE NOW:

» Fix your network connections » Amazing Pi hacks » Run Linux in the cloud

THE LATEST NEWS

ESSENTIAL GUIDES



DEFINITIVE REVIEWS

INDUSTRY INSIGHT

EXPERT OPINION

BUY YOUR ISSUE TODAY

Print edition available at www.imagineshop.co.uk

Digital edition available at www.greatdigitalmags.com

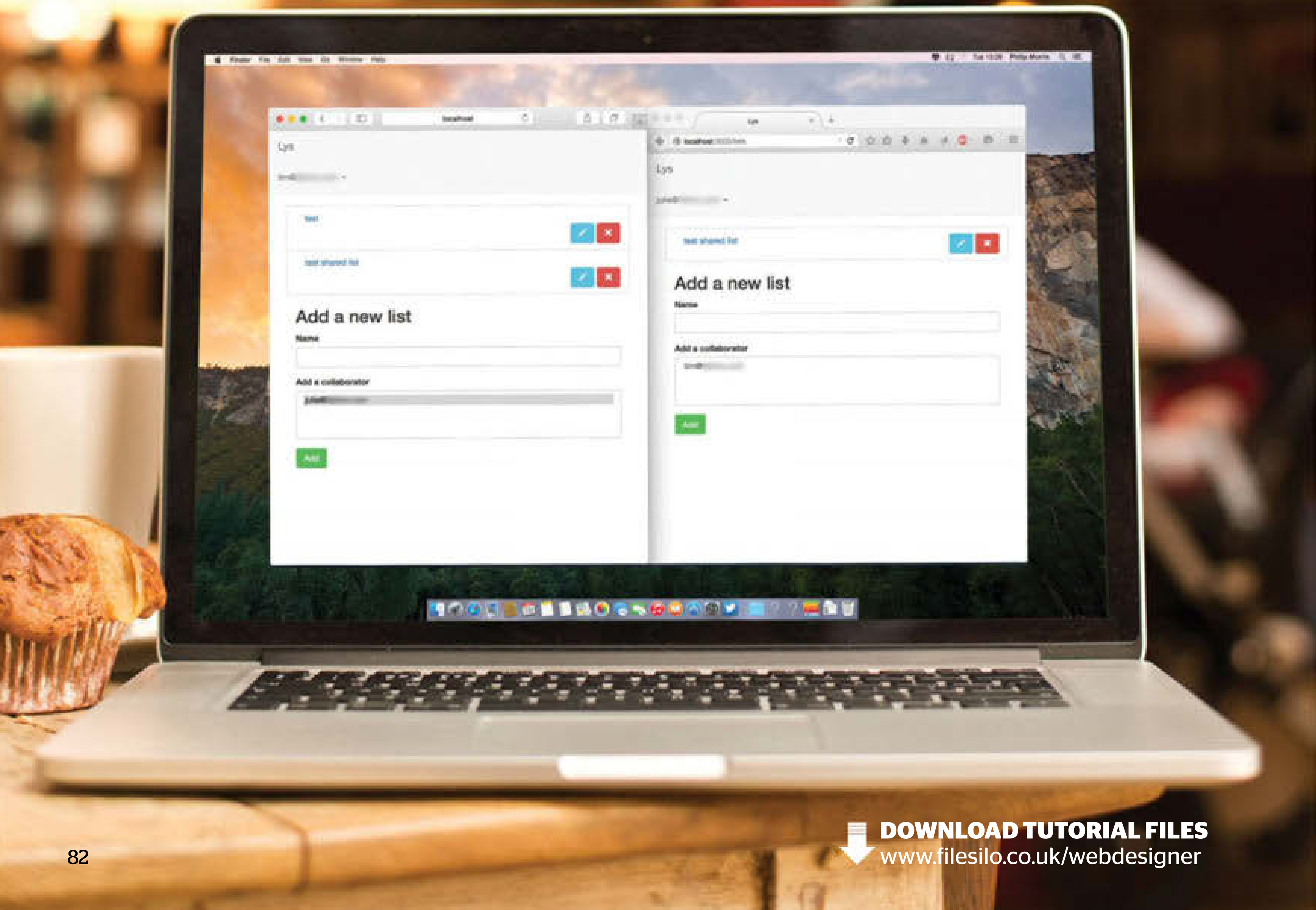
Available on the following platforms

Print Tablet Phone Desktop

facebook.com/LinuxUserUK twitter.com/LinuxUserMag

Build a reactive web app with Angular-Meteor

Combine the best of both worlds with Meteor and Angular to unleash three-way binding



DOWNLOAD TUTORIAL FILES
www.filesilo.co.uk/webdesigner

Meteor is a great way to add 'reactiveness' to your app. That means that as soon as a collection is updated all connected clients receive that data. Meteor uses a library called Blaze to perform live updating from the server. Blaze requires no effort from the developer and 'just works', but by design Blaze is simpler than Angular as it has a gentler learning curve. A side effect of this is that it makes architecting a large app in Blaze challenging.

Enter Angular. It's been around for six years now and has a myriad of 'best practices'. Whether you're building a larger app or have an existing Angular app that you wish to port, Meteor and Angular are a good team. Of course, Angular isn't the only option but it's the one that we're going to focus on today.

To see how the two work together we're going to build a collaborative list app. It'll take advantage of Meteor's reactivity and Angular's structure. By the end of the tutorial you'll have a good understanding of how they complement each other.

As of version 1.0 Meteor supports Windows but it will require a different installer, aside from that the steps should be identical.

1. Install Meteor

If you haven't installed Meteor already then copy the curl command. If you're on Windows then there's a standalone installer at **install.meteor.com/windows**. Then create a new project with 'meteor create', this will create three files and a .meteor folder. Run it with the meteor command (this is shorthand for 'meteor run').

```
$ curl https://install.meteor.com/ | sh
$ meteor create lys
$ cd lys
$ meteor
```

2. Angular Meteor

In a single step we've already created the project and got it running via a server! The secret glue to hooking Meteor and Angular up is to use 'angular-meteor' headed by Uri Goldshtein and nine other core contributors. We can add it to the project with 'meteor add' while running and Meteor will automatically restart.

```
$ meteor add urigo:angular
```

3. Lys markup

One of the files that Meteor created for us was lys.html. This is where the head and body content of our app will reside. We'll keep it simple for now with an Angular include pointing to a file we'll create shortly. The 'base' tag is required for Angular's routing service to work in HTML5 mode.

```
<!-- lys.html   ->
<head>
<title>Lys</title>
<base href="/">
</head>
<body>
<main ng-app="lys" ng-include="'index.ng.
html'"></main>
</body>
```

4. Module definition

Meteor automatically loads files and because of this it has a defined load order. Files in a folder called lib are loaded before other files so this makes it a good place to put our module definition file. Note that the files within the client folder are only delivered to the client and are not run on the server.

```
// client/lib/app.js
angular.module('lys', ['angular-meteor']);
```

5. Showing a list

We'll display a list of lists on the page in a section controlled by the ListController. The 'track by' expression is important here because it means that each item won't have an internal $$hashKey object added to it. This is important because properties starting with $ are protected in MongoDB and can't be added.

```
<!-- lys.html   ->
<section class="list-container" ng-
controller="ListController">
<ul>
<li ng-repeat="list in lists track by
list._id"></li>
</ul>
</section>
```

6. Toggle buttons

Within each list item we will now add two buttons: this will be one button to toggle editing and one button to remove it from the list. We are also using Angular's 'switch' directive to change which elements to show when we are editing. Because the input is tied to a model, updates will immediately be visible to Meteor and thus to all other users.

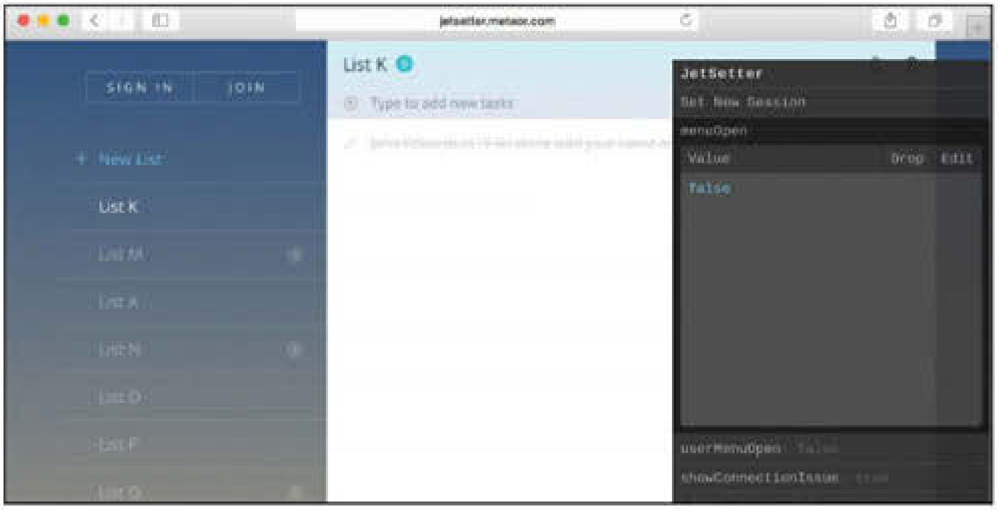
```
<!-- lys.html   ->
<span ng-switch="edit">
<input ng-switch-when="true" ng-model="list.
name">
<p ng-switch-default>{{list.name}}</p>
</span>
<button ng-click="edit = !edit">Edit</
button>
<button ng-click="remove(list)">Remove</
button>
```

7. List controller

Our list controller is pretty standard but as well as passing in $scope we're also injecting $meteor. This is what angular-meteor provides through the extra dependency that we added to the module which gives us Angular-ified Meteor methods. Note that we're

Best of both worlds

You can still use Blaze in your application with Angular-Meteor, more documentation can be found at **angularjs.meteor.com**. You can change the delimiter from '{{' to '[[' as well.



