27 PAGES OF EXPERT TUTORIALS

# web designer

FREE 84 MINUTES OF TYPE VIDEOS

HTML5 CSS3 jQuery JS JavaScript /WebDesignerMag /WebDesignerUK

## FIRST-CLASS JAVASCRIPT
Top tips to improve, maintain and test your code

## CREATE WITH CSS4 & HTML5.1

## WEB 3D 2.0
How WebGL 2.0 & JS will add a new dimension

## OPTIMISE IMAGES
Keep the quality, lose the weight

## EXCLUSIVE
Behind the scenes with Clearleft

## POWER UP YOUR CSS WITH SASS
Get started with the king of preprocessors

HOW TO USE THE LATEST SPECS IN YOUR PROJECTS TODAY

Future
Digital Edition
GreatDigitalMags.com
ISSUE 258

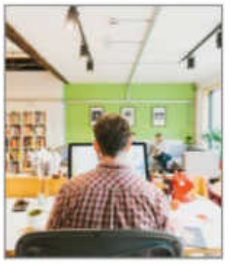
# Welcome to the issue

**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

**Steven Jenkins**

## Highlight



**We like to give back to the community that supported us in the first place. Sharing knowledge is what drives the web**

Creative veterans and forward thinkers Clearleft talk to **Web Designer**. **Page 32**

## Welcome to the new web

**The web never stands stills and the very reason for this is the ever-evolving specifications, technologies and tools.** At the core of all web design are the stalwarts HTML and CSS. When these get updates, things really start to change. In this issue's lead feature (page 40) we take a closer look at how CSS4 is evolving and what changes have been made to the HTML5 specification. Find out what new selectors you can get your hands on and how to use them. There is a closer look at CSS Variables and advanced animation as well as a peek at CSS Grids. Discover what's new in HTML5.1, what's been given the heave-ho and what has been changed. Plus, find out how to use newcomer <picture> and what is the current state of the HTML 5.2 specification.

The other kingpin in the web design triumvirate, JavaScript, is present and correct. This issue it's all about delivering first-class code. Discover the latest tools and techniques that will keep your projects on track and help make sure your code doesn't become unwieldy. Helping keep 3D code clean and easier to use is WebGL and the imminent arrival of WebGL 2.0 is going to make adding a new dimension better still. WebGL Workshop founder Carl Bateman reveals what's happening, the best tools to get started and cutting-edge examples to inspire you. Plus, we have all the usual suspects. Enjoy the issue.

Keeping up to date with what's available to you as a developer is vital. It means that you can improve your workflow, create features and more

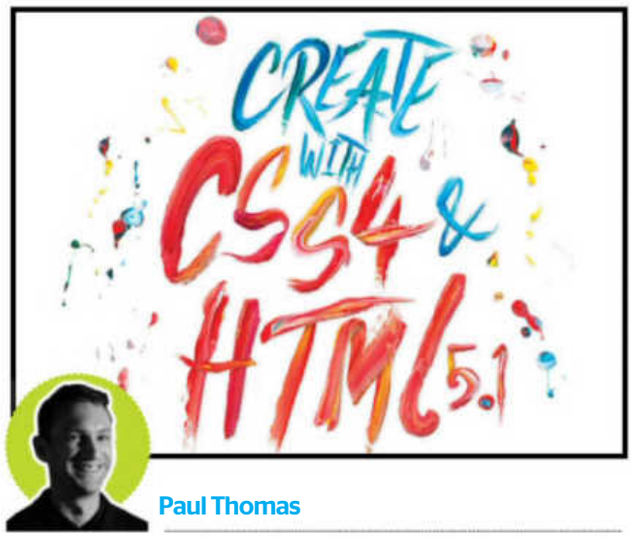
**Follow us on Twitter for all the news & conversation @WebDesignerMag**

**Visit our blog for opinion, freebies & more www.gadgetdaily.xyz**

# This issue's panel of experts

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

> Incremental releases mean that new features can be adopted by browser vendors more quickly, so that we don't have to wait so long to use them






### Paul Thomas

Paul is the senior developer at creative Bristol agency Green Chameleon. This issue he focuses on the latest developments in CSS4 and HTML 5.1. He reveals how to use the most useful selectors, along with CSS variables, advanced animation and the HTML 5.1 you need to use. **Page 40**

### Carl Bateman

Carl is the founder of WebGL Workshop and Khronos London Chapter lead. He is an experienced developer and this issue he talks about the current state of WebGL, best practices, what's up with 2.0 and shows cutting-edge examples. **Page 72**

### Matt Crouch

Matt is a front-end developer based in London. With the web getting more powerful, JavaScript applications are more complex than ever. In this issue, he offers key pointers on how to keep those projects manageable. **Page 64**

### Mark Shufflebottom

Mark is a Professor of Interaction Design at Sheridan College of Advanced Learning. This issue he will be creating organic, watercolour transition effects with the WebGL extension Aquarelle. **Page 50**

### Tam Hanna

Tam's development career has seen him wrestle with a variety of stacks and languages. This issue he takes a look at a selection of tools and techniques that can optimise images for faster-loading pages. **Page 84**

