

27
PAGES OF
EXPERT TUTORIALS

web

FREE 145 MINS OF
CSS VIDEO

designer™

 HTML5  CSS3  jQuery  WordPress

www.webdesignermag.co.uk



**BUILD
WITH WEB
COMPONENTS**

JAVASCRIPT 6
WHAT'S NEW?

Experts reveal the best new features of
the web's No. 1 scripting language

JS 6

UNIQUE CSS LAYOUTS
USE CSS STATES TO BREAK FREE OF THE STANDARD

BUILD A BRAND
MUST-KNOW RULES TO CREATE A DISTINCT IDENTITY

FREE DESIGNER
TOOLS RESOURCES

WEB TEMPLATES, FONTS,
BACKGROUNDS & EFFECTS

Synology®



THE MOST REQUESTED
SYNOLOGY NAS EVER IS HERE!

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™
buyer.co.uk](http://broadbandbuyer.co.uk)

Award & Recognition

[expertreviews](http://expertreviews.com)
BEST BUY



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



I pinch myself when I look at our client list

Web Designer talks to AnalogFolk, creatives using digital to make the analog world better. Page 40

Look to the future with ES6 and Web Components



JavaScript is an unsung hero in the world of web development and design. It's everywhere making sure that sites are working exactly as expected. When a new version hits the streets its big news. It may not be obvious to all, but behind the scenes it will be making the web a better and more efficient place. This issue we get industry experts to run through the major new features on offer. They explain what they do, how they work and how they will affect future builds. Plus, we have a tutorial on page 76 offering a practical look at some of the features in use.

Looking to the now and future of HTML, it is difficult to avoid Web Components. With browser support fully imposed on Chrome and the others very close, now is the time to embrace them. Turn to page 46 to discover the four core elements that make up Web Components, what they do, the best tools to build with and the best resources out there. Plus, we have included a 22-step tutorial on how to create your own web component. Sticking with the future theme we reveal the must-know rules to create a distinct and unique brand. It's much more than just a logo, as you can find out on page 70.

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.

ES6 is going to empower JS developers, offering well-organised code that is easily scalable, testable and understandable

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

55 Designer resources

- Video Tuition** - CSS3: 145 Minutes of expert video guides from CartoonSmart (www.cartoonsmart.com)
- Backgrounds** - 32 Hi-res Tokyo Aerial Cityscapes from SparkleStock
- Templates** - 3 Template packs from w3 layouts; flat invitation form, Midget responsive portfolio template, web app log-in screen template
- Fonts** - Kingsbridge, Hoverunit and Kilsenburg from Typodermic



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...

☞ ES6 will introduce new syntax and features to JavaScript, designed to make life easier for developers and solve common programming problems ☞



David Boyer

David has over ten years of experience and is a senior developer within NHS Wales using ColdFusion, PHP and Node.js to develop web-based health applications. This issue he casts his eye over the new features in the latest and greatest version of JavaScript. **Page 28**

Mark Shufflebottom



Mark is a professor of Interaction Design at Sheridan College, Ontario. This time out he unveils the art of code-free PhoneGap builds. Discover how to take control via a visual interface with the help of the new Desktop app. **Page 54**



Luke Guppy



Luke is a senior front-end developer at Redweb and has been in the industry for the best part of a decade. This issue he reveals how Web Components are the future of HTML and how they should be used in future builds. **Page 46**



Tom Wittlin



Tom is the creative director at online retail specialists Folk. He has been creating award-winning brands for over 10 years. This issue he reveals the must-know rules that contribute to creating a unique, attention-grabbing identity. **Page 70**



Neil Pearce



Neil is a designer and instructor for the Envato network. He is also a keen exponent of CSS. This issue he demonstrates how to break free of the standard rectangular layout using CSS Shapes. **Page 58**

Sean Tracey



Sean is a creative technologist living and working on the South coast. He loves playing with the latest JS APIs he can get his hands on. This issue he reveals how to build fast animations with the Famous framework. **Page 62**

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 16**

Tim Stone



Tim is a lead front-end developer at Redweb and has a first-class degree in Interactive Media Production. This issue he shows how to streamline your front-end workflow using jspm for snappier performance. **Page 76**

Tam Hanna



Tam is a veteran in the world of coding and has worked with a host of coding languages. This issue he tackles the art of creating iOS and Android native apps from a single code base using the NativeScript framework. **Page 82**

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

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
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

James Watts, Fabio Torlini, Mark Billen, David Boyer, Matt Gifford, David Howell, Luke Guppy, Mark Shufflebottom, Neil Pearce, Sean Tracey, Tom Wittlin, Tim Stone, Tam Hanna

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

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.



© Imagine Publishing Ltd 2015
ISSN 1745-3534



Receive £25k
tax-free to retrain as
a computing teacher

and help develop
the great minds
of tomorrow.



Switching to a great career as a computing teacher is more achievable than you think. You could receive a £25k* tax-free bursary or a £25k* scholarship provided by BCS, The Chartered Institute for IT. Alternatively, you could earn a salary of up to £25k* while you train. Then you can inspire young people to fulfil their potential.

Apply now.

Visit education.gov.uk/teachcomputing

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](https://twitter.com/WebDesignerMag) 📍 www.webdesignermag.co.uk

Quick look...

- 08 Reader survey 2015**
Want to meet the **Web Designer** team? Shape future issues of the mag? Then take the **Web Designer** survey for a chance to make it happen
- 10 Is it really the end for IE?**
Has the Microsoft browser as we know it finally drawn its last breath? **Web Designer** investigates
- 12 Webkit: The best must-try resources out there**
Discover the libraries and frameworks that will make your site a better place to visit
- 13 Patterns on PHP**
CakeFest organiser James Watts unravels the mystery of design patterns
- 16 Lightbox**
A showcase of inspirational sites and the techniques used to create them
- 34 On a mission**
Have you got what it takes to be a *Kingsman* knight? UNIT9 reveal how they created a compelling online interview
- 40 Digital experience, analog world**
AnalogFolk tell **Web Designer** how they are helping the world's leading brands make meaningful connections with their customers
- 46 Web Components: The future of HTML**
Get expert advice, indepth insight and practical guidance for building sites for the future today
- 70 Build your own brand**
Unlock the rules to create a unique and distinct identity for your business
- 88 Hosting listings**
An extensive list of web hosting companies. Pick the perfect host for your needs
- 98 Next month**
What's in the next issue of **Web Designer**? Find out what you can look forward to

THE NEW JAVASCRIPT STANDARD

Cover focus

28



40
ProFile: AnalogFolk
Creating value in people's lives via digital experiences



20
Lightbox: Reebok
Admire the beauty of the Be More Human project

FileSilo

94 Get the latest must-have resources and videos

A comprehensive collection of free designer resources!

- 145 Minutes of CSS3 video guides
- 32 Tokyo cityscape backgrounds
- 10 Glitch art Photoshop actions
- 3 Commercial fonts from Typodermic



Visit the **WEB DESIGNER** online shop at imagineshop.co.uk for back issues, bookazines and DVDs

<tutorials>

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

- 54 **Code-free PhoneGap development**
Make files run and behave as a native app while designing and developing with the new PhoneGap Desktop app
- 58 **Build unique CSS layouts**
Employ the power of CSS Shapes to break free of the standard rectangular design pattern
- 62 **Fast animations with Famo.us**
Implement the JavaScript library to take advantage of CSS3 and build a dynamic image gallery

BUILD A BRAND



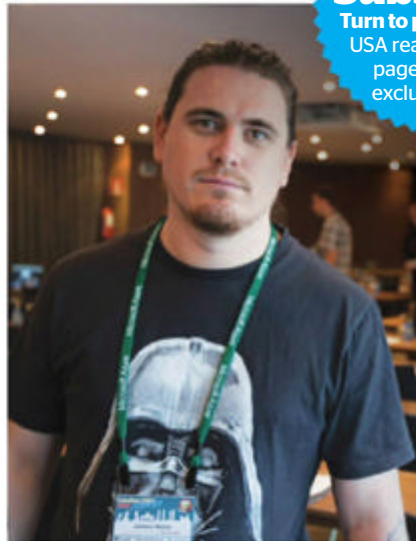
70

Discover the must-know rules to create a unique and distinct identity

<header>

The tools and trends to inspire your web projects

- 08 **Reader survey 2015**
Want to shape future issues of the mag? Then take the **Web Designer** reader survey for a chance to make it happen
- 10 **Bye bye Internet Explorer**
Will designers and developers finally be free of the shackles of the long-standing Microsoft browser?
- 12 **Webkit: The best must-try resources out there**
Need a new framework or library? Then this is where you need to start
- 13 **Comment: James Watts**
Design patterns: a generic solution to common problems



Design patterns are an invaluable ally when building clean, consistent code which scales

James Watts

Never miss an issue
Subscribe
Turn to page 68 now
USA readers turn to page 87 for an exclusive offer



web workshop

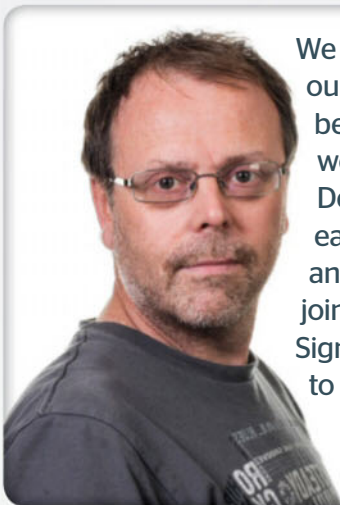
- 66 **Responsive fixed pages borders**
studiodyxe.com
Add a consistent coloured edge to a design

web developer

- 28 **JavaScript 6: What's new?**
Get ready for the latest standard in scripting. Discover the latest features and how to use them
- 76 **Manage dependencies with jspm**
Bring frictionless browser package management to your projects using the CLI
- 82 **Build a native web app**
Use NativeScript framework to make apps for iOS, Android and Windows Phone

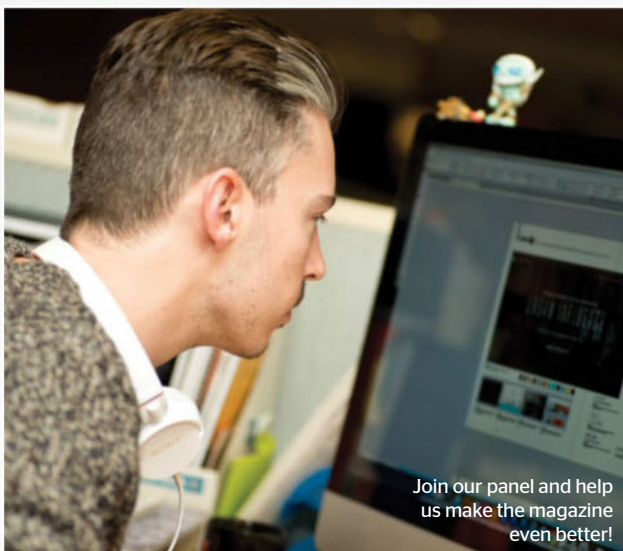
Join our ^{web} designer Reader Panel today!

Take our 3-minute survey at webdesignermag.co.uk/survey and win a place on our panel



We love making Web Designer and it's our mission to make it the best it can be. However, we're not perfect so we wondered how we could make Web Designer even better. That decision was easy, ask the readers for their help. By answering a few questions, you could join our first ever Web Designer panel. Sign up to help us to help you.

Steve Jenkins
Editor



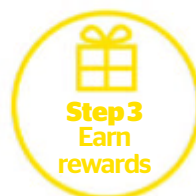
Join our panel and help us make the magazine even better!





Complete our survey to join our **Web Designer** panel and:

- See new ideas and changes before anyone else
- Get invited to attend special events in your area
- Test third-party products for review
- Have a chance to meet the Web Designer team
- Shape the future issues of the magazine
- Enter into exclusive competitions



Take the chance to get closer to the team than ever before

Complete our survey and win your place today
webdesignermag.co.uk/survey

<header>

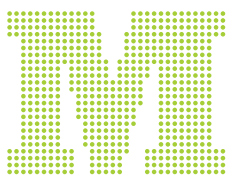
The tools and trends to inspire your web projects

If you have a creative project, new web product or great designer story, contact the editorial desk

✉ webdesigner@imagine-publishing.co.uk 🐦 @WebDesignerMag

Is it really the end for Internet Explorer?

Has the Microsoft browser as we know it finally drawn its last breath? What will replace it? Web Designer investigates



Microsoft Internet Explorer, three little words that strike fear into the heart of many a designer and

developer. The countdown to its demise has been a long and very slow process, but it seems that the browser is finally going to meet its maker in 2015.

Internet Explorer's popularity has been on the wane for years and attempts to bring it in line with web standards has failed. Later versions of the browser had little chance of success. Were IE8, 9 or 10 any good? It doesn't matter because less and less people were inclined to use them. With Firefox, and predominately Chrome, providing stiff competition, there was a shift away from IE.

Microsoft was making valiant attempts to resurrect the browser, but it seemed that

they never quite grasped what users wanted in a browser. Chrome was what users wanted and IE never even came close to its functionality, finesse and focus on development and experimentation.

There is a case that when you have an established brand that you stick with it as people will recognise it. However, as hard as Microsoft tried, the name was tarnished. The solution, and one that seemed blindingly obvious, is drop the name and start afresh with a new name and more open browser.

And, this is exactly what Microsoft finally hopes to achieve with its all-new browser currently known as Project Spartan.

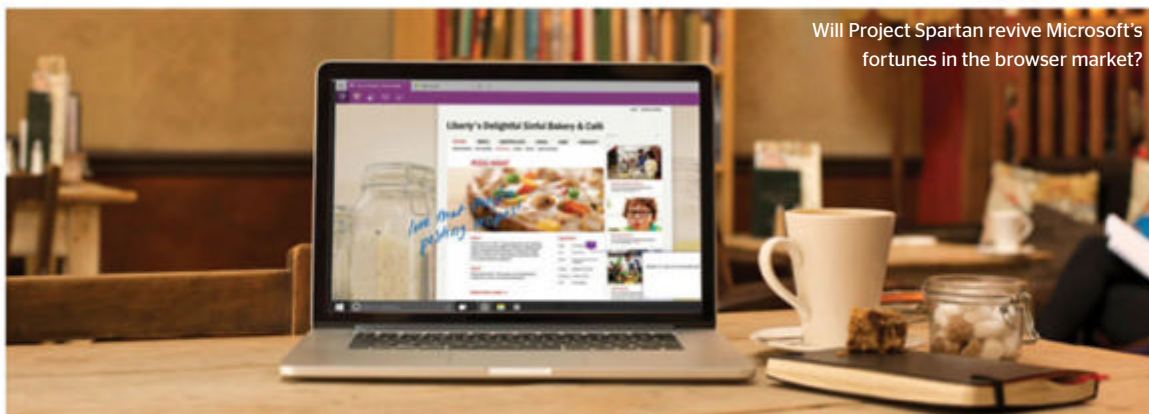
What is the new browser going to offer to tempt away the vast army of Chrome (mainly) and Firefox users? Plus, they can't forget mobile, like IE did.

Project Spartan is currently still in development, but Microsoft has revealed enough via preview builds for users to get a taste of what is to come in the future.

Let's start with the interface. If you have used Windows 8 or Windows Phone you will be familiar with what's on offer, lots of squares and the popular flat styling. What lies under the hood? Features worth a mention are Cortana integration, a reading mode and web notes. Cortana's voice recognition keeps it in line with other browsers, a reading mode is a simple, but hardly stunning addition. Web notes provides on-screen annotation. Is it a feature you would use? We will reserve full judgement until the finished article reaches the masses. But, are we excited? Not yet.

Getting back to IE, will it be missed? The answer for all designers and developers, bar a few die-hard fans, and perhaps those who have worked on IE will surely be good riddance. We know we won't miss it, but we know that people will still use it. It is worth noting that Microsoft has not totally given up on IE, it will be maintained for legacy reasons. But, please people, for the sake of designers and developers everywhere, please give it up.

🌐 Chrome was what users wanted and IE never even came close to its functionality and finesse 🌐



Will Project Spartan revive Microsoft's fortunes in the browser market?

Get Web Designer digital editions

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

Web Designer's latest digital addition to our burgeoning collection is **HTML5 Arcade**. Get an insight into the different games engines out there, plus discover how to start creating your very own with Phaser, Quintus and ThreeJS. Simply follow the expert tutorials.

Keeping our latest digital special company is **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

Making the Planet a Priority

aitw.dpw.lacounty.gov/?p=2356

1 To reinforce the message presented on the infographic, the background uses a recycled paper texture to add extra ambience.

2 The muted colour palette is ably accompanied by simple but instantly recognisable graphics and numerous facts and figures that really emphasises the amount of waste.



Colour picker

Hot hexadecimal codes

Finally got distract

bit.ly/1lZrb2z



#EE2037 #EE3559 #EF4983 #CA65A7 #A089C0

Dew blue

bit.ly/1aNCdmb



#56585D #3A383E #1D1F34 #212A5C #66849F #C0E5E6

Typesetter

The best fonts you need



Neon Display

myfonts.com/fonts/superfried/neon

A retro font that is perfect for re-creating neon sign graphics.

ABC abc

Muller

fontfabric.com/muller-font

A big font family that offers up some impressive flexibility.

Graphics

Great visuals to inspire



Grayscale vector

on.be.net/1AU7dAz

We are big fans of Joshua M. Smith's black-and-white illustrative work and what's not to like about a *Star Wars* and samurai combo?

Themematic

Style up your WordPress



F Magazine

demo.themeruby.com/fmagazine

A multitalented theme that offers neat well-defined layouts with full width images sliders and section to add a contemporary feel.

Sites of the month



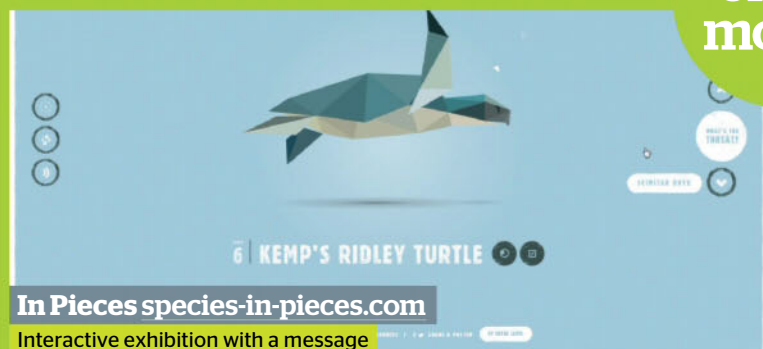
HBM FiberSensing fibersensing.com

Fullscreen sensibilities with big bold fonts



Stupid Studios stupid-studio.com

Fullscreen images matched with fullscreen animations



In Pieces species-in-pieces.com

Interactive exhibition with a message



Riga Ghetto Holocaust Museum rgm.lv

A sombre subject given great digital treatment

<webkit>

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



neovim

neovim.org

The Vim editor has been around for a while, but neovim is the next generation. It's still in development, but it's one to watch.



Burger

github.com/mblode/burger

Need a neat, easy-to-implement hamburger menu? Then try Burger, a minimalistic hamburger menu complete with slide-in fullscreen navigation.

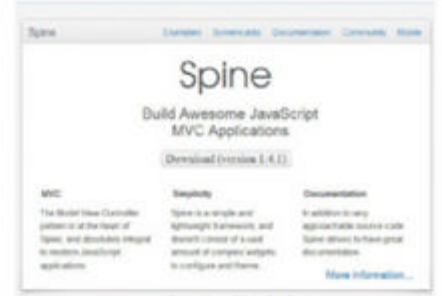


Titon

titon.io/en/toolkit

Titon is a collection of extensible front-end user interface components with a strong focus on the responsive and mobile devices. It uses HTML, CSS, JavaScript and jQuery to power a host of elements. Need to create a responsive and touch-friendly carousel? Need to build an off-canvas menu with multiple animation options? Want to create breadcrumb navigation? The framework has all this and much, much more.

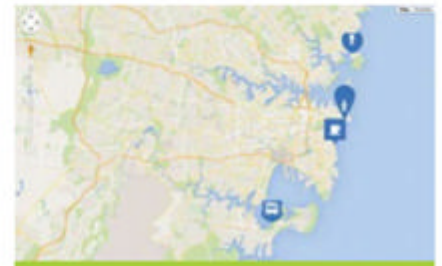
To appreciate what's on offer, there are plenty of demonstrations and extensive documentation to explain the finer points of use. Make sure to check out the Getting Started guide.



Spine

spinejs.com

A simple and lightweight framework for building JavaScript web applications. Spine's MVC structure "gets out of your way".



Map Icons

map-icons.com

A collection of markers and map-specific icons that work perfectly with the Google Maps API and Google Places API.

TOP 5 JS chart tools

Make your data beautiful with this collection of browser-friendly resources

01



Chart.js

chartjs.org

Chart.js uses HTML5 Canvas to help create one of six different chart styles. Each can be animated with plenty of customisation options.

02



ZingChart

zingchart.com

Create good-looking interactive real-time charts using JavaScript and styled with CSS. Offers over 20 chart types and hundreds of variations.

03



Canvas.js

canvasjs.com

An easy-to-use HTML5 JavaScript charting library built on the Canvas element. It creates a fast, lightweight and responsive end product.

04



AMCharts

amcharts.com

The online chart maker has a selection of different tools for building beautiful creations. Start with a template for quick and easy builds.

05



jChartFX

jchartfx.com

The jChartFX library uses HTML5, CSS and SVG to deliver "aesthetically superior charts". JSON support means data can be retrieved from most sources.

Design patterns in PHP

CakeFest organiser James Watts unravels the mystery of design patterns and how they provide a generic solution to common problems



For anyone who's not familiar with the concept of a design pattern in software development, it's basically a generic solution to a common problem. You can think of it as a blueprint,

acting as a guide of how to solve certain problems or scenarios when developing software. You may have even been introduced to design patterns if you've ever heard the terms factory, adaptor or strategy.

From the world of HTML and CSS, you may have implemented a design pattern already. For example, the separation of structure and presentation, where the CSS rules and declarations are segregated from the markup structure and semantics of HTML, is actually a design pattern. Design patterns can also enable other developers to quickly understand or become associated with the way software has been developed if they can identify with certain patterns from the onset. This is of huge benefit to teams where agility at the development level is important, as assumptions can be made about the architecture or code base if patterns have already been applied.

In contrast, there are antipatterns, which are quite the opposite: design patterns which cause problems and obstructions in code and architecture. For example, concepts known as div soup or divitis are in fact antipatterns. There are also some very specific antipatterns in PHP, such as the use of globals, defining multiple classes in a single file or liberal usage of eval. These are also sometimes referred to as 'smells' in code.

But, where did design patterns come from? The prevalent usage and application of design patterns didn't become popular until after 1994, when a book titled *Design Patterns: Elements of Reusable Object-Oriented Software* was published. This book really formalised and laid a foundation for design patterns to be taken into the mainstream of software



JAMES WATTS
Director at CakeDC
cakedc.com

James Watts has been involved in open source since around 2008, and using PHP for the last ten years. He's a core member of CakePHP, organiser of CakeFest and is currently the acting director of CakeDC. He's also an author, a speaker, community manager, husband and father.

development, and still remains a core book for anyone who is serious about building robust code which can stand the test of time.

As an example, let's look at the factory pattern, which is probably one of the most popular patterns found in any language. This pattern is used to create instances of classes, without exposing the instantiation details. So, first, the problem:

```
<?php
$car1 = new Ford('blue', 4);
$car2 = new Toyota('red', 2);
```

Here we've created two cars. One is a blue, four-door Ford, while the other is a red, two-door

Toyota. Both of these classes extend a base Car class. Notice how we've directly instantiated each of these with their class name, passing the details for each. The problem here is that we have no control over the instantiation logic when creating these objects, so if they're created at 200 locations over the code base, and we want to make a change, we'd have some work on our hands.

The factory pattern resolves this issue by introducing a single point of creation. The previous code would change to the following:

```
<?php
$car1 = CarFactory::create('Ford',
'blue', 4);
$car2 = CarFactory::create('Toyota',
'red', 2);
```

If we now want to change how cars are created, or introduce additional implementation specifics we need only to change that on the factory.

In my more than ten years of experience in professional PHP development I've found design patterns to be an invaluable ally when building clean and consistent code which scales. It's important to note that scale doesn't just mean infrastructure and performance, it's also affected by the development team's ability to grow and advance with the code base. A messy core, full of one-time hacks and creative patches to get things working can be a business' Achilles heel when the time comes to building out.

These tricks got the necessary results but can become the bottlenecks when features have to work around the mess, and the onboarding of new recruits becomes slow and expensive due to all of the artistic approaches taken in the code.

The use of design patterns can not only increase the quality of a code base, but also improve the way you approach problems. In an industry where the NIH (Not Invented Here) syndrome is widespread, building on the shoulders of giants and leveraging the experience of many who have come before you can give you a notable advantage when it comes to writing clean and elegant code like a pro.

WORDPRESS INSIDE OUT

WP Engine's Fabio Torini explains the importance of WordPress hosting

Q. What are the advantages for WordPress users if they were to take targeted WordPress hosting?

Dedicated WordPress hosting partners provide expertise and specialism that generalist hosting companies can't provide. By using a dedicated WordPress hosting provider, you're given access to the right infrastructure and also 24/7 access to WordPress expertise and trained professionals.

WP Engine goes even further, we provide a platform that is specifically optimised for WordPress. We provide benefits such as staging areas for testing, backups, reporting, security services and analysis. The WP Engine platform has approximately 4,000 rules that means that WordPress operates both faster and more securely.

Q. WordPress is the dominant force when it comes to web publishing platforms. But what should it offer?

WordPress is without doubt the most popular third-party CMS platform. Its usability, functionality and innovative community make it ideal for organisations looking for a reliable, flexible platform. That said, one area where WordPress has room for improvement is its search functionality, this does negatively impact the overall user experience.

To counter this, WP Engine is launching a new Elastic Search solution too. This will enable users to search more easily via the WordPress platform.

When considering the wider market, the biggest competitive challenge for WordPress does not actually come from differentiating itself against other vendors. It is to persuade IT decision makers and web developers that WordPress is competitive alongside custom build sites.

Q. How well does WordPress scale up and what issues can arise when a



FABIO TORLINI
EMEA MD, WP Engine
wpengine.com

As the MD of WP Engine, Fabio has over 15 years of experience in

the hosting and cloud industry, in addition to a thorough understanding of all aspects of IT B2B marketing, branding, PR, channel and management.

vendor decides to scale up their site to enterprise level?

WordPress is going through a journey, from a trail-blazing blog platform to a trusted CMS for SMBs and start-ups. In 2015, WordPress will firmly establish itself as a CMS capable of enterprise scale. For WP Engine we're already seeing this shift take place. We have numerous enterprise organisations using WP Engine and WordPress for their sites, including Network Rail and AMD. One of the largest sites we manage and host peaks at over 70 million visitors per month - a number not to be taken lightly!

As vendors choose to scale their business, the key is to work with

partners that can scale up alongside them and offer enterprise-class WordPress support and platforms. WordPress itself won't scale, so businesses need to look to specialists who can fully manage this process. WP Engine therefore creates clustered environments that enable businesses to scale up as needed.

A common challenge faced as brands looks to scale up is downtime. If the current hosting isn't suited to high volumes of traffic, the site will crash. It's therefore important that the hosting partner is able to provide support and quickly switch to a more suitable package that will provide for larger traffic volumes.

Q. The customer experience is a critical component in the success of any hosting enterprise. What needs to be offered to ensure success?

Top of the list is a specialisation in WordPress - with regular updates, changes to functionalities and new plugins on offer, high-level expertise is hugely critical to success.

Choosing a WordPress hosting provider that has people who code for WordPress at the core of their business is the easiest way to get access to this expert knowledge.

WordPress specialist hosting providers also have the luxury of accessing and testing new versions of WordPress before they launch. The hosting provider can offer early insight, and ensure that the update works when it's input into a vendors business.

Finally, the support provided on an ongoing basis is key. For enterprise-level organisations in particular, having 24/7 access to a support team of WordPress experts and a dedicated account manager is essential.

Q. As the web evolves how do you think hosting will keep up? Or, conversely will hosting get ahead of the web and influence its evolution?

The hosting world is full of rapid evolution. Customers are demanding more from their vendors - hardware management is not enough.

They want more in terms of support via cloud-based platforms, and so hosting providers will need to increase the level of service they provide. At WP Engine, we go above and beyond to manage end users, as we want to ensure they're utilising our expertise to get the best out of WordPress.

Web and cloud-based technologies make it essential for SLAs to be all encompassing across the entire platform, and not just for the infrastructure available.



Authorized
Reseller

NOW YOU CAN AFFORD A BRAND NEW MAC



LESS THAN
£2 PER DAY



YOU DO THE MATHS

ALL OUR FLEXI-LEASE PACKAGES INCLUDE:

MAINTENANCE

- **3 Years** full cover for all hardware with remote and telephone support, plus loan equipment

LEASE RENEWAL

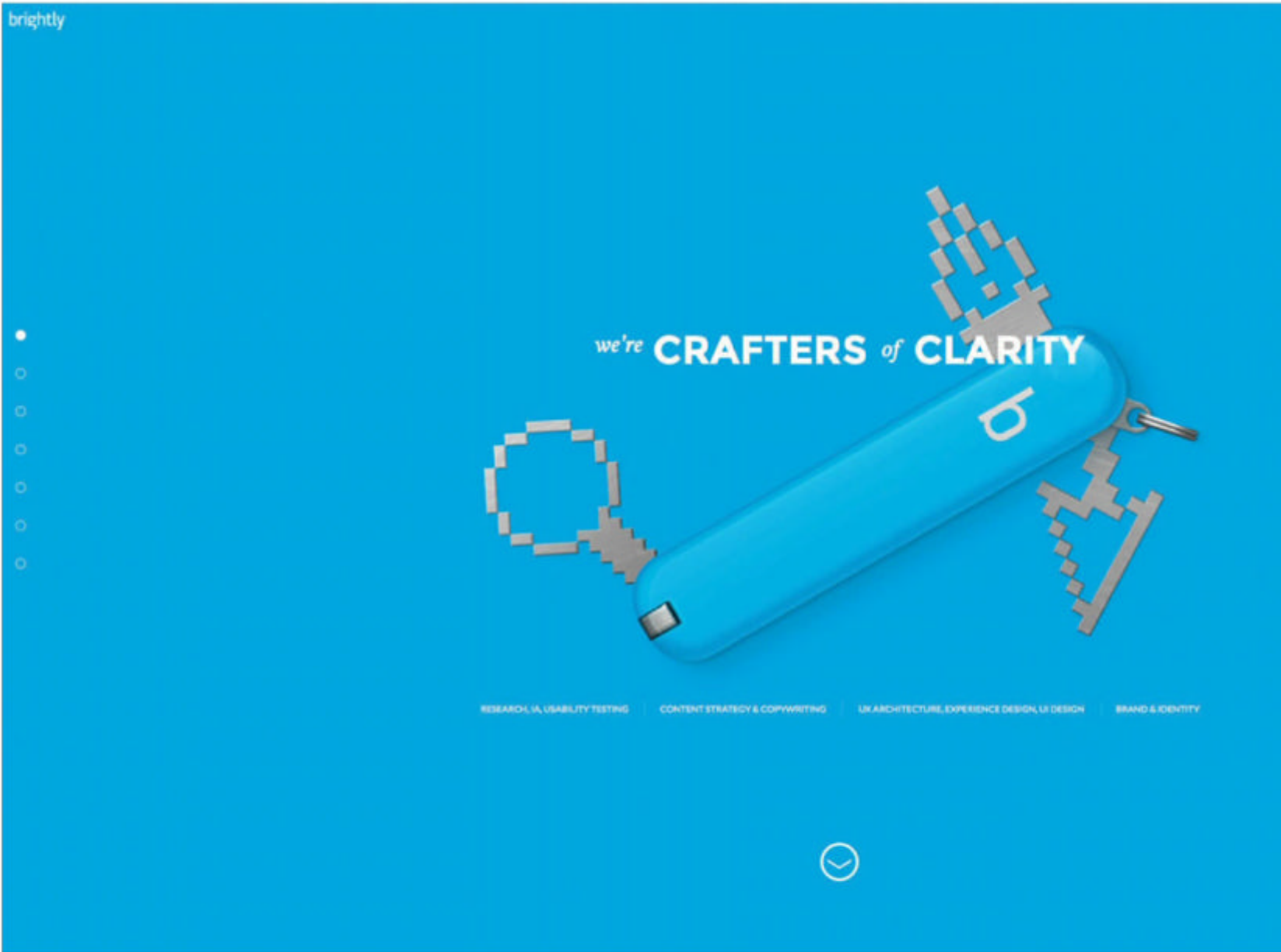
- after only **2 years** we will offer you a renewal to latest equipment and options to keep or return the old machine

OWNERSHIP

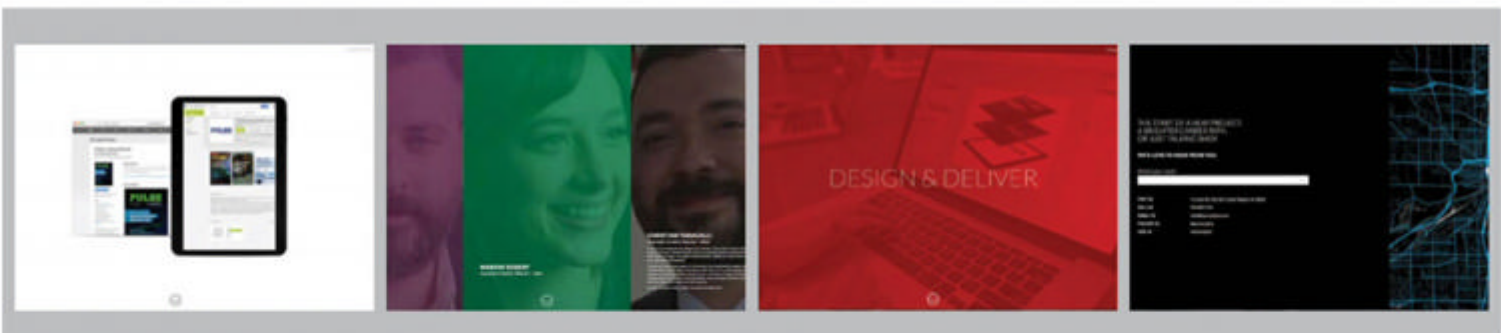
- at the end of the lease the equipment is yours for just **£1**

020 7111 1643
WWW.HARDSOFT.CO.UK

HARDSOFT
Established for over 25 years



<Above>
 • The site's main page thread is long and vertical, with sections also prompting horizontal navigation to show further content



<Above>
 • Project case studies are frequently illustrated with bold, crisp device shots showing websites and apps in action

<Above>
 • Each member of the brightly team is introduced using colourful, oversized portraits with expandable biographies

<Above>
 • Short looping video clips of the team at work provide full-motion backdrops for describing brightly's work process

<Above>
 • The site boasts one of the coolest reskins of an embedded Google Map on its stylishly minimal contact page



brightly digital agency

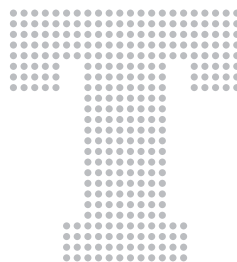
teambrightly.com

Development technologies WordPress, HTML5, CSS3, jQuery, SVG



Designer **brightly**
teambrightly.com

Notions of team and the importance of creative process are slickly celebrated within agency brightly's latest designer dotcom



The latest in a long line of noteworthy agency websites so far this year, this stellar effort from Michigan-based studio brightly is a shining example. Operating out of Grand Rapids, the website exhibits a rather apt, tangible sense of "team" that floods out from this cascading

WordPress-powered template.

Opening on an animated PNG of an unfurling Swiss Army knife, suitably reconfigured for digital deployment, the message is multidisciplinary. These "Crafters of Clarity", who are known for delivering projects across a broad industry spectrum, have a creative story to tell. Built around a long central 'page' that

behaves more like a vertical slider, sections branch off to screen right producing a waterfall-like feel to the navigation. Horizontal threads detailing team biographies, working process and featured work within web, mobile and software are all colourfully represented. "For our new site, we wanted to convey that we're more than designers; we're problem solvers," explains Christian Faragalli, associate creative director at brightly. "We've built a strong team of researchers and strategists, and those forces drive our process. It's allowed us to tackle projects of all size and complexity, and our site needed to reflect that. We also wanted to have a little fun. The idea was to make a cutting-edge site with clear messaging, easy navigation, and bits of visual storytelling throughout. We used bold colours and clear typography to achieve a classic look with a modern twist."

abcABC
1234567890

<Above>
• Abril Fatface by TypeTogether, available on Google Fonts is used for the down arrow icons

abcABC
1234567890

<Above>
• Montserrat by Julieta Ulanovsky available via Google Fonts kicks off the typography on brightly

The idea was to make a cutting-edge site with clear messaging, easy navigation, and bits of visual storytelling throughout



Swing open a structured animation with CSS

Justin Gauthier, UX designer at brightly explains how the Swiss Army knife on teambrightly.com was “crafted” using a combination of JavaScript and CSS

01 Exported assets

The technique starts with exporting the three, semitransparent knife blade PNG images in their final animation states. This means less code is needed to correctly place the images in the browser. The knife handle should be exported in its starting state.