Left
Mongol has a sister project called JetSetter which is a package that manages and visualises Session variables while developing

Top left
The buttons use icons from Twitter Bootstrap's glyphicon component aided with a title attribute

Top right
With a single line the collection is retrieved from the database and the work we've done so far renders it

referencing a variable called Lists, it's not in this file so where is it coming from?

```
angular.module('lys').
controller('ListController', ['$scope',
'$meteor',
function ListController ($scope, $meteor) {
$scope.lists = $meteor.collection(Lists).
subscribe('lists');
$scope.users = $meteor.collection(Meteor.
users).subscribe('users');
}
]);
```

8. List collection

Lys.js is another file that Meteor created for us when it created the project. This means that it's shared with both the client and the server. We're creating a global variable called Lists which is a database collection (a little like a table). This is what we referenced in our controller in the previous step.

```
// lys.js
Lists = new Mongo.Collection('lists');
```

9. List permissions

By default Meteor enables anyone to edit the database from the client (via the insecure package) and the database sends everything to the client (via the autopublish package). We'll remove these later but we need to specify under what conditions a list can be inserted, updated or removed. This will be based on either being an owner or collaborator.

```
Lists.allow({
insert: function (userId, list) {
return userId && list.owner === userId;
},
update: function (userId, list) {
return userId === list.owner || list.
collaborators.indexOf(userId) >   1;
},
remove: function (userId, list) {
return userId === list.owner || list.
collaborators.indexOf(userId) >   1;
}
});
```

10. Publish lists

As well as setting permissions we will also need to publish 'lists' which is what the client actually subscribes to. We only want to publish the lists that the logged in user has access to though so we will need to create a query here. Queries are built up through objects and arrays. Here we will look for lists where we're either the owner or a collaborator.

```
Meteor.publish('lists', function () {
return Lists.find({
$or: [{
$and:[{
owner: this.userId,
}, {
owner: {
$exists: true
}
}]
}, {
$and:[{
collaborators: {
$in: [this.userId]
}
}]
}]
});
});
```

11. Publish users

Similar to publishing lists we'll also want to publish all of our users. This is so that you can add collaborators to your lists. Note that we're returning everything (no conditions here) but only returning the field that will be used. This means that you can't accidentally leak data that the client doesn't need.

```
Meteor.publish('users', function () {
return Meteor.users.find({}, {
fields: {
emails: 1
}
});
});
```

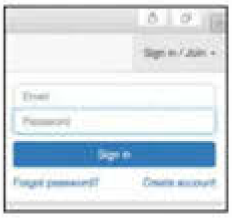
12. Add or remove packages

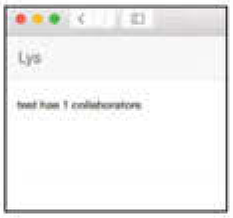
Next we'll add basic account handling with 'accounts-password' so people can log in (yes, it's that simple!). Then add Twitter Bootstrap and a version of accounts-ui which is specific to Bootstrap 3. We'll also remove the insecure and autopublish packages as described earlier. Chaining packages by adding a space between them is a useful and quick way to act on multiple packages.

```
$ meteor add accounts-password
twbs:bootstrap ian:accounts-ui-bootstrap-3
$ meteor remove insecure autopublish
```

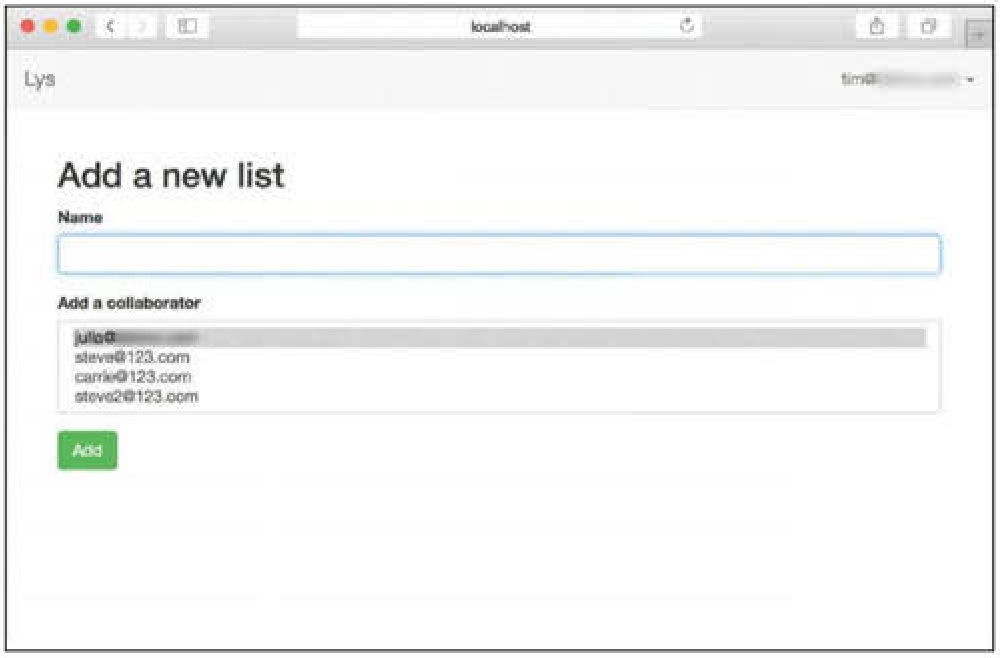
Still not convinced?

The Angular-Meteor Manifesto (**angularjs.meteor.com/manifest**) has a list of additional points that may convince you and also a roadmap for what's on the horizon.





test has 1 collaborators



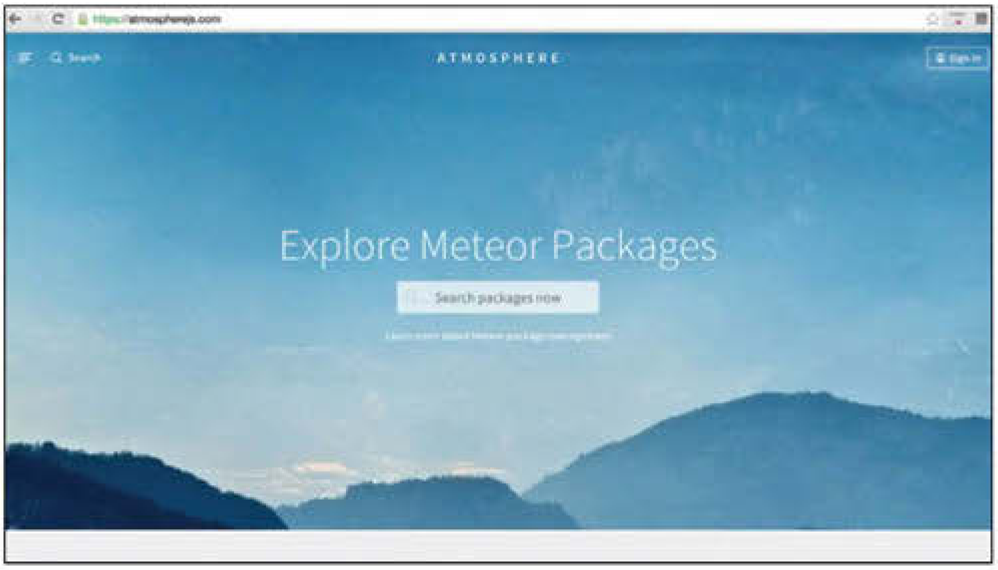
Add a new list

julio@[redacted]
steve@123.com
carrie@123.com
steve2@123.com

Top left
The log-in buttons partial is very simple to use if you're happy with the default markup and styling

Top right
Although basic, the list detail view demonstrates how routing can be used in conjunction with state

Right
The collaboration list is filtered to remove the current user out of the list but this could be done server-side



Where to find community code for your project

Using Angular and Meteor you have the opportunity to use packages from both communities. The de facto place for finding Meteor packages is Atmosphere (**atmospherejs.com**). Angular doesn't have a single place for modules but a good place to look is Angular Modules (**ngmodules.org**). Atmosphere is the Meteor equivalent of npm and also includes handy things like the command to paste in to add the package.

In development mode, Meteor can easily serve about 5MB of files (Firefox reckons it's closer to ten) but in production mode this is radically decreased to 240KB. This is done by running your app with the production flag, '$ meteor - -production'. Remember that each package that you add is going to increase this so ensure that you're using the package for the right reasons.

13. Add login buttons

Accounts-ui gives us a partial called loginButtons. This is the same as the double curly brace syntax that Angular uses but this is Meteor's Blaze template engine. It's important that the two aren't confused. The rest of the structure is for Bootstrap's navbar component.

```
<!-- lys.html  ->
<nav class="navbar navbar-default">
<div class="container-fluid">
<div class="navbar-header">
<a class="navbar-brand" href="#">Lys</a>
</div>
<div class="nav navbar-nav navbar-right">
{{ > loginButtons }}
</div>
</div>
</nav>
```

14. Show form

$root here is the same as $scope.$root and accesses $rootScope without having to explicitly pass it to the template. The next steps will expand this form. $root also has the getReactively method which watches the variable for changes and notifies Meteor's Tracker library.

```
<form class="col-md-7" ng-show="$root.
currentUser">
<h2>Add a new list</h2>
</form>
```

15. Name input

The form will create an object we'll novelly call 'newList'. One of the pieces of data that we'll need is the name of the list so we'll make a form text input to capture this data. Note that we don't have to specify the type because the default type for an input is text.

```
<label for="nameInput">Name</label>
<input ng-model=“newList.name”
id=“nameInput” class=“form-control”
required>
```

16. Collaborator list

The collaborator field will let the user select multiple users (but not themselves) to add as a collaborator via email address. Angular's select directive makes this trivial but the syntax is daunting. The key part is the ng-options attribute which says, "use the email address as the text and call each item in the users array user".

```
<label for="users">Add a collaborator</
label>
<select class="form-control" id="users"
name="users" ng-options="user.emails[0].
address for user in users | removeSelf"
ng-model="collaborators" multiple>
<option value="">Choose a collaborator</
option>
</select>
```