### Leon Brown

Leon is a freelance web developer and trainer who assists web developers in creating efficient code for projects. This issue he reveals a host of techniques, as seen on the top-class websites featured in our Lightbox. **Page 16**

### Neil Pearce

Neil is a web and UI designer from Essex. He has been contributing to the magazine for many years. In this issue Neil takes time to introduce the popular CSS preprocessor Sass and show how to use it. **Page 54**

### Sarah Maynard

Sarah Maynard is a freelance WordPress developer with a passion for usability and accessibility. This issue she reveals how to create bespoke page layouts with the plug-ins ACF Pro and ACF Page Builder. **Page 58**

### Mark Billen

Mark is a freelance technology journalist with a background in web design and development. This issue he jumps on the Lightbox pages and selects some of the best-looking and best-built sites on the web. **Page 14**



web designer

**Future Publishing Ltd**
Richmond House, 33 Richmond Hill
Bournemouth, Dorset, BH2 6EZ
☎ +44 (0) 1202 586200
Web: **www.gadgetdaily.xyz**
www.greatdigitalmags.com
www.futureplc.com

**Editorial**
Editor **Steven Jenkins**
steve.jenkins@futurenet.com
☎ 01202 586233
Designer **Harriet Knight**
Photographer **James Sheppard**
Senior Art Editor **Will Shum**
Editor in Chief **Amy Hennessey**

**Contributors**
Louise Christiansen-Rolls, Craig Murphy, Carl Bateman, Leon Brown, David Howell, Tam Hanna, Paul Thomas, Matt Crouch, Neil Pearce, Mark Shufflebottom, Sarah Maynard, Mark Billen, Carrie Mok, Philip Morris, Phil King

**Advertising**
Digital or printed media packs are available on request.
Commercial Sales Director **Clare Dove**
☎ 01225 442244
clare.dove@futurenet.com
Advertising Manager **Mike Pyatt**
☎ 01225 687538
michael.pyatt@futurenet.com
Account Director **George Lucas**
george.lucas@futurenet.com
Advertising Sales Executive **Chris Mitchell**
chris.mitchell@futurenet.com

**International**
Web Designer is available for licensing. Contact the International department to discuss partnership opportunities.
Head of International Licensing **Cathy Blackman**
☎ +44 (0) 1202 586401
cathy.blackman@futurenet.com

**Subscriptions**
For all subscription enquiries:
webdesigner@servicehelpline.co.uk
☎ 0844 848 8413
☎ Overseas +44 1795 592 878
www.imaginesubs.co.uk
Head of subscriptions **Sharon Todd**

**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

**Circulation**
Circulation Director **Darren Pearce**
☎ 01202 586200

**Production**
Production Director **Jane Hawkins**
☎ 01202 586200

**Management**
Finance & Operations Director **Marco Peroni**
Creative Director **Aaron Asadi**
Editorial Director **Ross Andrews**

**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, 5 Churchill Place, Canary Wharf, London, E14 5HU
☎ 0203 787 9060 www.marketforce.co.uk

**Distributed in Australia by** Gordon & Gotch Australia Pty Ltd,
26 Rodborough Road, Frenchs Forest, New South Wales 2086
☎ + 61 2 9972 8800 www.gordongotch.com.au



ISSN 1745-3534

HETZNER ONLINE

# Network security Firewall.


£ 0
Firewall
free of charge

## e.g. Dedicated Root Server PX61-NVMe

Intel® Xeon® E3-1275 v5
Quad-Core Skylake Processor
64 GB DDR4 ECC RAM
2 x 512 GB NVMe Gen3 x4 SSD
Guaranteed 1 Gbit/s bandwidth
100 GB Backup Space
30 TB traffic inclusive*
No minimum contract
Setup Fee £101.00

monthly £ 50

* There are no charges for overage. We will permanently restrict the connection speed if more than 30 TB/month are used. Optionally, the limit can be permanently cancelled by committing to pay £1 per additional TB used.

## Free Firewall for Your Dedicated Root Servers!

Hetzner Online's stateless firewall is a free security solution for your dedicated root server. Starting now on the customer interface Robot, you can use the firewall feature to define your own filtering settings for traffic, such as the originating IPv4 address or TCP/UDP sender port. With this feature, Hetzner Online helps you protect your dedicated root server from Internet dangers. **And it is naturally free of cost.**

## www.hetzner.de/gb

inside issue 258

# contents

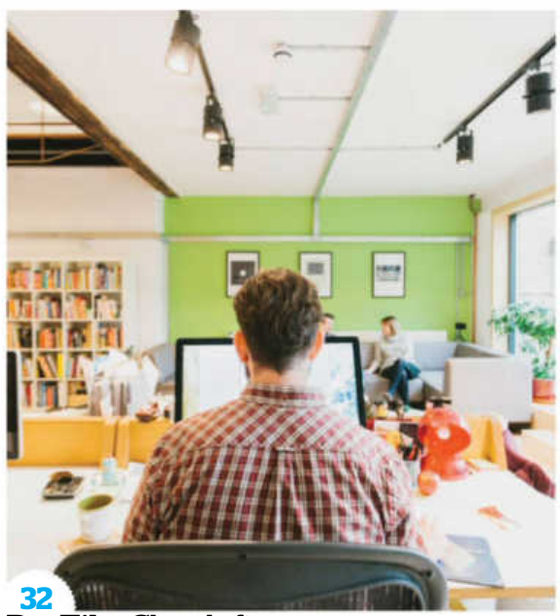
## Cutting-edge features, techniques and inspiration for web creatives