02 Write a specific structure

Each blade is absolutely positioned relative to the handle. Since the knife as a whole rotates, we use a knife-container <div> for the whole element. CSS transformations for rotating the knife are applied here, while supporting heading elements fall outside this rotating <div>. Then wrap everything in a “Stage” <div> to provide scope to the animation and specifically style elements without affecting other site styles. Each blade needs a unique class name since there are multiple positions, with each blade moving independently:

```
001 HTML:
002 <div id="Stage">
003 <h1><span>we're</span> Crafters
<span>of</span> Clarity</h1>
004 <div class="animation-container">
005 <div class="knife-container">
006 
007 
008 
009 
010 </div>
011 <h2>We have the tools to solve any
problem</h2>
012 </div>
013 </div>
```

03 Position the elements

Next we add CSS transformations to position the knife handle in its final state. Once the handle is in its final state, you can position the blades where they need to be by using -transform rotate and -transform-origin. After this is applied, you can see if your HTML structure is working by seeing whether the blades are rotating with the knife handle. Then apply absolute positioning to the blades and place them in their desired position.

```
001 CSS:
002 .knife-container {
```

```
003 position: relative;
004 z-index: 2;
005 width: 100%;
006 height: 190px;
007 /* Vendor prefixes here */
008 transform-origin: 71% -2%;
009 /* Vendor prefixes here */
010 transform: rotate(-30deg);
011 }
012 .knife-container img {
013 position: absolute;
014 top: 0;
015 left: 0;
016 z-index: 1;
017 }
018 .knife-container .knife { z-index:
2; }
```

04 Define the starting point

For animation start points we add closed CSS classes for each blade. Next, move the rotate transform from .knife-container into a separate .knife-rotate class. These classes will be used in the JavaScript function to launch and restart the animation. Use the same CSS -transform rotate and -transform-origin to place the knife blades into closed states, and -transform scale to reduce the size and hide the blades behind the handle.

```
001 CSS:
002 .knife-container .glass {
003 top: -279px;
004 left: -114px;
005 /* Vendor prefixes here */
006 transform-origin: 100% 100%;
007 }
008 .knife-container .glass-closed {
009 /* Vendor prefixes here */
010 transform: rotate(123deg)
scale(0.72);
011 }
012 .knife-rotate {
013 /* Vendor prefixes here */
014 transform: rotate(-30deg);
015 }
```

05 Apply CSS transitions

After defining the start and end points of the animation, add the transition effects to the blades and the knife-container <div> to smoothly animate from beginning to end. Since the blades flip out sequentially, apply a CSS -transition-delay in addition to the CSS -transition to create the effect:

```
001 CSS:
002 .knife-container .arrow, .knife-
container .glass, .knife-container .pen
{
003 /* Vendor prefixes here */
004 transition: 0.55s ease-in;
005 }
006 .knife-container .glass {
007 top: -279px;
008 left: -114px;
009 /* Vendor prefixes here */
010 transform-origin: 100% 100%;
011 /* Vendor prefixes here */
012 transition-delay: 0.2s;
013 }
```

06 Set the stage

Once the transition effects are added to the knife handle and all of its blades, add the closed CSS class to the blades in the HTML. This class is necessary because on page load, the knife should be closed:

```
001 
002 
003 
```

07 Switch the classes

Once the stage is set, write a JavaScript function to be called on page load. The function simply removes the closed class on all of the blades and this then adds the knife-rotate class to the knife container, which rotates the knife:

```
001 JAVASCRIPT:
002 function openKnife() {
003 $(".knife-container").
addClass("knife-rotate");
004 $(".animation-container h2").
addClass("hide-caption");
005 $(".pen").removeClass("pen-
closed");
006 $(".glass").removeClass("glass-
closed");
007 $(".arrow").removeClass("arrow-
closed");
008 }
```


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!

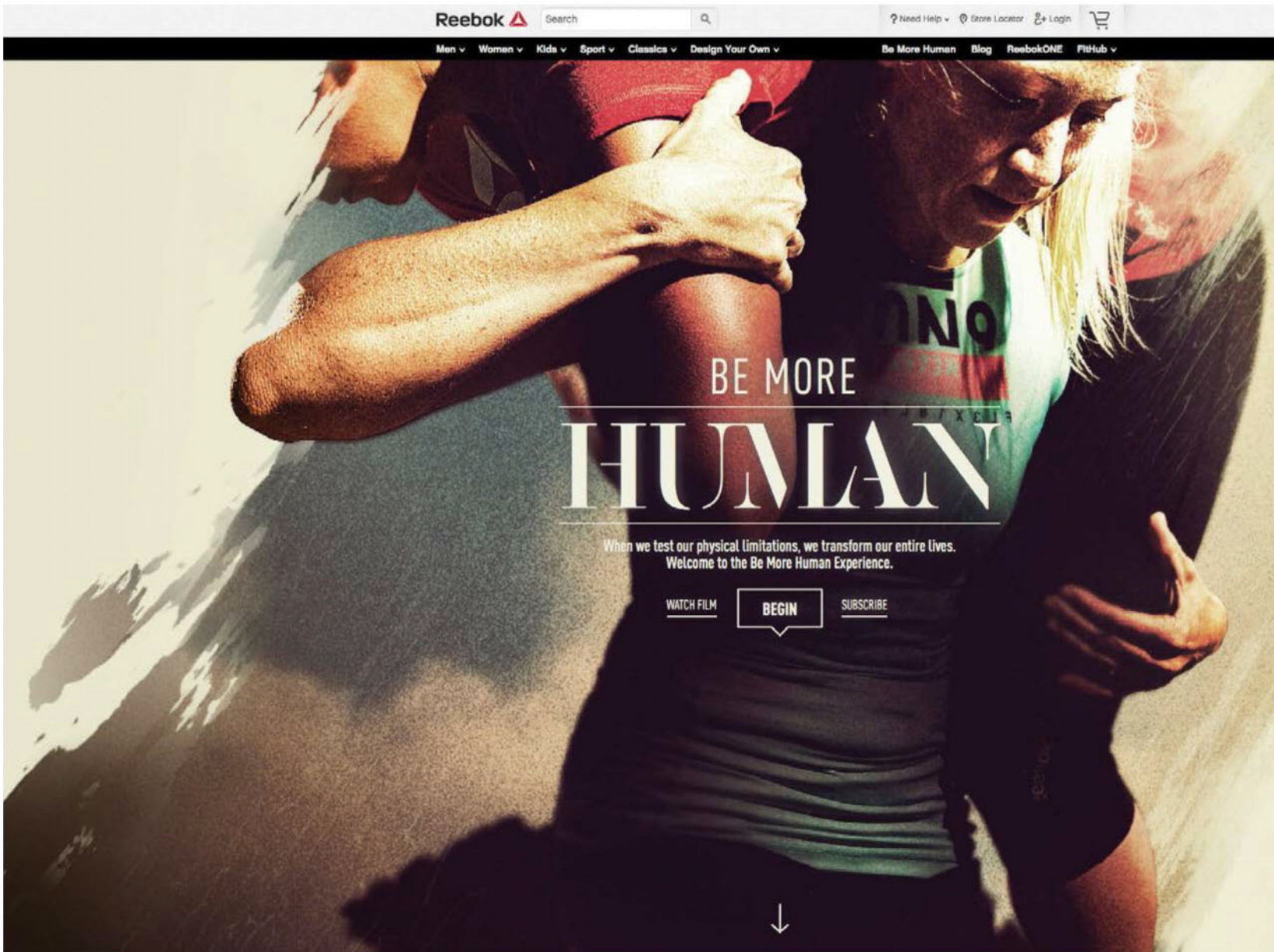
*** IT'S A CONTEST!!**

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

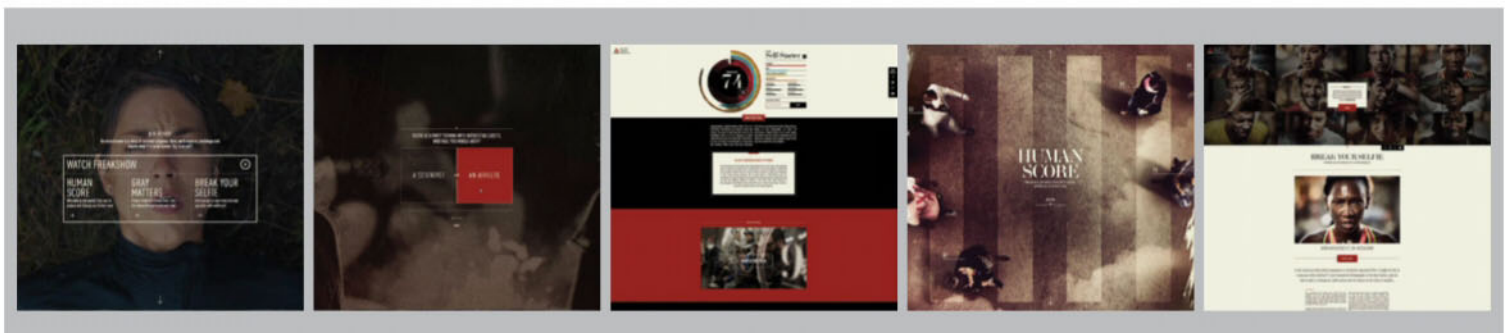
by



CoffeeCup



<Above>
 • The site promotes Reebok by leveraging a notion that testing our physical limitations will “transform our entire lives”



<Above>
 • Backed by the campaign film via HTML5 video, visitors can quickly navigate throughout the experience

<Above>
 • The Human Score test asks a series of hypothetical questions to assess your views on fitness and character traits

<Above>
 • A results page gives participants a dynamic, colour-coded visual score breakdown, with fitness feedback

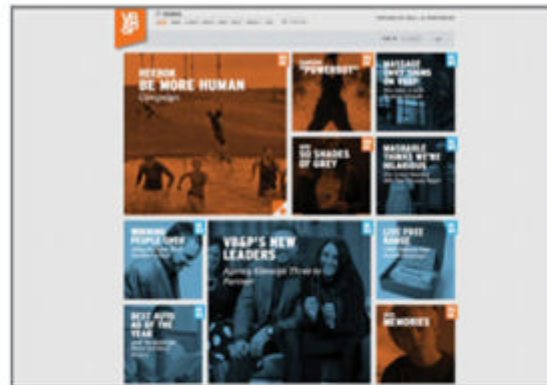
<Above>
 • Stylised sporting photography and people shots are combined with a watercolour-like aesthetic

<Above>
 • The Break Your Selfie section asks questions inspired by Sandro, Lucie Foundation’s Photographer of the Year

Reebok - Be More Human

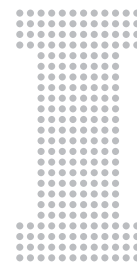
fitness.reebok.co.uk/be-more-human

Development technologies HTML5, CSS3, jQuery, Demandware



Designer **Venables Bell & Partners**
venablesbell.com

Reebok encourages fitness fans to “Be More Human” and share their stories of physical endurance via this high-profile campaign



If there's one thing that defines us as humans, it's the desire to push ourselves forward with challenging endeavours. Believe it or not, it isn't the desire or ability to take selfies. This idea is kind of central to this rich promotional experience from Reebok and the people who are behind the campaign at San Francisco consultancy Venables Bell & Partners.

The site basically hopes to persuade visitors to “Be More Human” across a series of interactive tools beginning with a Human Score multiple-choice quiz. Clicking through each question reveals an overall grade and personality assessment, giving the content a

genuine personal twist. A Gray Matters section illustrates the impact of fitness on the brain, before Break Your Selfie harnesses Instagram to share shots of physically exhausted fans. The result is a highly polished, slick site design that adopts an intriguing philosophical approach to conveying the brand message.

“The Be More Human Experience is an online destination launched to dimensionalise our new campaign for Reebok,” reveals Michael Sison, design lead at Venables Bell & Partners. “Revealed days before Superbowl XLIX along with a coinciding TV spot, the site is an exploration on the correlation between fitness and humanity through the use of video, interactive tests, 3D visualisers and social media. Designed as an adaptive website, Be More Human works great on just about any device, from desktops to smartphones.”

[The Be More Human] site is an exploration on the correlation between fitness and humanity



abcABC

1234567890

<Above>
• Reefont by House Industries in Bold and Regular appears to provide that rounded Reebok look

abcABC

1234567890

<Above>
• Q Stencil by Jonas Hecksher and the Playtype foundry styles the distinctive stencil headings



Create animation elements that slide into view on scroll

An increasingly popular effect observed on Reebok's Be More Human site is achievable via the use of a jQuery plugin and simple CSS animation

01 Identify the effect

Something you'll notice on Reebok's Be More Human site is a popular trick for pushing page content into view as you scroll. On a page such as the one for the Gray Matters section (fitness.reebok.co.uk/be-more-human/#/page/gray-matters), you'll see the effect put to good effect. For emulating the UI behaviour of mobile apps, we'll look at a method for achieving a similar effect with jQuery and CSS.



02 jQuery Visible plugin

In this instance the effect almost appends content on first scroll only, as if you are 'building' the page dynamically with fresh content. One of the first problems to overcome here is whether your statically defined HTML elements are visible within the browser's viewport. A clever little jQuery plugin called jQuery Visible by Sam Sehner is of real use here and available to download via GitHub at github.com/customd/jquery-visible.

```
001 <script src="http://code.jquery.com/jquery-1.11.2.min.js"></script>
002 <script src="jquery.visible.js"></script>
```



03 Detect element visibility

With this plugin downloaded and attached to your jQuery-equipped page, call a single line of code to ascertain if a page element is visible to the user inside the current viewport. By passing various optional selectors and parameters, the plugin can detect whole

or partial element visibility within the viewport as well as filtering element visibility as defined within CSS:

```
001 // Returns true if whole element is visible:
002 $('#element').visible();
003 //Returns true if just part of the element is visible:
004 $('#element').visible(true);
005 //Filters CSS visibility state:
006 $('#element:visible').visible();
007 //Filters CSS visibility and checks viewport:
008 $('#element:visible').visible(false, true);
```

04 Elements already visible?

The solution we'll examine derives from a Chris Coyier (css-tricks.com) method for toggling CSS classes on a series of identical, relatively positioned elements populating a page. By calling the jQuery Visible plugin to check the initial viewport visibility of each element pertaining to a parent class (.content). We can attach a class to override any slideIn animation we'll add to those made visible on scroll.

```
001 CSS:
002 .inView {
003   opacity: 1;
004   -webkit-transform: translateY(0);
005   -ms-transform: translateY(0);
006   transform: translateY(0);
007   -webkit-animation: none;
008   animation: none;
009 }
010 JAVASCRIPT:
011 $(".content").each(function(index, element) {
012   if ($(element).visible(true, 'vertical')) {
013     $(element).addClass("inView");
014   }
015 });
016
```

05 Check visibility on scroll

The second piece of JavaScript is triggered on window scroll. It grabs the current vertical scroll bar position, and compares to a previous value so we only perform the effect below the viewport and not above. If true, check the latest newly visible elements, attaching a slideIn class that contains our desired animations:

```
001 JAVASCRIPT:
002 var prevScrollTop = 0;
003 $(window).scroll(function(event) {
004   var scrollTop = $(this).scrollTop();
005   if (scrollTop > prevScrollTop){
006     $(".content").each(function(index, element) {
007       if ($(element).visible(true, 'vertical')) {
008         $(element).addClass("slideIn");
009       }
010     });
011     prevScrollTop = scrollTop;
012   }
013 });
```

06 The CSS animation classes

We want the relatively positioned elements to slide up into view so a translateY(0) for vertical motion is desired by however many pixels. Duration and timing are again variable but a 'forwards' animation-fill-mode is critical to maintaining the new styles upon animation. By also setting opacity first to zero, then one (full) in our @keyframes class, we can have it gradually become opaque as it returns to a defined position on the y axis:

```
001 CSS:
002 .slideIn {
003   opacity: 0;
004   -webkit-transform: translateY(200px);
005   -ms-transform: translateY(200px);
006   transform: translateY(200px);
007   -webkit-animation-name: slideIn;
008   animation-name: slideIn;
009   -webkit-animation-duration: 1.5s;
010   animation-duration: 1.5s;
011   -webkit-animation-fill-mode: forwards;
012   animation-fill-mode: forwards;
013 }
014 /*.inView class here or below next class!*/
015 @keyframes slideIn {
016   to {
017     opacity: 1;
018     -webkit-transform: translateY(0);
019     -ms-transform: translateY(0);
020     transform: translateY(0);
021   }
022 }
```




nativespace

ONLY
£14.99
PER MONTH

NEW
FOR WEB DEVELOPERS

GET READY FOR BLAST OFF...

FOR THE FIRST TIME EVER!



SECURE CLOUDHOSTING
for WEB DEVELOPERS



ENTERPRISE GRADE CLOUD AT
SHARED HOSTING PRICES

'MISSION CRITICAL' hosting packages



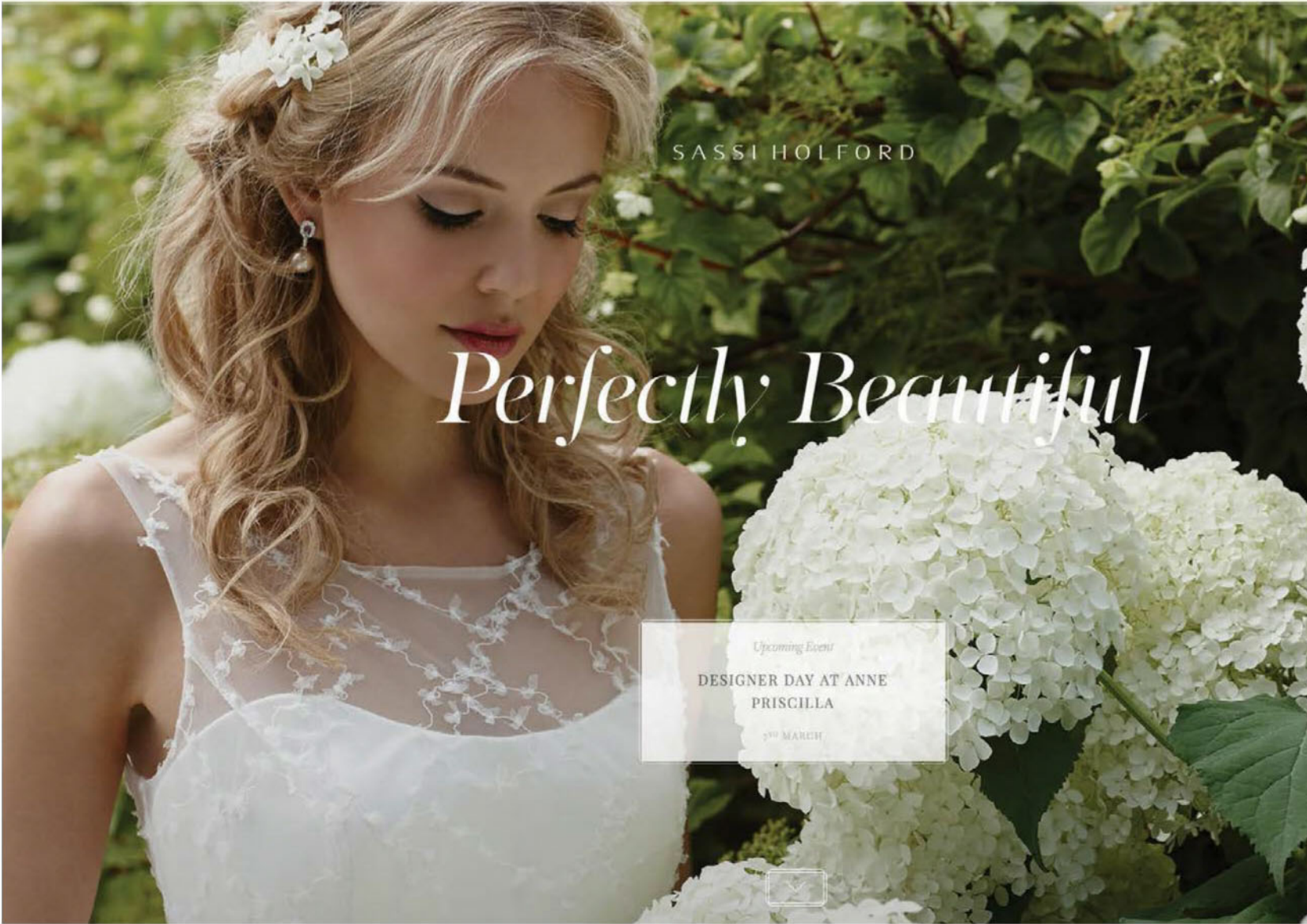
FULLY SCALABLE:
Grows with your client base



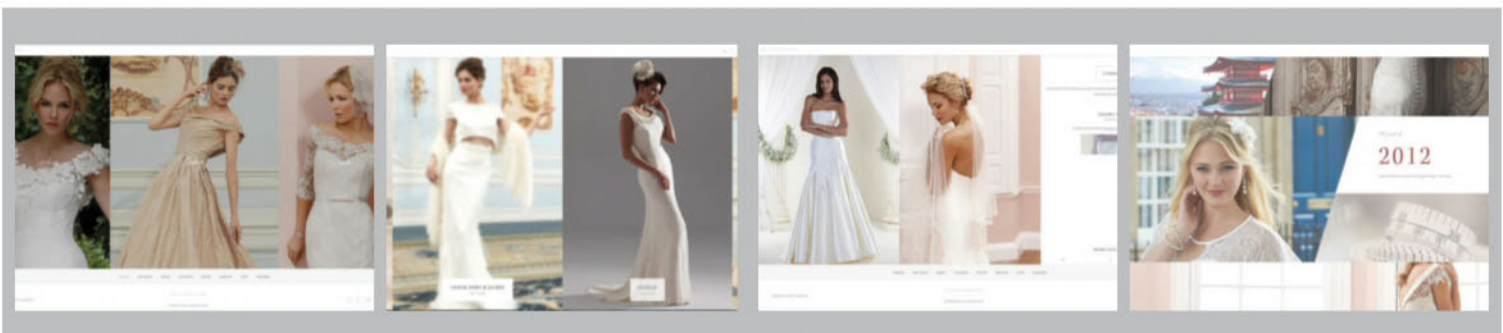
HOST UNLIMITED DOMAINS

LICENCED WITH CPANEL ... FULLY MANAGED ... ULTRA HIGH SPEC ... ULTRA LOW PRICE

LORD



<Above>
 • Sassi Holford is an English-based designer and her website proudly exhibits her handmade, contemporary bridal gowns



<Above>
 • Section categories for the key dress collections are navigated with rollover photographic panels and subtle motion

<Above>
 • Scrolling carousels showcase available gowns, using blurring effects to shift focus to the selected design

<Above>
 • Beautiful photographic assets tell a much bigger story than the economic use of text employed throughout

<Above>
 • Parallax scrolling, transitions and SVG-based animations provide the eye-catching content a sense of engaging depth



Sassi Holford

sassiholford.com

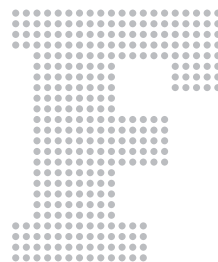
Development technologies HTML5 video, AJAX, SVG animation



Designer **Green Chameleon**

craftedbygc.com

This beautiful new website happily marries the romance of high-end bridal fashion with cutting-edge front-end design



or any blushing bride, looking divine on her big day isn't optional. The Sassi Holford site is all about that excitement around finding that perfect wedding dress, and that sense of occasion from perusing a showcase for handmade gowns. What follows is a heavenly coupling of brochure photography and sumptuous page transitions. Tasteful typography and an editorial sensibility for negative space and stunning oversized images, immediately sets the tone for an immersive, personal experience.

The designers at Bristol agency Green Chameleon, already a Lightbox favourite, clearly worked closely with the client to

understand the brand. By first gaining a sense of Sassi Holford's values, processes and target audience, they could deliver a site that emulated the "flawless" reality of choosing a Sassi dress. "Our aim was to replicate this quality throughout the website, paying close attention to the fine details and delivering a platform that showcases the dresses in as much detail as possible, providing the closest look a bride can get without actually trying the dress on," explains Green Chameleon's design director Nathan Riley. "To achieve this we kept the design of the site minimal, using a crisp white/grey colour scheme and sleek UI elements, letting their photography do the talking. Our focus from a technical standpoint was to really push the front-end development, utilising AJAX page transitions, SVG animations, HTML5 video and a number of subtle interactions site-wide."

abcABC

1234567890

<Above>
• Freight Display Pro by Phil's Fonts and found on TypeKit is used in Italic for subheadings

abcABC

1234567890

<Above>
• Playfair Display by Claus Eggers Sørensen available via Google Fonts adds class to headings

Our aim was to replicate this quality throughout the website, paying close attention to the fine details

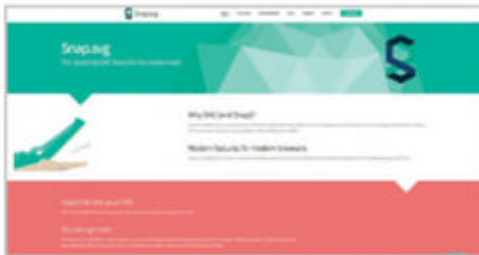


Animate a loading screen with SMIL, Snap.svg and Modernizr

Paul Thomas, senior front-end developer at Green Chameleon, describes an SVG animation technique with fallbacks as used on Sassiholford.com

01 Introduce the effect

We are going to examine the loading effect as used on sassiholford.com. This will be achieved using any SMIL-supported browsers other than IE, for which we'll create a fallback using Snap.svg (snapsvg.io). The live demo and full code can be found at bit.ly/19JB9ZL.



02 So why use SMIL?

The benefit of using SMIL (Synchronized Multimedia Integration Language) is that you don't need to download heavyweight JavaScript files. You can use them in symbols (eg icons) although this only seems to be supported in Chrome. Please note, for the sake of brevity CSS vendor prefixes have been left out.



03 Create the SVG

Create the SVG in your editor of choice, in this case we've used Inkscape (inkscape.org). Convert the 'type' into a 'path' without embedding the font and combine the paths so that a single path is a whole word, and not a single letter. Embed as a plain SVG into your HTML so we can manipulate it with CSS and JavaScript:

```
001 <div class="flex-container">
002 <svg viewBox="0 0 400 55"
003 id="loading">
004 <path id="loading-path" d="m ..."/>
005 </svg>
006 </div>
```

04 Setting the stage

We've embedded the SVG and added a simple layout. There's a background image with 'background-



size: cover;' and the loader is placed in the centre of the stage using the flex-container <div>. The SVG is set to take up 20% of the screen width, but this is variable:

```
001 body {
002 position: relative;
003 background-size: cover;
004 background-position: 50% 0;
005 background-repeat: no-repeat;
006 background-image: url(bg.jpg);
007 }
008 .flex-container{
009 width: 100%;
010 height: 100vh;
011 display: flex;
012 justify-content: center;
013 align-items: center;
014 background: rgba(255, 0.3);
015 }
016 #loading{
017 display: block;
018 width: 20%;
019 height: auto;
020 }
```

05 Create the Snap.svg version

We are creating a gradient that is a single colour (R189, G189, B189) but has three stops with opacity (0, 1, 0) respectively. Make the gradient move by animating the direction (x1, x2) from between: x1=-1, x2=0 to x1=1, x2=2. This starts the gradient off the path to the left, and moves it through off the path to the right, so it disappears, while maintaining its proportions:

```
001 var loading = Snap.
002 select('#loading');
003 var gradient = loading.
004 gradient('l(0, 0, 1, 0)rgba
005 (189,189,189,0):0%-
006 rgba(189,189,189,1):50%-
007 rgba(189,189,189,0):100%');
008 loading.attr({fill: gradient });
009 var animGrad = function(){
```

```
010 gradient.attr({ x1: -1, y1: 0, x2:
011 0, y2: 0 });
012 gradient.animate({ x1: 1, y1: 0,
013 x2: 2, y2: 0 }, 1500,
014 mina.linear, animGrad);
015 };
016 animGrad();
```

06 Create the SMIL version

Make a static gradient for animating in SMIL. This is the same gradient we applied in the Snap.svg version, but using <defs> directly in the SVG itself. Add the SMIL animation by animating the same x1 and x2 values directly in the SVG:

```
001 <defs>
002 <linearGradient id="gradient"
003 x1="0" y1="0" x2="1" y2="0">
004 <stop offset="0%" style="stop-
005 color:rgb(255,255,255); stop-opacity:0"
006 />
007 <stop offset="50%" style="stop-
008 color:rgb(255,255,255); stop-opacity:1"
009 />
010 <stop offset="100%" style="stop-
011 color:rgb(255,255,255); stop-opacity:0"
012 />
013 </linearGradient>
014 <animate attributeName="x1"
015 dur="1500ms" values="-1; 1;"
016 repeatCount="indefinite" />
017 <animate attributeName="x2"
018 dur="1500ms" values="0; 2;"
019 repeatCount="indefinite" />
020 </defs>
```

07 Toggle JavaScript or SMIL

Don't run the JavaScript if you can use SMIL - use Modernizr to detect for SMIL support and if it's not found, run JavaScript. Use a loader like Require.js to download the Snap.svg library if SMIL isn't supported, saving unnecessary download where SMIL's supported:

```
001 if(! Modernizr.smil ){
002 /* all JavaScript from previous
003 steps goes here */
004 } else {
005 /* fill the svg path with our SMIL
006 animated gradient */
007 $('#loading').attr('fill',
008 'url(#gradient)');
009 }
```




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](https://twitter.com/blacknight)

**GET RESULTS
THROUGH
IMPROVED
KNOWLEDGE**

Visit us at DjangoCon 2015
in Cardiff to learn more about
our unrivalled PostgreSQL
training and support packages.

Our courses are rated "excellent" by
100% of our clients and are highly
relevant to Django developers and
users. Book your place and gain access
to expert knowledge of the core code.

+44 (0)870 766 7756
2ndquadrant.com/knowledge

2ndQuadrant 
Professional PostgreSQL

there's no fancy advert here
just the details that you need to read

we are a **trusted** provider, **reliable** hosting for over 12 years
we **care** about your website, **daily** backups that are held for 7 days
we are **real** people providing real support, **caring** for our customers
we make things easier, **FREE** web applications with point and click installation

for **feature-packed** web hosting

<http://ukwebsolutionsdirect.co.uk>

WDM readers get 15% off hosting services with coupon WDM15



THE NEW JAVASCRIPT STANDARD

Upgrade your applications with ECMAScript 6 and
boost your scripts with the newest features

Practical ES6

Check out the
jspm tutorial on
page 76 to see
some of the new
features in use.

ES6 AND HOW IT IMPACTS JAVASCRIPT

JavaScript is available in a variety of forms today like that of web browsers, native applications, server-side scripting and even more exotic environments. As a language, it is steered by a standard called ECMAScript, which sets the core syntax and features of JavaScript. As each version of the ECMAScript standard is ratified, JavaScript engines move to support the changes detailed within it. This provides JavaScript with a general compatibility between implementations of the language. Code written to work in web browsers, that aren't tied to anything browser specific (like the DOM), should work in other JavaScript engines like Node.js.

ECMAScript 6 (referred to as ES6) is the next version of this standard for JavaScript and is nearing its final release, meaning that the features are set and that implementations are fully detailed and unlikely to change in any significant way. Browsers, scripting engines and other applications have already started adding support for ES6 with Firefox and Chrome leading the way. Once it's been officially finalised, expect full support of the standard to be taken up quickly by the major applications. ES6 will introduce new syntax and features to JavaScript, designed to make life easier for developers and solve common programming problems. The goals state that it intends to make the language better for writing complex applications, libraries and code generators so they've included a module system that is reuseable and can be used to improve code organisation. Class support is now built in with its own syntax, and methods have been added to help manage asynchronous code, such as promises and generators.

At lower levels there are new variable types to provide alternatives to objects and arrays for your data, known as 'sets' and 'maps'. Building strings from several variables has also been improved through a new template string syntax, which can also be used to easily create multiline strings without being split. Defining functions has also received attention through a new 'fat arrow' syntax, which can help with the often confused nature of the 'this' scope in relation to the function being called. Recursive functions can take advantage of performance improvements by returning at their end. Block-scoped variables, constants, default parameters and much more make this quite the upgrade for JavaScript.

What does this all mean for developers? On offer is better code organisation, plenty of new shortcuts for typical tasks, standard formats for sharing libraries and much more. There's also nothing holding you back from using the major features of ES6 now. Applications are available that will take your ES6-based code and compile it into ES5, making your code compatible with a wide range of JavaScript engines that are in use today.



Matt Gifford – consultant developer, Monkeh Works Ltd (monkehworks.com)

“ES6 is going to empower JavaScript developers, giving us the means and flexibility to write modular, well-organised code that is easily scalable, testable and understandable. Well-structured code will improve application architecture and will have a direct positive impact on performance.”

SUPPORT

Some ES6 features are commonplace, others are unsupported, but there are workarounds available

Browsers

Firefox and Chrome have made significant progress with support for these new features of JavaScript. They now support over 40 per cent of the new standard (at the time of writing). Especially in the area of new variable types, improve function syntax and promise support. Internet Explorer support is lower with IE11 only covering 11 per cent, but the next release will bring in wider coverage.

Scripting engines

The Node.js engine covers a quarter of the ES6 standard in its recent stable release. However, a recent fork of the project, named io.js, has pushed

this to over 40 per cent. There are discussions going on to possibly bring those changes back into Node.js or the two projects to work together, so expect ES6 support to grow further within these engines.

Transpilers

Due to lack of ES6 support or missing features in certain JavaScript environments, applications have been created that take your ES6-based code and compile it into ES5-compatible code. Through polyfills, these transpilers use alternative ways to replicate ES6 behaviour. Your original source code will always be available as you wrote it, but you'll have a new copy to distribute to ES5 environments. Babel (babeljs.io), at the time of writing, can support 76 per cent of the ES6 standard within ES5 engines. As the ES7 standard is being formed, effort has gone into Babel to provide early support of some features.

A CLOSER LOOK AT THE CODE

2. These setters enable a tighter control of what data may be set and even perform actions based on it, while the getters can provide the instance data in several formats.

3. Static methods aren't present on an instance of the class but rather the class itself. This can be used for any unique logic that hasn't been tied to a single instance.

4. Creating a new instance of a class is simple enough through the use of the 'new' keyword. Data can also be provided to the constructor if it is required here.

1. The constructor is executed when a new instance is created from the class, providing the ability to set anything up that we might need within the instance.



THE MAJOR FEATURES OF VERSION 6 CLASSES

New to JavaScript, classes provide a shortcut to OO-type code organisation

With ES6, JavaScript will finally have classes. These aren't new in the programming world and it has also been possible to replicate classes in JavaScript already, but having them built into the language by default will encourage a standard way of using object-orientated code, making it easier for you to write large applications and also understand other people's code.

A class at its core is used to create a template for an object, which can contain its own data and methods. ES6 will support several concepts that work with classes to aid in organisation of your code and provide functionality. Inheritance is included, so you can base a class off another existing one and make use of it, within your new class via the 'super' variable. Instance

methods are standard, and they're available when you create an object from the class template, but static methods can also be created and used on the class template itself. A constructor method is provided, enabling you to perform tasks on an instance of your class template, whenever one is created. Finally, there are getters and setters and these will provide the ability to capture, with a class method, when data is set or retrieved against your instance.

New function syntax

The 'this' scope can sometimes cause confusion within JavaScript. With the established function syntax, the 'this' scope can be different depending on where the function is being executed. If you wanted to retain a function's 'this' scope, you would have to bind a value to it with '.bind(this)'. The new syntax will create a function bound to where it was created, with the original 'function' method remaining available.

```
this.sides = 6;
var rollNumbers = (dice) => {
  var rolls = [];
  for (let i = 0; i < dice; i++) {
    rolls.push(Math.floor(Math.random() * (max + 1 - min)) + min);
  }
  return rolls;
};
```

Template Strings

Normally, to construct a string in JavaScript based upon set text and variable values, you either concatenate them using the plus operator (+) or join them together through an array. ES6 provides a new syntax known as template strings. These will enable you to create a string with variables inside it, avoiding the need of having to stitch it together. They also provide some other useful features, such as being able to create

“With the established function syntax, the 'this' scope can be different depending on where the function is being executed”

Let and const

Declaring variables is now more flexible. Constants are now available by prefixing with 'const', enabling you to declare and set a constant value to an identifier. If any attempt to change the value is made, an error will occur, safeguarding the data. You can now also create a variable that will only exist within the block, statement or expression it is declared within through use of the 'let' keyword instead of the normal 'var'. This helps with creating safer, self-contained code and avoids overwriting variables used within the same function. With considered use this could also result in lower memory usage, as 'let' variables could be freed sooner.

```
const apiKey = process.env.API_KEY;
apiKey = 'something'; // This would throw an error
for (let i = 0; i < 10; i++) {console.
log(i);}
// The i variable only exists within the
"for" loop
```

Modules

One major missing feature from previous versions of JavaScript has been a built-in module system. As ES6 intends to provide a better language for building complex applications and libraries, it makes a module system an important piece of the puzzle. The JavaScript community has created it's own alternatives such as Asynchronous Module Definition (AMD) and CommonJS (CJS) but with ES6, JavaScript will finally have one built in.

With the module system, anything within a file won't be available outside of it, unless you export it. Through use of the 'export' keyword, you can specify variables (data, functions, classes) that you wish to make use of elsewhere. Then, from another file, you can import some or all of those exposed items. This forms the basis of organising your application code, splitting it apart and making it more reusable.

```
// mag.js
const title = 'Web Designer';
export default function() {
return {
title: title,
data: getMagData()
};
};
// index.js
import Mag from 'mag';
var data = Mag();
```

Function parameter additions

Functions having parameters is nothing new within JavaScript, but ES6 has added support for default values and gathering some arguments as an array. A function in JavaScript can have numerous parameters available, but when calling the function you aren't required to supply every single one. Those arguments

MAKING A PROMISE AJAX REQUEST

An example of how to use promises to wrap asynchronous requests

In this example promises are primarily used from ES6, with a few other minor features here and there. Targeting the browser with our code, we have XMLHttpRequest available for making AJAX requests for further data. In this example we need to pass an application object a list of countries and languages that are available in JSON-based files.