17. RemoveSelf filter

In the last step we referenced a filter called removeSelf and this filter will return all of the users that don't match the currently logged in user's ID. We are using a technique called 'duck typing' here by inferring from the fact that it has a length property – it'll be an array and also have a filter function.

```
angular.module('lys').filter('removeSelf',
function () {
return function (users) {
if (users.length) {
eturn users.filter(function (user) {
return user._id !== Meteor.userId();
});
} else {
return users;
}
};
});
```

18. Add button

To finish our form we'll create a button which will add newList to the Meteor collection. The code for this is a little more involved so to keep things tidy we'll call a method on the controller to deal with it rather than write it all inline. The semicolon is optional but insert them in case we add more, it's the new 'be kind rewind'.

```
<button class="btn btn-success" ng-
click="add(newList);">Add</button>
```

19. Add function

The 'add' function maps each collaborator to just their ID so rather than include the entire object for each collaborator we only store their ID which we can look up later on. We set the list owner to the currently logged in user by accessing the root scope and we will then push this to the list. This push will also automatically update the database for us!

```
$scope.add = function (list) {
list.collaborators = $scope.collaborators.
map(function (user) {
return user._id;
});
list.owner = $scope.$root.currentUser._id;
$scope.lists.push(list);
};
```

20. Remove a list

Removing a list is far simpler than adding it because all we need to do is call the remove function on the collection. This receives the list object and voila, it's removed from the database and each connected client sees the list vanish. Meteor collections have all sorts of exotic methods for you to play with like updateCursor and unregisterAutoBind.

```
$scope.remove = function(list) {
$scope.lists.remove(list);
};
```

21. Jumbotron message

You may have noticed that we had to create a 'special' HTML file suffixed with .ng.html. This is because otherwise Meteor will try to parse the Angular expressions and break. Next we'll create a message for logged out users explaining the benefits of what they'll be able to do with an account. This works because currentUser is 'null' when there is no user.

```
<!-- index.ng.html   ->
<div class="col-md-12">
<div class="jumbotron" ng-hide="$root.
currentUser">
<div class="container">
<p>Login to create and share lists with
your friends and family!</p>
</div>
</div>
</div>
```

22. Angular routing

So we can now create a list and add collaborators to it. Let's introduce routing to our little app. Create a file called 'routes.js' inside of the client folder. HTML5 mode rewrites URLs to 'normal' URLs which fallback to hashbangs in older browsers. The next steps will sit inside this function.

```
// client/routes.js
angular.module('lys').
config(['$urlRouterProvider',
'$stateProvider', '$locationProvider',
function ($urlRouterProvider,
$stateProvider, $locationProvider) {
$locationProvider.html5Mode(true);
$urlRouterProvider.otherwise('/lists');
}]);
```

23. List route config

Angular's state provider comes from the ui-router package. Here we can specify what is rendered at a given URL and which controller it should be hooked up to. More information on controlling state can be found at **github.com/angular-ui/ui-router/wiki**.

```
$stateProvider
.state('lists', {
url: '/lists',
templateUrl: 'client/lists/views/lists.ng.
html',
controller: 'ListController'
})
```

24. List detail config

Likewise we'll now be doing the same for the details view. Note that this will take a URL parameter denoted by the colon. ListId will be the ID of the list in the database which we will use to get further information about it. Being able to specify the template and controller separately is extremely useful, especially if we want to reuse any of the templates.

```
.state('listDetails', {
url: '/lists/:listId',
templateUrl: 'client/lists/views/list-
details.ng.html',
controller: 'DetailsController'
});
```

25. Details controller

The second controller in our app will get the list we want and a list of users. We're going to show the number of collaborators but you could extend it to allow the addition or removal of collaborators to the list. We name the DetailsController function (convention would be to keep it anonymous) for readable stack traces when we are debugging.

```
angular.module('lys').
controller('DetailsController', ['$scope',
'$meteor', '$stateParams',
function DetailsController ($scope, $meteor,
$stateParams) {
// next step
}
]);
```

26. Meteor reactivity

$meteor.object wraps a Meteor object to give it 'reactivity'. That is, the ability to update when other users' version of that object updates. Note also the second argument of 'false' to the Meteor collection. This toggles automatic client-side saving which we want to disable for 'users', by default this is true.

```
$scope.list = $meteor.object(Lists,
$stateParams.listId).subscribe('lists');
$scope.users = $meteor.collection(Meteor.
users, false).subscribe('users');
```

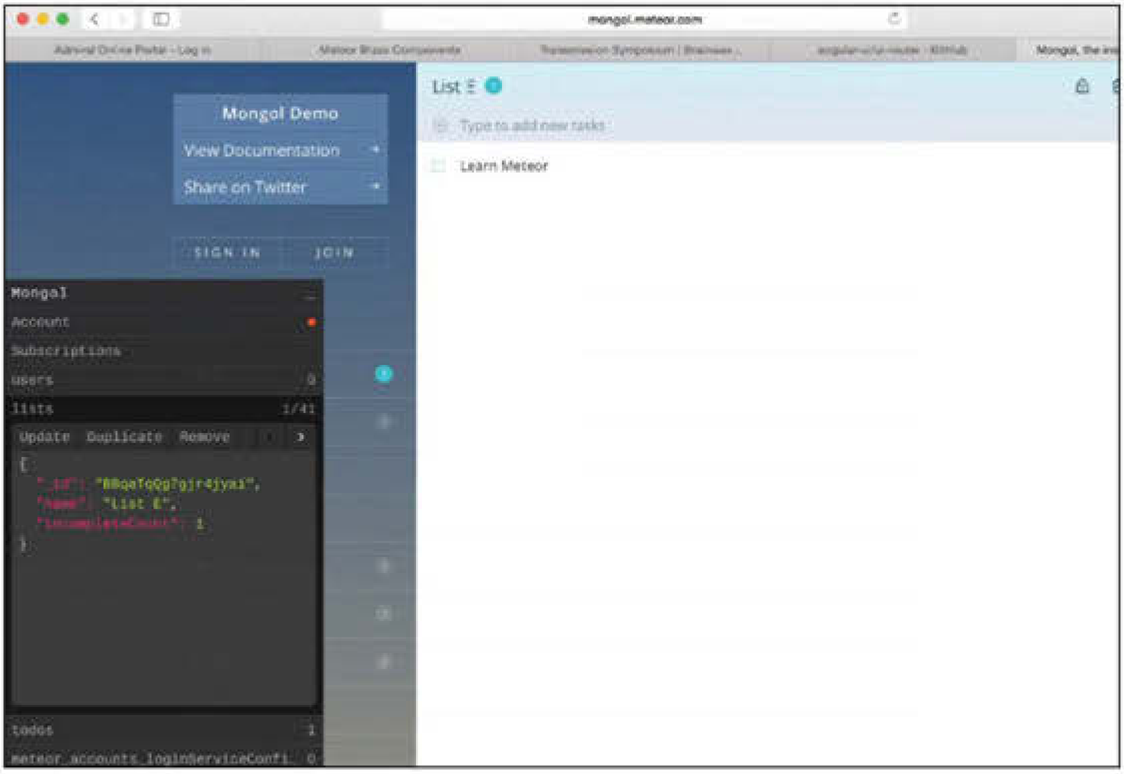
27. Render to page

Lastly we output the name of the list and the number of collaborators into the template. As an exercise for the discerning reader, see if you can get a list of email addresses to display for each collaborator. Harnessing Angular's principles with Meteor is a powerful way to architect your app.

```
<p>{{list.name}} has {{list.collaborators.
length}} collaborators</p>
```

Monitor and debug with Mongol

An invaluable package while developing Meteor apps is msavin's Mongol (**mongol.meteor.com**). Add it to your project with '$ meteor add msavin:mongol' and wait for Meteor to restart. You shouldn't notice anything different until you hit Cmd/Ctrl+M in the browser and a black window pops up in the bottom left. Mongol gives you real-time visibility of all the collections that are currently published to the page. This includes users! Here you can add, edit, duplicate and remove documents (analogous to records). It's useful for working out if something has made it to the database or not and can help with debugging. Mongol's smart enough to remove itself when built or deployed unless you're in debug mode.



```
{
  "_id": "BBqaTqQg7gjr4jyai",
  "name": "List E",
  "incompleteCount": 1
}
```

FREE 195 MINS OF CSS VIDEO
web designer
28 PAGES OF EXPERT TUTORIALS
www.webdesignermag.co.uk
HTML5
CSS3
jQuery
WordPress
14 PAGE GUIDE ADVANCED ANGULAR INDUSTRY TIPS & TUTORIALS
POWER UP CSS PREPROCESSORS
·SASS ·LESS ·STYLUS
EXPERT ADVICE TO MASTER PRO LEVEL TOOLS
BUILD WITH REACT NATIVE
A LOOK INSIDE FACEBOOK'S NEW FRAMEWORK
DYNAMIC CSS MODALS
ANIMATE POP-UP WINDOWS
WHAT IS WEB OF THINGS?
WHY IOT WILL ADOPT WEB STANDARDS
INTERACTIVE GRAPHICS HTML 5
UNLEASH THE P5.JS LIBRARY FOR DRAWING IN THE BROWSER
67 DESIGNER RESOURCES
WEB TEMPLATES, FONTS, TEXTURES & ACTIONS
ISSUE 236
£5.99

web designer FREE 145 MINS OF CSS VIDEO
JAVASCRIPT 6 WHAT'S NEW?
BUILD WITH WEB COMPONENTS
JS 6
web designer FREE 88 MINUTES OF CSS3 VIDEO
GOOGLE ANALYTICS
ANIMATED 3D MENUS
THE NEXT WAVE OF RESPONSIVE DESIGN
PHP MADE EASY
66 DESIGNER RESOURCES
FASTER REACT
web designer FREE 30
RUBY ON RAILS
JQUERY SPECIAL EFFECTS
ANGULAR IMAGE GALLERY
web designer FREE 82 MINUTES OF SASS VIDEO
CSS 3D
HOW TO ADD NEW DEPTHS TO DESIGN
65 DESIGNER RESOURCES
HOW THE HOBBIT SITE WAS BUILT

Hosting listings

Keep an eye on the latest packages and deals with our comprehensive list of service providers

Got a deal you think we should list?