The files could be requested one at a time, but this would not be as fast as requesting both at once (asynchronously). This would take some effort to manage normally, but with ES6 promises, it becomes a lot simpler. The first and most important step is to wrap a call to the XMLHttpRequest object with a promise. When a new promise is created, it provides two functions to define whether the promise was successful or failed via 'resolve' and 'reject'. Anything passed to 'reject' will cause the promise to fail and fire any function provided to the .catch(fn) method (of the returned promise). If the request worked, we'll attempt to parse the JSON and fulfil the promise with 'resolve', passing it the parsed data object.

Now, any call to our XMLHttpRequest object wrapper (fetch), will return a new promise for the requested URL. This promise will remain unfulfilled (pending) until it either shows an error (rejected) or alternatively, it shows that it has received the response and has parsed correctly. Using the returned promises' .then method with a function will gain access to the parsed data.

To easily make multiple requests at the same time and wait for all of them to return, a further promise can be created. Promise.all will take an array of other promises (and values) and wait for all of them to resolve before it fires anything attached to .then. It can also manage errors across all of those multiple promises.



EXPERT COMMENT



BRIAN LEROUX

Developer at Adobe
bit.ly/1wadOsK
@brianleroux

ECMAScript 6 is the JavaScript we've been waiting for. It reveals the hidden beauty of JS, taking cues from contemporary dynamic languages and adding its own shortcuts paved from common paths tread by the developer community. The best part - we can start using ES6 today in io.js, and with the help of Babel, anywhere that runs ES5 (which is just about everywhere).

a string over multiple lines in your source code, something that isn't normally possible without splitting your string apart and joining it with the plus operator at the end of each line.

```
var luckyNum = Math.round(Math.random() *
100);
var output = 'Today's lucky number will be
${luckyNum}.';
```

THE NEW JAVASCRIPT STANDARD

that aren't supplied would normally remain 'undefined', but now there's the ability to set default values for them. ES6 also enables declaring of some function parameters and then having the rest grouped as an array. This is useful if you want to let your function take a flexible amount of arguments, but have a few initial set ones.

```
function plural(word, amount, suffix='s') {
  return (amount === 0) ? word :
  `${word}${suffix}`;
};
plural('test', 5); // returns 'tests'
function setNames(userId, ...names) {
  // names would equal ['Bob', 'Robert',
  'Bobby']
};
setNames(42, 'Bob', 'Robert', 'Bobby');
```

Promises

Sometimes, in a JavaScript environment, you'll have to do something asynchronously. This could be in a web browser where you need to make an AJAX request, or in Node.js when making a request to a database, or handling web requests. The standard way to handle such

tasks would be to provide a callback function, which would be executed when the task was complete (like data being received from the AJAX request).

Promises are an alternative approach to handling this asynchronous tasks, which can make code much more readable. In comparison, callbacks can sometimes lead to pyramid-type code where callbacks are nested by several levels. Promises work as you might expect. You create a task (usually asynchronous in nature) which returns an unfulfilled promise. When the task completes, the promise is fulfilled and it'll execute anything attached to its '.then()' method. If a task fails with an error, the '.catch()' method will be fired. Promises can be easily passed around, enabling you to check its state and require one promise before others are completed.

```
'g'
function initApp() { `${word}${suffix}`;
return new promise(function(resolve, reject)
{
// App initialization...
if (app.complete) return resolve(app.id);
reject(new Error( 'App failed to
complete'))); });
```

```
}
initApp()
.then(function(appId) {
// Now setup the UI
})
.catch(function(error) {
// Handle the error
// Handle the error
});
```

EXPERT COMMENT



JACK FRANKLIN
Engineer, GoCardless
jackfranklin.co.uk
@Jack_Franklin

I'm personally really excited about ES6, not only for the features it brings but because it's getting such backing from developers already. We're seeing tools spring up to let us experiment and use these new features today in production applications, without having to wait for browser support to catch up.

FROM ES6 TO ES5

Take your ES6 code into ES5 environments

Babel is a JavaScript-based application to take your ES6 code and make it compatible with ES5 JavaScript engines, that are missing some or all of the features introduced by ES6. Grunt can be used to manage this task and hold any configuration it may need. In the screenshot below, the steps are shown for creating a project and then the Gruntfile for executing Babel against the ES6 code.

Once the Gruntfile is ready, ES6-based source code can start to be created within the src directory. When ready, a single Grunt command will use Babel to build the ES5 code. Babel attempts to keep the resulting code as clean as possible, but it will be using workarounds for those missing ES6 features. This includes using polyfills to add support and providing source maps for easier debugging.



We don't keep secrets



Tips & Tricks

Learn the truth about iPhone, iPad, Android, Photoshop and more with the Tips & Tricks series' expert advice and tutorials

BUY YOUR COPY TODAY

Print edition available at www.imagineshop.co.uk

Digital edition available at www.greatdigitalmags.com

Available on the following platforms



 [facebook.com/ImagineBookazines](https://www.facebook.com/ImagineBookazines)

 twitter.com/Books_Imagine



ON A MISSION

UNIT9 reveal how they made a compelling online experience for Kingsman knights with 20th Century Fox

Project **Kingsman:
Become A Knight**

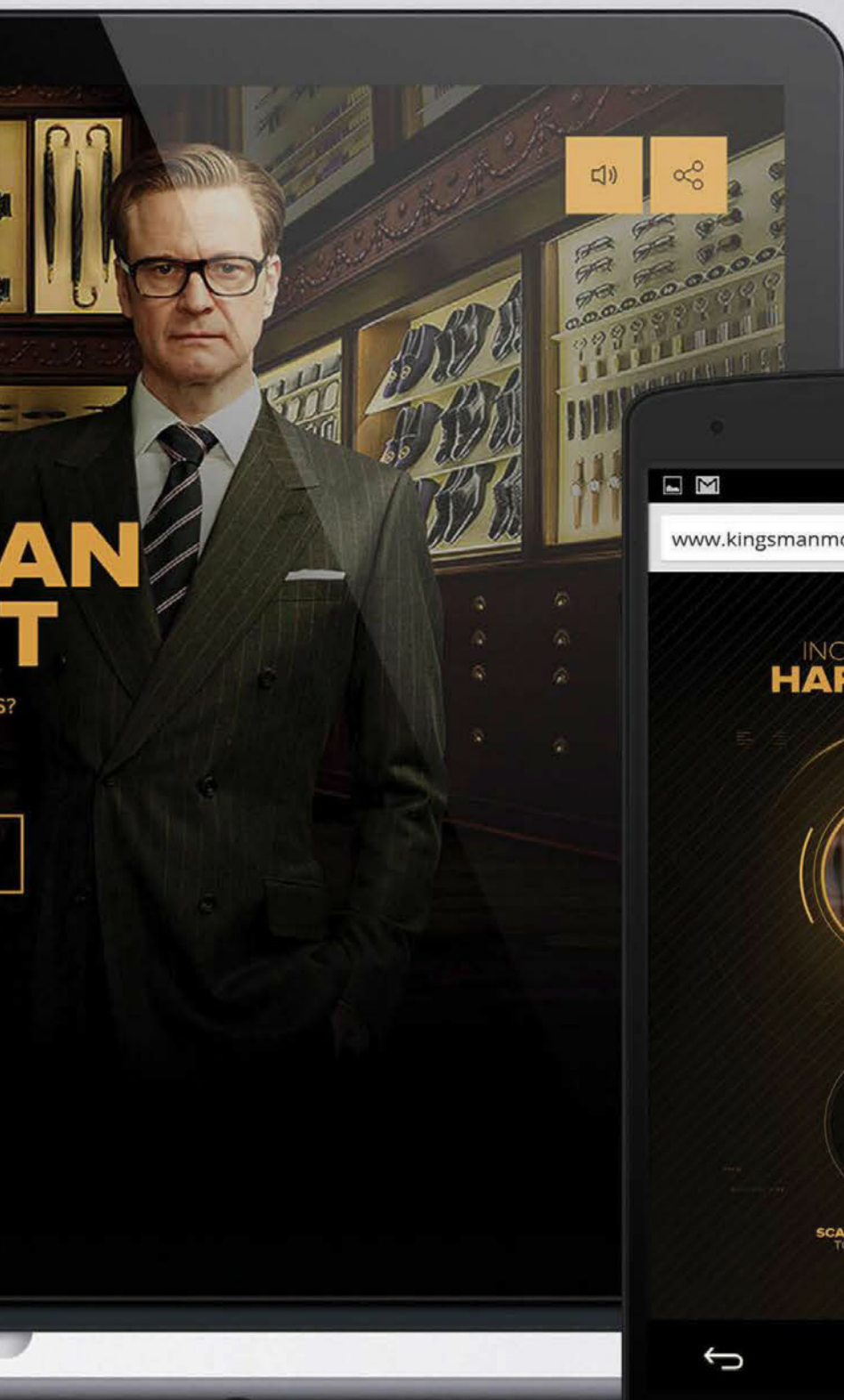
Web www.kingsmanmovie.com

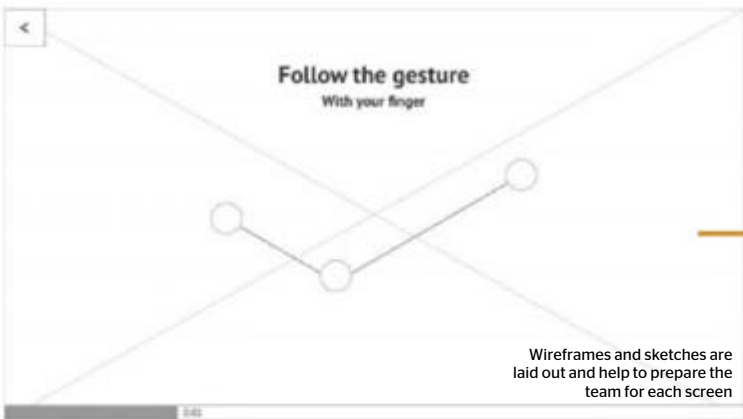
Agency **UNIT9**
Web unit9.com

Duration **2 months and 2 weeks**

People involved **25**

Total hours **2,100 hours**





Wireframes and sketches are laid out and help to prepare the team for each screen





Combining spy missions with the techy gadgets of the film was the perfect digital cocktail

When it comes to spies and the British secret service surely there is nothing that beats James Bond. You may think that, but *Kingsman* does a very good job of making you forget 007, and this is only the film we are talking about. Throw in the digital experience and you have a classic combination that perfectly demonstrates how to create a compelling digital experience.

To bring the whole process to fruition the makers of *Kingsman*, 20th Century Fox, teamed up with UNIT9 director Anrick Bregman and a team of storytellers, designers and developers. The mission: to give people the chance to interact with the *Kingsman*'s set of lovable characters. How does this happen? The experience starts via the interactive site, connecting a smartphone to a

desktop and using it to control the action happening on the larger screen. The two devices are linked and can exchange up to 25 interactions per second. The best parts are the touch and swipe actions, and a virtual compass that controls the outcome of missions. The user gets the opportunity to show off their skills as a potential Kingsman agent. Colin Firth's character Harry Hart needs your help and he wants you to find hidden data and steal intel from the enemy. This is where the experience gets interesting. The first mission, The Rainmaker, takes the user into a fight scene from the film and even lets users fight as Harry Hart. For the second mission it's over to St. Pancras station to grab more important intel and capture real-life images. How did UNIT9 build the mission? The project developers used the Street View Image API while the design team had to retouch each image one by one. A total of 12,000 images had to be tweaked in all.

What would want to make UNIT9 take up such a mammoth task? Art director Craig Parkinson reveals how 20th Century Fox wanted to move forward and engage new audiences, a challenge they couldn't resist: "From the first conception of this project we knew it was going to be a great opportunity to create something special for 20th Century Fox. Building on their traditional advertising methods, Fox was hungry to move forward and partner up with UNIT9 to create an exciting new digital platform for *Kingsman: The Secret Service*."

"We knew that we were aiming for a younger, more engaged generation of film lovers. The movie is quite 'techy', with a focus on gadgets and cool weaponry, wrapped in an exciting espionage action-packed setting. So we wanted to somehow combine the techy coolness of being a spy with a digital experience that would resonate with a young and excited audience."

"The film's two pillars, fashion and gadgets, became our prime focus. The film stars the young and talented actor Taron Egerton as the character Eggsy. He becomes the unlikely candidate to join Kingsman under the watchful eye of Harry Hart, played by Colin Firth. Eggsy has to conquer a series of nearly impossible tasks in order to join the secret service. We knew that the film depicts Kingsman as being an extremely tough job to get, in fact the movie sells the test as 'the most dangerous interview in the world' and we thought it would be fun to test people in the same way. We had a great kick-off session where we came up with lots of ideas for 'extreme' job interview tasks that each user would need to pass to become a Kingsman knight."

"At UNIT9 we have a very fast pace of working and a strong work ethic, which meant that we were soon creating visuals to express these concepts. We spent roughly two weeks refining these ideas and working closely with Fox to create the initial designs."

Collaboration, conversations and constant communication are what makes a project work. With these key components in place UNIT9 had a clear vision as executive producer Alessandro Pula explains: "It is very important to establish a bond with the client from the very

FASHION, STYLE AND FIGHTING

Idea and concepts are the first step on the road to creating a masterpiece. UNIT9 didn't have to start from the ground up, they already had a story to work with. Creative technologist Dirk Van Ginkel and creative strategist/UX designer Laura Cortes reveal how missions redefined the experience: "*Kingsman: The Secret Service* tells the story of a supersecret spy organisation that recruits young people into the agency's ultracompetitive training program. We decided to split the experience into missions, each with its own objective, which is the way they're structured in the film as well. These missions were inspired by elements that embody the film: fashion and style, fighting, and time-sensitive tasks. Using this concept as our starting point we decided to create a story that felt personal and always connected to the user. With the film being so focused on cool, hidden gadgets we immediately realised how fantastic it could be to transform the audience's phone into an actual *Kingsman* gadget and create a scenario where the user feels part of an international top secret service. This became our creative vision for the project and helped guide the direction we wanted to take."

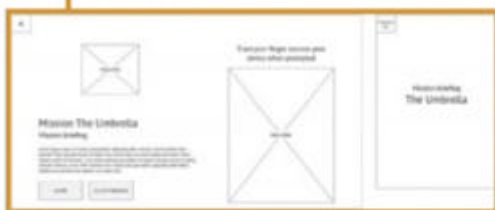
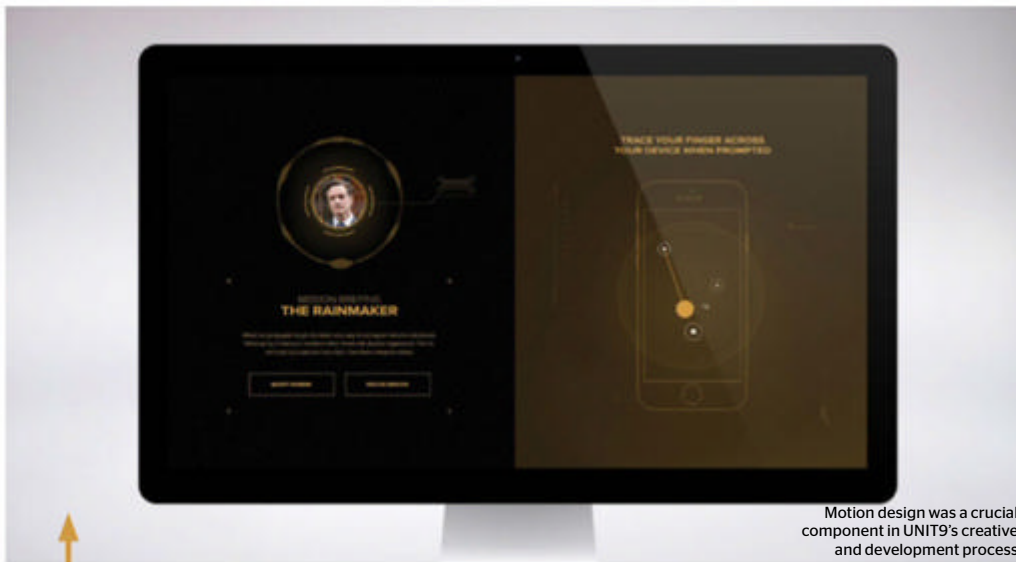
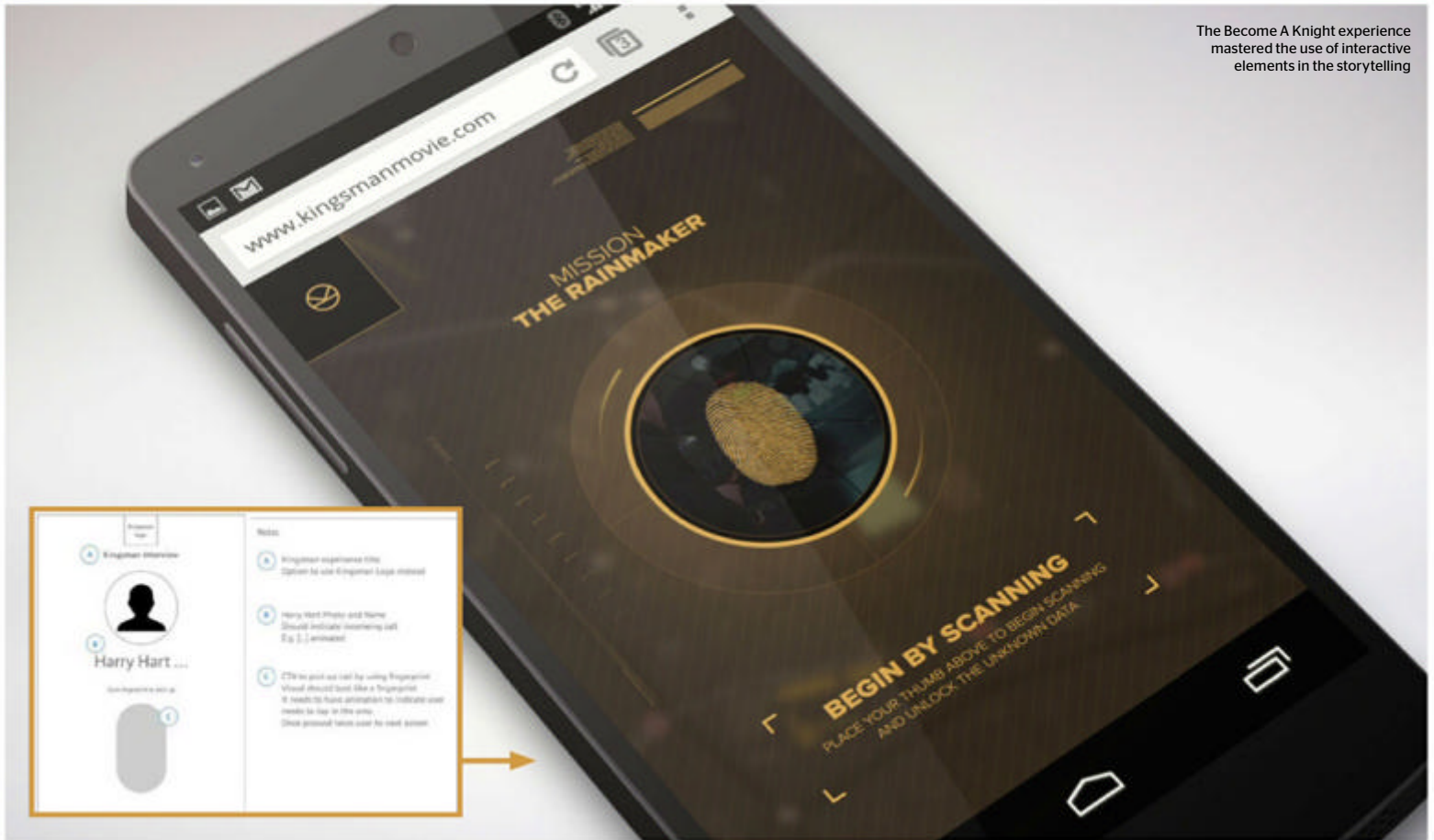
"In mission one, The Rainmaker, we emphasised the fighting aspect by using footage of the film where Harry Hart teaches some manners to a pub full of hooligans. By correctly swiping shapes that correlate to the fighting move, users help unfold the scene and feel close to the actual action. Mission two enabled us to deepen the use of gadgets and use the second screen to find Valentine, the bad guy, in King's Cross."

first call, because that's where you start building the partnership. There are a lot of things that are happening at the same time in this phase. After we receive a brief, our scope is to understand exactly what the client is looking for to help them build something relevant, so the conversation focuses mostly on the storytelling, how the user should interact with it and what kind of technologies should be used to make that happen. That conversation needs to take into account the client's KPIs, timing and budget.

"Once the objectives are clear and we are on the same page with the client, the next step is for us to provide an initial creative and technical direction based on those conversations. The creatives start crafting the story and the user journey, always pushing for something new. The tech team is always involved in this phase as we need to make sure that what we are proposing is feasible within the time frame we've been given to build the app. The design team explores a direction based on the director's brief to

“We spent roughly two weeks refining these ideas and working closely with Fox to create the initial designs”

The Become A Knight experience mastered the use of interactive elements in the storytelling



come up with the look and feel of the experience, which in this case needs to be in line with the look and feel of the film. All these are conversations that see the client involved, at any level, all the way through."

"We had a great deal of fun with the graphical representation of this project" enthuses art director Craig Parkinson, "it was an absolute pleasure to work on the

creative... The main focus being on the cool techy gadgets gave us a great scope to explore a really interesting and tech-inspired UI. We pushed into the realms of what an actual spy intelligence interface could look like. We used the gold-colour pallet inspired from the film's overall visual tone and created a consistent style that lead into project's art direction. This resulted in a very cool, sleek and sophisticated look we feel a Kingsman would be proud of.

"One particular concept I am personally very proud of, and which we thought was interesting, was the idea of installing spy software onto your device. Initially we created a lot of different designs focusing on mobile, further exploring this concept of downloading and installing software. However due to some technical limitations of what could be achieved on mobile in a very short delivery time, we had to shift our focus away from mobile to predominantly concentrate on the desktop experience. This, however, was good news as it meant we could achieve a great deal more in terms of tech and introduce motion design. Right from the start we worked very closely with our own motion design team here at UNIT9, who helped us bring this techy, spy-inspired design to life. Using motion during this early creation process helped us to think about how the subtle transitions of the user interface could be pushed in interesting ways. This process of creative and motion working together gave us the freedom to prototype exciting transitions, which in turn gave the developers a very clear idea of what we wanted to achieve and the standard of design we were aiming for.

"Overall, it was a great process. We set the bar really high with what we wanted to achieve, so the final designs

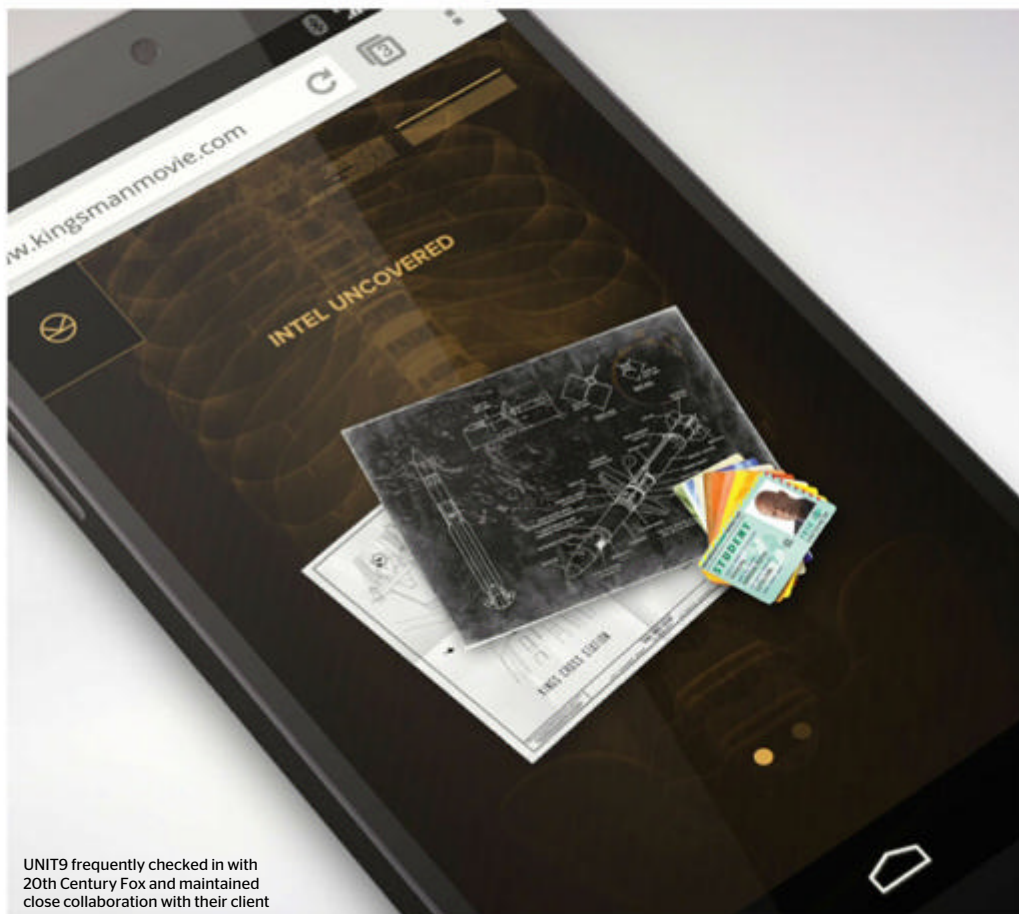
“The final result uses an HTML Canvas with predefined paths for each of the cars, so it is light... and also quite dynamic”

really pushed technology. There are always technical limitations to some of the things we want to create, especially with such an ambitious project. The key to delivery in our eyes was using motion to explain how we wanted to bring this project to life. We found that just by explaining the design concepts verbally, our ideas could be lost in translation. It was sometimes difficult for the developers to understand what we were wanting to achieve. In our minds we had a clear idea of all the cool things we wanted to do, but as creatives, we didn't necessarily focus on all of the technical restrictions. Showing our ideas with full animation as opposed to flat comp mockups made it much easier to understand what we wanted to create and easier for the development team to then reproduce."

Behind the scenes it is the back-end technologies where the challenges lie for developers. But it wasn't the more traditional code bases that were the test for the UNIT9 team, the biggest hurdle was the browser. "Everything else happens in the browser, so that's where we found most of the challenges," tech lead Bruno Imbrizi revealed. "The development process was very concentrated on the client side. There is not much happening on the server side, except for some logic to manage the socket connection between devices when users choose to play with both desktop and mobile phone for example. We used almost everything modern browsers have to offer.

"Usually there is more than one way to do something and we had to experiment with different methods to see what would work better. An example is the main screen which has a map of London in the background. The final result uses an HTML Canvas with predefined paths for each of the cars, so it is light in terms of assets and also quite dynamic since paths are selected randomly when the page is loaded. But... we first tried a Google Maps element with a custom theme and later a big video in which the cars would go around in a five-second loop. Iterations like this happened throughout the project. Perhaps the most challenging part was the mission 'Steal The Intel' which takes the user to King's Cross station in London. We used the Google Street View API, which provided a lot of the functionality required in the mission, but we also had to write a lot of code. To create the custom panoramas we used more than 12,000 images."

Project delivery is very dependent on the client. It can be a simple pass over or a more involved process. The



UNIT9 frequently checked in with 20th Century Fox and maintained close collaboration with their client

UNIT9 team explains how they approached this task with 20th Century Fox: "The moment we deliver a project is a very delicate one as there are many moving parts. Once the site is built and all the pieces are in place, we need to make sure that it works and looks as expected at every level, so we go in for a final review with the client and if everything is okay, we get the green light to start the testing and the quality assurance process that on a project like this can take up to two weeks as the app needs to be tested on a variety of devices, computers, mobile phones, iPads.

"Furthermore, on Kingsman the mobile phone and the computer also had to work together for some of the challenges. This is a process where also the client is involved usually, which is great because the more people testing the app, the more thorough the fixing of the issues and the stability of the app. When we are done

with the testing phase and there are no more issues to solve, we go through another round of approvals and then we are ready to push the project live. We do our best to make sure the site is perfect when we release it into the wild, but the reality of coding is that anything can happen. Therefore we keep a very close eye on it whilst it's live, ready to jump on anything that could come up and resolve it, whether it's a bug or a functional or design last minute update."

THE BIG QUESTION...

Which character in Kingsman would you be and why?



"I would have to choose Eggsy. Simply because he originated from a bit of a rough background like myself. And when I say rough I simply mean Manchester. Or maybe the pug, JB? He was by far the star of the show!"

CRAIG PARKINSON,
ART DIRECTOR



"Gazelle! Who wouldn't want to be able to slice someone's throat with their legs?"

LAURA CORTES,
CREATIVE STRATEGIST/
UX DESIGNER

ONE HUNDRED PER CENT COMMITMENT

The client relationship varies depending on who an agency is in collaboration with. UNIT9 was in with the big boys in the shape of 20th Century Fox, and there were no intermediaries to take the responsibility. It was all down to them.

"Working on a small project or a big one requires the same amount of focus and dedication," states executive producer Alessandro Pula. "What changes though is the amount of people involved both on our side and the client side to take care of and therefore the amount of risks involved. Normally as a production company we work on projects driven by

the advertising agency, which takes care of all the client relationship at a higher level, filtering the conversation for us.

"In a project like this, we worked directly with the client, which comes with a variety of challenges, because at that point you are responsible for the project and the relationship one hundred per cent. It is also good though because it puts us in a position where we can start building the project from scratch and be free to explore the possibilities that the project can offer, both from a storytelling and technical point of view."



Digital experiences, analog world

With a desire to enhance and expand the experiences of everyone that connects with their work, AnalogFolk have continued to develop their world-class skills with innovative digital design



AnalogFolk

who AnalogFolk
what AnalogFolk are an independent global creative agency with offices in London, Sydney and New York
where 20 Rosebery Avenue, London, EC1R 4SX
web analogfolk.com

Key clients

AkzoNobel
Canon
Lucozade Ribena
Suntory
Pernod Ricard
Sainsbury's

Founded in 2008, AnalogFolk live by one driving force: to create value in peoples' lives via the digital work they build for their clients. What is clear is that AnalogFolk certainly innovate, but at all times this is influenced by their desire to help the world's leading brands make lasting and meaningful connections with their customers.

"The agency was born out of a desire to do work for brands that would add value to people's lives in some way," explained Matt Dyke cofounder and chief strategy officer. "Bill (Brock - cofounder and chief executive officer) and I met whilst working at the digital agency Tribal DDB. Incredibly, it was during our very first meeting together that we discussed the need for a new model of agency, which we then left to start up three years later."

Matt continued: "My background was in advertising account planning and before that brand research. Bill had a background in software development before moving into digital marketing. We think that it was the combination of this experience that really shaped the unique capability of AnalogFolk versus other agencies from one of these worlds or the other."

Naming a new agency is always an important aspect of any business to get right. Matt explained their approach to this vital decision: "Everybody assumes the agency name is an ironic take on the fact that our creative work is digitally led. However, it is actually inspired by our belief that the human world we live in is analogue and that digital technology should be used to improve it for us, not replace it. It won't surprise you to know that AnalogFolk was immediately available as a domain name. When we originally sat down to brainstorm names for the agency our only criteria was that it



New technologies, like the Oculus Rift, are being embraced by AnalogFolk all the time

should begin with 'A' so that we would appear first in any industry listings. We've no idea if that has helped us over the years!"

As a calling card an agency's website is a vital component of its marketing. Fame Razak, chief technology officer outlines the approach AnalogFolk has taken: "First impressions are everything, our website will be the first port of call for many potential clients and employees to AnalogFolk. Whether they read about a new hire or a pitch win in the trade press or heard about us through colleagues, we'll be judged on our ability to create a user-friendly and well-designed site with well-written and relevant content.

"Simply, if we can't do it for ourselves, then we certainly can't do it for others. It's only been a year since it was launched but we are already evolving our website. There's no chance of it getting stagnant, with new content in-line with the launch of a new piece of tech, an industry event or a blog [about] our latest hack on **make.analogfolk.com.**"

How an agency attracts its clients can be manifold. Often a mix of strategies delivers the best results, as Bill Brock explained: "I'd love to say that 100 per cent of our new business opportunities are proactive approaches that come without a pitch, but that's not how it works in our business in 2015.

"Whilst we get our fair share of these straight appointments by reputation, we also have a comprehensive business strategy to target and win clients who share our values," Bill continued. "Besides good, old-fashioned networking, our most fruitful strategy is to unleash our market-leading strategy and technology teams to publish thought leadership white papers or host custom workshops to help perspective clients tackle common digital business challenges. We're always up for an



industry insight

Bill Brock,
Founder and chief
executive officer

“I pinch myself when I look at our client list. With brands like Absolut Vodka, Chivas Regal and Sainsbury’s on our roster I’d understand if people felt we had a few tricks up our sleeves. But the truth is that we’ve just worked hard”

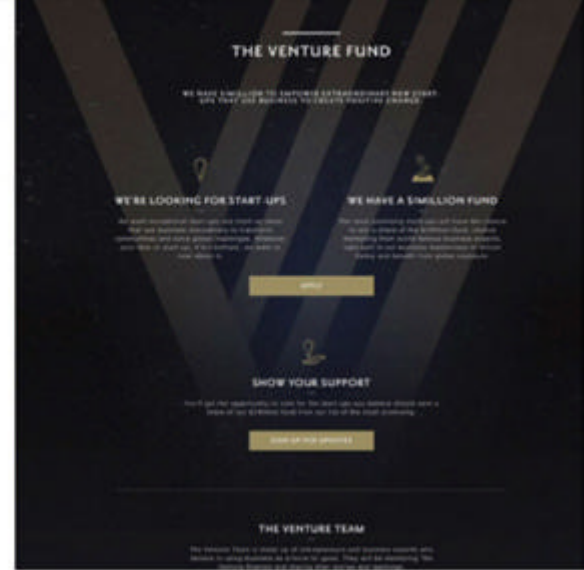
opportunity to show our stuff! “I pinch myself when I look at our client list. With world-class brands like Absolut Vodka, Chivas Regal and Sainsbury’s on our roster I’d understand if people felt we had a few tricks up our sleeves. But the truth is that we’ve just worked hard, have never taken anything for granted and always see the opportunity in any project.”

Agencies that quickly become known in the marketplace are sometimes seen as exponents of a particular kind of digital design. Agencies fight the industry’s desire to label them in this way. However, agencies certainly develop their own DNA. Is there a project that reflects the AnalogFolk ethos? Matt explained: “We are very proud of the diversity of our output, it would be very hard to pigeonhole what we do into any one type of digital marketing. However, the common start point for all our work is ‘what value can we create for people?’ A recent example of this would be the Sainsbury’s ‘Spooky Speaker’ app. Sainsbury’s had a range of kids’ Halloween costumes they were going to sell in their stores, but there was nothing about them that stood out from the others. So, rather than simply advertise them, we decided to increase their value to people

by designing a mobile app that distorts your voice to the character of that costume when you hold it up to your mouth. Alongside over 40,000 downloads and five-star app reviews, it had a direct uplift in sales on that range of costumes.”

The digital design landscape has become incredibly diverse as new technologies appear almost on a weekly basis. Fame outlined the approach AnalogFolk takes on technology: “We work on a wide variety of projects, each requiring a different configuration of tools. From a technology perspective, there [are] multiple platforms and technologies, from open source to commercial software and because of this, we keep an open mind to the tools we use for communicating ideas to our clients. A prototype can save a thousand meetings and get projects moving quickly,

whether it is a paper prototype stitched together in the POP app or key animations and sequences animated quickly in After Effects, prototyping is an essential tool helping to visualise the ambition of the project. HTML5 and CSS3 are very capable technologies for creating semantic content and presenting it efficiently. Being relatively new standards, they will live long and prosper but

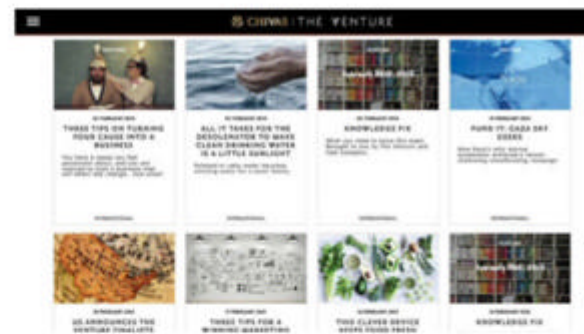


Chivas Regal, The Venture

theventure.com/gb/en/

Chivas Regal awarded AnalogFolk its global digital account in 2013. Along with digital creative, the agency is responsible for social media and editorial content online.

In 2014 AnalogFolk launched The Venture, a new brand platform to inspire and enable young, ambitious professionals to use business as a force for good. The platform was launched in a high-profile ad campaign produced by Havas Worldwide starring Chiwetel Ejiofor and Oscar Isaac. The Venture



website is a fully responsive web platform, making use of HTML5 and CSS3 animations and effects, providing an engaging and immersive experience for users. Housing a host of functions from the online application processes, to regularly updated editorial content about social entrepreneurship, the website



provides a consistent experience across both modern and legacy browsers.

AnalogFolk partnered with Fast Company to create four pieces of content a week for a year, which are published via the site to owned social channels and through paid media.

AnalogFolk

jQuery, released in 2006, is no longer appropriate for web application development. JavaScript has evolved and modern Model View Controller (MVC) frameworks such as Angular and Backbone are now being used for complex apps that would be very difficult to build with jQuery alone.

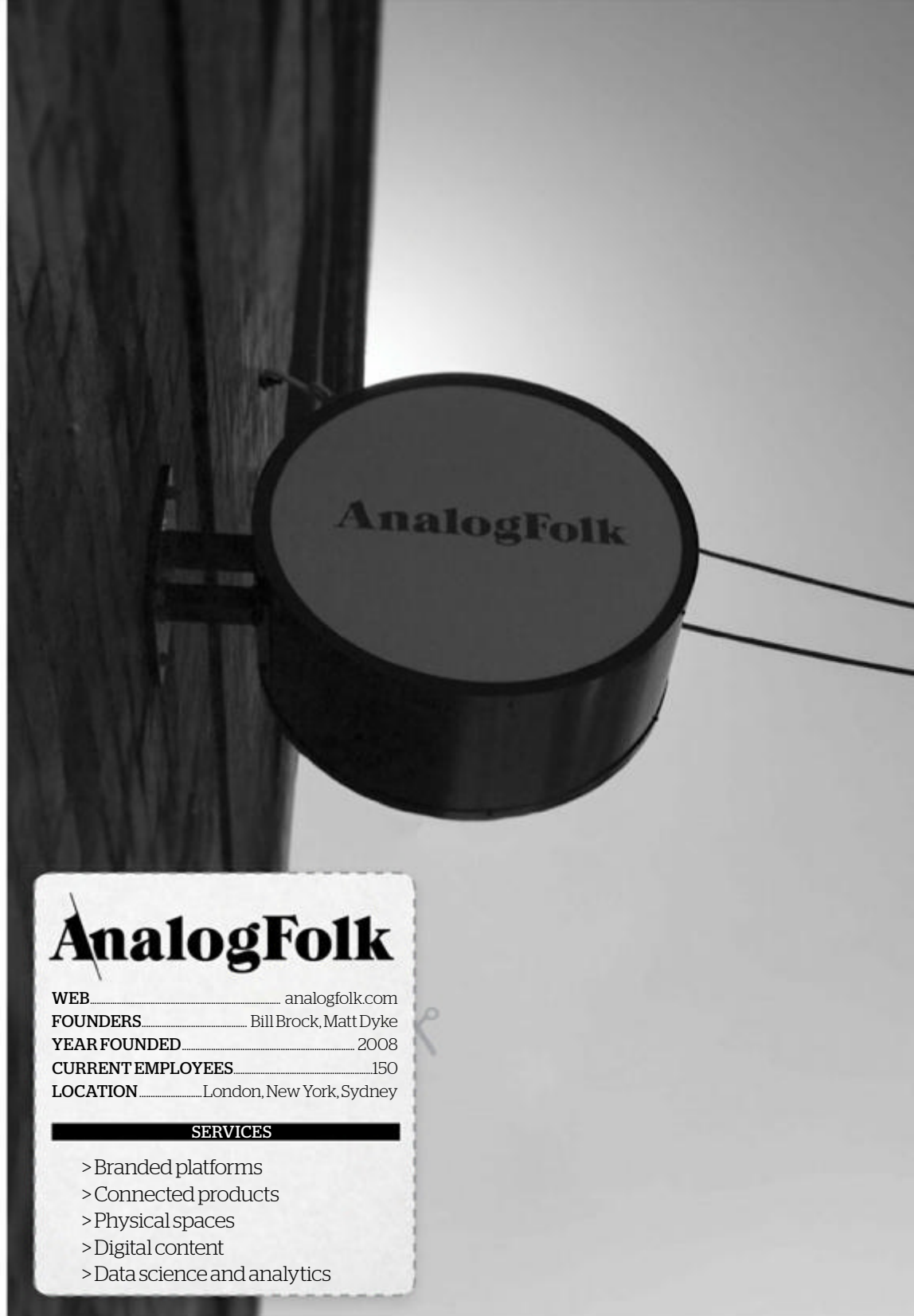
"These technologies are no longer restricted to desktop browsers; wearable devices, kiosks, TV, robots and Internet of Things (IoT) devices, can all render HTML5 and CSS3 content. [When] coupled with ECMAScript 6 (a subset of JavaScript), we are able to deliver engaging experiences across a wide range of channels."

And how does the massive expansion of mobile devices influence how AnalogFolk develops their applications? Fame continued: "Responsive is a great approach and we use it often at AnalogFolk. But, imagine trying to complete a complicated eight step form using your mobile with a 3G connection. That's where responsive might not be the approach to take; sure you can hide an application form with CSS but you are left with bloated code and a poorly optimised experience on that 3G connection. In some cases, you might want to consider an adaptive delivery approach and let the backend serve the most appropriate and optimised content. Ultimately, we keep in mind our users' key objectives and the business needs."

Fame also said: "It seems like something new is announced every day and we're always finding time to experiment with emerging technologies. Wearable devices will be more and more popular and the Apple Watch will give this trend a boost. IoT is taking off and we've loved creating prototypes with the Arduino but the Spark Electron will offer a very simple way for beginner hardware developers to build exciting prototypes with a built-in data chip, that can connect to the cloud and communicate with other services such as **IFTTT.com**."

Social media has also had a massive impact on how digital design is consumed and shared. How does AnalogFolk factor in the influence that social media now has? "As consumers, we each live in a media filter bubble that is curated for us by all the links we click, the media we interact with and the social media we follow," said Matt. "Social media is uniquely capable of quickly propagating content via the endorsement of people that will willingly play this role if the content is interesting enough. It is a modern marketing myth that this is completely organic. Social media is as much a paid channel as any other media property now, albeit with tremendous upside if your content takes off."

AnalogFolk has rapidly expanded and now has 150 talented creative innovators working out of three offices. How does the agency decide who is a good fit for them? Fame explained: "Beyond talent, new folks needs an informed point of view on the digital world, an open mind and a love of their craft. Having a strong opinion is a must, but it has to be



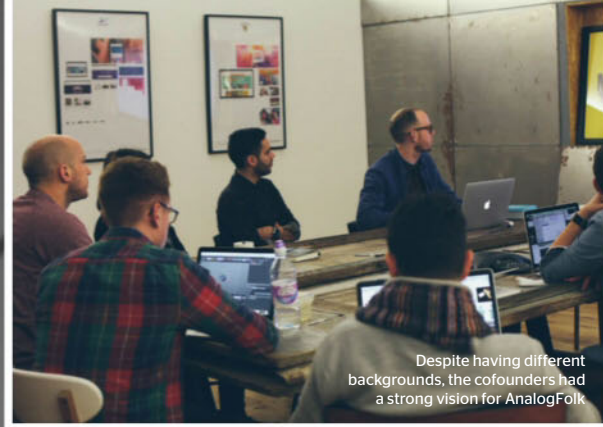
one backed up with knowledge. Tell me about a problem you proactively went about fixing and then show me the prototype you built to solve it. The industry is a great place to be in right now with so many opportunities for folks to learn and network, don't be scared to get involved."

And what does the future hold for AnalogFolk? "We've come a long way since Matt and I set the agency up in 2008, just seven short years ago," concluded Bill. "We continue to grow, and are incredibly proud of the journey the agency has taken, the folk we've hired, the work we've produced and the direction AnalogFolk is heading

in. "We'll always focus our efforts on our ethos of using digital to make the analogue world better, and so we'll always concentrate our energy on interactive experiences. We are 'established' now and so we've grown up somewhat. However, we're independent and we control our own destiny, so we'll always have a start-up mentality."

With a roster of clients that include some of the world's best-known household names, Sainsbury's, Canon and Lucozade to name a few, enhancing our analogue world will continue to be the driving force behind all that AnalogFolk will accomplish over the next few years.

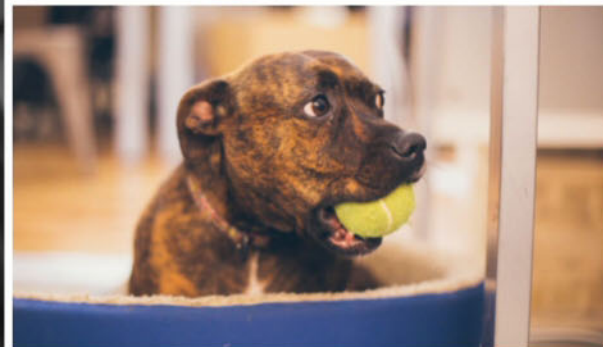
⌘ We'll always focus our efforts on our ethos of using digital to make the analogue world better ⌘



Despite having different backgrounds, the cofounders had a strong vision for AnalogFolk



Make.analogfolk.com is the agency's platform for rapid prototypes and hacks



FUTURE

HTML

Web Components, the new standard in modular HTML development



Tom Dudfield
Senior developer
@tomdudfield

Web Components are a huge step towards a truly modular way of developing front-end interfaces. This mirrors design practices that have been so successful and stable within C# development for some time.



Tim Stone
Lead front-end developer
@timofetimo

Web Components will soon become the de facto way to code reusable widgets. Support isn't universal because it relies on implementing four standards, but you can use it today with polyfills.



Luke Fribbens
 Development director at IA Digital
 @iadigital

Web Components are a developer's dream. We have seen componentisation and its advantages in other parts of development for years - this is the next big step. The best thing since HTML5!



Matthew Bowden
 Creative director at Vitality Health
 @matthewbowden

Tuck away the code we love to tinker with and you're left with components exhibiting consistent behaviour, rendering correctly and most importantly, enabling a coherent brand experience across applications.



Simon Hutchings
 Owner of Visualise Graphics
 @Simonhutchings

Web Components provide so much more control of our markup - this is a fundamental shift in how we develop for the web.

Web Components are a totally new way of building parts of a website that replaces HTML, CSS and JavaScript and are only really used by the mysterious, masters of the web, who tell us all how such things should be done, right? Wrong. Web Components have been around for a long time, hiding just beneath the surface of our beloved HTML. They are simply a combination of HTML, CSS and JavaScript packaged together as an HTML element.

Web Components actually account for many of the huge strides that have been made with HTML5. One of the best examples is the <video> element, by simply adding a src value to this, your video is displayed in the browser with button controls and time indicators, you have a player from just this one HTML element. These are all actually Web Components, we just never thought to stop and look at them all. When you saw this as part of the HTML5 specification you may have thought, 'Wow that's great', but have you ever wondered how this was all built? How are the components we use and take for granted every day actually created?

Plus, we can't just define HTML spec for ourselves, can we? Well that is exactly what we can do. Browsers have been working really hard, over the past year or so to give developers access to this hidden part of HTML. We are not quite there, but the direction taken by all browsers to do just this is a good sign of things to come. There are also a host of JavaScript polyfills available to give us the methods needed to take full control of our own HTML.

Web Components are constructed with four core elements of support that browsers have been working on. Custom Elements enables the creation and manipulation of custom HTML elements. HTML Imports can import packaged HTML, CSS and JavaScript into an HTML page, whereas HTML Templates holds content or information that is not rendered on page load, but is available to render at runtime. Finally, Shadow DOM enables the structure of a web component to be separated from the DOM of the page providing encapsulation. These four elements will be explained in more detail on the next page.

The importance of this change in HTML development cannot be overstated. The evolution of HTML as a language, although excellent, has been pretty slow going. This is mainly due to it being in the hands of the browser vendors, with possible developments/ideas submitted by the general development community. These browsers then prioritise areas of development and gradually define new specifications and build support for these into their applications. But the ability to build our own Web Components gives the entire community control to directly contribute to the evolution of HTML.

The four core elements

Custom Elements </>

Probably the most obvious aspect of a web component is the ability to create, modify and control HTML elements. The introduction of Custom Elements into browser standards enables us to break away from the limited vocabulary that HTML offers. This means we can create true markup semantics that are really meaningful in context of the application, whatever that may be. Gone are the days of endlessly nested <div> tags with a littering of classes trying to suggest a structure as Custom Elements provide us with the ability to write truly beautiful HTML.

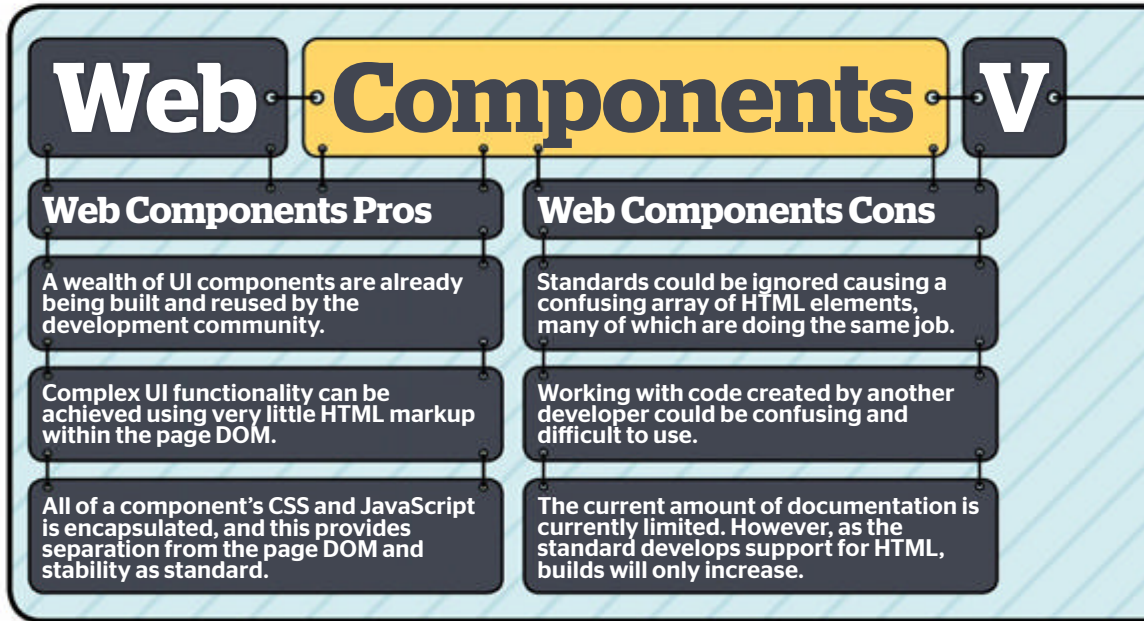
There are a range of new JavaScript methods that enable you to create, register, style and add JavaScript properties and methods all within one new HTML element, and these are outlined really well at html5rocks.com/en/tutorials/webcomponents/customelements.

One thing that is worth noting is that standards have been defined for the naming conventions of Custom Elements. Any element must contain a hyphen, this ensures that Custom Elements are easily identified as such amongst core HTML elements, as well as mitigating the risk of any conflicts with future core HTML elements.

HTML Templates

Templating HTML markup is not a new concept, there are many frameworks that provide this type of approach to building a user interface. They all provide a more streamlined way of managing blocks of HTML, but HTML Templates as a defined part of HTML spec is much more elegant.

The new <template> tag can be used to bundle HTML, CSS and JavaScript together and keep all of its contents encapsulated. Think of these blocks of code as reusable snippets that are easily shared with other developers and projects. The time spent writing the same HTML structures will be dramatically reduced without the need of a framework to do so. Content can be used within a template that resides outside it within a DOM element.



This can be achieved by using the <content> tag within your template and then attaching that template to the desired element with content.

Anything within a template is not rendered in the browser on load but simply made available to be integrated with the DOM using JavaScript. Any content within a template can be used and reused as you wish and if bound to new Custom Elements within it, then the Shadow DOM can be kept out of the DOM as we know it.

This is where HTML Imports come in, the <link> tag can be used with a 'rel' attribute value of 'import', and this tells the browser to import the contents of the source file into your page. It seems just like an include but when we combine this with the HTML Templates we can keep our template code blocks, including any CSS and JavaScript, in separate files. Our working code becomes just as well written as that seen by the browser, and templates are even easier to share and reuse.

HTML Imports

So if we are creating a load of new Custom Elements and templates and bringing these components together, our HTML markup is going to get cluttered and confusing really quickly. This could rapidly lead to unmanageable markup and even more verbose code than before we were using these new techniques. One of the key benefits of using Web Components is the elegance of the code that can be produced, it's all good for your code in the browser to be beautiful, but if the code you are working with is a mess then something is obviously isn't quite right.

Shadow DOM

So last but not most definitely not least, we have the Shadow DOM, this is arguably the most powerful part of Web Components. Each HTML element in the DOM, whether it is core or custom, can have its own DOM hidden within it.

The JavaScript method createShadowRoot() can be used to create this for a new Custom Element. A template that has been imported can then in turn define the structure of this Shadow DOM. It works by bringing all four elements together and giving us a fully functional web component to work with.

Tutorial

Hands on with Web Components

This tutorial will take you through using and creating Web Components within your projects.

As browsers are still undergoing development to support Web Components, Polymer will be used. Polymer is a component polyfill and library created by Google (polymer-project.org). This adds support to all modern browsers as well as a host of preset components ready to use.

1. Install Node.js and Bower

The recommended way to use Polymer is to install using Bower via Node.js. Make sure Node.js is installed and install Bower. The -g flag installs Bower globally too.

```
npm install -g bower
```

2. Add Polymer to your project

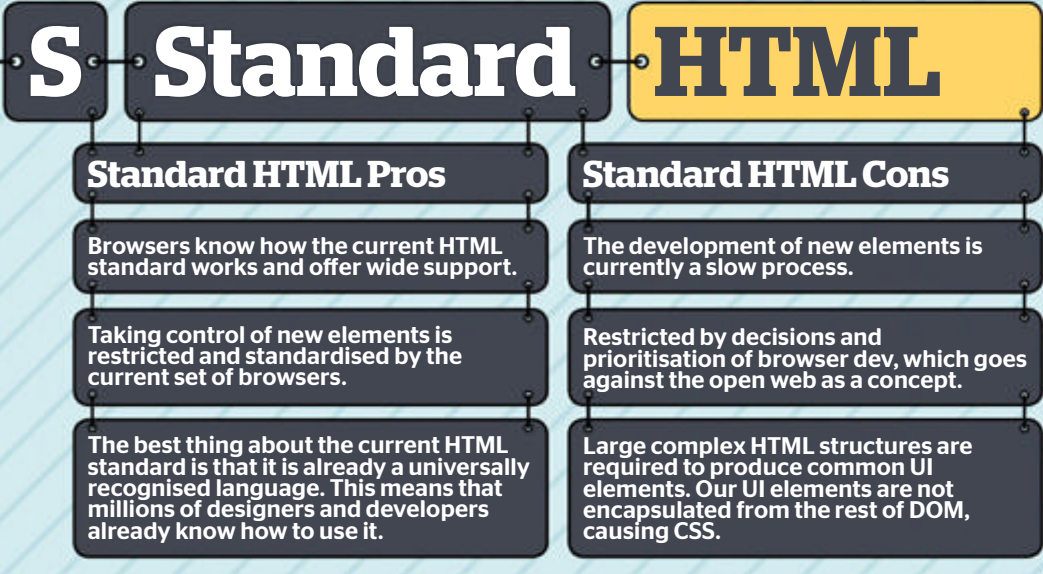
Now use Bower to install Polymer within your project. Make sure you are within your project directory and then install it in your CLI as shown. The 'bower init' command will ensure a bower.json file is created. The '-save' flag will add Polymer as a dependency within this file.

```
D:\Web Designer Mag\Web Components Feature\  
my-component  
bower init  
bower install --save Polymer/polymer#^0.5
```

3. Create your first element

Use an HTML Import to use Polymer to create your Custom Element using <polymer-element>. Add this as a new HTML file. The name of this element is added as an attribute and the contents held within a <template> tag.

```
<link rel="import" href="../bower_  
components/polymer/polymer.html">
```

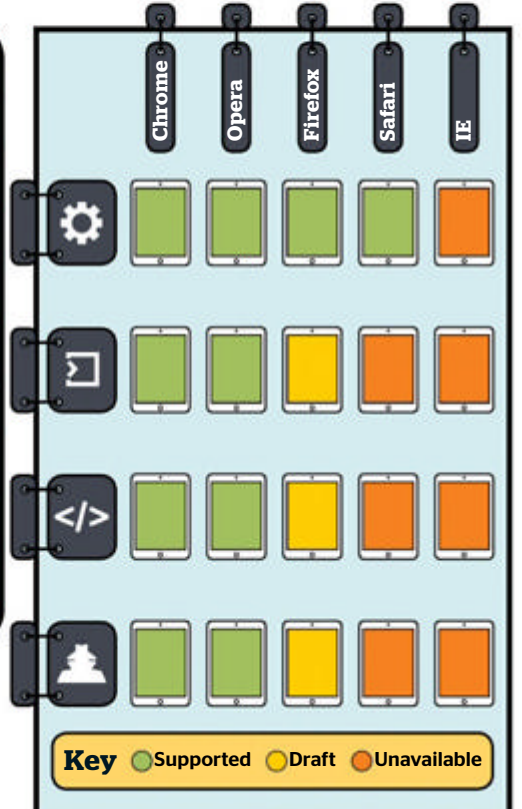
There is much more going on here than it may seem, the Shadow DOM is not just a way of hiding code, it actually deals with the encapsulation issues inherent in HTML, CSS and JavaScript.

For example, if you styled a selector that just happened to match a selector within your Shadow DOM you would expect it to be affected. But the Shadow DOM protects its contents from this, keeping a defined scope for the component. This is vital in enabling the production of reusable components. The door is not completely closed though, you can style the contents of a Shadow DOM by using the `::shadow` selector in CSS, or alternatively style an element from its own Shadow DOM using the `:host` selector.

The Shadow DOM seems like black magic at first, but when you use it you'll notice that it's used everywhere.



Check out Polymer's Elements collection for ready-made web components bit.ly/1NJDrUG



Browser support

The specifications for Web Components from W3C are still in progress but browsers are working to these. Chrome and Opera are leading the way with stable support for all four aspects of Web Components. Firefox have full support for Templates and both Custom Elements and Shadow DOM support are close. Safari supports Templates, and Internet Explorer are collating and prioritising their development roadmap for Web Components. All these browsers are working to the same goal, which is the future of HTML as we know it. It can be difficult to stay up to date with support of so many development paths, the site jonrimmer.github.io/are-we-componentized-yet provides links to information on the paths.

“Custom Elements provide us with the ability to write truly beautiful HTML”

```
<polymer-element name="my-signup" noscript>
<template>
<p>Hi this is <strong>my-signup</strong>.
You are looking at the Shadow DOM</p>
</template>
</polymer-element>
```

4. Create your app

Add an `index.html` file to your project root. Load the `webcomponents.min.js` polyfill from `bower_components`. Import your new Custom Element and then reference this. When viewed in a browser you will see your Shadow DOM content and be able to inspect this using Chrome.

```
<!DOCTYPE html>
<html>
<head>
<script src="bower_components/
webcomponentsjs/webcomponents.min.js"></
script>
<link rel="import" href="elements/my-signup.
html">
</head>
<body>
<my-signup></my-signup>
</body>
</html>
```

5. Style your element

Within your element file, add a `<style>` tag to the template and adjust how the `<p>` tag is displayed. Add another `<p>` tag to your `index.html`.

```
//elements/my-signup.html
<polymer-element name="my-signup" noscript>
<template>
<style>
p {
font-family: Verdana, sans-serif;
font-size: 1.2em;
color: #912A07;
```

Four great building tools

TOOLS, RESOURCES AND DEMOS THAT HELP ESTABLISH A NEW CODING APPROACH

Polymer POLYMER-PROJECT.ORG

The Polymer project was started by Google's development team and makes the Web

Components landscape

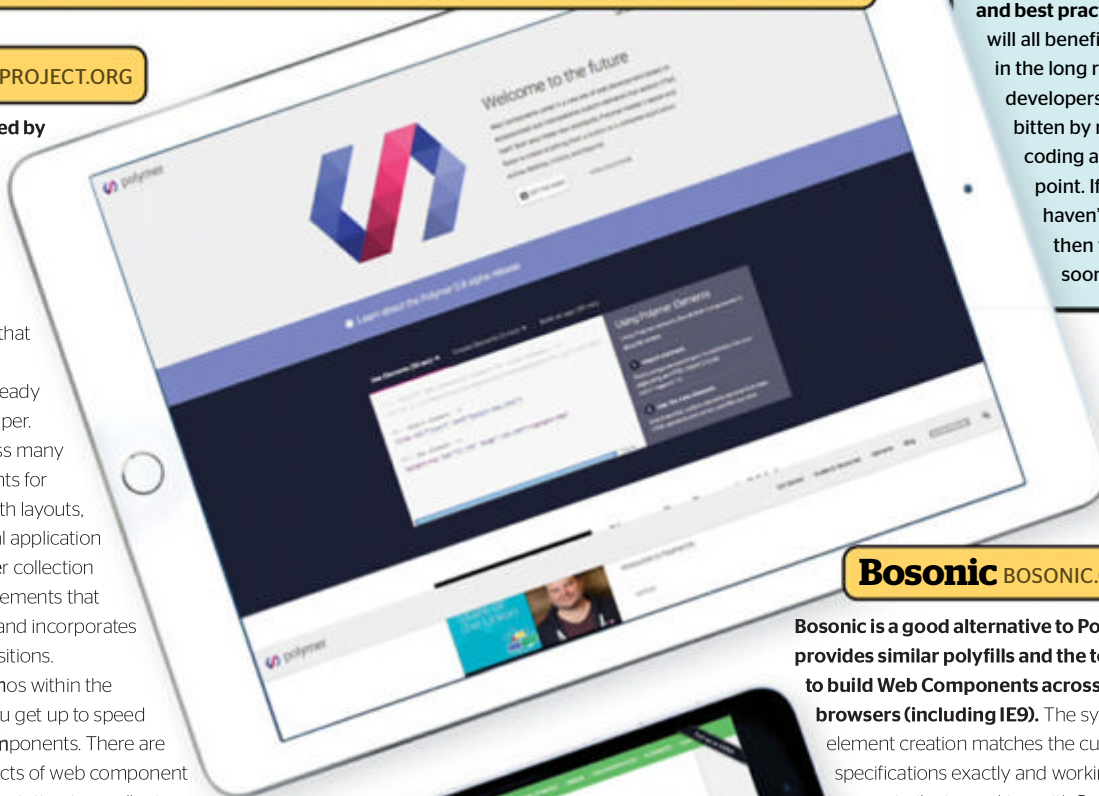
much easier to move to. Any polyfills required to support Web Components across browsers is immediately available. On top of that Polymer ships with two element collections of prebuilt elements ready for production use - Core and Paper.

The Core elements encompass many useful, single purpose components for core functionality. These work with layouts, inputs, data handling and general application structure components. The Paper collection covers much more UI-focused elements that are highly visual and interactive, and incorporates controls, interactions and UI transitions.

There are a wide range of demos within the Polymer project that can help you get up to speed with using and building Web Components. There are tutorials demonstrating key aspects of web component builds and the actual API documentation is excellent.

The Polymer project team have even produced a Designer tool for quickly prototyping apps using Polymer and the ability to save experiments within GitHub (polymer-designer.appspot.com).

Polymer is built based on the (still-in-progress) W3C specifications for Web Components, with the view that when browsers have adopted full support the transition from the polyfills to native support will be smooth. All Polymer code is available on GitHub (github.com/polymer), and benefits from ongoing open source development. Any bugs can be reported directly and visibility of upcoming releases is readily available.



Understand the standards

Before you dive in, make sure that you get your head into the standards and best practices. We will all benefit from this in the long run, all developers have been bitten by maverick coding at some point. If you haven't already, then you will be soon enough.

Bosonic BOSONIC.GITHUB.IO

Bosonic is a good alternative to Polymer, it provides similar polyfills and the tools required to build Web Components across even older browsers (including IE9).

The syntax for the element creation matches the current W3C specifications exactly and working with it is very similar to working with Polymer.

There are a range of elements available for production. This is all available on GitHub github.com/bosonic/bosonic and open to community contributions.

The getting started guide and documentation is structured in a step by step, making it great for learning about

Web Components. If you want to get involved in developing and contributing to a Web Components framework,

Bosonic is a good choice. It is well structured and open but without the scale of contributors as Polymer (at present).

```

}
</style>
<p>Hi this is <strong>my-signup
</strong>. You are looking at the Shadow
DOM</p>
</template>
</polymer-element>
//index.html
<body>
<p>This text is within the page DOM and is
unaffected by component styles.</p>
<my-signup></my-signup>
</body>
    
```

6. Style the element host

If you add .host as a selector to your element styles, you can style the parent element of your Shadow DOM, ie style the Custom Element tag in your index.html from within the element itself.

```

<style>
p {
font-family: Verdana, sans-serif;
font-size: 1.2em;
color: #912A07;
}
</style>
    
```

7. Style the element from your app

Any new Custom Element can be styled from its parent page (in this case this is index.html). If you add styles to this page for your new element they will overrule those set using the .host selector. But the elements within your Shadow DOM will still be protected.

```

<style>
my-signup {
background-color: red;
}
my-signup p {
color: white; /* this will not work */
}
    
```




WebComponents.org WEBCOMPONENTS.ORG

This resource site brings together the best practices, updates and developments of everything Web Components related. It provides a wealth of useful content, from presentations and podcasts given by web component framework developers to the current state of native browser support.

The explanations around the base polyfills (used by all libraries) can give good insight into the work that has already gone into enabling us to work with Web Components today. There are also key links to galleries, libraries and the web component development community, many of which are instrumental in the development of the tools we have a ready mentioned.

One of the best things to take a look at on this site are the presentations, they are often an opportunity to see how other developers are using Web Components or what they are trying to achieve. This is a great place to keep coming back to and will help you to keep up to date in the Web Components landscape.

The core vision of this site is to provide a reference point for everyone in the open development community, ensuring that Web Components best practices are defined and followed, making life easier for all of us.

X-Tag X-TAGS.ORG

X-Tag is the offering from the Mozilla team, and is a small library providing the ability to build, extend and manage your Web Components. This is built from the same W3C polyfills that are provided by the Polymer project with extensions to offer IE9 support.

This library uses JavaScript to register new elements and this script references any HTML Template elements that are to be used, opposed to the Polymer approach of the <script> being integrated into a <polymer-element> alongside the HTML Template.

One of the most comprehensive examples of use is Mozilla's Brick (mozbrick.github.io), a collection of ten UI components that can be reused in production for building interactive UIs. It is actually available as a Bower installation and is a really quick way to start using these components in your project. This library could rapidly evolve, but at present it doesn't have the uptake that Polymer has.



Internet Explorer 9

Remember Polymer does not support Internet Explorer 9, so if you do actually need to support this older browser for your project then X-Tag or Bosonic would be a better choice of library.

Essential polyfilling

SUPPORT MUST BE DONE CONSISTENTLY

As with any new browser technology, it takes time to establish a stable specification, consistent rollout across all browser vendors, and the best practices to be defined by the community. Web Components are no different, but it does seem that lessons have been learnt from previous technology developments. There is a base set of Web Components polyfills available on webcomponents.org/polyfills that are actually used consistently by web component libraries. Some of these aforementioned libraries extend them but this core set remains.

This concerted effort to work to a best practice approach means that as native support increases, the removal of such polyfills will be easier. Currently, these fixes are a necessity, but based on the rapid progression of support and the wide range of reusable components already available, it won't be long before they are no longer needed.

```
};
</style>
```

8. Overruling shadow styles

If you use ::shadow you will find that any styles defined within a component can be altered.

```
<style>
my-signup {
background-color: red;
}
my-signup::shadow p {
color: white; /* this will work */
```

```
};
</style>
```

9. Build the signup form

Add the signup form markup and default styles to the template. The styles don't affect the rest of our page so base CSS selectors can be used without conflicts.

```
<polymer-element name="my-signup" noscript>
<template>
<style>
:host {
display: inline-block;
```

10. Add values to your inputs

Extend the input elements to have values that are set to namespaces within handlebars templating syntax. This will make any value entered available within the context of our component. You should see the 'name' value within the <p> tag as you type.

```
<p>{{name}}</p>
<div class="form-row">
<label>Name:</label>
<input type="text" id="signup_name"
value="{{name}}" />
</div>
```

The community

One key difference between the development community for Web Components and that of other new technologies is the sheer amount of people that Web Components does and will affect. It will be

impossible for anyone working in web development to completely avoid using Web Components and it is easy to underestimate the impact this may have on the industry. With all this in mind the collaboration of everyone involved is vital to ensure that the road to full web component support and take up is as smooth as it possibly can be.

Keep up to date

Keep an eye on the progress of W3C specifications, browser support for all of the key aspects of Web Components and the elements that are added to libraries in any updates. All of these areas will continue to drive and define the best practices that will end up being followed by all, keeping us on the same dev path.

Share what you find

Don't assume that just because the blog post you have found is a few months old, you are the last to find it. Share it with your network and talk to your colleagues about what you find out. Only by doing this and encouraging others to do the same will the community as a whole be able to drive forward maintaining the rate of progression we have seen over the past year.

Using Web Components

Even if you start on a small scale, Web Components are all ready for use in your projects. They provide excellent code separation and reusability which is one of the key

goals of many application frameworks of today. The main difference is that this approach will be standardised across browsers in the future.

Share your code

Most libraries of Web Components provide the means for users to submit their own elements for inclusion, if you have written something useful, it will probably be useful to someone else too. Additionally most Web Components projects are available on GitHub, if you find or fix issues then do so here so that everyone can benefit.

There is an established community of developers who are already contributing to the Web Components landscape. If you aren't sure about something then they can help and answer your questions. We will probably see a lot more conference talks and presentations on the subject over the next two years, so keep an eye out for those as well.

“Most Web Components projects are available on GitHub, if you find or fix issues then do so here so that everyone can benefit”

11. Introduce some script

Add a `<script>` tag to your element. Within this, call the core 'Polymer' class and set a default value for the `{{name}}` property we've defined in the markup. Remove the `noscript` attribute from the `<polymer-element>` tag.

```
<polymer-element name="my-signup">
  <template>...</template>
  <script>
    Polymer({
      name: "Luke" // you can set a default value
      here
    })
  </script>
</polymer-element>
```

12. Pass data to your component

If we know the user's name we can prepopulate it. Add an element attribute and include it as a listed attribute on our Polymer element (the attributes value can be a comma separated list to define multiple attributes).

```
//elements/my-signup.html
<polymer-element name="my-signup"
  attributes="user">
  ...
</polymer-element>
//index.html
<my-signup name="Luke"></my-signup>
```

13. Watching properties

Polymer has a range of changed watchers that are excellent for reacting to property changes within your

component and doing something with them. Add the `nameChanged` method to your script and whenever it is changed, the name will be logged out in console.

```
<script>
  Polymer({
    nameChanged: function() {
      if(this.name){
        console.log('Hi ' + this.name + ' I am
        listening!')
      }
    }
  })
</script>
```

14. Watching groups

Another change watcher Polymer provides is the 'observe' method. This can watch multiple properties and fire when any of them change. Add an observe list to both email properties and bind these to a validate method. The values will be logged out when changed.

```
<script>
  Polymer({
    observe: {
      email: 'validate',
      email_confirm: 'validate'
    },
    validate: function(oldVal, newVal){
      console.log(newVal)
    }
  })
</script>
```

15. Match the email fields

This can now be used to validate that both email fields match. Adjust the 'validate' function to check the fields against each other and create a new `email_match` property within the component for tracking the fields.

```
validate: function(oldVal, newVal){
  if(this.email === this.email_confirm){
    this.email_match = true;
  }
  else{
    this.email_match = false;
  }
  console.log(this.email_match);
}
```

16. Display an error

Add an error message within the element to be displayed to the user. Use the 'hidden' attribute bound to our `email_match` property. This only appears when the emails don't match. We have field match validation here.

```
<div class="form-row">
  <label>Confirm Email:</label>
  <input type="email" id="signup_email_
  confirm" value="{{email_confirm}}" />
  <span hidden?="{{email_match}}">Your email
  addresses do not match</span>
</div>
```

17. Set default validation

You may notice that the error message is appearing before the user has even started to enter an email

Inspecting the Shadow DOM

HOW CAN WE WORK WITH THE HIDDEN DOM?

Accessing the Shadow DOM feels a bit like spying but it is actually quite simple. In Chrome developer tools access the settings using the cog on the right of the inspection panel. Under Elements check 'Show user agent Shadow DOM', then restart the browser. Now when you inspect the HTML of the page you will see a new node wherever a Shadow DOM is present named #shadow-root, you can expand this node to see what is actually in the Shadow DOM. Without this it would be very difficult to debug any of the problems within our Web Components.

Additionally this gives us the ability to interrogate other Web Components that we haven't created ourselves. By doing so we can ascertain how to adjust the styles of a core HTML component such as changing the colours of the HTML5 video player, or enable us to understand how a Custom Element has been built. It's worth taking some time to dig around looking for the Shadow DOM and seeing what you find, you may be surprised how prevalent its use is already.



address, this is because the email_match property does not exist until these properties are changed. Add a default value for this as true within your Polymer script.

```
<script>
  Polymer({
    email_match: true
    ...
  })
</script>
```

18. Style the validation

Add some further styles within your element to display the error message to go alongside the 'Confirm your email' field. Again there is no need for a specific error class here as this is the only instance of a tag within the component.

```
span {
  font-family: Verdana, sans-serif;
  font-size: 0.8em;
  display: block;
  width: 294px;
  margin-left: 148px;
```

19. Add a submit method

Now add a submit function and bind to an on-click event on the button. Within this function bring the name and email properties together so that they are ready to be posted to a HTTP service.

```
submitForm: function(){
  this.formData = {
```

20. Use Polymer core elements

Polymer has a range of core elements that can be used. We will utilise the core-ajax element and this will enable us to POST the form data to a web service. All you have to do is import the core-components-page.html into your element and add the element to your HTML with the form data set as the body attribute.

```
<link rel="import" href="../bower_
components/core-component-page/core-
component-page.html">
```

21. Make the POST url dynamic

To use this signup element in multiple instances you will need to be able to change the POST url. Make sure that you add this as an attribute on the element in index.html, extend the listed attributes on the polymer element itself and finally reference this property in the core-ajax element.

```
//index.html
<my-signup name="Luke" url="http://
dummypostservice.com"></my-signup>
```

22. POST your data to a service

Finally add a handleResponse function to your script and call the 'go' method on the <core-ajax> element within your formSubmit function. This will POST the formData and the response will then be returned to the given method (you will need a POST service available to fully run this last step, but you can stub it using Node.js).