Whether you're a hosting firm keen to promote your products or a happy customer who wants a favourite provider to be listed, drop us a line with the details!

webdesigner@imagine-publishing.co.uk

NAME AND URL	PACKAGE	PHONE NUMBER	COST PER YEAR	WEB SPACE	MONTHLY BANDWIDTH	POP3 ACCOUNTS	DATABASE SUPPORT	SHOPPING CART	VIRUS FILTER	FIREWALL	PHONE SUPPORT	EMAIL SUPPORT	WEB CONTROL PANEL	SERVICE LEVEL AGREEMENT
111WebHost 111webhost.com	Budget Web Hosting Pack	N/A	£10	1GB	1GB	5	✓	✗	Add-on	N/A	✗	✓	✓	✗
	WordPress Web Hosting Pack	N/A	£20	5GB	2GB	100	✓	✓	Add-on	N/A	✗	✓	✓	✗
	Start-up Web Hosting Pack	N/A	£25	5GB	Unlimited	100	✓	✓	Add-on	N/A	✗	✓	✓	✗
	Unlimited Web Hosting Pack	N/A	£50	Unlimited	Unlimited	Unlimited	✓	✓	Add-on	N/A	✗	✓	✓	✗
	Unlimited Multi-site Web Hosting Pack	N/A	£300	Unlimited	Unlimited	Unlimited	✓	✓	Add-on	N/A	✗	✓	✓	✗
123-reg (www.123-reg.co.uk)	Starter	0845 859 0018	£29.88	1GB	5GB	20	✗	✗	✗	✓	✓	✓	✓	✗
123-reg (www.123-reg.co.uk)	Plus	0845 859 0018	£59.88	5GB	50GB	500	✓	✗	✗	✓	✓	✓	✓	✗
123-reg (www.123-reg.co.uk)	Pro	0845 859 0018	£107.88	10GB	100GB	750	✓	✗	✗	✓	✓	✓	✓	✗
123-reg (www.123-reg.co.uk)	Bus Pro	0845 859 0018	£179.88	20GB	Unlimited	1,000	✓	✗	✗	✓	✓	✓	✓	✗
123-reg (www.123-reg.co.uk)	Plus (MS)	0845 859 0018	£59.88	2GB	25GB	100	✓	✗	✗	✓	✓	✓	✓	✗
123-reg (www.123-reg.co.uk)	Pro (MS)	0845 859 0018	£107.88	5GB	50GB	500	✓	✗	✗	✓	✓	✓	✓	✗
123-reg (www.123-reg.co.uk)	Bus Pro (MS)	0845 859 0018	£179.88	10GB	150GB	1,000	✓	✗	✗	✓	✓	✓	✓	✗
2020Media (www.2020media.com)	Light User	0870 321 2020	£45	20MB	1GB	3	✓	✓	✓	✓	✓	✓	✓	✗
2020Media (www.2020media.com)	Everyday	0870 321 2020	£100	200MB	10GB	15	✓	✓	✓	✓	✓	✓	✓	✗
2020Media (www.2020media.com)	Business/Pro	0870 321 2020	£275	500MB	20GB	50	✓	✓	✓	✓	✓	✓	✓	✗
2020Media (www.2020media.com)	JAVA Tomcat	0870 321 2020	£300	100MB	3GB	15	✓	✓	✓	✓	✓	✓	✓	✗
2020Media (www.2020media.com)	ASP.Net	0870 321 2020	£275	100MB	3GB	15	✓	✓	✓	✓	✓	✓	✓	✗
4D Hosting (www.4dhosting.com)	Big Web Hosting	0207 183 0602	£60	5GB	Unlimited	20	✓	✓	✓	✓	✓	✓	✓	✓
4D Hosting (www.4dhosting.com)	Bigger Web Hosting	0207 183 0602	£120	10GB	Unlimited	75	✓	✓	✓	✓	✓	✓	✓	✓
4D Hosting (www.4dhosting.com)	Biggest Web Hosting	0207 183 0602	£180	20GB	Unlimited	100	✓	✓	✓	✓	✓	✓	✓	✓
4D Hosting (www.4dhosting.com)	Big VPS	0207 183 0602	£204	75GB	500GB	Unlimited	✓	✓	✓	✓	✓	✓	✓	✓
4D Hosting (www.4dhosting.com)	Bigger VPS	0207 183 0602	£360	100MB	1,000GB	Unlimited	✓	✓	✓	✓	✓	✓	✓	✓
4D Hosting (www.4dhosting.com)	Biggest VPS	0207 183 0602	£720	200GB	1,500GB	Unlimited	✓	✓	✓	✓	✓	✓	✓	✓
Blackfoot Hosting Ltd (www.blackfoot.co.uk)	Home	N/A	£40	500MB	5GB	5	✓	✓	✓	✓	✗	✓	✓	✗
Blackfoot Hosting Ltd (www.blackfoot.co.uk)	Business	N/A	£50	1GB	20GB	100	✓	✓	✓	✓	✗	✓	✓	✗
Blackfoot Hosting Ltd (www.blackfoot.co.uk)	eCommerce	N/A	£100	2GB	40GB	200	✓	✓	✓	✓	✗	✓	✓	✗
Blackfoot Hosting Ltd (www.blackfoot.co.uk)	Professional	N/A	£150	3GB	60GB	300	✓	✓	✓	✓	✗	✓	✓	✗
Blacknight (www.blacknight.com)	Minimus	35359 918 3072	£45	10GB	200GB	Unlimited	✓	✓	✓	✓	✓	✓	✓	✗
Blacknight (www.blacknight.com)	Medius	35359 918 3072	£79	20GB	400GB	Unlimited	✓	✓	✓	✓	✓	✓	✓	✗
Blacknight (www.blacknight.com)	Maximus	35359 918 3072	£45	30GB	600GB	Unlimited	✓	✓	✓	✓	✓	✓	✓	✗
CityCM (www.citycm.couk)	City250	0330 223 0120	£10.99	250MB	5GB	Unlimited	✓	✓	✓	✓	✓	✓	✓	✓
CityCM (www.citycm.couk)	City500	0330 223 0120	£14.99	500MB	10GB	Unlimited	✓	✓	✓	✓	✓	✓	✓	✓
CityCM (www.citycm.couk)	City2000	0330 223 0120	£29.90	2GB	50GB	Unlimited	✓	✓	✓	✓	✓	✓	✓	✓
CityCM (www.citycm.couk)	City10000	0330 223 0120	£49.90	10GB	100GB	Unlimited	✓	✓	✓	✓	✓	✓	✓	✓
CityCM (www.citycm.couk)	CityR10	0330 223 0120	£120	10GB	50GB	Unlimited	✓	✓	✓	✓	✓	✓	✓	✓
CityCM (www.citycm.couk)	CityR20	0330 223 0120	£180	20GB	100GB	Unlimited	✓	✓	✓	✓	✓	✓	✓	✓
CityCM (www.citycm.couk)	CityR30	0330 223 0120	£240	30GB	150GB	Unlimited	✓	✓	✓	✓	✓	✓	✓	✓
CityCM (www.citycm.couk)	City VPS15	0330 223 0120	£180	15GB SSD	400GB	Unlimited	✓	✓	✓	✓	✓	✓	✓	✓
CityCM (www.citycm.couk)	City VPS30	0330 223 0120	£300	30GB SSD	600GB	Unlimited	✓	✓	✓	✓	✓	✓	✓	✓
CityCM (www.citycm.couk)	City VPS75	0330 223 0120	£420	75GB SSD	600GB	Unlimited	✓	✓	✓	✓	✓	✓	✓	✓
Designwasp (http://designwasp.com)	Starter	0844 372 9848	£30	3GB	20GB	1,000	✓	✓	✗	✓	✗	✓	✓	✗
Designwasp (http://designwasp.com)	Home	0844 372 9848	£40	10GB	50GB	10,000	✓	✓	✗	✓	✗	✓	✓	✗
Designwasp (http://designwasp.com)	Business	0844 372 9848	£60	Unlimited	Unlimited	Unlimited	✓	✓	✗	✓	✗	✓	✓	✗
Designwasp (http://designwasp.com)	Windows Unlimited	0844 372 9848	£60	Unlimited	Unlimited	Unlimited	✓	✓	✗	✓	✗	✓	✓	✗
Designwasp (http://designwasp.com)	Windows Home	0844 372 9848	£40	10GB	50GB	1,000	✓	✓	✗	✓	✗	✓	✓	✗
Designwasp (http://designwasp.com)	CheapHost	0844 372 9848	£10	300MB	100MB	5	✗	✗	✗	✓	✗	✗	✓	✗
Digital Gibbon Ltd (http://digitalgibbon.com)	Cheeky Chimp	NA	£0	500MB	Unlimited	5	✓	✗	✓	✓	NA	✓	✓	✗
Digital Gibbon Ltd (http://digitalgibbon.com)	Digital Gibbon	NA	£12	5GB	Unlimited	10	✓	✗	✓	✓	NA	✓	✓	✗
Digital Gibbon Ltd (http://digitalgibbon.com)	Silverback	NA	£24	Unlimited	Unlimited	Unlimited	✓	✓	✓	✓	NA	✓	✓	✗
Digital Gibbon Ltd (http://digitalgibbon.com)	Wordpress hosting	NA	£12	5GB	Unlimited	10	✓	✗	✓	✓	NA	✓	✓	✗
Donhost (www.donhost.co.uk)	Reseller Unix	0845 226 5566	£399.99	Unlimited	Unlimited	Unlimited	✓	✗	✓	✓	✓	✓	✓	✗
Donhost (www.donhost.co.uk)	Reseller Windows	0845 226 5566	£499.99	Unlimited	Unlimited	Unlimited	✓	✗	✓	✓	✓	✓	✓	✗
Donhost (www.donhost.co.uk)	Enterprise	0845 226 5566	£89.99	1GB	2GB	250	✓	✗	✓	✓	✓	✓	✓	✗
Donhost (www.donhost.co.uk)	Commerce	0845 226 5566	£179.99	2GB	5GB	500	✓	✗	✓	✓	✓	✓	✓	✗
Donhost (www.donhost.co.uk)	Designer	0845 226 5566	£119.99	1GB	2GB	250	✓	✗	✓	✓	✓	✓	✓	✗
Donhost (www.donhost.co.uk)	Developer	0845 226 5566	£259.99	2GB	5GB	500	✓	✗	✓	✓	✓	✓	✓	✗

Featured host of the month: 111WebHost 111webhost.com

Getting more out of one of our quality listed hosting providers

111WebHost is a provider committed to making website hosting easy to use while offering the best possible value. It provides solutions for small- and medium-sized businesses as well as individuals – and web designers and developers are high on their list. 111WebHost offers a range of top-value packages starting at the ridiculously low price of £1 per month. For this, users get 1GB of webspace and 1GB of monthly traffic. For those looking for more, there is 5GB of webspace and unlimited traffic for just £2.50 a month. Unlimited webspace and traffic is available for just £5 a month. They also provide specialist hosting for WordPress, Joomla and Drupal.

NAME AND URL	PACKAGE	PHONE NUMBER	COST PER YEAR	WEB SPACE	MONTHLY BANDWIDTH	POP3 ACCOUNTS	DATABASE SUPPORT	SHOPPING CART	VIRUS FILTER	FIREWALL	PHONE SUPPORT	EMAIL SUPPORT	WEB CONTROL PANEL	SERVICE LEVEL AGREEMENT
eHosting (www.ehosting.com)	Starter	0844 999 4100	£23.88	1GB	25GB	10	✗	✗	✗	✗	✗	✓	✓	✓
eHosting (www.ehosting.com)	Personal	0844 999 4100	£59.88	2.5GB	Unlimited	50	✓	✗	✗	✗	✗	✓	✓	✓
eHosting (www.ehosting.com)	Expert	0844 999 4100	£95.88	5GB	Unlimited	250	✓	✗	✗	✗	✗	✓	✓	✓
eHosting (www.ehosting.com)	Virtual	0844 999 4100	£227.88	50GB	Unlimited	Unlimited	✓	✗	✗	✗	✓	✓	✓	✓
Equiphase (www.equiphase.net)	Bronze	0121 314 4865	£30	200MB	2GB	10	✓	✓	✗	✓	✗	✓	✓	✓
Equiphase (www.equiphase.net)	Silver	0121 314 4865	£42	400MB	5GB	20	✓	✓	✗	✓	✗	✓	✓	✓
Equiphase (www.equiphase.net)	Gold	0121 314 4865	£72	800MB	10GB	100	✓	✓	✗	✓	✗	✓	✓	✓
Equiphase (www.equiphase.net)	Platinum	0121 314 4865	£114	1,200MB	40GB	200	✓	✓	✓	✓	✓	✓	✓	✓
Eurofasthost.com (www.eurofasthost.com)	Email Only	02380 249 823	£40	1GB	2GB	10	✗	✗	✓	✓	✓	✓	✓	✓
Eurofasthost.com (www.eurofasthost.com)	Essential	02380 249 823	£75	2GB	5GB	10	✗	✗	✓	✓	✓	✓	✓	✓
Eurofasthost.com (www.eurofasthost.com)	Superior	02380 249 823	£140	5GB	10GB	25	✓	✓	✓	✓	✓	✓	✓	✓
Eurofasthost.com (www.eurofasthost.com)	Premium	02380 249 823	£250	10GB	25GB	100	✓	✓	✓	✓	✓	✓	✓	✓
Evohosting (www.evohosting.co.uk)	Starter	N/A	£29.99	500MB	1GB	3	✓	✓	✓	✓	✗	✓	✓	✓
Evohosting (www.evohosting.co.uk)	Home	N/A	£54.99	2.5GB	30GB	50	✓	✓	✓	✓	✗	✓	✓	✓
Evohosting (www.evohosting.co.uk)	Business	N/A	£79.99	6.5GB	Unlimited	Unlimited	✓	✓	✓	✓	✗	✓	✓	✓
Evohosting (www.evohosting.co.uk)	eCommerce	N/A	£159.99	30GB	Unlimited	Unlimited	✓	✓	✓	✓	✗	✓	✓	✓
Fasthosts (www.fasthosts.co.uk)	Personal	0808 168 6777	£32.87	5GB	Unlimited	50	✓	✗	✓	✓	✓	✓	✓	✗
Fasthosts (www.fasthosts.co.uk)	Business Bronze	0808 168 6777	£58.38	50GB	Unlimited	500	✓	Option	✓	✓	✓	✓	✓	✗
Fasthosts (www.fasthosts.co.uk)	Business Silver	0808 168 6777	£76.39	Unlimited	Unlimited	Unlimited	✓	Option	✓	✓	✗	✓	✓	✗
Fasthosts (www.fasthosts.co.uk)	Business Gold	0808 168 6777	£101.89	Unlimited	Unlimited	Unlimited	✓	✓	✓	✓	✗	✓	✓	✗
Fasthosts (www.fasthosts.co.uk)	WD Starter	0808 168 6777	£149.99	20GB	Unlimited	Unlimited	✓	Option	Option	✓	✗	✓	✓	✗
Fasthosts (www.fasthosts.co.uk)	WD Advanced	0808 168 6777	£199.99	Unlimited	Unlimited	Unlimited	✓	Option	Option	✓	✗	✓	✓	✗
Giacom (www.giacom.com)	Business Pro	0800 542 7500	£199	100MB	2GB	100	✓	✓	✓	✓	✓	✓	✓	✓
Heart Internet (www.heartinternet.co.uk)	Starter Professional	0845 644 7750	£29.80	2.5GB	10GB	1,000	✗	✗	✗	✓	✗	✓	✓	✓
Heart Internet (www.heartinternet.co.uk)	Home Professional	0845 644 7750	£89.99	Unlimited	Unlimited	10,000	✓	✓	✗	✓	✗	✓	✓	✓
Heart Internet (www.heartinternet.co.uk)	Business Professional	0845 644 7750	£129.99	Unlimited	Unlimited	Unlimited	✓	✓	✗	✓	✗	✓	✓	✓
Heart Internet (www.heartinternet.co.uk)	Reseller Professional	0845 644 7750	£349.99	Unlimited	Unlimited	Unlimited	✓	✓	✗	✓	✗	✓	✓	✓
Hostway (www.hostway.co.uk)	Silver	0808 180 1880	£79.50	150MG	3GB	5	✗	Option	✓	✓	✗	✓	✓	✗
Hostway (www.hostway.co.uk)	Gold	0808 180 1880	£139.50	300MB	5GB	10	✓	Option	✓	✓	✓	✓	✓	✗
Hostway (www.hostway.co.uk)	Gold Plus	0808 180 1880	£189.50	450MB	10GB	30	✓	Option	✓	✓	✓	✓	✓	✗
Hostway (www.hostway.co.uk)	Platinum	0808 180 1880	£359.50	600MB	20GB	50	✓	Option	✓	✓	✓	✓	✓	✗
Hostway (www.hostway.co.uk)	Platinum Plus	0808 180 1880	£599.50	1.2GB	40GB	10	✓	Option	✓	✓	✓	✓	✓	✗
Hostway (www.hostway.co.uk)	Email Plus	0808 180 1880	£49.95	50MB	N/A	5	N/A	N/A	N/A	✓	✗	✓	✓	✗
LD Hosts (http://ldhosts.co.uk)	LD Budget (Linux)	07891 235858	£11.88	1GB	1GB	500	✓	✓	✗	✓	✓	✓	✓	✗
LD Hosts (http://ldhosts.co.uk)	LD Home (Linux)	07891 235858	£23.88	10GB	10GB	5,000	✓	✓	✗	✓	✓	✓	✓	✗
LD Hosts (http://ldhosts.co.uk)	LD Pro (Linux)	07891 235858	£41.88	50GB	50GB	20,000	✓	✓	✗	✓	✓	✓	✓	✗
LD Hosts (http://ldhosts.co.uk)	LD Unlimited (Linux)	07891 235858	£60	Unlimited	Unlimited	Unlimited	✓	✓	✗	✓	✓	✓	✓	✗
LD Hosts (http://ldhosts.co.uk)	Windows Home	07891 235858	£17.88	2GB	2GB	1,000	✓	✓	✗	✓	✓	✓	✓	✗
LD Hosts (http://ldhosts.co.uk)	Windows Unlimited	07891 235858	£60	Unlimited	Unlimited	Unlimited	✓	✓	✗	✓	✓	✓	✓	✗
Media Temple (http://mediatemple.net)	Shared-Server Pro	+1 310 841 5500	£64	2GB	1TB	1,000	✓	✓	✓	✓	✓	✓	✓	✓
Media Temple (http://mediatemple.net)	Shared-Server Advanced	+1 310 841 5500	£109	5GB	1.5TB	5,000	✓	✓	✓	✓	✓	✓	✓	✓
NameHog (www.namehog.net)	Email Only	01604 212 904	£11.99	2.5GB	15GB	10	✗	✗	✓	✓	✓	✓	✓	✓
NameHog (www.namehog.net)	Starter	01604 212 904	£36.99	10GB	150GB	Unlimited	✗	✗	✓	✓	✓	✓	✓	✓
NameHog (www.namehog.net)	Home Pro	01604 212 904	£59.99	25GB	Unlimited	Unlimited	✓	✓	✓	✓	✓	✓	✓	✓
NameHog (www.namehog.net)	Business	01604 212 904	£109.99	Unlimited	Unlimited	Unlimited	✓	✓	✓	✓	✓	✓	✓	✓
Namesco (www.names.co.uk)	StartUp Plus	0845 363 3632	£89.99	500MB	5GB	10	✓	✓	✓	✓	✓	✓	✓	✓
Namesco (www.names.co.uk)	Business	0845 363 3632	£149.99	2,000MB	20GB	100	✗	✗	✓	✓	✓	✓	✓	✓
Namesco (www.names.co.uk)	Business Plus	0845 363 3632	£89.99	3,000MB	30GB	200	✓	✓	✓	✓	✓	✓	✓	✓
Namesco (www.names.co.uk)	Designer	0845 363 3632	£239.88	1GB	20GB	200	✓	✓	✓	✓	✓	✓	✓	✓
Namesco (www.names.co.uk)	Designer Plus	0845 363 3632	£479.88	Unlimited	30GB	Unlimited	✓	✓	✓	✓	✓	✓	✓	✓
NETCETERA Netcetera www.netcetera.co.uk	DEVELOPER	0800 808 5450	£32.89	1GB	Unlimited	500	✓	✓	✓	✓	✓	✓	✓	✓
	ONE	0800 808 5450	£109.99	5GB	Unlimited	1000	✓	✓	✓	✓	✓	✓	✓	✓
	RESELLER	0800 808 5450	£274.89	Unlimited	Unlimited	1000	✓	✓	✓	✓	✓	✓	✓	✓
	VM500 Server	0800 808 5450	£300	20GB	Unlimited	Unlimited	✓	✓	✓	✓	✓	✓	✓	✓
	2200DC Server	0800 808 5450	£720	160GB	Unlimited	Unlimited	✓	✓	✓	✓	✓	✓	✓	✓
	3000DC Server	0800 808 5450	£1,200	2x500GB	Unlimited	Unlimited	✓	✓	✓	✓	✓	✓	✓	✓
	2600QC Server	0800 808 5450	£1,800	2x500GB	Unlimited	Unlimited	✓	✓	✓	✓	✓	✓	✓	✓