Discover more

EXPLORE A WEALTH OF COMPONENT EXAMPLES TO USE, ADAPT OR MAKE YOUR OWN

Custom Elements

CUSTOMELEMENTS.IO

With over 900 elements, Custom Elements is a huge gallery of Web Components created by the development community. If you need an element fast, this is the place. It is open to submissions and each element is available to fork on GitHub for any adaptations.



Component Kitchen

COMPONENT.KITCHEN

The Component Kitchen combines some good tutorials alongside a comprehensive library of Custom Elements. Each element has its own page with usage information and fully working demo. Within the Developers section of the site you can find out how to register your own elements with Component Kitchen including setting up your demo page.



Polymer Elements

BIT.LY/10ICULO

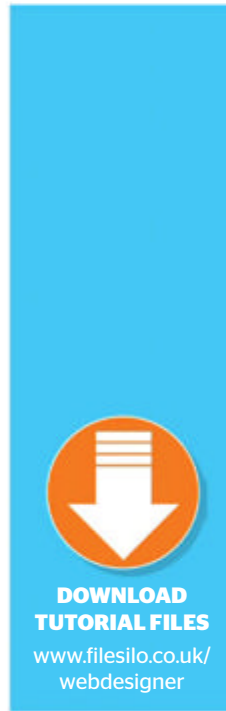
Although not as large as the previous two component libraries, Polymer Elements is excellent and consistently built. The documentation for each of these is very easy to follow and the range available provides most base application functionality and UI interactions to help you build engaging Web Components-based solutions right now.



Build a PhoneGap memo app with photos

Pass files onto your phone quickly and make them run native with PhoneGap Desktop

tools | tech | trends PhoneGap Desktop, PhoneGap Developer, Brackets **expert** Mark Shufflebottom



Recently the PhoneGap team created an app for phones that could pair with the desktop using command-line tools to sync the development environment together with a live testing environment on the

phone. This enables designers and developers to see instant refreshes of their app on the device so that code can be tested immediately without having to endure a lengthy build process.

So what's so special about this tool? Firstly, testing phone-specific features, like using the camera, especially if you are a PhoneGap Build user, took some time to get to the device. Now when you save a file such as HTML or CSS, the phone automatically reloads the content and you can test your app as soon as it's transferred. Secondly, this is a simple user interface that enables you to manage multiple PhoneGap projects on your computer, making development much easier. Thirdly, iOS developers can get a taste of their app on the device without having to pay out for a developer licence until they are ready to start building.

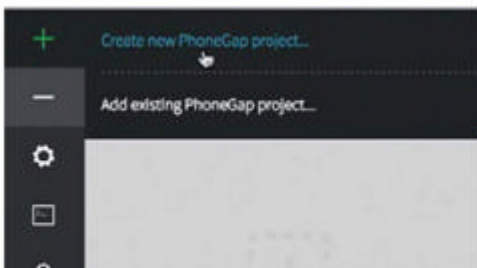
01 Get started

First step is to get the PhoneGap Desktop App for the platform of your choice. Visit phonegap.com/blog/2014/12/11/phonegap-desktop-app-beta and under 'Get the app' click on either Windows or Mac to get your version of the app. After it has downloaded follow normal install procedures for your computer.



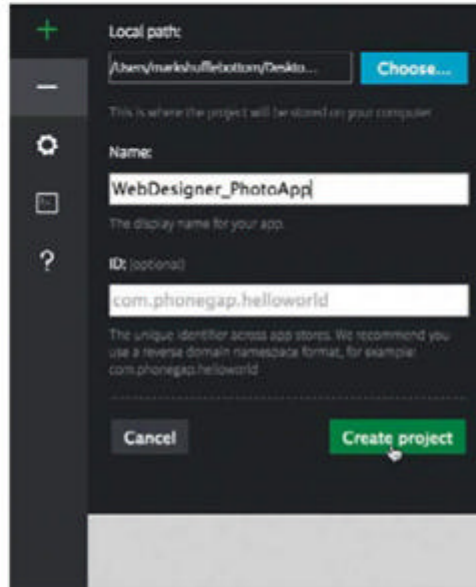
02 Open the app

After installation, the app is usually just extracted, so move it into your Applications folder and then double-click it to start. The interface is extremely minimal because there is not a lot of options for the app. Click the plus symbol to create an app and click 'Create new PhoneGap project...'



03 Name the app

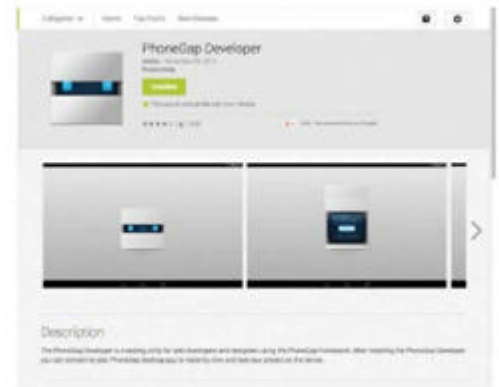
Click on the Choose button and browse to the folder that you want to save the PhoneGap App to. Once you've done this click OK and then add a name for the app. It's not necessary at this stage as it can be added later, but you can add a reverse domain name for the app. 'Click Create Project' to continue.



04 Get the Developer App

You have created your first project and you need to see this on your phone, so go to your respective phone's app store and search for 'PhoneGap Developer

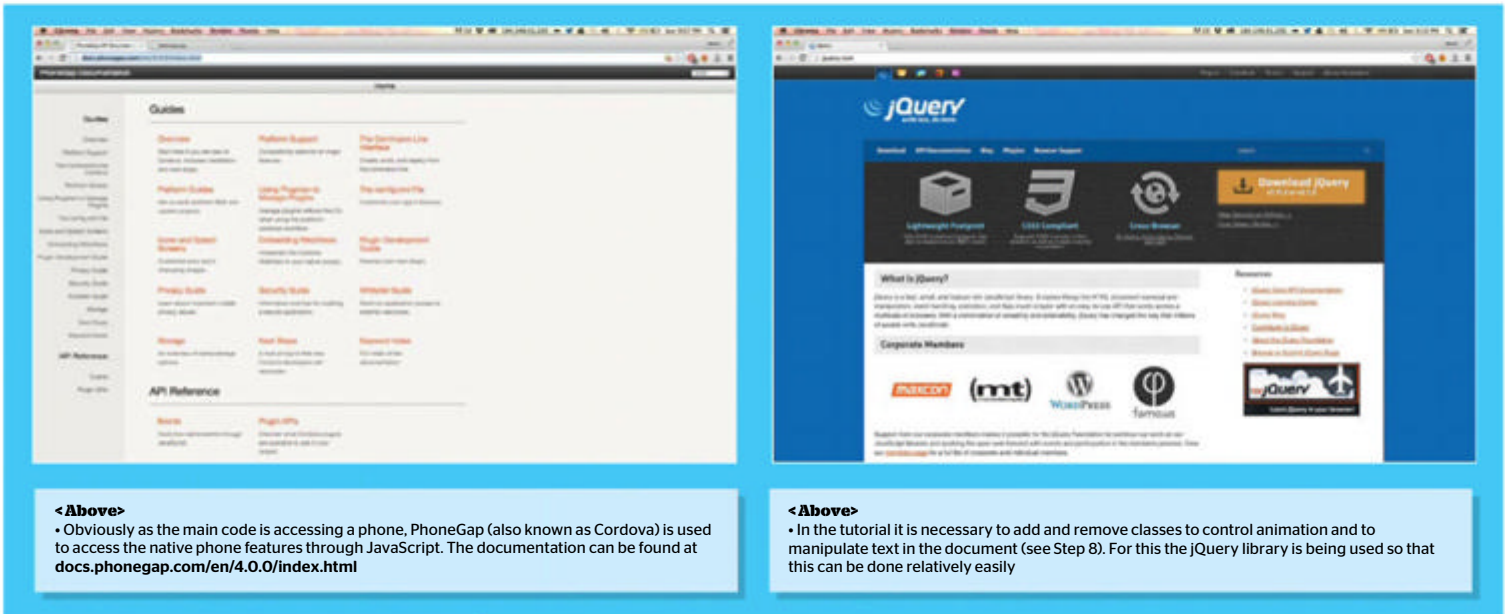
App'. Download this app to your device. Once downloaded you can start this app on your device. When this is running you will see that you are asked to put in the server address.



05 Connect the two together

On the desktop app there is a green bar at the bottom, this tells you the address that the server is running at. You will need to be on the same Wi-Fi network as your phone. Put the address shown into your phone and hit Connect. On the screen you will see a green flashing message stating 'device is ready'.





<Above>

• Obviously as the main code is accessing a phone, PhoneGap (also known as Cordova) is used to access the native phone features through JavaScript. The documentation can be found at docs.phonegap.com/en/4.0.0/index.html

<Above>

• In the tutorial it is necessary to add and remove classes to control animation and to manipulate text in the document (see Step 8). For this the jQuery library is being used so that this can be done relatively easily

Desktop to mobile

When you have the PhoneGap Desktop App open it launches a server in the background, powered by Node.JS. This server sends the files to the phone as if they are a native app on the phone.

06 Your app on the phone

What you are seeing on your screen is the HTML, CSS and JS from the demo app being placed on your phone as an app. From the tutorial files on Filesilo, copy the WWW folder from the Start folder over the existing WWW folder. This just has the font, CSS and HTML in place so that we can create the app with JavaScript.



07 Create the app

Open the index.js file from the js folder and add the code where there is a comment '//add code here' in the receivedEvent function. These four variables help us create the app which tell the app that the overlay effect is not on, an array is set up to store variables and that a variable will hold the picture.

```
001 var overlay=false;
002 var memos = [];
003 var key = "memos";
```

```
004 var pic;
005
```

08 Detect user input

When the user presses the 'new' button in the HTML, this checks to make sure the overlay effect has not been set, the 'add' div section is made visible and set to animate in with a bounce from the left. As this is now out, the rest of the interface is hidden and therefore the overlay is true.

```
001
002 $("#new").click(function() {
003   if (overlay == false){
004     $('#add').removeClass('hidden');
005     $('#add').addClass('animated
006     bounceInLeft');
007     overlay = true;
008   }
009 });
```

09 Other button inputs

When the user clicks on the submit button they are going to create a memo so here, the appropriate createMemo function is called. In the case of pressing the photo button the accessCam function will be called to start using the phone's camera. In both cases the code for these functions hasn't been added yet.

```
001
002 $("#submit").click(function() {
003   createMemo();
004 });
005 $("#photo").click(function() {
006   accessCam();
007 });
008
```

10 Access the phone's camera

In this function the camera on the phone is called and in most phones, it will bring up the native phone's camera app. Here the quality of the finished JPEG is set at 75 quality and the target width and height is set at 100 pixels so that it will fit within the interface that has been created in the HTML.

```
001 function accessCam(event) {
002   navigator.camera.getPicture(onSuccess,
003   onFail, {
004     quality: 75,
005     destinationType: Camera.DestinationType.
006     DATA_URL,
007     targetWidth: 100,
008     targetHeight: 100
009 });
```

11 Successful photo capture

If the user has successfully taken a photo, the image is passed to the success function as a base64 image in the JPEG format - it is stored in the pic variable and displayed on screen. If there is a problem the onFail function is called and an alert will then explain the problem that has occurred.

```
001
002 function onSuccess(imageData) {
003   pic = imageData;
004   $('#preview').prepend(' pipe.position.x &&
```

```
004 memos.push(memo);
005
```

14 Store the data

The storeMemos function is called which will write the memo into the user's local storage as JSON. The interface needs to be updated and the overlay removed. The appropriate classes are removed and then the animation for moving the overlay out is added. A listener checks that this finishes, and then the overlay is hidden and the variable set to false.

```
001
002 storeMemos();
003 $("#add").removeClass('animated bounceInLeft').addClass('animated bounceOutLeft');
004 $("#add").one('webkitAnimationEnd mozAnimationEnd MSAnimationEnd oanimationend animationend', function() {
005   $("#add").removeClass('animated bounceOutLeft').addClass('hidden');
006   overlay = false;
007   return false;
008 });
009
```

15 Clean up the input

As the overlay interface is animated off the screen the memo needs to be added to the page, so the appropriate function is called. The text field is cleared of the current memo so if the user wants to add another memo they see an empty text input area and the image preview is removed as well.

```
001 addMemoToPage(memo);
002 $("#memo").val('');
003 $("#preview").empty();
```

```
004 }
005
```

16 Add to the page

The function to add the memo to the page is being created now and here the memo's element in the HTML is grabbed and a new list element is added within that unordered list. The list element is given the image and the appropriate class that styles it on the page.

```
001
002 function addMemoToPage(memo) {
003   var memosUl = document.getElementById("memos");
004   var li = document.createElement("li");
005   li.innerHTML = '';
006 }
```

17 Add the text

After the image has been displayed the text is added into a paragraph element. If this is the first list element the list is added, however if there are already list elements, this text and image are added at the top so that new memos appear at the top.

```
001
002 li.innerHTML += '<p>' + memo.text + '</p>';
003 if (memosUl.childElementCount > 0) {
004   memosUl.insertBefore(li, memosUl.firstChild);
005 }
006 else {
007   memosUl.appendChild(li);
008 }
009 }
010
```


Example

Take a photo and retrieve it as a base64-encoded image:

```
navigator.camera.getPicture(onSuccess, onFail, { quality: 50,
  destinationType: Camera.DestinationType.DATA_URL
});

function onSuccess(imageData) {
  var image = document.getElementById('myImage');
  image.src = "data:image/jpeg;base64," + imageData;
}

function onFail(message) {
  alert('Failed because: ' + message);
}
```

Take a photo and retrieve the image's file location:

```
navigator.camera.getPicture(onSuccess, onFail, { quality: 50,
  destinationType: Camera.DestinationType.FILE_URI });

function onSuccess(imageURI) {
  var image = document.getElementById('myImage');
  image.src = imageURI;
}

function onFail(message) {
  alert('Failed because: ' + message);
}
```

CameraOptions

Optional parameters to customize the camera settings.

```
{ quality : 75,
  destinationType : Camera.DestinationType.DATA_URL,
  sourceType : Camera.PictureSourceType.CAMERA,
  allowEdit : true,
  encodingType : Camera.EncodingType.JPEG,
  targetWidth : 100,
  targetHeight : 100.
```

Cut and paste PhoneGap

If you are new to using PhoneGap it might seem like an uphill struggle knowing what the commands are and how to use them. The documentation online is excellent though and you will find many examples of code that you can simply cut and paste into your work to get it going. For example, looking at the camera API will give you code that can be dropped in. Once you have the basic functionality it simply becomes a case of building up the code until you have enough functionality for your app to work. Remember when creating apps they should have one main purpose so try and keep it to that. You could always add extra functionality once it is released and you start to receive feedback.

18 Store the memos

Now that the image and text have been added onto the screen, the code here sets about storing the memo's permanently by converting them into a JSON object. This is then stored in the local storage so that next time the app is launched the data can be retrieved and shown to the user.

```
001 function storeMemos() {
002 var jsonMemos = JSON.stringify(memos);
003 localStorage.setItem(key, jsonMemos);
004 }
005
```

19 Load the memos

When a user starts the app there needs to be a way to check if there are memos from a previous occasion and show those on the screen. This function lets that happen by checking the local storage and if there is something in there, these memos are sent to the addMemoToPage function from Step 16 and 17.

```
001 function loadMemos() {
002 var jsonMemos = localStorage.getItem(key);
003 if (jsonMemos != null) {
004 memos = JSON.parse(jsonMemos);
005 for (var i = 0; i < memos.length; i++) {
006 addMemoToPage(memos[i]);
007 }
```

```
008 }
009 }
010
```

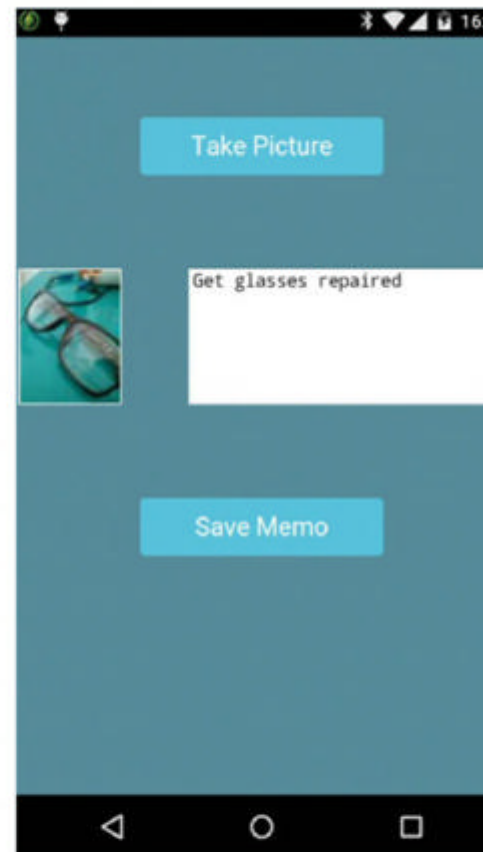
20 Check local storage

In this section of the code the 'if' statement checks if there is local storage available, if there isn't then the user is alerted to that. If there is local storage then the memos are loaded into the display by calling the loadMemos function. Save your file now and this will automatically call the PhoneGap App on the device to reload your project.

```
001
002 if (!window.localStorage) {
003 alert("The Web Storage API is not
supported.");
004 } else {
005 loadMemos();
006 }
007
```

21 Check it in action

You will see a relatively empty interface but if you click the plus icon in the top-left corner of the screen, the overlay will appear and you will then be able to snap a picture and write a note. Hitting submit will save this so that next time you launch the app this data will be here ready for you.



Construct custom web layouts with CSS Shapes

Build layouts using shapes, and wrap content without affecting content flow

tools | tech | trends HTML, CSS
expert Neil Pearce



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



here are so many remarkable and exciting things that we designers can now achieve using just CSS. Not only can we animate elements to jazz up our pages, but we can now create some really interesting shapes using CSS

transforms. These shapes will not affect the flow of the content inside or around them. That is, if you create a triangle with CSS, for example, the shape created does not define or affect the way the text inside it flows, or the way inline text flows around it.

With the introduction of CSS Shapes into the web design workflow, wrapping content in custom nonrectangular shapes and re-creating print designs and layouts on the web becomes a breeze! CSS Shapes enables us to wrap content around custom paths, which lets us break free from the constraints of the rectangle we have been accustomed to.

So in this tutorial we'll take a look at how we can use CSS Shapes to create a custom layout. We won't be looking at all the new properties, but rather just the ones that enables us to create an interesting layout based on a visit to a world-famous landmark.

01 Get started

As with all new projects, let's begin with a new HTML document and start adding in our HTML markup. We're going to create just the one style sheet and then below that we will use HTML5 Shiv to compensate for IE users when using HTML5 elements.

```
001 <head>
002 <meta charset="utf-8" />
003 <title>CSS shapes</title>
004 <link rel="stylesheet" type="text/css"
href="css/style.css" />
005 <meta name="viewport"
content="width=device-width, initial-
scale=1.0">
006 <!--[if IE]>
007 <script src="http://html5shiv.googlecode.
com/svn/trunk/html5.js"></script>
008 <![endif]-->
009 <!--[if gte IE 9]>
010 <style type="text/css">
011 .gradient {
012 filter: none;
013 }
014 </style>
015 <![endif]-->
016 </head>
```

02 The page wrap

Now that we have the head meta information done, let's start adding some structure to our HTML file. First, we will add in a few divs that will give us some control when positioning elements on the page with CSS.

The container div may be the only one we'll be using, but we'll wrap that within a page wrapper in case we need it.

```
001 <div id="wrapper" >
002 <div class="container">
003 </div><! END container >
004 </div><! End wrapper >
005
```

03 Shape up

You can apply a shape to an element using one of the shapes properties and its function. In this case, we'll be adding the polygon function to the background image of Stonehenge so we can create an interesting shape. So just underneath the container div, place a div with a class name of 'shaped'.

```
001 <div class="shaped"></div>
002
```

04 Content and title

Our page is going to have a heading and we are going to make the first word "the" slightly smaller than the other words. So to accomplish this, we will need to wrap it within a tag. Our page title, along with everything else for the page, will be placed within a main content element.

```
001 <div class="content">
002 <h1><span>The</span> Visit To <br/>
Stonehenge</h1>
003 </div>
004
```

05 Add in paragraphs

Our page is going to have a few paragraphs, so within the content div put about four or five <p> tags and fill them with either dummy text or go ahead and add in some proper information relating to our topic. We've left a gap after the second <p> tags, because in the next step, we will be adding in a few extra elements such as an arrow and profile picture.

```
001 <p>Lorem ipsum dolor sit amet, consectetur
adipiscing elit.</p>
002 <p>Lorem ipsum dolor sit amet, consectetur
adipiscing elit.</p>
003 <p>Lorem ipsum dolor sit amet, consectetur
adipiscing elit.</p>
004 <p>Lorem ipsum dolor sit amet, consectetur
adipiscing elit.</p>
```

06 Arrow, circle and profile image

As mentioned in the last step, we are going to add in some additional elements within our paragraphs. What we want to achieve here is a green circle with an arrow pointing to a profile image. And within the green circle we will have some text saying "this is me!". Then we can use this profile image to demonstrate our second CSS shape in a later step.

```
001 <div id="arrow"></div>
002 <div class="circle"><p>This is me!</p></div>
003 
004
```




- <Top left>**
 - With most of the markup added, we are starting to see our page come together
- <Top right>**
 - Now we have the profile image added as well as the content blockquote and footer
- <Bottom left>**
 - The page has some structure to it now and our first shape has been created on the main image
- <Bottom right>**
 - We've now styled the main content and things are looking a lot more cleaner

Custom paths

CSS Shapes enable web designers to wrap content around custom paths like circles, ellipses and polygons.

07 Put in a blockquote and footer
 We are going to finish up our HTML by adding in a blockquote and footer to the bottom of our page. The blockquote could be styled, so we encourage you to play around with that idea, but we will leave the footer alone as it's there for semantic purposes only.

```
001 <div class="blockquote">
002 <blockquote></blockquote>
003 </div>
004 <footer>
005 <p>&copy; Copyright 2015 The Visit To Stonehenge</p>
006 </footer>
```

08 Box sizing
 Now create a new CSS file and call it 'styles.css', and place it in its own folder called 'CSS'. At the top of your CSS file, we are going to add in the box-sizing rule. This will apply a natural box layout model to all elements, but enabling components to change, and it's now considered best practice to add this to your CSS.

```
001 * {
002 -moz-box-sizing: border-box;
```

```
003 -webkit-box-sizing: border-box;
004 box-sizing: border-box;
005 }
```

09 Set the width and height
 Modern websites nowadays have fullscreen sections on their homepage and this is what we want. To achieve this, we set the viewpoint's height and width to 100 per cent. Then we are going to position the container relative, which will now enable us to absolute position any element anywhere on the page.

```
001 .container{
002 overflow:hidden;
003 height: 100vh;
004 width: 100vw;
005 position: relative;
006 }
```

10 Background image
 This page is about Stonehenge, so it's obvious we want a main image of Stonehenge. We are going to attach this to the 'Shaped' class, set the viewpoint and make sure that the image is centred from the top and doesn't repeat. We then need to make sure this is floated to the right.

```
001 .shaped{
002 height:100vh;
003 width:40vw;
004 float:right;
005 background: black url(../imgs/stonehenge.jpg) center top no-repeat;
```

```
006 background-size: cover;
007 }
```

11 Create our first shape
 Having added our main image, let's now create our first shape. Each shape is defined by a set of points. Some functions take points as parameters, but they all eventually draw the shapes as a set of points on the element. We are going to use the polygon function here and create points to create our desired shape. Using the clip-path property will clip all the parts of the image that are outside the defined shape.

```
001 -webkit-clip-path: polygon(0 0, 100% 0, 100% 100%, 30% 100%);
002 -webkit-shape-margin: 20px;
003 }
```

12 Style the content
 Now we need to add some styles to our main content. This is going to be pretty straightforward and by looking at the CSS rule, you can see clearly what we are doing here. You might want to play around with the padding and font size to get a slightly different look, but that's up to you.

```
001 .content {
002 padding: 30px;
003 color: black;
004 font-size:15px;
005 text-align: justify;
006 line-height: 23px;
007 font-family: Verdana, Arial, sans-serif;
```



- <Top left>**
 - The page title is now styled and our second shape is now created for the text
- <Top right>**
 - The page title is finished off by making the word "My" small
- <Bottom left>**
 - We can now start to style our profile image and make it into a circular shape
- <Bottom right>**
 - Now with the green circle and arrow added, we have finally finished our page

Polygon shape

The minimum number of pairs to specify a polygon is three, which is of course a triangle shape.

```
008 }
009
```

13 Page title

The main page title is going to be nice and big, so it grabs attention straight away. We've used a Google Font called Sintony. Now, this is where we use our second shape. By using 'shape-outside: polygon()' we can create another shape that targets the main text. To make things interesting, we are going to shape it the opposite way to our main image.

```
001 .content h1{
002 font-family: 'Sintony', sans-serif;
003 font-size:50px;
004 line-height:1;
005 float:left;
006 width:350px;
007 height:100vh;
008 margin-top:0;
009 padding-top:20px;
010 color: #3f3f3f;
011 -webkit-shape-outside: polygon(0 0, 100% 0,
012 40% 100%, 0 100%);
012 }
```

14 Finish up the page title

Now that we have used the page title (h1) to shape our content, the only thing left to do is simply to make the word "My" nice and small by adding some styles to the 'span' tag. Then we can pull in the word "visit" by adding some negative right margin.

```
001 .content h1 span{
002 font-size:.5em;
003 display:inline-block;
004 margin-right:-10px;
005 }
```

15 Style our profile image

Because this page is about someone visiting Stonehenge, it would make sense to add a small profile image to the page. And doing so will give us another chance to play with CSS Shapes. But first we are going to add some styles to our image and float it to the left.

```
001 .profile_img {
002 float: left;
003 width: 180px;
004 height: 180px;
005 border-radius: 50%;
006 margin-right: 15px;
007 padding: 0px;
008 }
009
```

16 Shape our profile image

At the moment our content is pushed over to the right-hand side of our image and there's no shape to it

other than a square shape cut out of the text. But we want the text to flow around our image in a circular shape. We can achieve this by adding 'shape-outside: circle()'; and some margin and border.

```
001 -webkit-shape-outside: circle();
002 -webkit-shape-margin: 10px;
003 border: 10px solid #f1f1f1;
004 }
```

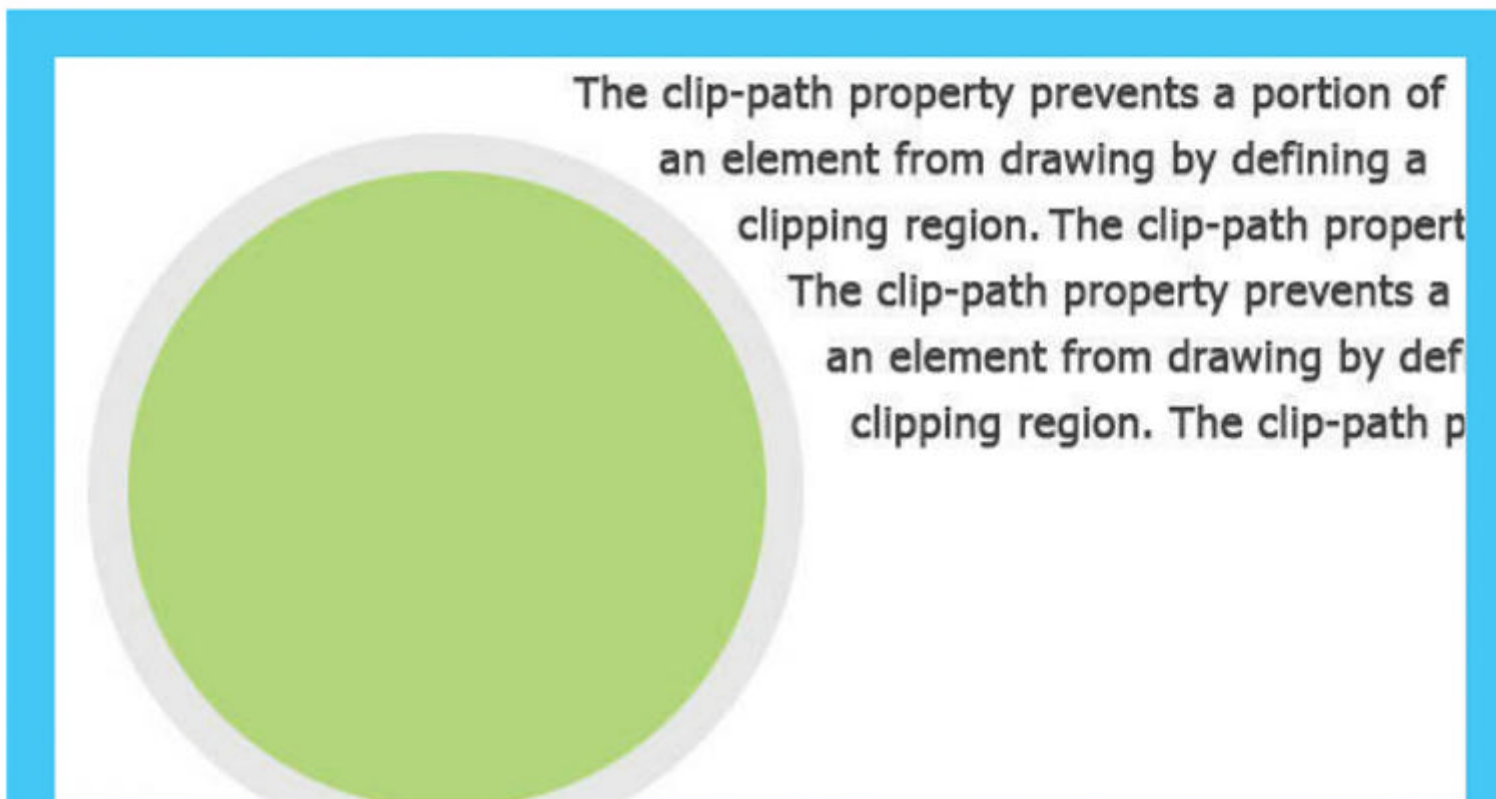
17 Circle styles

To practice with creating other shapes using CSS, we are going to add a green circle that floats next to our profile image. This circle is created by using border-radius. Then we will give it a nice green colour that represents the feel of our main Stonehenge image, and position it accordingly.

```
001 .circle {
002 width: 100px;
003 height: 100px;
004 background: #9fb876;
005 -webkit-border-radius: 130px;
006 border-radius: 130px;
007 position: absolute;
008 top: 350px;
009 left: 180px;
010 border: 10px solid #f1f1f1;
011 }
```

18 Circle content

The green circle is going to have a little bit of text within it, by targeting its <p> we can make sure the text is



The clip-path property prevents a portion of an element from drawing by defining a clipping region. The clip-path property prevents a portion of an element from drawing by defining a clipping region. The clip-path property prevents a portion of an element from drawing by defining a clipping region.

Understanding the clip-path property

The clip-path property takes the same shape functions and values as the shape properties. If we pass the same polygonal shape we used in the shape-outside property to the clip-path property, it will clip all the parts of the image that are outside the defined shape.

The clip-path property is supported with prefixes and will work with the Chrome-webkit- prefix added. The clip-path property is an excellent companion to the shape properties, as it helps visualise the created shapes and clip out any parts of the element that are outside the defined shapes. You will probably find yourself using it a lot in conjunction with the shapes properties.

styled how we want it and fits within the circle nice and tidy. We're going to set the colour to the same colour as our page title.

```
001 .circle p {
002 margin: 20px 0;
003 font-size: 1em;
004 text-align: center;
005 color: #3f3f3f;
006 }
```

19 The arrow

Creating other shapes without using the new CSS shape functions is good fun. In this next step we will create a small arrowhead purely using CSS and we will then position it between our main profile image and the green circle. You can play around with the rotate value using transform: rotate(), but setting it to 5deg should work well too.

```
001 #arrow {
002 position: absolute;
003 width: 0;
004 height: 0;
```

```
005 top: 380px;
006 left: 300px;
007 border-top: 18px solid transparent;
008 border-right: 18px solid #9fb876;;
009 -webkit-transform: rotate(5deg);
010 }
```

20 Finish up the arrow

Now that we have created a small arrowhead, we can now create its tail. So again using border and border-radius we can create an arrowtail and position it absolute, using 'transform: rotate(45deg)' to place it exactly where we want it to be. So now we have a nice little arrow pointing to our profile image.

```
001 #arrow:after {
002 content: "";
003 position: absolute;
004 border: 0 solid transparent;
005 border-top: 6px solid #9fb876;;
006 border-radius: 20px 0 0 0;
007 top: -24px;
008 left: -18px;
009 width: 24px;
```

```
010 height: 24px;
011 -webkit-transform: rotate(45deg);
012 }
```

21 Style the blockquote

To finish this page off, we are going to add some simple styling to our blockquote. So all we are going to do here is centre our text so it looks nice and tidy. By all means add more styles to it to see if you can achieve something slightly different.

```
001 .blockquote {
002 text-align: center;
003 }
004
```


22 Final thoughts

The current CSS Shapes specification is merely the first step. Soon, new options will give us more control over creating shapes and wrapping content in and around them, making it a lot easier for us to turn our mockups into live designs with just a few lines of code. So this tutorial is just the first steps to bigger and better page layouts using the power of CSS Shapes.

Get faster animations by using Famous

Build an image gallery super quickly with Famous' surfaces, modifiers and animations

tools | tech | trends Famous, your favourite IDE, a modern web browser
expert Sean M. Tracey



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



Ever since the advent of computing, we've been pushing for more power, more speed, more memory, more storage.

Even though it arrived 40 odd years after the first computers, the web is not free of this demand, we need more! Then JavaScript was added to a browser, but still we've been pushing our browsers to do things quicker and quicker - from face detection to generating whole 3D worlds, we can do it all. One thing, however, still remains a little slow - DOM manipulation. Manipulating the DOM in almost any way triggers redraws and reflows that can seriously affect performance.

Now that we have CSS3, JS and hardware acceleration, there are ways around this and we now have the tools to squeeze the best out of every cycle, but as with all things browser-based, each approach and browser has its quirks. But now, we have Famous, a JavaScript library that abstracts away all of the mucking about with DOM reflows and animations. We make the web app with the transitions we need, and it makes the transitions with the 60FPS we deserve.