Hosting listings

Keep an eye on the latest packages and deals with our comprehensive list of service providers

Continued...

NAME AND URL	PACKAGE	PHONE NUMBER	COST PER YEAR	WEB SPACE	MONTHLY BANDWIDTH	POP3 ACCOUNTS	DATABASE SUPPORT	SHOPPING CART	VIRUS FILTER	FIREWALL	PHONE SUPPORT	EMAIL SUPPORT	WEB CONTROL PANEL	SERVICE LEVEL AGREEMENT
Netplan (www.netplan.co.uk)	Shared 100	0207 1000 424	£60	100MB	1GB	5	✓	✓	✓	✓	✓	✓	✓	✓
Netplan (www.netplan.co.uk)	VS100	0207 1000 424	£600	5GB	5GB	5	✓	✓	✓	✓	✓	✓	✓	✓
Netplan (www.netplan.co.uk)	VS200	0207 1000 424	£1,000	10GB	10GB	10	✓	✓	✓	✓	✓	✓	✓	✓
Netplan (www.netplan.co.uk)	VS300	0207 1000 424	£2,000	15GB	50GB	20	✓	✓	✓	✓	✓	✓	✓	✓
Netplan (www.netplan.co.uk)	Dedicated Servers	0207 1000 424	£3,000+	73GB+	1,500GB	100+	✓	✓	✓	✓	✓	✓	✓	✓
PurplePaw (www.purplepaw.co.uk)	Email	N/A	£25	100MB	1GB	10	✗	✗	✓	✓	✗	✓	✓	✗
PurplePaw (www.purplepaw.co.uk)	Play	N/A	£40	100MB	1GB	10	✗	✗	✓	✓	✗	✓	✓	✗
PurplePaw (www.purplepaw.co.uk)	Plus	N/A	£65	750MB	5GB	25	✓	✓	✓	✓	✗	✓	✓	✗
PurplePaw (www.purplepaw.co.uk)	Power	N/A	£95	2GB	10GB	100	✓	✓	✓	✓	✗	✓	✓	✗
PurplePaw (www.purplepaw.co.uk)	R3 Reseller	N/A	£660	10GB	50GB	Unlimited	✓	✓	✓	✓	✗	✓	✓	✗
Reddex UK (www.reddexuk.com)	Reddex Design Starter	0843 289 4625	£18	1GB	Unlimited	10	✓	✗	✓	✓	✓	✓	✓	✓
Reddex UK (www.reddexuk.com)	Reddex Design Business	0843 289 4625	£59.88	10GB	Unlimited	100	✓	✓	✓	✓	✓	✓	✓	✓
Streamline.net (www.streamline.net)	Trial (3 month)	0844 941 1000	N/A	10GB	Unlimited	1,000	✓	Option	✓	✓	✓	✓	✓	✗
Streamline.net (www.streamline.net)	Starter	0844 941 1000	£23.88	500MB	Unlimited	20	✓	Option	✓	✓	✓	✓	✓	✗
Swish Hosting (www.swishhosting.co.uk)	Email	08445 67 69 71	£18	-	Unlimited	Unlimited	✗	✗	✓	✓	✓	✓	✓	✓
Swish Hosting (www.swishhosting.co.uk)	Windows Hosting	08445 67 69 71	£66	-	Unlimited	Unlimited	✓	✓	✓	✓	✓	✓	✓	✓
Tidy Web Hosting (www.tidywebhosting.co.uk)	Entry	0844 884 9100	£25	100MB	1GB	Unlimited	✓	✓	✓	✓	✓	✓	✓	✗
Tidy Web Hosting (www.tidywebhosting.co.uk)	Home	0844 884 9100	£50	500MB	5GB	Unlimited	✓	✓	✓	✓	✓	✓	✓	✗
TwentyHost (www.twentyhost.co.uk)	Basic5S	0845 641 0776	£24	100MB	1,500MB	25	✓	✓	✓	✓	✓	✓	✓	✓
TwentyHost (www.twentyhost.co.uk)	Standard5S	0845 641 0776	£45	200MB	3,000MB	50	✓	✓	✓	✓	✓	✓	✓	✓
TwentyHost (www.twentyhost.co.uk)	Business5S	0845 641 0776	£70	500MB	7,500MB	100	✓	✓	✓	✓	✓	✓	✓	✓
TwentyHost (www.twentyhost.co.uk)	Advanced5S	0845 641 0776	£110	1,000MB	15,000MB	200	✓	✓	✓	✓	✓	✓	✓	✓
UK2.NET (www.uk2.net)	Starter Hosting	0808 168 2427	£24	500MB	1GB	✗	✓	✓	✓	✓	✓	✓	✓	✗
UK2.NET (www.uk2.net)	Business Hosting	0808 168 2427	£54	Unlimited	Unlimited	Unlimited	✓	✓	✓	✓	✓	✓	✓	✗
VARiHOST (www.varihost.net)	WordPress Basic	0208 144 7057	£47.88	2GB	10GB	10	✓	✗	✗	✓	✓	✓	✓	✓
VARiHOST (www.varihost.net)	WordPress Plus	0208 144 7057	£71.88	4GB	100GB	100	✓	✓	✗	✓	✓	✓	✓	✓
VARiHOST (www.varihost.net)	WordPress Extra	0208 144 7057	£119.88	unlimited	Unlimited	Unlimited	✓	✓	✗	✓	✓	✓	✓	✓
WebFusion (www.webfusion.co.uk)	Fusion Professional	0845 130 1602	£107.40	5GB	50GB	1,000	✓	✗	✓	✓	✓	✓	✓	✗
WebFusion (www.webfusion.co.uk)	Fusion Business	0845 130 1602	£179.40	10GB	150GB	1,500	✓	✗	✓	✓	✓	✓	✓	✗
WebFusion (www.webfusion.co.uk)	Fusion Developer	0845 130 1602	£227.40	20GB	300GB	5,000	✓	✗	✓	✓	✓	✓	✓	✗
Z-Host (z-host.co.uk)	Z-100	N/A	£15	100MB	10GB	Unlimited	✓	✓	✓	✓	✗	✓	✓	✓
Z-Host (z-host.co.uk)	Z-200	N/A	£21	100MB	10GB	Unlimited	✓	✓	✓	✓	✗	✓	✓	✓
Z-Host (z-host.co.uk)	Z-500	N/A	£42	500MB	20GB	Unlimited	✓	✓	✓	✓	✗	✓	✓	✓
Z-Host (z-host.co.uk)	Z-1000	N/A	£60	1GB	40GB	Unlimited	✓	✓	✓	✓	✗	✓	✓	✓
Zen Internet (www.zen.co.uk)	Bronze (Linux)	0845 058 9000	£47.88	2GB	10GB	10	✓	✓	✓	✓	✓	✓	✓	✗
Zen Internet (www.zen.co.uk)	Silver (Linux)	0845 058 9000	£95.88	5GB	50GB	25	✓	✓	✓	✓	✓	✓	✓	✗
Zen Internet (www.zen.co.uk)	Gold (Linux)	0845 058 9000	£143.88	10GB	100GB	50	✓	✓	✓	✓	✓	✓	✓	✗
Zen Internet (www.zen.co.uk)	Platinum (Linux)	0845 058 9000	£239.88	50GB	250GB	100	✓	✓	✓	✓	✓	✓	✓	✗
Zen Internet (www.zen.co.uk)	Reseller (Linux)	0845 058 9000	£479.88	150GB	500	250	✓	✓	✓	✓	✓	✓	✓	✗

Golden rules to top hosting

We identify and explain the key criteria for success...

1 The best resources for you

Selecting your ideal package is largely determined by the kinds of resources and quantity of features you require from your hosting solution. Key aspects such as web space and monthly bandwidth are important for those services likely to be subjected to heavy amounts of traffic, which is why package solutions for enterprise applications are typically much more expensive. The general rule of thumb is not to buy more than you need or underestimate potential requirements.

2 Competitive and reliable

The hosting market is big business and hosting providers do try to drive prices down in a bid to entice your custom. Use our chart to compare costs, but be sure to visit the vendor websites to keep track of the latest deals, as they do change. Remember that low price should not always be a deciding factor and that paying a premium for a more reliable, trusted and experienced vendor can offer you much better value for money in the long term.

3 Putting you in control

Modern hosting is all about giving customers the power to set up, monitor and maintain their web space with minimal fuss. Most commercial vendors offer access to award-winning and intuitive control panels that enable you to log in remotely and intuitively tweak your account – without the need to relay complicated instructions down the phone. Be sure to find out from your potential host as much as you can about the control panel and request a demo.

4 Fantastic customer support

If all else fails and you need some extra help to get your hosting back online, then a commitment to future customer support is key. Many vendors offer a service-level agreement which outlines what you can expect here, however most will be more explicit about whether phone support is included or email contact is preferred. Think about what you need for your own peace of mind and factor good, comprehensive technical support against the price.

ADVANCED PHOTOSHOP
FREE RESOURCES WORTH $225+
The magazine for Adobe Photoshop professionals
PAINT WITH MASKS
PRO TYPE SECRETS
LEARN TO CREATE REFLECTION MAPS
DYNAMIC PEN TOOL TRICKS
CREATE AMAZING LOW-POLY EFFECTS WITH THE EXPERT SHAPE TECHNIQUES INSIDE
HOW TO ILLUSTRATE WITH IMPACT
FREE TEXTURES WORTH $63
COMMERCIAL PHOTOSHOP
CREATE A 3D INFOGRAPHIC
EXPERT GUIDE TO PHOTOMANIPULATION
PHOTO EFFECTS
BLEND MODES
DESIGN RETRO-STYLE LOGO TEXTURES
EXPERT TRICKS HISTOGRAMS
FREE RESOURCES WORTH £100+
FREE RESOURCES WORTH £1,200
FREE TEXTURE PACKS WORTH $63
ADOBE BRUSH
KEY SKILLS FOR PHOTO COMPOSITES

{code club}

Can you volunteer for Code Club?

Code Club is a nationwide network of volunteer-led after school coding clubs for children aged 9-11.

We need people who know how to program computers to volunteer to run a club at their local primary school, library or community centre for an hour a week.

We create the projects for our volunteers to teach, the projects we make teach children how to program by showing them how to make computer games, animations and websites.

Get involved, let's teach the next generation to code!

Visit **www.codeclub.org.uk** to find out more

trade secret

noun {c} ~tred si kriht - A secret formula, method, or device that gives one an advantage over competitors.

Low cost address lookup for your website

- ➔ Buy online from £15 + vat
- ➔ Eliminate spelling mistakes
- ➔ Reduced abandoned carts
- ➔ 80% reduction of keystrokes entering addresses
- ➔ Create a professional image for your clients
- ➔ Simple integration with SDK & working examples

From less than 1p per lookup

For more details visit postcode-software.net or call 0845 83 82 666

«clikpic» Websites for photographers

Try the new Clikpic for only £40 pa includes ecommerce

- Stunning range of new templates.
- New admin system makes it even easier to use.
- Stylish slideshows and carousels throughout.
- New blog styles with extra options.
- Easy upload and link to your social media.
- And much more besides!

Compatible with mobile phones and tablets

Visit www.clikpic.com for a FREE 14 day trial

BLACKNIGHT SOLUTIONS .HOST Pioneer

I spy with my little eye **domains ending in...**

.black .buzz .care **.club**
.guru .ink .host **.london**
.ninja **.party** .press .rocks
.science .social **.website** .wiki

Think
Different

Think
Context

Think
Available

www.blacknight.host sales@blacknight.host @blacknight

Domains
.COM just £5.80/pa

Hosting
From £2.99/pm + FREE Domain

Cloud
SSD Server from £5/pm

Servers
Save 50% for 3 months

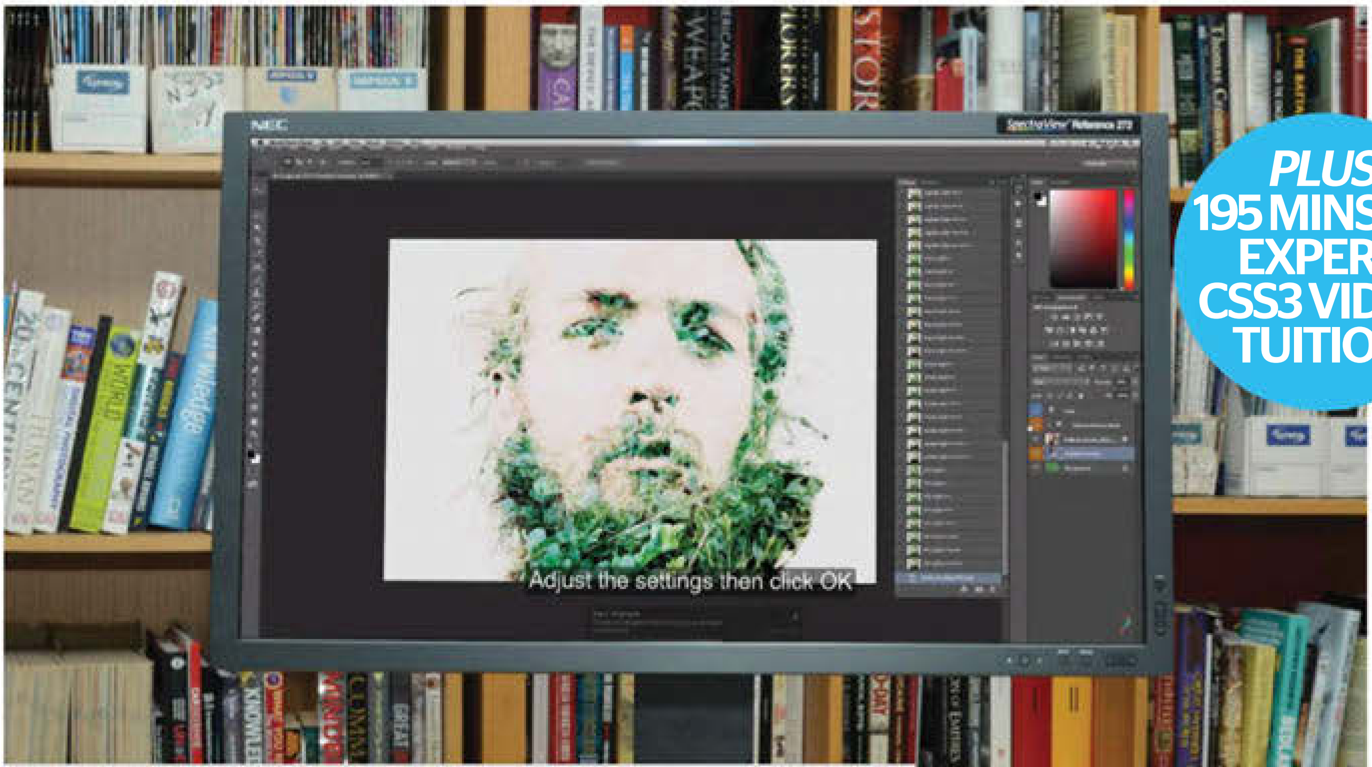
Datacentre
Rackspace from £35/pm

netcetera®

☎ 0800 808 5450
netcetera.co.uk/wdm

YOUR FREE RESOURCES

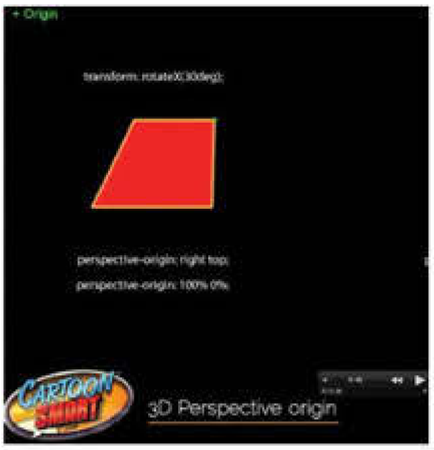
LOG IN TO WWW.FILESILO.CO.UK/WEBDESIGNER AND DOWNLOAD THE LATEST COLLECTION OF MUST-HAVE VIDEOS AND RESOURCES



PLUS 195 MINS OF EXPERT CSS3 VIDEO TUITION

DOUBLE EXPOSURE PHOTOSHOP KIT

COMBINE TEXTURES & GRADIENTS FOR KILLER GRAPHICS


+ Origin

transform: rotateX(30deg);

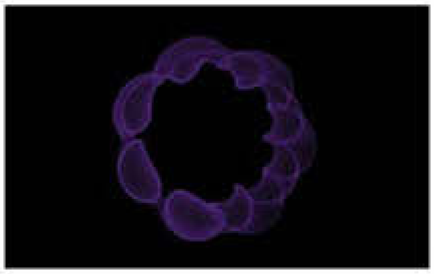
perspective-origin: right top;

perspective-origin: 100% 0%;

EXPERT CSS3 VIDEOS

TRANSITIONS AND ANIMATION

TOTAL RESOURCES 67



TUTORIAL FILES

THIS ISSUE'S ASSETS

YOUR BONUS RESOURCES

ON OUR BRAND NEW FILESILO THIS ISSUE YOU WILL FIND AN UNMISSABLE COLLECTION OF VIDEO GUIDES AND RESOURCES

- 195 Minutes of expert CSS3 video guides from CartoonSmart
- 40 Brick Mosaics Actions from SparkleStock (worth $5)
- 3 Web templates: Creative Pix photography template, Trail responsive and Umbrella mobile template from w3layouts
- The Mesmerize and Manbow font from Typodermic fonts
- Double Exposure Photoshop kit from SparkleStock (worth $14)