01 Set it up

We're going to need Famous.js to get started with the image gallery project for our tutorial. For simplicity's sake, you can download and retrieve the starter project from our dedicated FileSilo site (filesilo.co.uk/webdesigner). In the root of the folder is a simple Node.js/Express app. Just enter:

```
001 npm install; node server.js;
002
```

and you'll be able to view the project files at <http://localhost:8118>;

02 Get global

In the src folder, open the main.js file. To help us keep track of the bits of Famous that we will end up using, we're going to create some references for the global objects that Famous declares as we import them into our project.

```
001 var Engine = famous.core.Engine;
002 var Modifier = famous.core.Modifier;
003 var Transform = famous.core.Transform;
004 var ImageSurface = famous.surfaces.
ImageSurface;
005 var ContainerSurface = famous.surfaces.
ContainerSurface;
006 var EventHandler = famous.core.
EventHandler;
007 var Transitionable = famous.transitions.
Transitionable
008
```

03 Create a context

In order to add things to our Famous app, we will need to first create a context for our Famous app. The Famous context is kind of similar to a <canvas> context, but it works more like a virtual DOM than a series of points. We can create a context by adding the following line to main.js

```
001 var mainContext = Engine.
createContext();
002
```

04 Make the gallery

We will need to use a lot of images to really tax the Famous rendering engine. The assets provided on FileSilo contain 20 Creative Commons images that can be used to create a mock gallery. Replace these with your own images to create a more personal gallery. In main.js, we create an images array with the name of the files that we are going to add to our project and then some simple variables that will then help us to position them inside of our window:

```
001 var images = ["1.jpg", "2.jpg", "3.jpg",
"4.jpg", "5.jpg", "6.jpg", "7.jpg", "8.jpg",
"9.jpg", "10.jpg", "11.jpg", "12.jpg", "13.
jpg", "14.jpg", "15.jpg", "16.jpg", "17.jpg",
"18.jpg", "19.jpg", "20.jpg"];
002 var thumbSize = Math.ceil(window.innerWidth
/ 7),
003 columns = window.innerWidth / thumbSize,
004 rows = window.innerHeight / thumbSize,
005 idx = 0,
```

```
006 count = 0;
007
```

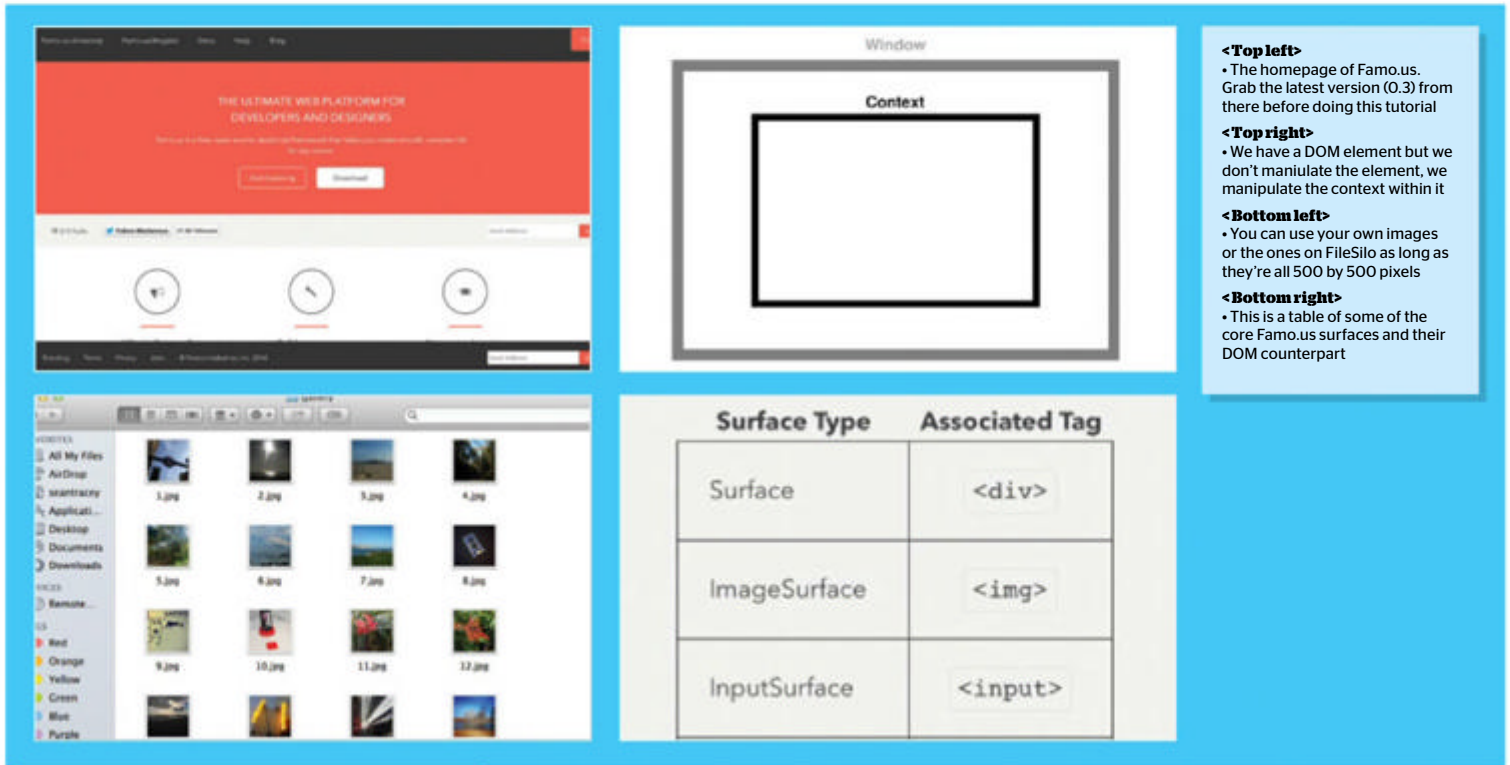
05 The ImageSurface

With Famous, we don't need to create any nodes in the DOM, we create surfaces that correspond to various DOM elements instead. We can then adjust the properties and appearance of the surfaces before we add them to our context. This will enable Famous to optimise element rendering ahead of time. We can create an image surface for each of our gallery images like the snippet below:

```
001 for(var y = 0; y <= rows; y += 1){
002 for(var x = 0; x < columns; x += 1){
003 var img = new ImageSurface({
004 size : [thumbSize, thumbSize],
005 content : '/assets/img/gallery/' +
images[idx],
006 classes: ['pointer']
007 });
008
```

06 The various surfaces

There are seven different types of surfaces for us to make use of. The standard surface equates to a <div>, imageSurface - , inputSurface - <input type="text">, textareaSurface - <textarea>, submitInputSurface - <input type="submit">, canvasSurface - <canvas> and videoSurface - <video>. Note that you can interface with each of the created elements just as you would in any ordinary HTML, JavaScript or CSS web page, but the problem there is that it's not like Famous will get to do its



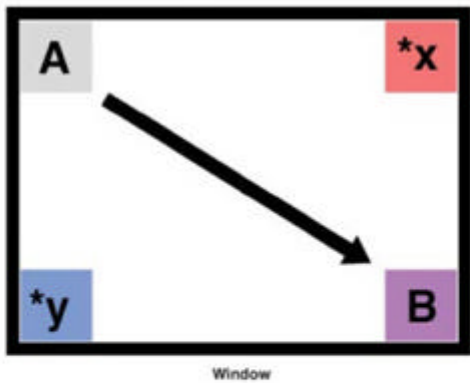
- <Top left>**
 - The homepage of Famo.us. Grab the latest version (0.3) from there before doing this tutorial
- <Top right>**
 - We have a DOM element but we don't manipulate the element, we manipulate the context within it
- <Bottom left>**
 - You can use your own images or the ones on FileSilo as long as they're all 500 by 500 pixels
- <Bottom right>**
 - This is a table of some of the core Famo.us surfaces and their DOM counterpart

Slow DOM
 The DOM was designed to be syntactically understandable. Due to its tree-like nature, any change to one part means that everything else has to be checked and redrawn for changes almost every time.

nifty optimisations. By interfacing with these elements through Famo.us, we're trusting that we are going to get the best performances out of using this method.

07 Understand the modifiers
 Modifiers will enable us to modify the properties of a surface in Famo.us. We can use them to create animations, transform positions, size and skews, and we can also chain them together to generate cumulative effects! That sounds pretty neat huh? Here, we will create two modifiers, one for positioning our ImageSurface in our context, and then another modifier for animating into its rightful place.

```
001 var posMod = new Modifier({
002   origin : [0, 0]
003 });
004 var animateToPosition = new
Modifier({transform : Transform.identity});
005 animateToPosition.setTransform(Transform.
translate(x * thumbSize, y * thumbSize, 0),
{curve : 'easeInOut', duration : 1000});
006
```



08 Animate with modifiers
 Our animateToPosition modifier is how we tell a surface to move about with our context once the modifier has been added to it. First, what we will do is we create an object to access the Modifier method. Next we will then call the setTransform() function to do exactly that: set the transform we want the modifier to perform. The first argument is a Transform object which will describe the transformation, the second argument are the properties for timing the animation. In this instance we are creating a transformation that will last one second, and this will accelerate and then slow down to complete the transform.

09 Piping events
 We're working through a finite amount of images that we want to use to fill our context, and as such, we need to update some simple indexes that will end up making our lives a little simpler:

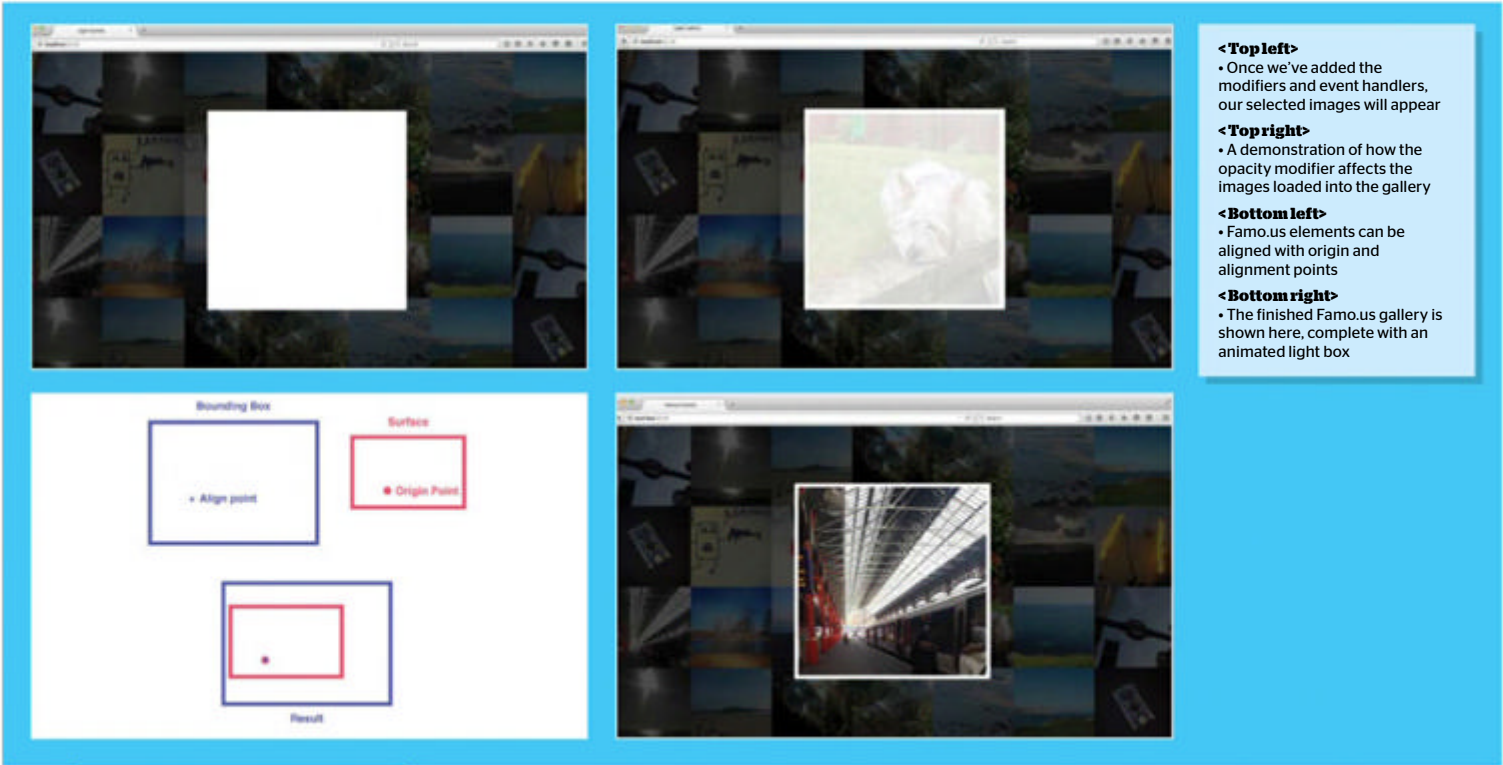
```
001 idx += 1;
002 count += 1;
003 if(idx > images.length - 1){
004   idx = 0;
005 }
006
```

If we run out of new images, start again.

10 Bind to EventHandlers
 When we click or tap an image, we want to see a bigger version of it come up. This sounds pretty simple, just addEventListener(), right? Well Famo.us doesn't use event binders. Instead, Famo.us we will 'pipe' (or pull) events to and from our surfaces. First, we create an EventHandler (which is one of the Global Famo.us objects that we imported at the start), then we 'pipe' that EventHandler through to the surface that we want to interact with, and in this scenario, we want it to interact with our ImageSurface:

```
001 var eventHandler = new EventHandler();
002 img.pipe(eventHandler);
```

11 Create an overlay
 Famo.us EventHandlers can both emit and listen for events. This is great as it enables us to use interactions to trigger behaviours on elements that aren't directly associated with the object interacted with (which is basically everything in Famo.us) and keep our code nice and modular. Binding functions to events is very similar to addEventListener().



- <Top left>**
 - Once we've added the modifiers and event handlers, our selected images will appear
- <Top right>**
 - A demonstration of how the opacity modifier affects the images loaded into the gallery
- <Bottom left>**
 - Famo.us elements can be aligned with origin and alignment points
- <Bottom right>**
 - The finished Famo.us gallery is shown here, complete with an animated light box

Famo.us and a virtual DOM

Famo.us uses contexts and surfaces to keep a virtual structure of how things should be drawn on the screen and in what order. Famo.us uses actual DOM objects to display content, but prioritises the rendering of those DOM elements rather than reducing the number of reflows or redraws needed.

```
001 eventHandler.on('click', function(e) {
// ...Code to do stuff... });
002
```

12 Fade in

Much as when we created our image surfaces, for our overlay we need to create a different kind of surface: a ContainerSurface. Where an ImageSurface is equivalent to an element, a ContainerSurface is a <div> with the sole purpose of containing other surfaces/elements. For our overlay, we need two container elements, one for the semiopaque black background, which goes over all of our gallery images, and nested within that another container to contain the larger version of our image.

```
001 var overlay = new ContainerSurface({
002 size : [window.innerWidth, window.
```

```
innerHeight],
003 properties : {
004 background : "rgba(0,0,0,.7)"
005 }
006 });
007 var focusHolder = new ContainerSurface({
008 size: [500, 500],
009 properties: {
010 background: 'rgba(255,255,255,1.0)'
011 }
012 });
013
```

13 Set surfaces

Earlier, we used transforms to animate the position of an element on our context, but we can also animate opacity and sizes with a similar bit of code. If we create another modifier, opacityMod, we can use it on any of our surfaces, and, when applied to a surface the modifier will cause it to fade in from zero opacity (transparent) to one opacity (opaque).

```
001 var opacityMod = new Modifier({
002 transform : Transform.identity,
003 opacity : 0
004 });
005 opacityMod.setOpacity(1, {duration :
1500});
```

14 Layer it up

Unlike the DOM, Famo.us does not have a concept of flow. Elements are positioned on top of one another without regard for their own position or that of

elements around them. Although it defeats the purpose of Famo.us to implement browser flow, we can use modifiers to control the points from which our surfaces draw themselves and how they position themselves relevant to their parents. In this case, we want everything to be in the middle of our screen. By changing our origin to [0.5, 0.5] we're setting our surfaces to draw from the centre of themselves, and by setting 'align' to [0.5, 0.5] we're setting those surfaces to align their origin points to the centre of the parent surface:

```
001 var centerModifier = new Modifier({
002 align: [0.5, 0.5],
003 origin: [0.5, 0.5]
004 });
005
```

15 Hide overlays

Now that we have all of our surfaces and modifiers, it's time to add them all together so they display properly when we click on a thumbnail. First we want to add the centerModifier to our overlay (so that everything is centrally aligned).

```
001 overlay.add(centerModifier).
add(focusHolder);
002
```

16 Get focused

Next, we add our focusHolder (the white div element) to our overlay, to which we then add the centerModifier (one more time), the opacityMod modifier and then the image. This will create a fading in effect for

our image after we've selected it from the gallery. Finally, we will add the centerModifier with overlay to our mainContext, which will draw it to our screen.

```
001 focusHolder.add(centerModifier).
add(opacityMod).add(imgSurf)
002 mainContext.add(centerModifier).
add(overlay);
003
```

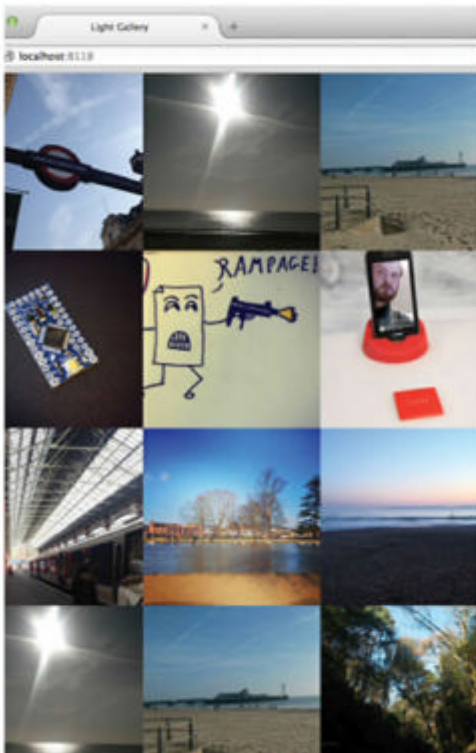
17 Hide overlays

Once our overlay displays, we want to be able to hide it. That's easy, all we need to do is create another EventHandler to delete the element. When the handler is triggered (in this case by clicking outside of the element) we look up the parent container and remove its child element (our white box) from the DOM.

```
001 var exitHandler = new EventHandler();
002 overlay.pipe(exitHandler);
003 exitHandler.on('click', function(e){
004 console.log(overlay);
005 overlay._element.parentNode.
removeChild(overlay._element);
006 });
007
```

18 Full circle

That's it. We've now covered the basics of Famo.us' surfaces, modifiers and properties. We've only scratched the surface here - there are tons of things that Famo.us is capable of doing at superhigh frame rates. Go! Make something awesome.



Famo.us, nesting and the DOM

So, how does Famo.us stop the browser from doing what it was designed to do? By drawing the DOM. If you open your web inspector tools and look at the DOM structure of our gallery, you'll see that almost every DOM node is a *sibling* node to every other node - they exist as a child node only to the Famo.us container, not to each other (unless we do so deliberately). By not being nested within DOM elements, Famo.us can prioritise the order in which things are drawn somewhat, it can also make sure that any kind of drawing that needs to happen only happens in the next available frame (with requestAnimationFrame) rather than halting everything else and trying to force the changes as everything is moving. Hence, we get lovely smooth animations, rather than blocky, stuttery ones we're all so used to seeing across the web.

Code library

Make and destroy our gallery overlay

A lot of concepts can go into making our overlay, so let's go through and point out all of the different aspects

Create a new ContainerSurface with an almost opaque, black background the size of the window it's in. This is for when the image's thumbnail is clicked.

```
001 eventHandler.on('click', function(e) {
002 var overlay = new ContainerSurface({
003 size : [window.innerWidth, window.innerHeight],
004 properties : {
005 background : "rgba(0,0,0,.7)"
006 }
007 });
008
```

Pipe any 'click' event that happens on the overlay to the exitHandler, so the overlay can be closed and a new image selected.

```
009 var exitHandler = new EventHandler();
010 overlay.pipe(exitHandler);
011 exitHandler.on('click', function(e){
012 console.log(overlay);
013 overlay._element.parentNode.removeChild(overlay._element);
014 });
015
```

Create the modifiers and holders that will contain the images on the screen.

```
016 var opacityMod = new Modifier({
017 transform : Transform.identity,
018 opacity : 0
019 });
020 opacityMod.setOpacity(1, {duration : 1500});
021 var focusHolder = new ContainerSurface({
022 size : [500, 500],
023 properties : {
024 background : 'rgba(255,255,255,1.0)'
025 }
026 });
027 var imgSurf = new ImageSurface({
028 size : [480, 480],
029 content : e.target.currentSrc
030 });
031 var centerModifier = new Modifier({
032 align : [0.5, 0.5],
033 origin : [0.5, 0.5]
034 });
035
```

Chain them all together in the order that you want to affect the elements. Remember, every modifier applied affects children applied after it.

```
036 overlay.add(centerModifier).add(focusHolder);
037 focusHolder.add(centerModifier).add(opacityMod).
add(imgSurf)
038 mainContext.add(centerModifier).add(overlay);
039 });
```

web workshop

Build a responsive fixed page border

inspiration studiodyxe.com

1 INSPIRATION

Keeping the content tidy

As with most sites that have scrolling content, images are animated onto the page as the user scrolls to that section of the site and they will then come into focus within the browser window. This keeps maximum impact for the animated elements as they are presented to the user. The site gives the impression of loading content into the same page by animating a div over the top, then calling the next page and removing the div when that page has loaded, removing that annoying flicker that can occur when going from page to page. This effect has been achieved with the animation jQuery plugin.

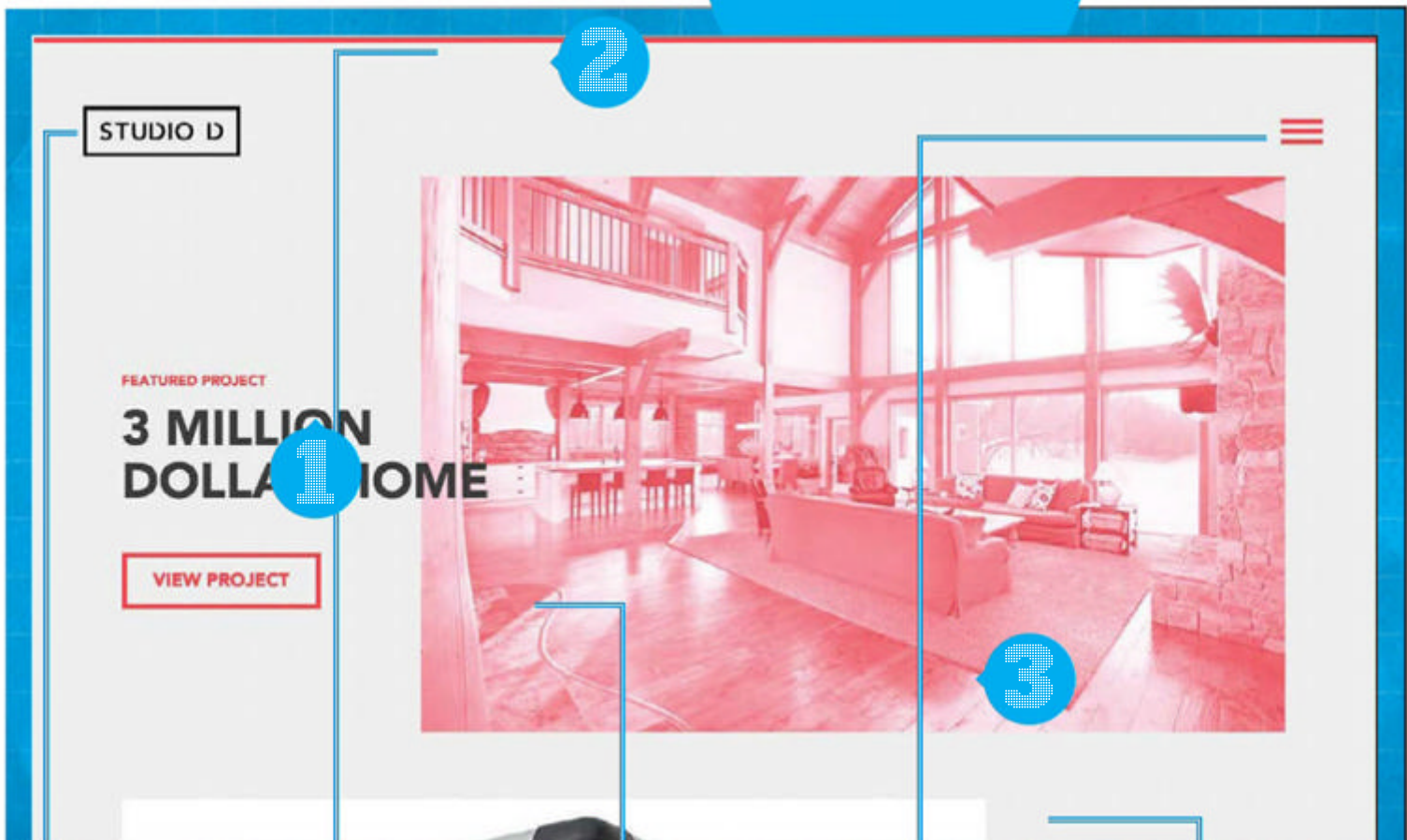
Simple navigation

On the homepage the navigation is hidden in an offscreen menu. As the user moves through content pages, the menu enables easier browsing and there are buttons for moving through the content.



Having a simple and easily defined style is essential for any website and Studio D YXE, or Studio D as they are known has managed to create this effortlessly. The key to achieving this has been to completely reduce the colour palette to a simple scheme of a light grey background, while a red focus colour appears throughout

the site for all of the buttons and menu elements. The red is then reintroduced for a stylised look to the photography. The main text is a very dark grey, almost black colour and with this very reduced palette, all images fit together with the complete look that is stylish, elegant and easily recognisable.



Back to the studio

The site logo found in the top-left corner also doubles up as another way to navigate. Clicking it on any page within the site will lead you back to the homepage.

Responsive border

The border around the edge of the page always fits the page exactly, no matter what size screen it's on and the rest of the content will then scroll below this.

Stand-out shades

The colour scheme of the site is kept to a light grey background, the text to a dark grey and a red accent colour is given to the main image to draw attention to the buttons and text.

Slide-in menu

There is a more traditional and familiar burger menu in the top-right corner of the screen that enables access to an off-screen menu. This menu then slides in from the top.

Active content

Scrolling down the page causes other content to animate into the viewport from the left and the right as the user reaches those sections of the site.



Flexible design portfolio

“The client wanted the capability to create custom layouts for each case study, so we created custom building blocks in the backend where the client could simply check off which elements they wanted for a given piece. This resulted in each portfolio piece looking unique while keeping a beautiful aesthetic.”

Dustin Weeres, creative director/owner, Island Collective

<comment>
What our experts think of the site

3

TECHNIQUE

Create the red images

On the website the images have been treated so that they are all similar. This is good for the web because there is less colour information so images will load faster.

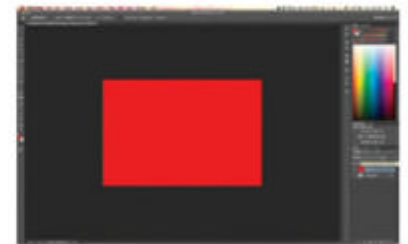
01 Convert to greyscale

Rather than converting it to greyscale and then back to RGB, just press Cmd/Ctrl+U to open the Hue/Saturation in Photoshop once you have selected your image. Drag the saturation back to lose the colour.



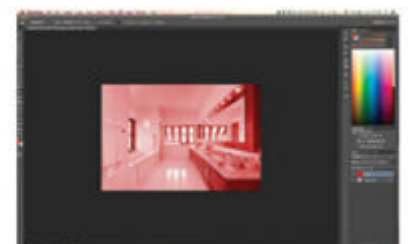
02 Add the colour

Create a new layer and make sure this is above the first layer. Choose a red colour or a colour of your choice and fill the new layer with this colour. By affecting the blending mode we will easily be able to get the look that we want.



03 Finish it off

Change the blending mode in the layer panel to Color and reduce opacity to 65 per cent. This lets some of the darker colours through so that you get a good tone. Now save this for web as a JPEG.



2

TECHNIQUE

The responsive border

01 Body content

The Studio D site has a great border that fits around the edge of the site and this is always there, no matter what content is on the screen. To create this, firstly an empty div tag is required with the class of 'wrapper', this will contain the border.

```
001 <div class="wrapper"></div>
```

02 For testing only

In order to test that this works as it should, we need some page content that's longer than the actual page. Here another div is added and this will be given a long height so that later on, the border can be seen working around the edge of the page.

```
001 <div class="long">A long piece of content</div>
```

03 Add the CSS

In the head section of the page, add an opening style tag so that the CSS can be applied to the elements. The first styling will be for the page itself. Setting the height to 100% and the background to a similar colour to Studio D helps get the look. Also a large padding around the edge is very useful.

```
001 <style>
002 html, body{
003 height: 100%;
```

```
004 heroText">
005 background-color: #f0f0f0;
006 padding: 20px;
007 }
```

04 Style the wrapper

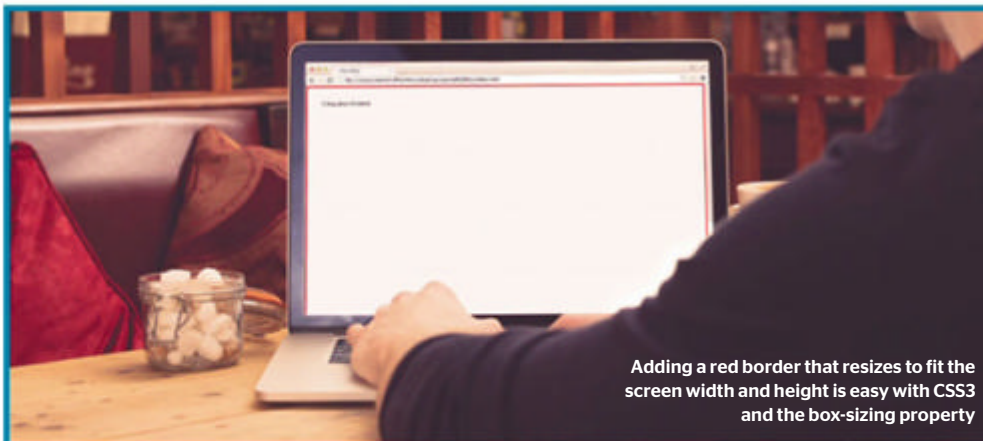
The wrapper class is going to contain the border, so it is set to be fixed on the page in the top-left corner. It's given a z-index higher than other page content and then made to fit the width and height of the page. The box sizing is set to include the border, then a 5px border is added all round.

```
001 .wrapper{
002 position: fixed;
003 z-index: 999;
004 top: 0; left: 0;
005 width: 100%;
006 height: 100%;
007 box-sizing: border-box;
008 border: #ff0000 solid 5px;
009 }
```

05 Finishing off

The class of 'long' is only here for testing purposes and it's given a height of 2,000 pixels which is larger than most monitors. Save this and test in the browser to see the red border always on the screen and fitting neatly into the browser window, even working responsively.

```
001 .long{ height: 2000px;}
002 </style>
003
```



Adding a red border that resizes to fit the screen width and height is easy with CSS3 and the box-sizing property

web designer SPECIAL SUBSCRIPTION OFFER

**SUBSCRIBE FOR 1 YEAR
& GET THE
2ND YEAR FREE**



**HALF
PRICE
OFFER**



CONTEMPORARY FEATURES
WRITTEN BY EXPERTS



BEHIND THE SCENES WITH THE
WORLD'S BIGGEST AGENCIES

Order hotline
0844 848 8413

Online at
www.imaginesubs.co.uk/wed



BUILD A BRAND

The key considerations for anyone starting a new brand, reinventing an existing one, or helping their client work to establish theirs

Use data to improve

Unless you're using analytics to bolster your content efforts, it's likely it'll fall on deaf ears. Data enables you to figure out what's working well, and not so well.

The playing field for brands is changing. We're now gradually seeing a shift, from a world of fear-based selling to a more positive, innovative place where brands talk to shoppers as people on the same level, as opposed to treating shoppers like they are minions following the herd.

While this may sound complicated to some, in reality, it's far simpler than ever before – because at the heart of this new approach is honesty and integrity as well as a desire to do what's right.

A brand is so much more than a logo and a typeface. That's merely the visualisation of it – and our role as web designers is expanding, as more people are using the web to define who they are, and not just as a shop window. The problem is that many brands can't define who they are.

1. Finding your 'Why'

"People don't buy what you do, they buy why you do it" says author and marketing and branding expert Simon Sinek. His TED Talk about 'Why' is one of the most important thought pieces of our generation when it comes to brand building.

Understanding your true core – the essence of why you do something – is as important for your customers to know as it is for you. It's astonishing how many brands think that the root of their business is 'to make money' But as Simon says: that's merely an outcome. You have to go deeper, down into the human psyche... and much like discovering who you are as a person, once you find that 'true north' inside you and your brand, it won't feel like you have to come up with ideas, you'll begin to naturally develop them through your own intuition.

Your 'Why' should be something that can be boiled down to a single sentence or phrase. It won't be for example: 'We provide expert premium architectural services to builders and planners throughout the UK'. That's *what* you do. A 'Why' might sound something more natural to a human way of speaking. For example let's look at the following: 'England is naturally beautiful, but buildings keep changing that. Every building we design considers its surroundings first in order to be in true harmony with nature.' This could then be boiled down to something simpler still. For example: 'Surroundings first'.

The difference here is obvious. To some it may sound fluffy, but if you truly care about your work, you'll care about the environment too. Would you rather work with a company whose primary focus is to make money?

To get to your 'Why', you simply keep asking yourself 'Why?' until you get to it. This typically will take around five iterations – but be patient, it'll come.

So we start with:

"We design buildings."

Why?

"Because it's what we do."

Why?

"Because we like doing it."

Why?

"Because we do..."

Why?

"Because... the landscape is damn ugly in places, and we wanted to make it look more beautiful...!"

Ah ha! Now we're getting somewhere.

2. What you do

So you're now clear on why you do something, the next item on the agenda is to know, with complete clarity, what you do. Saying 'we design buildings', isn't clear enough. Be more specific and ensure that every time you answer, it's in exactly the same way. Who do you design for; what types of buildings do you design? Are they specialist? What's the focus? And how can you say you're contributing in some way?

3. How you do it

This again, isn't as obvious as answering 'by placing bricks and mortar together'. It's about how you carry out your 'Why'.

So the question really becomes: 'How do you ensure landscapes are more beautiful places?' The answer might be: 'Through considering the environment first and foremost, before designing buildings with particular materials in mind that are sympathetic to the area. We also aim to solve the problems the customer is experiencing – lack of space, not enough natural light and so on.'

4. Who is your who?

Know your customer better than they know themselves. Create profiles on them, know their names, shopping habits, where else they buy, what they do for holidays and what books they read. But above all, be respectful. You'll find that many consistencies begin to emerge but only by fully understanding your audience can you join in their conversation and be sure you remain relevant – which in today's climate is vital.

“In a crowded marketplace, fitting in is a failure. In a busy marketplace, not standing out is the same as being invisible”

Seth Godin, author, speaker and marketer

TO FOLLOW

Simon Sinek

@simonsinek

Simon's theory of what makes brands great is clear, simple and calls for us all to realign our understanding of what a brand should be today.

Five brand essentials to consider



Brand refresh

You should really refresh every year. Most brands only consider it every three to five years, but customers are moving so quickly that you can't afford to stay the same for too long.



Social media

Love it or hate it, 72 per cent of all internet users are reportedly active on social media of some kind. So give strong consideration to how you'll be relevant on social. No more dormant Twitter accounts please!

SOURCE bit.ly/1j5P0tB



Search

Research is started with a search query by 48 per cent of people. Great content goes a long way, but be specific; talk about what matters and engage with your audience in a way that's relevant to them – not just about what you want to talk about.



eCommerce

According to the Office of National Statistics, 74 per cent of all adults bought goods or services online in 2014, up from 53 per cent in 2008.

SOURCE bit.ly/1oD7v8K



Mobile browsing

Mobile web surfing accounts for 52 per cent of our time according to Comscore. The rate at which its growing is faster than desktop use.

SOURCE trcn.ch/1ABeHuJ

Who else is out there?

LOOK AROUND, LEARN FROM OTHERS AND YOU'LL SAVE YOURSELF FROM HAVING TO REINVENT THE WHEEL

Who are you up against?

You may not care about the competition, or perhaps you're lucky enough to be the first in your market, but more often than not, it's imperative that we know who we're up against. But why? Obviously you don't want to copy everyone else, you need to sit apart - which is why it's critical you keep an eye on what the others are up to. By discovering the rule book they follow, and completely understanding the rules within it, it enables

you to be more creative, and use those rules as your springboard for inspiration.

So find out where competitors are positioning themselves. What's their unique offering? Who are they talking to? Is it the same people as you? How do they communicate; what's their tone? What's their price bracket? And above all, what's their service level?

Where possible, become a customer. Experience their service, quality and value. When you shell out your own

cash and expect a certain level of return, you'll soon see any problems that exist for you that may also exist for other people.

Where it's not possible to become a customer, pick the top ten competitors (or as many as you wish) and produce a simple strength and weakness chart for each one. You may find that many of the brands on there have the same items, but if there are some consistent weaknesses between them then you've found your opportunity to capitalise. It's also useful to define what each competitor's primary message is and how they compare and contrast. Take cars for example. Volvo has traditionally been about safety; Alfa Romeo about beautiful design, whereas Skoda's is value.

“It's also useful to define what each competitor's primary message is and how they compare and contrast”

Case study

TOMS

TOMS.COM

Founded in 2006, Blake Mycoskie created TOMS in a bid to provide shoes to impoverished children in Argentina, following his trip there. His model was simple, buy a pair of shoes, and we'll give one to a child in Argentina. The concept exploded and 10,000 pairs of shoes were sold within the first six months with the equivalent amount being sent to abroad to children.

Homepage

Whereas most eCommerce retailer's homepages are adorned with traditional marketing messages (sales, new collections or launches and so on), TOMS takes a completely different approach. Their core message is their One for One statement and introduces the customer immediately to the concept behind the brand.

Newsletter sign up

TOMS 'Why' flows consistently through everything, including newsletter sign-ups. This box (when visiting the US site) immediately presents you with a question, involving you immediately in a type of discussion; it's an opportunity to 'buy into' what they're about.

Discover 'Why'

Purpose at your brand's core breeds consistency, authenticity, and uniqueness - and it's a surefire way to connect with the right people at the right time.



What makes you different?

What are you famous for? Once you know what your competitors say about themselves, it helps you to find your own niche. What do you bring to the table that others aren't bringing, or alternatively, how are you doing it better? This should feed from your 'Why' - so go back to it. In the case of the architectural practice that was described earlier, they may choose to talk about their passion for the environment and how their work may feed from nature's design.

When you clearly understand your own purpose, and why you do what you do, you'll naturally be different. In the same way all individuals are unique, your natural approach to what you do will shine through. It's up to you to discover this and use it to your advantage.

Be useful

It's not enough in today's market to just be a clone of other brands. Bring something new and original to the table. Brands like Warby Parker in the US have dedicated themselves to donating to charity each time a purchase is

“[A ‘Why’] is the constant burning light that fires us up and moves us forward.”

Peter Docker, founder and MD of Why Not Unlimited



made of their glasses; that's more than marketing, that's a brand with real purpose.