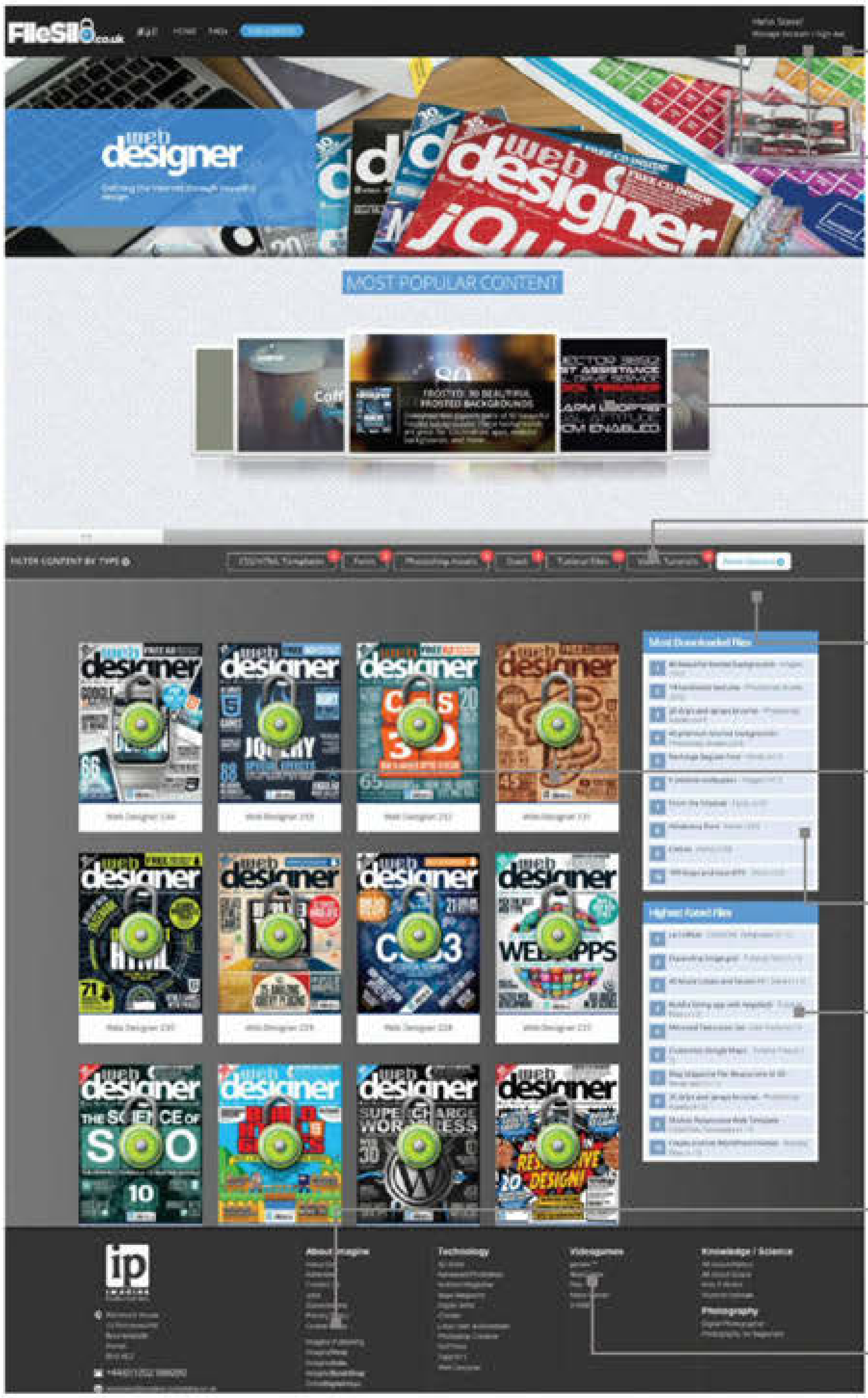
FileSilo

www.filesilo.co.uk/webdesigner

FILESILO – THE HOME OF PRO RESOURCES

DISCOVER YOUR FREE ONLINE ASSETS

- A rapidly growing library
- Updated continually with cool resources
- Lets you keep your downloads organised
- Browse and access your content from anywhere
- No more torn disc pages to ruin your magazines
- No more broken discs
- Print subscribers get all the content
- Digital magazine owners get all the content too!
- Each issue's content is free with your magazine
- Secure online access to your free resources



This is the new FileSilo site that replaces your disc. You'll find it by visiting the link on the following page.

The first time you use FileSilo you'll need to register. After that, you can use the email address and password you provided to log in.

The most popular downloads are shown in the carousel here, so check out what your fellow readers are enjoying!

If you're looking for a particular type of content like brushes or fonts, use the filters here to refine your search.

Can't find the resource you're looking for in these filters? Click on More Options to specify exactly what kind of resource you want.

Green open padlocks show the issues you have accessed. Red closed padlocks show the ones you need to buy or unlock.

Top Downloads are listed here, so you can get an instant look at the most popular downloaded content.

Check out the Highest Rated list to see the resources that other readers have voted for as the best!

Find out more about our online stores, and useful FAQs like our cookie and privacy policies and contact details.

Discover our amazing sister magazines and the wealth of content and information that they provide.

HOW TO USE FileSilo

EVERYTHING YOU NEED TO KNOW ABOUT ACCESSING YOUR NEW DIGITAL REPOSITORY

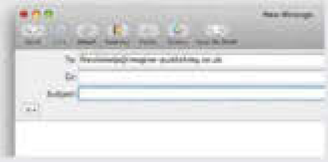
To get access to FileSilo, please visit www.filesilo.co.uk/webdesigner

01 Follow the instructions on-screen to create an account with our secure FileSilo system, or log in and unlock the issue by answering a simple question about the magazine. You can access the content for free with your issue.

02 If you're a print subscriber, you can easily unlock all the content by entering your unique Subscriber ID. You can find this on all correspondence from Imagine Publishing, including the clear plastic envelopes your magazine gets delivered to your door in every month.

03 You can access FileSilo on any desktop, tablet or smartphone device using any popular browser (such as Safari, Firefox or Google Chrome). However, we recommend that you use a desktop to download content, as you may not be able to download files to your phone or tablet.

04 If you have any problems with accessing content on FileSilo, or with the registration process, take a look at the FAQs online or email filesilohelp@imagine-publishing.co.uk.



SLIDE-IN MENU
OBSTER
Off canvas menu
Create a slide-in menu...
CSS SCROLLING BACKGROUND
Hello, we're Elespacio.
Create a constant scrolling background
Use CSS to add animation to backgrounds...
JQUERY
OVER THE HILLS
the Dead pirates
Line animations with jQuery
Create a Looney Tunes style opener...

MORE TUTORIALS AND INSPIRATION?

Web Designer is all about giving web designers and developers what they want. Make sure that you pay a visit to the Web Designer website where you will find a treasure trove of design and development tutorials. All the latest technologies and tools are covered including HTML5, CSS3, jQuery, PHP and responsive design. Plus, delve into the lives of the world's biggest agencies with our exclusive interviews. Find out how they design and develop some of the best websites seen on the web.

www.webdesignermag.co.uk

Issue 237 of web designer is on sale 25 June 2015 from GreatDigitalMags.com

UNLOCK THE POWER OF ANDROID

www.littlegreenrobot.co.uk

EVERY ISSUE • FREE RESOURCE DOWNLOADS

Android magazine

Available from all good newsagents and supermarkets

ON SALE NOW

› 20 Lollipop Fixes › What to expect from Android M › Join the mod generation

HARDWARE REVIEWS

ANDROID HACKING

TIPS & TRICKS

APP REVIEWS

HELP & ADVICE

BUY YOUR ISSUE TODAY

Print edition available at www.imagineshop.co.uk

Digital edition available at www.greatdigitalmags.com

Available on the following platforms

Print Tablet Phone Desktop

facebook.com/littlegreenrobot twitter.com/lgrobot

BE AN HTML

EXPERT

Eight pages of tips, techniques and tricks to help you to design and code like the pros

IS YOUR CONTENT STRATEGY KING?

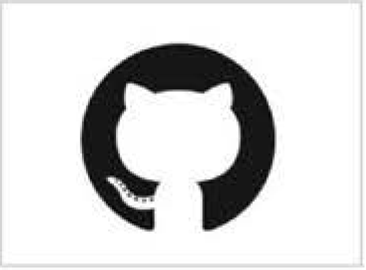
Find out what you need to know to make your content shine

SUPERCHARGE HTML5 VIDEO

Discover the power of Popcorn.js and how to add interactivity

BUILD APPS WITH ELECTRON

Combine HTML, CSS and JavaScript to create cross-platform apps



25 BEST GITHUB PROJECTS

The freshest frameworks, libraries and tools that you need to know

ALL IN YOUR NEXT

WEB DESIGNER

Issue 237 on sale

Thursday 25th June 2015

Synology®

Full HD 1080

dlna®

DS415 play

FOUR-BAY NAS SERVER FEATURING VIDEO TRANSCODING WITH LOSSLESS QUALITY

ENJOY FULL HD STREAMING QUALITY ON TV

Featuring 1080p Full HD video transcoding, DS415play satisfies multimedia enthusiasts' requirement for on-demand high-quality video streaming.

A TRUE DLNA® CERTIFIED MEDIA SERVER

Serving as a ready DLNA DMS (Digital Media Server) with the capability to host digital multimedia contents for remote playback, DS415play could stream music, photos and videos to DLNA-compliant devices.

BACKED UP WITH ENHANCED COMPUTING PERFORMANCE

Dual core CPU powered by a built-in Floating-Point Unit and hardware transcoding engine, speeds up thumbnail processing and enables photo-viewing. A refreshingly new experience.

Other models available include DS214play, DS213j and DS414j. To see the full product range visit **www.synology.com**

Where to Buy

amazon.co.uk dabs.com ebuyer.com broadband buyer.co.uk™

Award & Recognition

expertreviews BEST BUY

WORTH BUYING

EDITOR'S CHOICE

No Cowboys!