What's your story?

This doesn't necessarily have to be in words. Brands are difficult to define, they're about feeling. Try asking: how do you want people to feel? Take Waitrose for example. They do the same job as ASDA - they're both essentially grocery stores when we get down to the basics. They both have large shops, you push a trolley around and pay at a checkout, they both offer value,

mid-range and premium product ranges and they both have green signs. But what does Waitrose have to offer that ASDA doesn't? How do you feel when you walk into Waitrose compared to how you would feel when you walk into ASDA?

Find your core message, and stick to it. Keep on the same track and never deviate. You can tell the story in many different ways, but it needs to be the same story. In doing so, customers will come to recognise you as *that* brand - be it for quality, design, environment, philanthropy or efficiency.



Simplicity throughout

The navigation is as simple as their message. Just two primary links sit within the top masthead; Shop and 'How we give'. Nobody in their right mind would find this website difficult to use, which reinforces the understanding that TOMS is a brand with ease and simplicity at its core.

Earn rewards

To get more from the TOMS brand, users can sign up for Passport Rewards. Earn stamps for shopping, connecting to social media and attending events. This makes the user feel even more part of a brand they love.

SET YOUR GOALS

Goal setting should be very structured and followed diligently by all key stakeholders. Start with a ten-year goal and think the bigger picture, like turning over £5 million in ten years, to win five big, industry-recognised awards, to have ten per cent of the market - then work backwards eventually to smaller goals that will increase year on year and what you want to achieve this year, next year and so on. Write them all down and break down what that means for the next ten:

- Year 1: £100,000 turnover
- Year 2: £500,000 turnover and one award
- Year 3: £750,000 turnover, one award and one per cent of the market and so on.

Next, work back again this time working in quarters. Months are too short to see any effect of change, whereas a year can be too long; 90 days enables you to see real change. Set minigoals that are in line with achieving the annual goal and distribute them to the key stakeholders in the business for that quarter.

Hold weekly meetings like clockwork to ensure everyone is on target for completing their minigoals. Offer help and support to those that don't look as though they will achieve their's - ask what they need in order to make it happen.

Recommended reading:
Traction: Get a Grip on your business
by Gino Wickman

Build a brand

A brand in action: Mr Porter

www.mrporter.com

Forget the bottom line

It's tough to think about sometimes, but too much emphasis on cashflow can take the soul from your brand. Go with your gut, and if it doesn't feel right, chances are: it's not!

It's brands like the online male fashion retailer Mr Porter that make the branding game look effortless.

They're at the top, and deservedly so, because they know the consumer comes first. Economy experts like *The Financial Times* sing the website's praises and how well its digital presence works in conjunction with the brand, calling it "emblematic of the new order in men's fashion, an online space for men who are informed enough to throw together their own ideas about how to dress one click at a time". And it shows: they get a total of 2.5 million monthly unique visitors, and more than 25 million monthly page views.

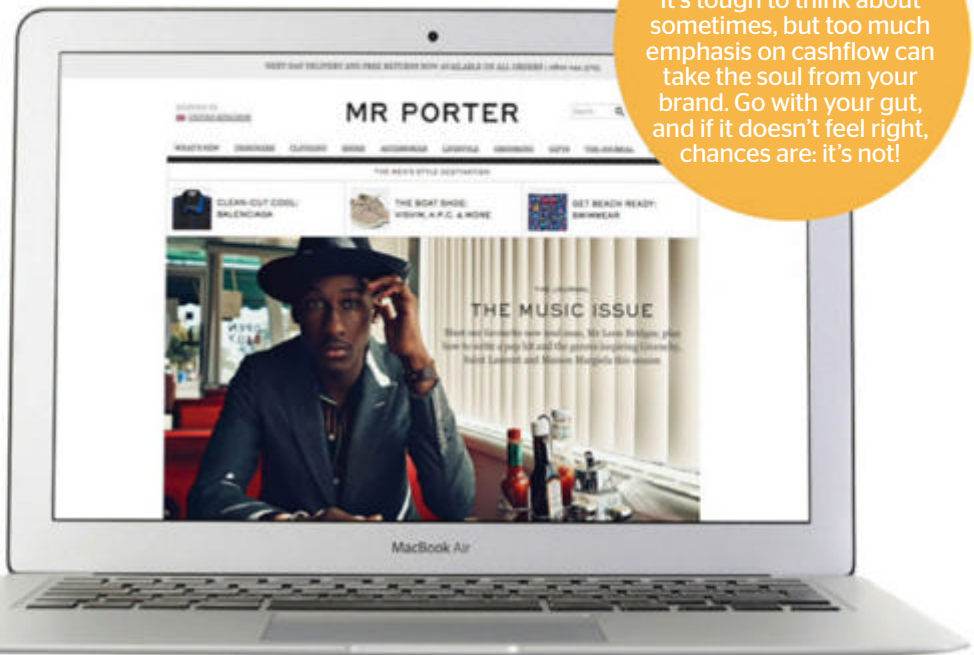
They empower men to make their own decisions quickly, easily and with incredible advice. They're authentic, consistent, cool and targeted. Just take a look for yourself if you're not already convinced.

TO FOLLOW

Dave Trott

@davetrott

Advertising legend and author of several books including the brilliant *Creative Mischief*, Dave Trott is from an old style of advertising where bold, blunt speaking works. His blog, *Predatory Thinking*, is less about advertising and more about intelligent thinking that makes brands succeed.



THE WEBSITE

Mr Porter's website is clean, stylish and easy to use. Bold, carefully kerned sans serif titling lends itself to the Wall Street, looks-conscious gents who use the site. Couple that with body copy that quietly hums of *Times*, and you've got yourself a business-time editorial site that's unpretentiously professional, and aspirational too. Content is never patronising, and speaks as a high-end tailor might speak to his client; knowing, sophisticated, and maybe a little tongue in cheek.

Five ways to stand out

THE ESSENTIALS FOR ESTABLISHING YOUR BRAND'S LOOK AND PRESENCE

Creating a strong brand requires a strong, visually striking and memorable appearance. It pays to be bold and to go all out to ensure that you stand out from your competitors. Below are five key considerations and ways they can be applied:

1. Tone of voice

It's often an overlooked area and left to the end of any branding project when someone asks 'Oh, can you write some copy for the website?' but copy can be the making or breaking of a business. It's very

much like judging the sort of conversations to start at a dinner party. With your mates, you'll ask about the latest date or drunken tales, whereas dinner with the in-laws requires a very different tact. Always keep the audience in mind and who you're talking to - but this doesn't mean it has to sound forced. Again refer back to the 'Why' that you created at the start and draw from there. Be authentic; tell the truth and you never have to remember anything. Finally make sure that you are consistent. From your website and emails to brand packaging and customer service teams verbal

communication style - the same message and tone of voice needs to flow through it all.

2. Typeface

This goes hand in hand with tone of voice. Once you've decided how you'll speak, how will that look on screen? How will the font represent your message? If your general vibe is light-hearted with a sense of humour, you wouldn't select a blackletter typeface. Again referring back to the dinner party analogy, if the tone of voice is the type of conversation you have,

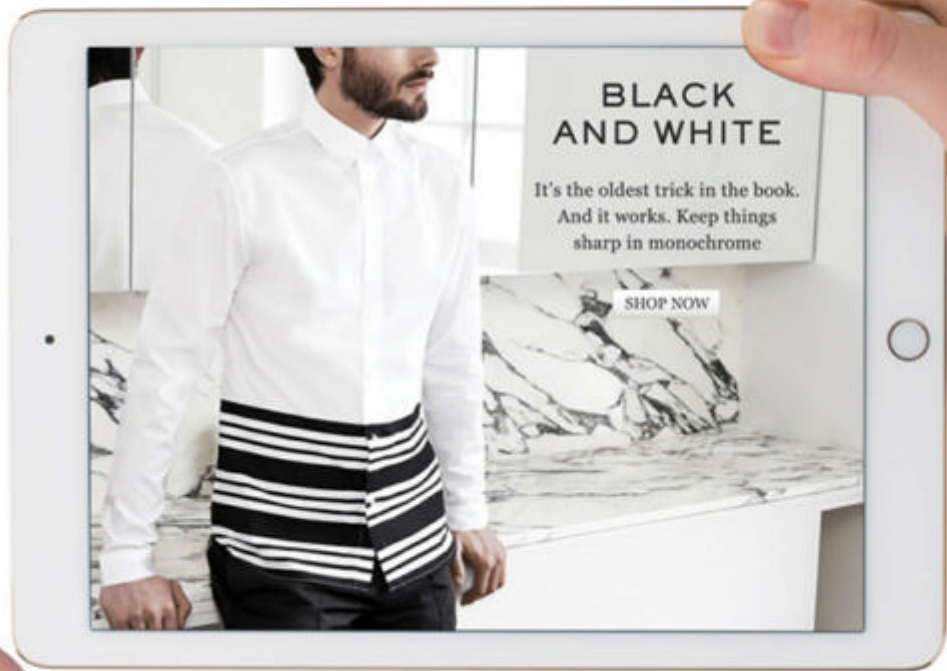
EMAILS

Mr Porter emails are enviably consistent with their website messaging, and everywhere else. You can simply tell it comes from a united source. Content is useful, engaging. It's not so copy heavy that it's taxing, but there's enough to build substance.

SOCIAL MEDIA

The key thing about a switched-on brand is that they know what their customers like.

That's why you'll find lamb shanks on Mr Porter's Facebook page and luxury getaways on their Twitter. Oh, and their tone of voice is impressively consistent the whole way through.



COLOUR

Mr Porter set themselves apart by knowing when to use and when not to use colour. Black and white leads the way across the entire brand universe, but the key is in the photography. A black and white scaffold lets any and every colour way shine through; it's not overpowering, and everything feels at home, whether it's a print, check, block or stripe.

PACKAGING

The visuals offline match the visuals online. Sounds simple, right? There are so many channels nowadays that it's easy to lose track of the smaller details. But these experts are on point. Not a space out of place. And it looks and feels as refined as the site to boot. The service is also exemplary; if they sound like a gentleman's outfitter, they should feel like a gentleman's outfitter. So, expect personal courier service to beat the Christmas rush, and presentation that makes you feel like a million dollars.

selecting the right typeface is dressing appropriately for the occasion.

3. Visual style

You have a beautiful logo, but visual style doesn't end there. What about photography? Illustration? A brand pattern? Find something simple and try it out with your brand on various different platforms. Will it work online, in print, in black and white, on a billboard from a distance? Think of O2 as a key example here. Their blue gradient and bubbles style is recognisable no matter where it's seen without having to see their logo or any writing.

4. Colour

Following on from visual style, pick a colour palette that won't date easily. Choose carefully and stick with classic colours, not 'of the moment' colours. Right now

there's a trend towards neons, largely thanks to the Nineties renaissance in dance music. Come next year, it'll have moved on and if you're stuck with 10,000 lurid day-glow tote bags that nobody wants, your finance people won't be pleased when you ask for additional budget to replace them with white ones.

5. Service

Above all, offer amazing service, and look after your customer. They're not coming to you just for your wares. The greatest identity and design work in the world can't save you if your customer service and delivery is shoddy and ill considered. You need look no further than eBay for proof of this. Nobody wants to deal with someone with a two-star rating, even if their eBay shop is beautiful. Be helpful and use every customer encounter as an opportunity to learn more about your audience.

Keeping track

A lot gets said about big data, but simply paying attention to the small details of what people are doing can be as important insight to your operations as any large numbers. Returning back to the rocks you set in the 90-day plan, see how they can be directly related to online activity. If one of the objectives was to reduce the number of returns on your eCommerce clothing store by ten per cent in Q1, and a proposed solution was to provide a more detailed sizing chart for customers to work out their exact size, then track that figure constantly to see if it solves the problem. If not, by how much did it change? Did it reduce at all? What other solutions could you try? What are you customers doing?

Manage your project's dependencies with jspm

Master frictionless package management and learn how to manage and bundle your projects

tools | tech | trends jspm, npm, JavaScript, Terminal.app
expert Tim Stone



For the past few years we've found ourselves jumping through hoops to bring a robust module system to JavaScript. Tools like RequireJS and Node paved the way in popularising modules and approaches like AMD, UMD, and CommonJS. ECMAScript 6 (the next version of JavaScript) promises to unify all of these

approaches with native modules.

Jspm provides a way to use these technologies today as well as all of the features you'd expect from a package manager. Jspm uses SystemJS to load ES6, AMD and CommonJS modules, this then works in tandem with Traceur or Babel to compile this ES6 code to ES5, which all modern browsers can run (as well as IE 9+). Jspm is a command-line tool so some prior experience with using Terminal.app or cmd.exe would be beneficial. Don't worry if you haven't written any ES6 prior to this, we'll explain as we go.

We're going to use Google Maps to create a map, load some data, and plot some markers on the map. Through this you'll learn some real-world examples of how ES6 modules can help decouple your code.

01 Install jspm

Jspm is installed via npm, this is a little strange as you're using a package manager to install a package manager. We'll head to nodejs.org to install it and do this globally so that it's available throughout the system.

```
001 $ npm install -g jspm
```

02 Set up the project

With jspm installed we'll create a new folder for our demo and navigate into it. The jspm CLI has an init command which asks a few questions about the structure of your app, you can likely just hit Enter to select the default. We've chosen to use the Babel transpiler over Traceur.

```
001 $ mkdir jspm-demo && cd jspm-demo
002 $ jspm init
003
```

03 Download data

The init process will create a few files for you, including package.json and config.js. You should rarely have to edit these directly as jspm does most of the maintenance of these files for you. We're going to be using this data source from http://bournemouthdata.io/data/buildings_listed_buildings.csv which you'll need to download a local copy of under a folder called 'data'.

04 index.html

Below is a reduced version of our index.html file. Jspm uses SystemJS to do the actual in-browser module loading. We then point the browser to our configuration file which passes important information to SystemJS like where to look for our modules and what to map the names to. It is this mechanism which enables shorter names than 'github:jquery/jquery'.

```
001 <!doctype html>
002 <title>jspm</title>
```

```
003 <div class="map-container"></div>
004 <script src="jspm_packages/system.js"></script>
005 <script src="config.js"></script>
006 <script>
007   System.import('app/main');
008 </script>
009
```

05 Import a module

Create a folder called 'app'; this will house our main JavaScript files. In a new file called main.js write the code below. We're making use of ES6's module loader as we're effectively saying 'I know that there's a file called Map.js at this location, import the Map export that it exposes'. Then we instantiate a new instance of it.

```
001 import { Map } from './Map';
002 new Map();
```

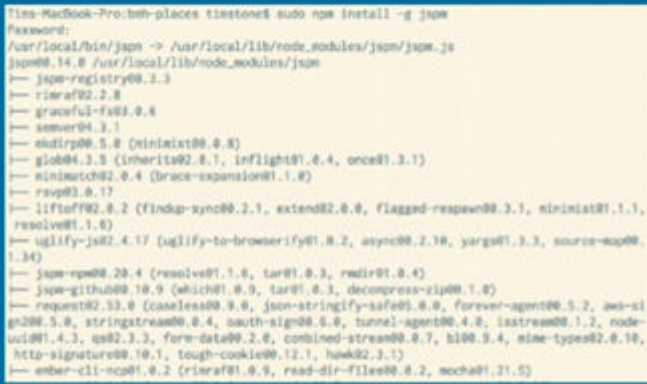
06 Install Google Maps

At the time of writing Google Maps does not provide a module-compliant API but sakren has written a wrapper for this so we can still use it within our app. Jspm's install mechanism is fairly flexible, it can look at its own internal registry, GitHub and npm. Note that we're providing a shorthand name of 'google-maps' to this.

```
001 $ jspm install google-maps=github:sakren/node-google-maps
```

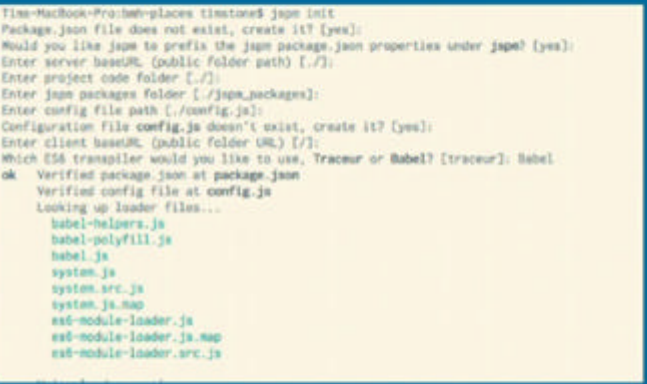
07 Import modules

We'll import the GoogleMaps module we just installed and we'll also import a file called keys which we'll write ourselves. Note that when we reference a file on the system we prepend it with '.', this tells the loader where to look relative to this file. Once imported, default modules can be named anything you wish.




<Above>

- Installing jspm is a cinch through npm. We're installing it globally as you'll probably want to use it across projects



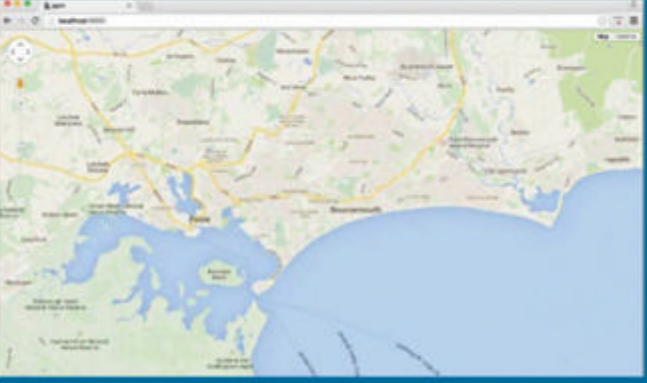
<Above>

- Initial setup of jspm is painless thanks to init as it configures your project to use package management



<Above>

- The only manual loading that actually needs to be done within the HTML here is configuring SystemJS



<Above>

- With the Google Map module installed and a little configuration we've got a map of Bournemouth rendering!

Debug with source maps
 Babel includes source maps for the generated files so you can debug from within the browser without having to interpret any of the transpiled output.

```
001 import GoogleMaps from 'google-maps';
002 import keys from './keys';
```

08 Store keys
 Create a file called 'keys.js'. We're going to use a different method of exporting this module just to show how flexible jspm is. This is the CommonJS way (which is probably familiar to you if you've used NodeJS before). We'll put any keys that we need here, and in this demo it's only the Google Maps one.

```
001 module.exports = {
002   googleMaps: 'YOUR_API_KEY'
003 };
```

09 Map class
 Our app has had an error because we've said that this file will export something called Map but so far we haven't. We'll rectify this by adding the following snippet to the bottom of Map.js. This creates an ES6 class called Map.

```
001 export class Map {
002 }
```

10 Constructor function
 ES6 classes have a function called constructor that is run when the class is invoked. Within a class we don't have to use the function keyword as it's denoted by the parentheses. We're setting values for the GoogleMaps library and then loading; the function passed to 'load' will be called once it is loaded.

```
001 constructor () {
002   GoogleMaps.LIBRARIES = ['places'];
003   GoogleMaps.KEY = keys.googleMaps;
004   GoogleMaps.load(this.createMap.bind(this));
005 }
```

11 Create the map
 When Google Maps loads we call this function which will instantiate a new Google Map instance within the element provided and configured to be

<Above>

- Our first couple of markers have been added but the geocode service hasn't found many results from the dataset of 251

<Above>

- The Google Places API has fared a little better, but there are still many points that haven't been plotted unfortunately!

<Above>

- Our map is complete and the markers animate into place, you could take this further showing a window per marker

<Above>

- The output from 'jspm inspect' is clear and concise and is certainly beneficial when familiarising yourself with a project

Learn ES6

Babel has a very good guide for learning about ES6 features at babeljs.io/docs/learn-es6 but having a practical application is a good way to learn.

centred on Bournemouth. We invoked this with `.bind(this)` because the value of 'this' changes within the callback to 'undefined'.

```
001 createMap () {
002   this.map = new google.maps.Map(
003     document.querySelector('.map-container'),
004     {
005       center: {
006         lat: 50.720806,
007         lng: -1.904755
008       },
009       zoom: 12
```

```
010   }
011   });
012 }
```

12 Load buildings

As well as the load callback there is also the `.onLoad` event, which you can pass a function to, that is fired after the initial load callback. In a similar way to before, we're going to call a method called `getBuildings` with the current value of 'this' which points to the Map class itself.

```
001 GoogleMaps.onLoad(this.getBuildings.bind(this));
```

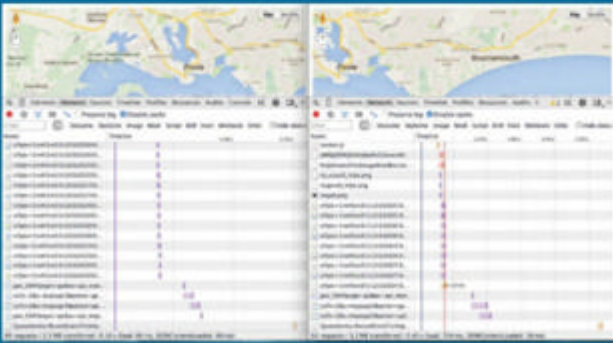
13 Import GetData

To help separate our application a little we're going to import a module to make the request for us. The `GetData` module we'll write will have a 'default' export as opposed to the named exports that we've seen previously (with the curly braces). Modules can export both default and named modules.

```
001 import GetData from './getData';
```

14 Install Papa Parse

The data that we want is in CSV format so we'll use a handy library called Papa Parse which specialises in all there is to know about parsing CSV.



Does jspm add value to your project? Part one

There is no doubt that this approach requires more code to be downloaded and executed than if you were to simply write this in ES5. The fact is that it'll be years before these features are natively supported in enough client devices to make this viable in production. Even if you exclude IE there are still many mobile browsers that may never support it. You have to make a judgement on whether the flexibility this approach gives you works with your product or client. Once minified and concatenated, our project goes from 89 requests at 2.5MB to 74 requests at 1.2MB (this includes everything that Google Maps loads and .map files which inflates the number of requests a lot). This is a good starting point for long-term projects and will help introduce you to concepts that'll help you in the future.

Package.json for jspm
The configuration options for managing all of the modules in your project are extensive and a great explanation of how they work can be found on GitHub at bit.ly/1DJOKyk.

We install it with jspm and it's hosted on GitHub so we specify the username and repository name of the project.

```
001 $ jspm install github:mholt/papaparse
```

15 Import Papa Parse

Without having to include anything through HTML we're able to import the newly installed Papa module. We then create an object called GetData with a csv method, which will return the Papa.parse function and execute the passed callback. Papa Parse splits the data out for us and also deals with making the AJAX request. Note the 'default' keyword.

```
001 import Papa from 'PapaParse';
002 var getData = {
003   csv: function (url, callback) {
004     Papa.parse(url, {
005       download: true,
006       complete: callback
007     });
008   }
009 };
010 export default getData;
```

16 Get the buildings

We then use it as follows, we pass the location of the file we wish to download and the function to execute on completion. Again we have to bind the current value of 'this' to the function and when we map out the data within 'process' the second argument effectively does the same thing as .bind(this).

```
001 getBuildings () {
002   function process (results) {
003     results.data.map(this.plotLocations, this);
004   }
005   GetData.csv(
006     '/data/buildings_listed_buildings.csv',
007     process.bind(this)
008   );
009 }
```

17 Geocode results

The data in the file gives us a name which looks like 'Albert Road, Former Theatre Royal', but to place a marker on the map we need to know its latitude and longitude values. To do this we need to use the Google geocode service which comes with Google Maps.

```
001 plotLocations (result, index) {
002   var geocoder = new google.maps.Geocoder();
003   geocoder.geocode(
004     {
005       'address': result[1]
006     },
007     );
008 }
```

18 Geocoder response

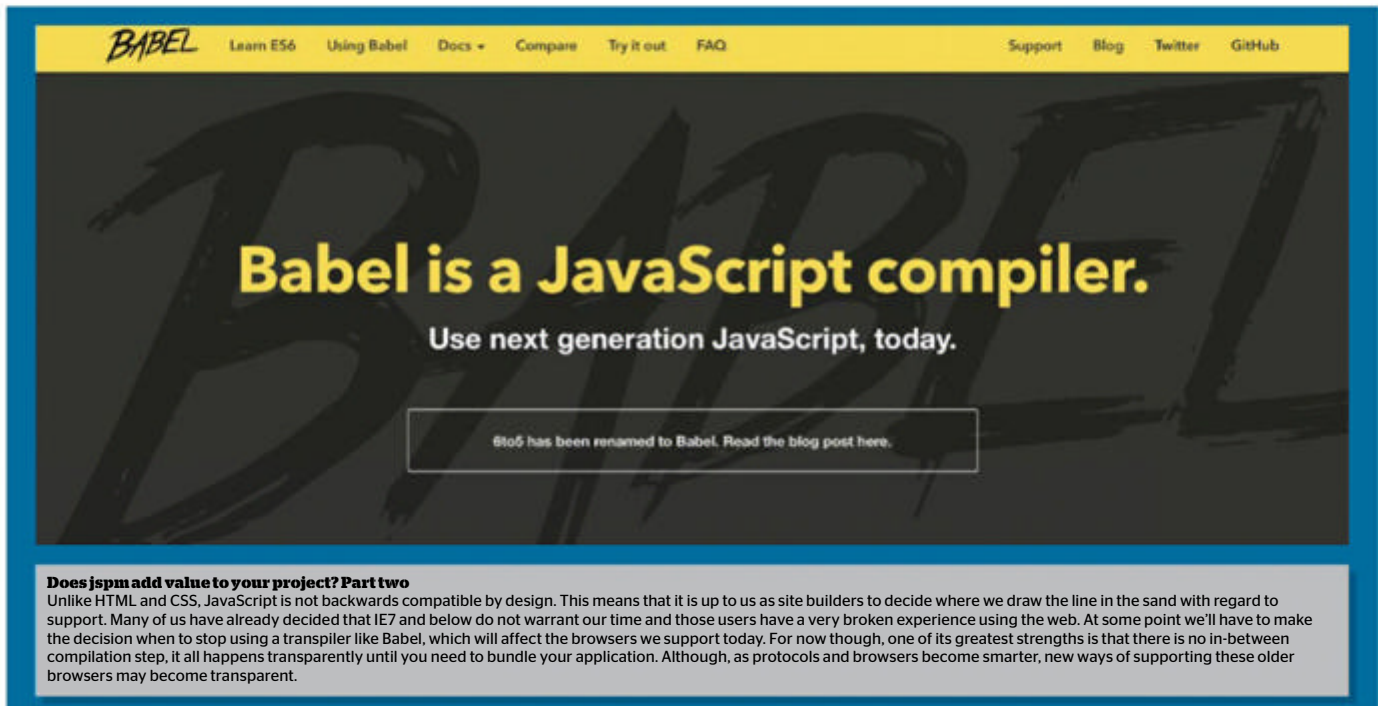
If we get a result back from the geocoder then we immediately create a new marker (we'll write this method shortly). If it doesn't find a result then we'll try to find one using the Google Places API which is more advanced than the geocoder. Google limits how many places results we can make in a second so we stagger the requests with 'setTimeout'.

```
001 function (place, status) {
002   if (status === google.maps.GeocoderStatus.OK) {
003     this.createMarker(result[1], place[0].geometry.location);
004   } else {
005     setTimeout(
006       this.searchPlaces.bind(this), 100 * index, result
007     );
008   }
009 }.bind(this)
```

19 Places configuration

SearchPlaces tries to find a location matching the passed query (eg 'Alum Chine Road, Westbourne Library') within a five-mile radius of the latitude and longitude passed to it (in this case, Bournemouth). By specifying this we ensure we don't get results for other countries! This is the initial configuration and the next step is the callback function.

```
001 searchPlaces (result) {
002   var places = new google.maps.places.PlacesService(this.map);
003   places.textSearch({
004     query: result[1],
005     location: {
006       lat: 50.720806,
007       lng: -1.904755
008     },
009     radius: 5
010   }, /* next step */
011 }
```



Does jspm add value to your project? Part two

Unlike HTML and CSS, JavaScript is not backwards compatible by design. This means that it is up to us as site builders to decide where we draw the line in the sand with regard to support. Many of us have already decided that IE7 and below do not warrant our time and those users have a very broken experience using the web. At some point we'll have to make the decision when to stop using a transpiler like Babel, which will affect the browsers we support today. For now though, one of its greatest strengths is that there is no in-between compilation step, it all happens transparently until you need to bundle your application. Although, as protocols and browsers become smarter, new ways of supporting these older browsers may become transparent.

20 Google Places response

If Google Places finds somewhere that matches then we just take the first result (there could be options) and we create a marker from that. Otherwise we output a message to the console. This uses ES6's string interpolation feature - a much cleaner way to embed variables in strings.

```
001 function (place, status) {
002   if (status === 'OK') {
003     this.createMarker(result[1], place[0].geometry.location);
004   } else {
005     console.log('Could not find ${result[1]}');
006   }
007   }.bind(this));
```

21 Create markers

Now write the createMarker method. This creates a new Google marker with a title property so when the user hovers over a pin they can see what is there, and we also animate the pins to drop onto the map.

```
001 createMarker (title, position) {
002   return new google.maps.Marker({
003     map: this.map,
004     title: title,
005     animation: google.maps.Animation.DROP,
006     position: position
007   });
008 }
```

22 Bundle for production

Our app is finally code complete! Unfortunately with all of those modules being loaded it's making far too many requests. Jspm has a 'self-executing' bundle process which combines and minifies all of the ES6-to-ES5 features and all of the modules into a single streamlined bundle.

```
001 $ jspm bundle-sfx app/main --minify
```

23 Not just JavaScript

The frontend isn't all about JavaScript, though. Jspm enables you to also install CSS files like Normalize or Bootstrap too. This will then include the CSS-loading plugin which does add some overhead. Something to note is that CSS ordering is not guaranteed so modular styles must be 'name spaced', either by a base class or unique ID.

```
001 $ jspm install normalize.css
```

24 Import normalize.css

To include a CSS module you use it identically to a JavaScript one. This makes including only the CSS that is used on a per-page basis much easier. Coupled with HTTP/2's features (where bundling will become an antiquated antipattern) this should lead to performance improvements.

```
001 import 'normalize.css';
```

25 Inspect and update

To see what packages are installed, and their dependencies, we can use 'inspect'. This will print the name and version of all the installed packages. As you'd expect from a tool like this you can also update all packages at once. Jspm will look at the package.json file to make sure only supported versions are installed, however multiple versions of the same package are supported.

```
001 $ jspm inspect
002 $ jspm update
```

26 Module maintenance

Jspm gives you very granular control over what versions of packages are installed and it's not rare to find situations where some libraries use older versions. You can manually override this by updating config.js. This can sometimes leave orphaned packages, those that are installed but no longer used and they can be purged with 'clean'.

```
001 $ jspm clean
```


FOR THE GNU GENERATION

www.linuxuser.co.uk



LinuxUser & Developer

Available from all good newsagents & supermarkets today

ON SALE NOW:

» Fix your network connections » Amazing Pi hacks » Run Linux in the cloud



BUY YOUR ISSUE TODAY

Print edition available at www.imagineshop.co.uk

Digital edition available at www.greatdigitalmags.com

Available on the following platforms



facebook.com/LinuxUserUK twitter.com/LinuxUserMag

Build a mobile web app with NativeScript's library

Use NativeScript to make apps for iOS, Android and Windows Phone and redefine mobile programming

tools | tech | trends JavaScript, Telerik Platform, Telerik companion app
expert Tam Hanna



Nitobi's PhoneGap redefined the world of mobile app development by bundling web content with a browser stub. NativeScript goes one step further: its developers provide a custom JavaScript Runtime which can access native operating system classes without cumbersome plugins. A separate set of abstraction classes provide a common interface, sparing developers the effort of maintaining different code bases for each available operating system.

NativeScript's approach is beneficial in that it permits you to use native controls directly: applications rendered in WebViews tend to look out of place. Furthermore, it simplifies the creation of custom logic for developers who are unfamiliar with Java and/or Objective C.

Even though Telerik has not yet released a final version of NativeScript, the open source product already shows significant progress and is more than deserving of your attention. A live licence is expected any time soon - so you better get coding while the coding is good!

01 Create an app

Sign up at telerik.com/nativescript first. Once logged in, click Create app to create a skeleton container. Click 'Create AppBuilder Native project' to start working with NativeScript and select NativeScript Blank (JavaScript). You can also pick a TypeScript-based sample or choose a more complex example.

02 Analyse the IDE

Telerik Platform will then create the project skeleton. Running the app on a device requires you to download the companion app (Telerik NativeScript) to your phone: it is available in the iOS App Store, on Google Play and in the Microsoft Store. Click Build and follow the instructions.

03 Inflatable GUIs

Now that the toolchain is in place, we can start to create a user interface declaration. Even though NativeScript permits the direct use of native control classes, using interpreted XML layouts tends to yield faster results. Double-click main-page.xml, and replace its contents with the code accompanying this step.

```
001 <Page xmlns="http://www.nativescript.org/tns.xsd" >
002 <StackLayout>
003 <Label text="R:" />
004 <TextField id="TxtR" width="200"/>
005 <Label text="U:" />
006 <TextField id="TxtU" width="200"/>
007 <Label text="I:" />
008 <TextField id="TxtI" width="200"/>
009 <Button id="CmdGo" text="Go" width="100"/>
010 </StackLayout>
011 </Page>
```

04 Add an event handler

User interaction makes widgets emit events. Button clicks should cause a recalculation of the values shown in the fields. This can be done by adding an

event handler to the button. Since the button control exposes a property, we can integrate our new event handler by changing a single line in the XML file.

```
001 <Page xmlns="http://www.nativescript.org/tns.xsd" >
002 <StackLayout>
003 . . .
004 <Button id="CmdGo" tap="buttonTap" text="Go" width="100"/>
005 </StackLayout>
006 </Page>
```

05 Export the logic

Accompanying logic is contained in main-page.js. The most basic version of an event handler consists of the declaration of a method body, which is then added to the exports object. The NativeScript runtime will not be able to find the event handler if it is not part of the exports object.

```
001 var view = require("ui/core/view");
002 function buttonTap(args) {
003 }
004 exports.buttonTap = buttonTap;
```

06 Find our widgets

The args-object provides a group of members which can be used for traversing the object tree resuscitated from the XML declaration. The snippet demonstrates the traversal procedure for the GUI - we pick out the three labels, which come as children of the parent of the button object.

```
001 function buttonTap(args) {
002 var sender = args.object;
003 var parent = sender.parent;
004 if (parent)
005 {
006 var txtR = view.getViewById(parent, "TxtR");
```


<Above>

- Click the big red button to get started, you can sign up to NativeScript or log in if you already have an existing account

<Above>

- The NativeScript IDE is ready and waiting - just click a file on the right in order to open it up for editing

<Above>

- Thanks to the abstraction classes, our XML file gets inflated into a user interface which looks almost native

<Above>

- The Android OS, for example, will hide the buttons if the notification is not positioned at the top of the event list

Tell me more!
 The official NativeScript documentation is stored on GitHub. Visit it at github.com/NativeScript/docs, and scroll to the bottom of the page in order to start reading. For another useful URL, try docs.nativescript.org.

```
007 var txtU = view.getViewById(parent, "TxtU");
008 var txtI = view.getViewById(parent, "TxtI");
009 }
010 }
```

07 Get calculating

Creating the actual logic is not particularly difficult. Start by analysing the contents of the text fields shown. If the R field is empty, it is reasonable to expect that the user has provided values for voltage and current - practical applications should perform more stringent checking of their input values.

```
001 function buttonTap(args) {
002   var sender = args.object;
```

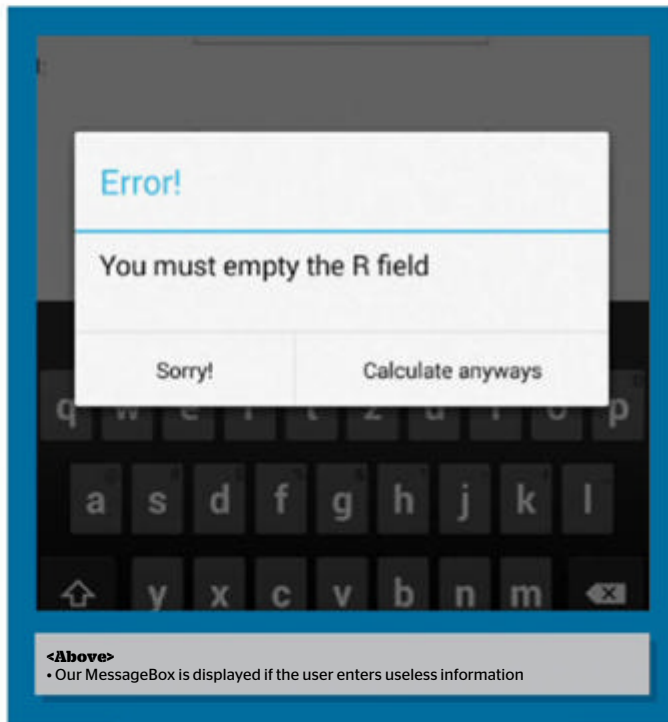
```
003   var parent = sender.parent;
004   if (parent)
005   {
006     . . .
007     if(txtR.text.length==0)
008     {
009       txtR.text=parseInt(txtU.text)/parseInt(txtI.text);
010     }
011   }
012 }
013 }
```

08 Sync the code

Telerik's online IDE stores changes automatically. Pushing changes to your Android phone is really easy: expand the notification, and click the Sync button. The NativeScript companion app will then proceed to downloading and committing any changes deemed important. Your phone does not need to be connected to a desktop for the downloads.

09 Complain about errors

NativeScripts libraries provide a wide variety of constructor objects which permit you to access their features. Our snippet obtains a reference to the dialog's object.



It's just JavaScript

Deploying large amounts of native code is a sure-fire way to reduce portability. Just check to see whether a JavaScript solution for your problem exists if you encounter a situation that is not already covered by the NativeScript libraries.

```
001 var dialogs = require("ui/dialogs");
002 dialogs.confirm({
003   title: "Error!",
004   message: "You must empty the R field",
005   okButtonText: "Calculate anyways",
006   cancelButtonText: "Sorry!"
007 });
```

10 False promises

Our message box runs its magic in an asynchronous fashion. Since blocking the GUI thread is not an option, a promise object is returned. We use its 'then' method to schedule a function for delayed execution

```
001 var dialogs = require("ui/dialogs");
002 dialogs.confirm({
003   . . .
004 }).then(function (result) {
005   if(result==true) //OK
006   {
007     txtR.text=parseInt(txtU.text)/parseInt(txtI.text);
008   }
009 });
```

11 Add another page

With that, work on the calculator page is all but done - it is now time to add a second page to the application. Click Add New file, and enter 'tamspage.js' into the file name field. Then, create a form definition by repeating the process with the file name tamspage.xml.

12 Initial page change!

It's time to change the page shown after application start-up. This can be accomplished by opening the app.js file which contains the declaration of the application object. Next, change the value of the mainModule property to match the newly created group of files.

```
001 var application = require("application");
002 application.mainModule = "app/tamspage";
003 application.start();
```

13 Operating system events

The application class provides a group of members which can be populated with custom logic for handling operating system events. The code on FileSilo shows the possible fields along with basic handlers illustrating the usage of the provided parameters. Handling the events is not mandatory.

14 Dynamic event handling

Tamspage is currently empty. Replace it with the code accompanying this step. The button is now declared without an inline event handler. Instead, the page is provided with a method to be invoked once the inflation is done.

```
001 <Page loaded="loadStartView">
002 <StackLayout>
003 <Button id="calcButton" text="Calculate a resistance"
004 width="600"/>
005 </StackLayout>
006 </Page>
```

15 Page switching

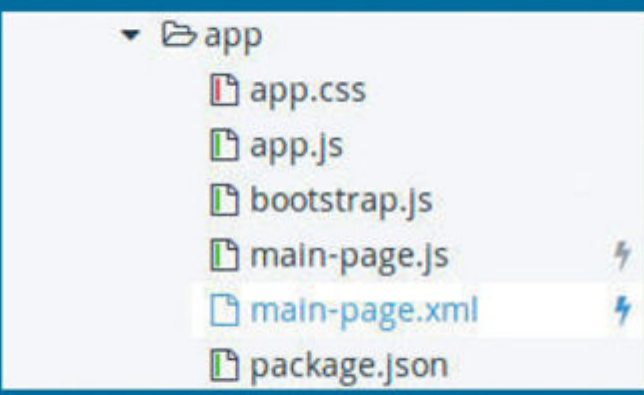
With that, it's time to switch the page in response to the button being pressed. OnLoad is responsible for setting the event handler as the form is loaded. The actual page switch is performed in the inline function provided.

```
001 var view = require("ui/core/view");
002 function onLoad(args) {
003   var page = args.object;
004   var my = view.getViewById(page, "calcButton");
005   my.on("tap", function () {
006     var frames=require("ui/frame");
007     frames.topmost().navigate("app/main-page");
008   });
009 }
010 exports.loadStartView = onLoad;
```

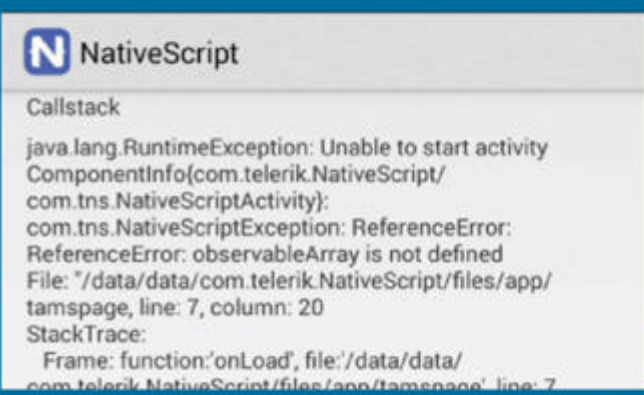
16 Data binding

In the following steps, our example will receive a small list which is to be populated from an array object. Start by adding the specified bit of markup to the tamspage XML file:

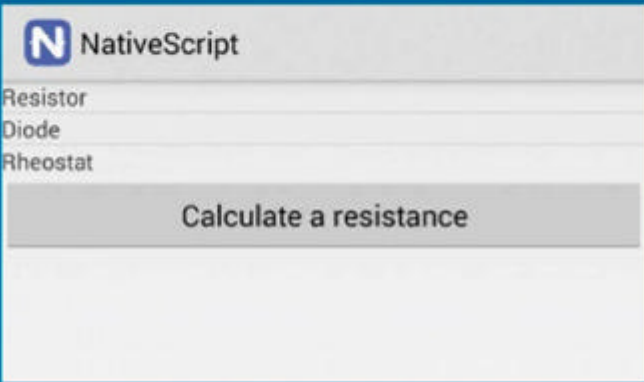
```
001 <Page loaded="loadStartView">
002 <StackLayout>
003 <ListView id="myList" loadMoreItems="listViewLoadMoreItems"
004 />
005 <Button id="calcButton" text="Calculate a resistance"
```

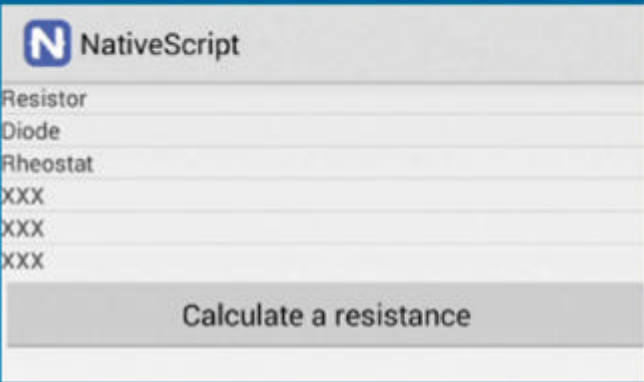
<Above>
• Tampage.js lives in the /app/ subfolder. This affects the string which is to be passed to the page-changing function



<Above>
• Providing invalid code or XML leads to an error such as the one shown in the figure screenshot above



<Above>
• Our list populates itself automatically using the values stored in the object that has been passed in as the model



<Above>
• Changes to the model's content are easily viewable as they are reflected in the user interface automatically

```
width="600"/>
005 </StackLayout>
006 </Page>
```

17 Create an observableArray

Data shown in lists has the unpleasant habit of changing from time to time. NativeScript can handle updates automatically if the underlying data storage is derived from the observableArray class. Load it via require, and use it as a wrapper around a set of different electrical components.

```
001 var observableArray = require("data/observable-array");
002 var labelModule = require("ui/label");
003 function onLoad(args) {
004   var page = args.object;
005   var myList = view.getViewById(page, "myList");
006   var items = new observableArray.ObservableArray(["Resistor",
007     "Diode", "Rheostat"]);
007   myList.items = items;
```

18 Create some elements

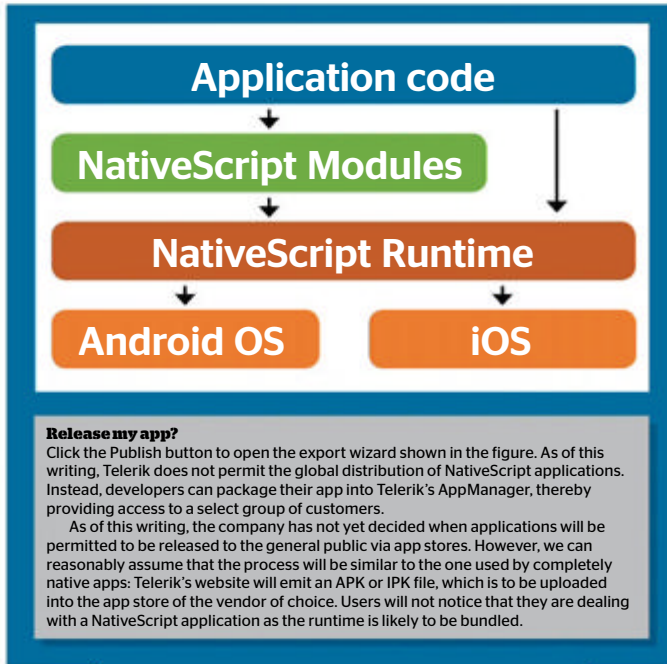
ListView invokes the itemLoading callback when new items need creating. It then receives an args.view parameter to be populated with the object(s) in the slot allocated. Now populate view with a simple label.

```
001 myList.on("itemLoading", function (args) {
002   if (!args.view) {
003     args.view = new labelModule.Label();
004   }
005   args.view.text = items.getItem(args.index);
006 });
```

19 Annoy the list

With that, it's time to modify the underlying storage. This is accomplished via a method which adds a value to the array. As the window object does not exist in NativeScript, Telerik chose to provide the set XXX methods via a helper object.

```
001 var timer = require("timer");
002 function onLoad(args) {
003   var page = args.object;
004   var myList = view.getViewById(page, "myList");
005   var items = new observableArray.ObservableArray(["Resistor",
006     "Diode", "Rheostat"]);
006   myList.items = items;
007   myList.on("itemLoading", function (args) {
008     if (!args.view) {
009       args.view = new labelModule.Label();
```



Yarr, kick a pirate!
 NativeScript is interpreted at runtime. It will only be a question of time until attackers figure out a way to extract code from applications. Obfuscating code before compilation can help though...

```
010 }
011 args.view.text = items.getItem(args.index);
012 });
013 timer.setInterval(function () {
014     items.push("XXX");
015 }, 3000);
```

20 GUI to code
 Our resistance calculator uses object references to retrieve the values needed for the computation. It would be much nicer if the updating was handled by the framework. We can accomplish this by creating a model class which is then bound to the form.

```
001 var dataField = {
002     valu: "24",
003     vali: "35",
004     valr: ""
005 };
006 function assignModel(args)
007 {
008     var page = args.object;
009     page.bindingContext = dataField;
010 }
011 function buttonTap(args) {
012     var sender = args.object;
013     var parent = sender.parent;
```

```
014     if (parent)
015     {
016         if(dataField.valr.length==0)
017         {
018             dataField.valr=parseInt(dataField.valu)/parseInt(dataField.
019             vali);
020         }
021     }
022     else
```

21 Bind to the model
 Setting the loaded property ensures that the initialisation code gets invoked as the app starts up. The text fields are bound to the individual fields of the model via binding strings. A double { informs the parser that a binding command is to follow, it is also possible to set up bindings entirely from code:

```
001 <Page xmlns="http://www.nativescript.org/tns.xsd"
002     loaded="assignModelToView">
003     <StackLayout>
004     <Label text="R:" />
005     <TextField id="TxtR" width="200" text="{{valr}}"/>
006     <Label text="U:" />
007     <TextField id="TxtU" width="200" text="{{valu}}"/>
008     <Label text="I:" />
009     <TextField id="TxtI" width="200" text="{{vali}}"/>
```

22 Make it observable
 The code from this step does not update the text box contents. JavaScript does not provide a way to notify objects about property changes, NativeScript addresses this by introducing a set of observable classes. Data models are based on this class with extra wiring, put to good use in our engine.

```
001 var observable = require("data/observable");
002 var dataField = new observable.Observable();
003 dataField.valu = "24";
004 dataField.vali = "35";
005 dataField.valr = "";
006 function buttonTap(args) {
007     . . .
008     if (parent)
009     {
010         if(dataField.valr.length==0)
011         {
012             dataField.set("valr",parseInt(dataField.valu)/
013             parseInt(dataField.vali));
014         }
015     }
```

23 Swipe to go back
 Most NativeScript controls provide a predefined listener object, it can notify your application of incoming events. We will use this feature to provide our users with a convenient way to get back to the main page of the application. The code is relatively simple: a listener is assigned to the page. Args contain further information about the gesture.

24 Styling
 Telerik's abstracted controls can be modified to suit your liking by adding CSS declarations to the application. The runtime will try to apply the specified design to the native controls as best as it can. Since the actual rendering is done by the operating system, one-to-one coverage can not always be achieved. More information on that is available via [docs.nativescript.org/styling.html](https://nativescript.org/styling.html).

**SPECIAL
OFFER**
USA
readers

Subscribe now and get **5 ISSUES FREE***

The only magazine you need to design and develop stunning websites



Subscribe now **Online**

Order by visiting:
www.imaginesubs.co.uk/wed
and enter USA3 to get this exclusive offer!

Telephone

Order by phone, just call:
+44 (0) 1795 592 878
and quote USA3



TERMS & CONDITIONS

*This is a US subscription offer; please don't forget to quote USA3 when ordering. You will actually be charged £80 sterling for an annual subscription. This is equivalent to \$120 at the time of writing, although the exchange rate may vary. Five free issues refers to the newsstand price of \$14.99 for 13 issues totalling \$194.87, compared with \$120 for a subscription. Your subscription will start from the next available issue. Latest offer is based on newsstand price of \$14.99 per issue compared with \$9.23 through this offer.

This offer expires 31 July 2015.



To advertise here contact Luke
 luke.biddiscombe@imagine-publishing.co.uk
 +44 (0)1202586431

Get your listing **highlighted!** Contact Luke
 luke.biddiscombe@imagine-publishing.co.uk
 +44 (0)1202 586431

Hosting listings

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

Keep an eye on the latest packages and deals with our comprehensive list of service providers

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 <small>.com</small> 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	✓	✗	✓	✓	✓	✓	✓	✗

To advertise here contact Luke
 luke.biddiscombe@imagine-publishing.co.uk
 +44 (0)1202586431

Get your listing **highlighted!** Contact Luke
 luke.biddiscombe@imagine-publishing.co.uk
 +44 (0)1202 586431

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	x	x	x	x	x	✓	✓	✓
eHosting (www.ehosting.com)	Personal	0844 999 4100	£59.88	2.5GB	Unlimited	50	✓	x	x	x	x	✓	✓	✓
eHosting (www.ehosting.com)	Expert	0844 999 4100	£95.88	5GB	Unlimited	250	✓	x	x	x	x	✓	✓	✓
eHosting (www.ehosting.com)	Virtual	0844 999 4100	£227.88	50GB	Unlimited	Unlimited	✓	x	x	x	x	✓	✓	✓
Equipphase (www.equipphase.net)	Bronze	0121 314 4865	£30	200MB	2GB	10	✓	✓	x	✓	x	✓	✓	✓
Equipphase (www.equipphase.net)	Silver	0121 314 4865	£42	400MB	5GB	20	✓	✓	x	✓	x	✓	✓	✓
Equipphase (www.equipphase.net)	Gold	0121 314 4865	£72	800MB	10GB	100	✓	✓	x	✓	x	✓	✓	✓
Equipphase (www.equipphase.net)	Platinum	0121 314 4865	£114	1,200MB	40GB	200	✓	✓	✓	✓	✓	✓	✓	✓
Eurofasthost.com (www.eurofasthost.com)	Email Only	02380 249 823	£40	1GB	2GB	10	x	x	✓	✓	✓	✓	✓	✓
Eurofasthost.com (www.eurofasthost.com)	Essential	02380 249 823	£75	2GB	5GB	10	x	x	✓	✓	✓	✓	✓	✓
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	✓	✓	✓	✓	x	✓	✓	✓
Evohosting (www.evohosting.co.uk)	Home	N/A	£54.99	2.5GB	30GB	50	✓	✓	✓	✓	x	✓	✓	✓
Evohosting (www.evohosting.co.uk)	Business	N/A	£79.99	6.5GB	Unlimited	Unlimited	✓	✓	✓	✓	x	✓	✓	✓
Evohosting (www.evohosting.co.uk)	eCommerce	N/A	£159.99	30GB	Unlimited	Unlimited	✓	✓	✓	✓	x	✓	✓	✓
Fasthosts (www.fasthosts.co.uk)	Personal	0808 168 6777	£32.87	5GB	Unlimited	50	✓	x	✓	✓	✓	✓	✓	x
Fasthosts (www.fasthosts.co.uk)	Business Bronze	0808 168 6777	£58.38	50GB	Unlimited	500	✓	Option	✓	✓	✓	✓	✓	x
Fasthosts (www.fasthosts.co.uk)	Business Silver	0808 168 6777	£76.39	Unlimited	Unlimited	Unlimited	✓	Option	✓	✓	x	✓	✓	x
Fasthosts (www.fasthosts.co.uk)	Business Gold	0808 168 6777	£101.89	Unlimited	Unlimited	Unlimited	✓	Option	✓	✓	x	✓	✓	x
Fasthosts (www.fasthosts.co.uk)	WD Starter	0808 168 6777	£149.99	20GB	Unlimited	Unlimited	✓	Option	Option	✓	x	✓	✓	x
Fasthosts (www.fasthosts.co.uk)	WD Advanced	0808 168 6777	£199.99	Unlimited	Unlimited	Unlimited	✓	Option	Option	✓	x	✓	✓	x
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	x	x	x	✓	x	✓	✓	✓
Heart Internet (www.heartinternet.co.uk)	Home Professional	0845 644 7750	£89.99	Unlimited	Unlimited	10,000	✓	✓	x	✓	x	✓	✓	✓
Heart Internet (www.heartinternet.co.uk)	Business Professional	0845 644 7750	£129.99	Unlimited	Unlimited	Unlimited	✓	✓	x	✓	x	✓	✓	✓
Heart Internet (www.heartinternet.co.uk)	Reseller Professional	0845 644 7750	£349.99	Unlimited	Unlimited	Unlimited	✓	✓	x	✓	x	✓	✓	✓
Hostway (www.hostway.co.uk)	Silver	0808 180 1880	£79.50	150MG	3GB	5	x	Option	✓	✓	x	✓	✓	x
Hostway (www.hostway.co.uk)	Gold	0808 180 1880	£139.50	300MB	5GB	10	✓	Option	✓	✓	✓	✓	✓	x
Hostway (www.hostway.co.uk)	Gold Plus	0808 180 1880	£189.50	450MB	10GB	30	✓	Option	✓	✓	✓	✓	✓	x
Hostway (www.hostway.co.uk)	Platinum	0808 180 1880	£359.50	600MB	20GB	50	✓	Option	✓	✓	✓	✓	✓	x
Hostway (www.hostway.co.uk)	Platinum Plus	0808 180 1880	£599.50	1.2GB	40GB	10	✓	Option	✓	✓	✓	✓	✓	x
Hostway (www.hostway.co.uk)	Email Plus	0808 180 1880	£49.95	50MB	N/A	5	N/A	N/A	N/A	✓	x	✓	✓	x
LD Hosts (http://ldhosts.co.uk)	LD Budget (Linux)	07891 235858	£11.88	1GB	1GB	500	✓	✓	x	✓	✓	✓	✓	✓
LD Hosts (http://ldhosts.co.uk)	LD Home (Linux)	07891 235858	£23.88	10GB	10GB	5,000	✓	✓	x	✓	✓	✓	✓	x
LD Hosts (http://ldhosts.co.uk)	LD Pro (Linux)	07891 235858	£41.88	50GB	50GB	20,000	✓	✓	x	✓	✓	✓	✓	x
LD Hosts (http://ldhosts.co.uk)	LD Unlimited (Linux)	07891 235858	£60	Unlimited	Unlimited	Unlimited	✓	✓	x	✓	✓	✓	✓	x
LD Hosts (http://ldhosts.co.uk)	Windows Home	07891 235858	£17.88	2GB	2GB	1,000	✓	✓	x	✓	✓	✓	✓	x
LD Hosts (http://ldhosts.co.uk)	Windows Unlimited	07891 235858	£60	Unlimited	Unlimited	Unlimited	✓	✓	x	✓	✓	✓	✓	x
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	x	x	✓	✓	✓	✓	✓	✓
NameHog (www.namehog.net)	Starter	01604 212 904	£36.99	10GB	150GB	Unlimited	x	x	✓	✓	✓	✓	✓	✓
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	x	x	✓	✓	✓	✓	✓	✓
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	✓	✓	✓	✓	✓	✓	✓	✓



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	✓	✓	✓	✓	✓	✓	✓	✓	✓

To advertise here contact Luke
 luke.biddiscombe@imagine-publishing.co.uk
 +44 (0)1202586431

Get your listing **highlighted!** Contact Luke
 luke.biddiscombe@imagine-publishing.co.uk
 +44 (0)1202 586431

Hosting listings

Keep an eye on the latest packages and deals with our comprehensive list of service providers

Discover what an **enhanced** listing can do for your business. Contact Luke on:

+44 (0)1202 586431

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)	Basic55	0845 641 0776	£24	100MB	1,500MB	25	✓	✓	✓	✓	✓	✓	✓	✓
TwentyHost (www.twentyhost.co.uk)	Standard55	0845 641 0776	£45	200MB	3,000MB	50	✓	✓	✓	✓	✓	✓	✓	✓
TwentyHost (www.twentyhost.co.uk)	Business55	0845 641 0776	£70	500MB	7,500MB	100	✓	✓	✓	✓	✓	✓	✓	✓
TwentyHost (www.twentyhost.co.uk)	Advanced55	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.

IMAGE IS EVERYTHING

www.advancedphotoshop.co.uk



ADVANCED PHOTOSHOP®

Available from all good newsagents and supermarkets

ON SALE NOW

> Blend modes > Expert tricks for histograms > Hot new CC tools > Ink & Slide



BUY YOUR ISSUE TODAY

Print edition available at www.imagineshop.co.uk

Digital edition available at www.greatdigitalmags.com

Available on the following platforms



[facebook.com/AdvancedPhotoshop](https://www.facebook.com/AdvancedPhotoshop)

twitter.com/advancedpshop

COMPILA
Web Hosting You Can Trust

Call Today
0843 506 8902

Spring Special offer
SAVE 50%
On Award Winning Hosting

50% OFF

100% Uptime Guarantee
These are just some of the benefits you will enjoy with Compila hosting:

- ✓ Free UK Domain
- ✓ Fast UK Web Servers
- ✓ Huge Web Space Limits
- ✓ Free 24/7 UK Support
- ✓ Free Software Apps
- ✓ and lots more ...

The Only UK Web Host To Offer A Lifetime Money Back Guarantee!

To claim your 50% discount simply visit www.compila.com and choose your plan To SAVE enter **save15webd** at the checkout

ONLY £0.58 + VAT pm

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

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

For more details visit postcode-software.net or call **0845 83 82 666**

From less than **1p** per lookup

«**clikpic**» Websites for photographers

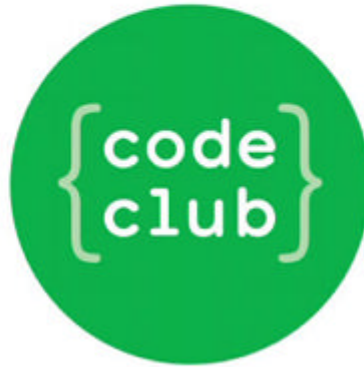
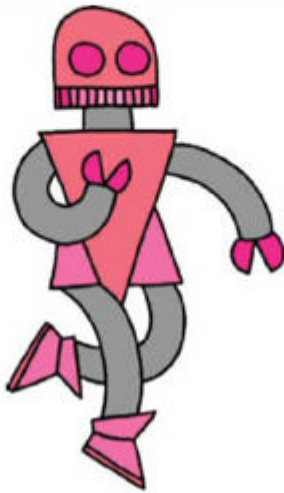
Try the **new** Clikpic for only **£40 pa** includes ecommerce



- 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!

Visit www.clikpic.com for a **FREE 14 day trial**

Compatible with mobile phones and tablets



Can you volunteer for Code Club?

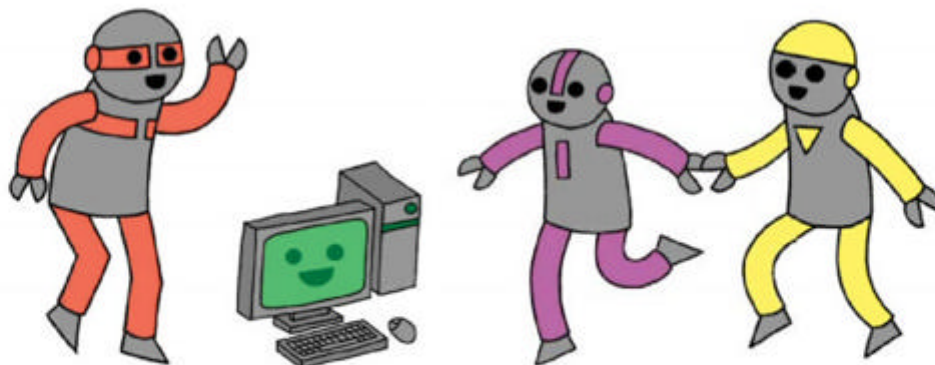
Code Club is a nationwide network of volunteer-led after school coding clubs for children aged 9-11.

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.

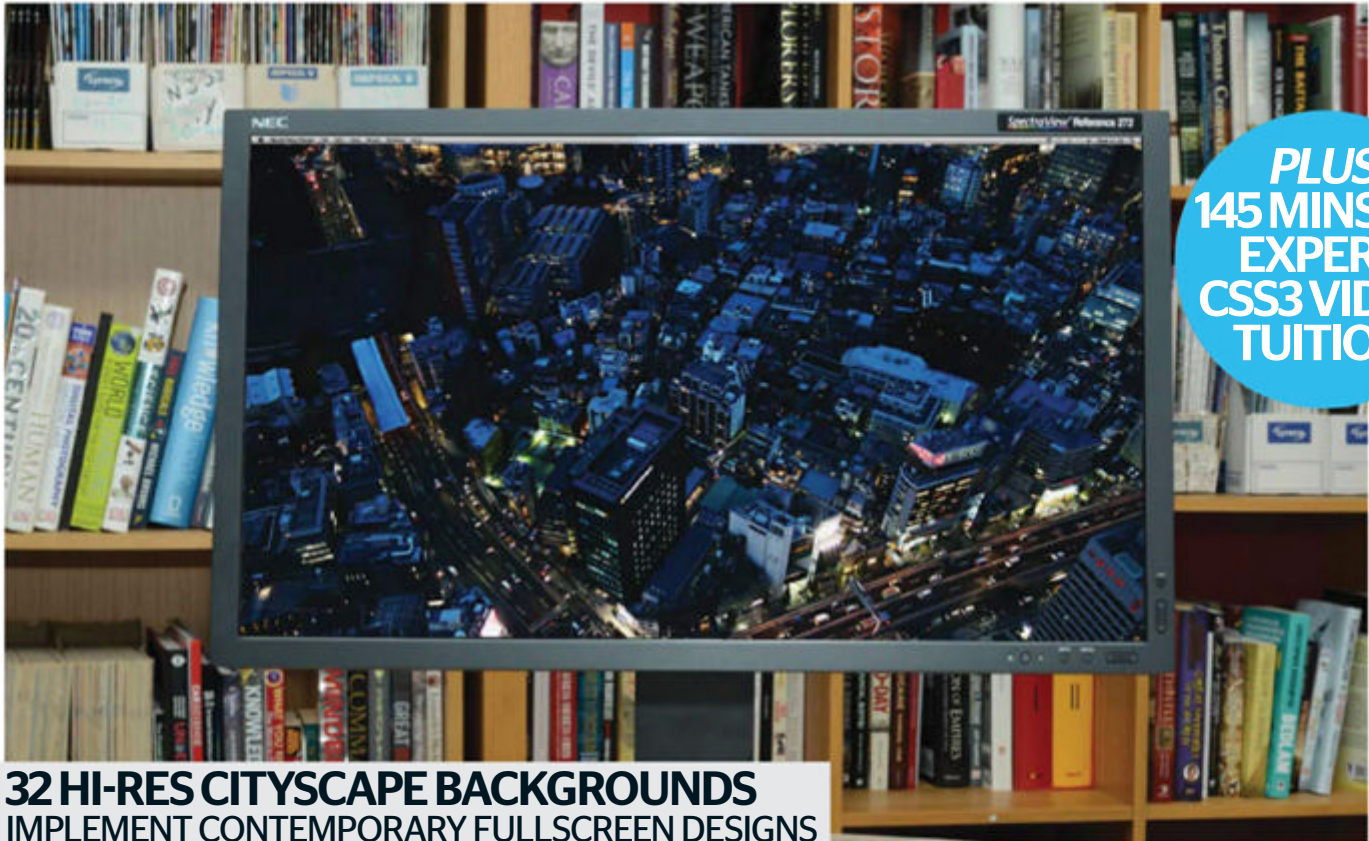
Get involved, let's teach the next generation to code!

Visit www.codeclub.org.uk to find out more

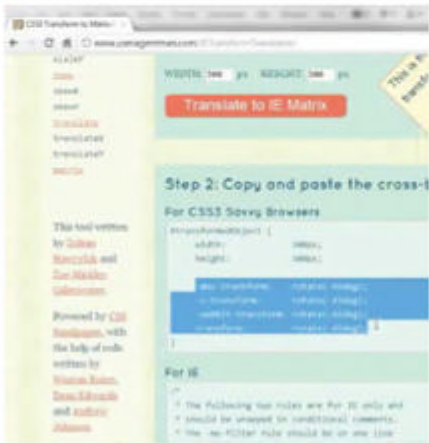


YOUR FREE RESOURCES

LOG IN TO WWW.FILESILO.CO.UK/WEBDESIGNER AND DOWNLOAD THE LATEST COLLECTION OF MUST-HAVE VIDEOS AND RESOURCES

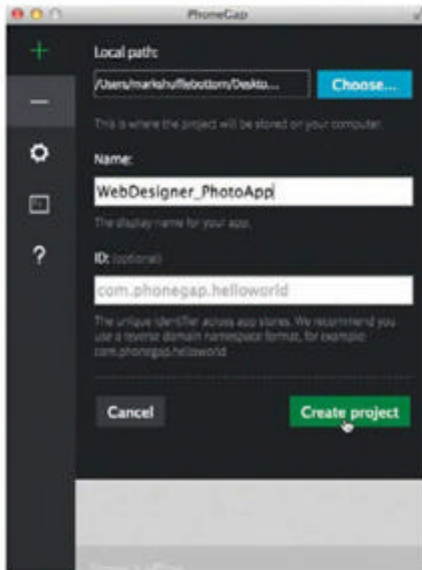


32 HI-RES CITYSCAPE BACKGROUNDS
IMPLEMENT CONTEMPORARY FULLSCREEN DESIGNS



EXPERT CSS3 VIDEOS
CHECK TRANSFORMS & MORE

TOTAL RESOURCES **55**



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

- 145 Minutes of expert CSS3 video guides from CartoonSmart
- 32 Hi-res Tokyo aerial cityscape backgrounds from SparkleStock (worth \$10)
- 3 Web templates: flat invitation form, Midget responsive portfolio template and web app log-in screen template from w3layouts
- The Kingsbridge, Hoverunit and Kilsnburg font from Typodermic
- 10 Glitch art Photoshop actions from Sparklestock

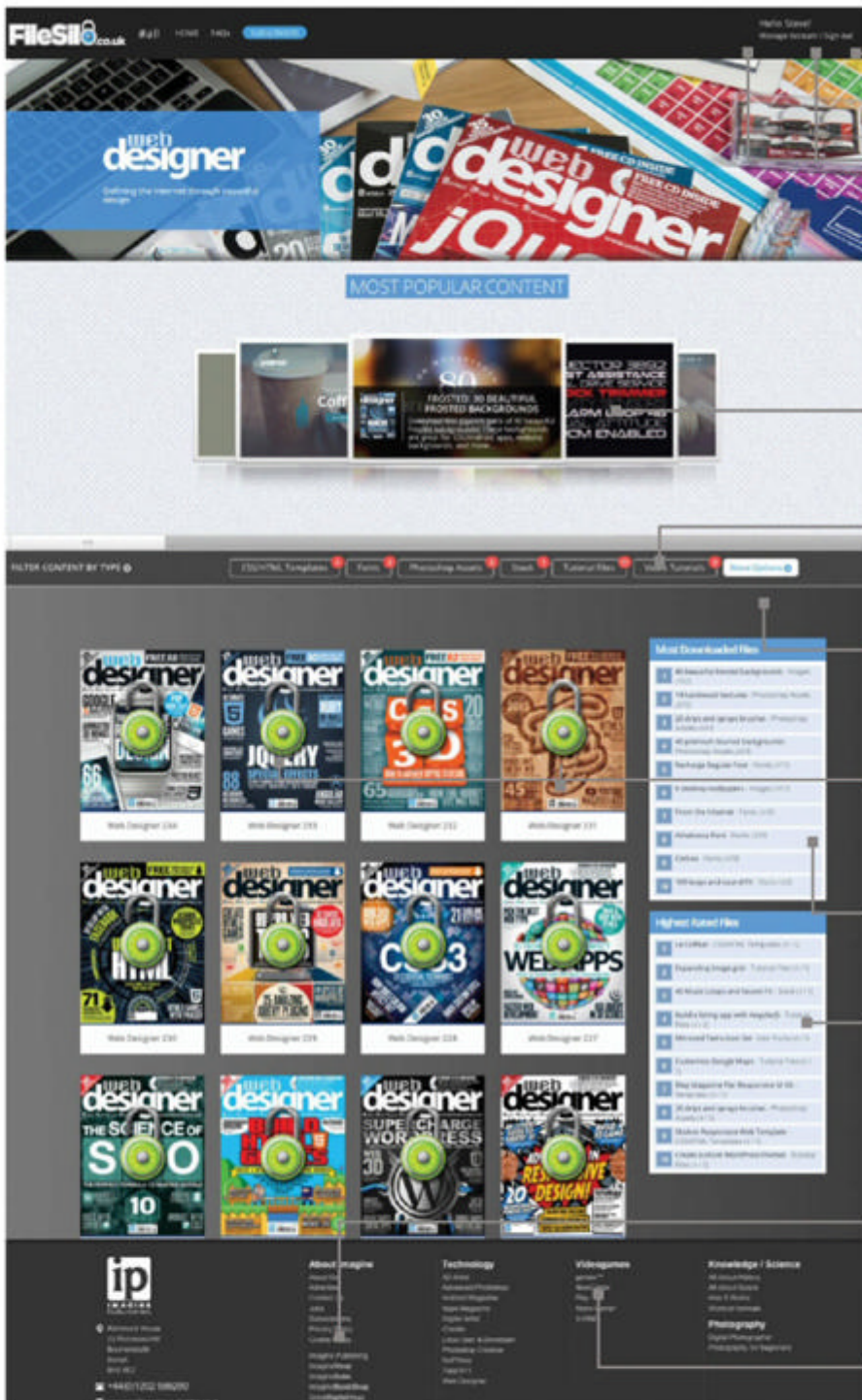


www.filesilo.co.uk/webdesigner

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

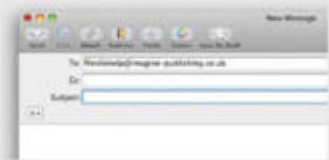


question about the magazine. You can access the content for free with your issue.

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 236 of **web designer** is on sale **28 May 2015** from GreatDigitalMags.com

UNLOCK THE POWER OF ANDROID

www.littlegreenrobot.co.uk



Android magazine

Available from all good newsagents and supermarkets

ON SALE NOW

> 50 hacks, apps and ROMs > Get Apple Watch features on Android > The future of Android

HARDWARE REVIEWS



ANDROID HACKING



TIPS & TRICKS



APP REVIEWS



HELP & ADVICE

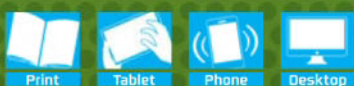


BUY YOUR ISSUE TODAY

Print edition available at www.imagineshop.co.uk

Digital edition available at www.greatdigitalmags.com

Available on the following platforms



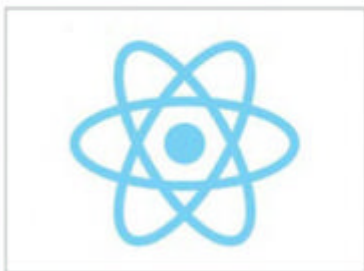
facebook.com/littlegreenrobot



twitter.com/lgrobot

ADVANCED ANGULAR

Nine pages of tips, techniques and tutorials
for getting the very best from the superheroic
JavaScript framework



**SAY HELLO TO
REACT NATIVE**
How to build apps with Facebook's
new addition to the React library



**INTERACTIVE
HTML5 GRAPHICS**
Use HTML5 Canvas to create cool
data visualisations with P5.js



**TURBOCHARGE
YOUR CSS CODE**
A comprehensive guide to using
preprocessors in the real world



**BETTER APPS
WITH METEOR**
Combine the fast app creation
platform with a front-end framework

Visit the **WEB DESIGNER** online shop at
imagineshop.co.uk
for back issues, books and merchandise

ALL IN YOUR NEXT
WEB DESIGNER
Issue 236 on sale
Thursday 28th May 2015

Where ideas take shape.

Bring your projects to life with Shutterstock's diverse library of over 50 million images, videos, and audio tracks. Flexible payment and subscription options make creating even easier. **Save up to 20% now.**

[shutterstock.com](https://www.shutterstock.com)

NEW! Convenient monthly billing • **NEW!** No daily download limit • Innovative search tools



shutterstock[®]



Accelerate the growth of your digital agency...



Telephone: 020 3004 9644

Email: info@iwdro.org

www.iwdro.org

We believe in better business online