Chat with the team and other readers and discuss the latest tech, trends and techniques. Here's how to stay in touch...
webdesigner@imagine-publishing.co.uk @WebDesignerMag www.gadgetdaily.xyz

## Quick look...







**ProFile:** Clearleft
Digital veterans reveal a little of their world

**Lightbox:** Mozaik
Mesmerising sound and vision

# FileSilo



## A comprehensive collection of free designer resources!

- 84 minutes of expert typography video guides from Pluralsight
- 60 Retro effects and frames
- 7 Photoshop dehaze effects






# <header>

## The tools and trends to inspire your web projects



"Using data wisely will enable you to market your products to the right customers at the right time via multiple channels"
**Louise Christiansen-Rolls**



# <tutorials>

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



# web workshop



# web developer

*CONTACT US AT:* webdesigner@imagine-publishing.co.uk | @WebDesignerMag

# Header

The tools, trends and news to inspire your web projects.

# How good is your mobile site?

## Your site might be mobile-friendly, but how well does it deliver on speed and experience?

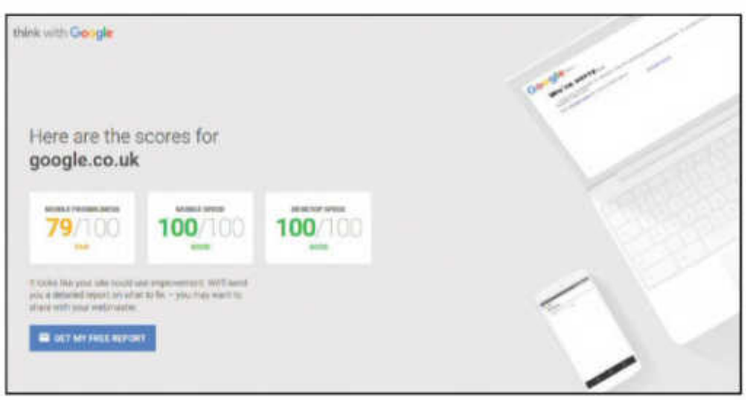
It's no revelation that mobile is the number one option for browsing the web. It overtook the desktop back in 2014. But exactly how important is the mobile experience to the success of a site or campaign? According to a 2016 report from comScore, 65% of 'digital' time is spent on mobile, while the share of desktop time is around 35%. This is an increase of around 5% year on year, while desktop is falling by a slightly slower rate. What else the report points out is that mobile app and smartphone app usage is also, unsurprisingly, on the rise.

This doesn't mean that the desktop is dead; for example, most eCommerce transactions still take place on a desktop. But, this statistic can be put down to the fact that the online buying experience is tailored very much to the desktop. The buying experience on mobile, especially smartphones, is still lagging behind. While more and more companies are investing in this area and great strides are being made, there is still a lot of work to do. Once the buying experience on the small screen equates to the desktop, or betters it, consumers will move to mobile as well, making the desktop further redundant. Another concern – especially for the older age groups – is security, with many still sceptical about using credit cards online.

According to the report, smartphone penetration sits at around 80% overall, but for the 18-34 year group it rises to around 93 per cent. This is the group that will be doing most of the online purchasing, but they don't necessarily have the spending power of older age groups.

All this info brings us back to the fact that mobile is more popular than ever and will only get more popular. What else it tells us is that if your brand and site are not fully committed to mobile then it is time they were. To be more successful, you need to be fully mobile-ready.

> *"Many brands and sites are committing to mobile, but simply not doing it well enough"*




*Find out how good your website is with the help of Google*

Many brands and sites are committing to mobile, but simply not doing it well enough. The user experience and download speeds are critical to success. Much like being on a desktop, if you don't get it right, users will simply look for a better alternative.

Google's Mobile Website Speed Test site, **testmysite.thinkwithgoogle.com**, is a quick and easy way to test your site. This offers results on Mobile Friendliness, Mobile Speed and Desktop Speed. All you need to do is add a URL and wait a few seconds. Try this on your own site and then compare it with some of the more popular on the web. We tested Google, the BBC, Instagram, the Daily Mail and our very own Gadget site. What we found from this small sample is that most sites are now mobile-friendly, but lack in the delivery department. Mobile speeds on all the sites (apart from Google) were poor, ranging from 48-62 out of 100. Don't panic: you can get quick look at where you need to improve a site and get a more in-depth report if desired. The changes aren't generally drastic, so in the main can be put into place quickly. Note that if you are thinking of buying a WordPress theme to build your site, make sure you test before buying: it may look pretty, but does it perform?

Google is not the only option in the market. Try out GTMetrix (**gtmetrix.com**) This "gives you insight on how well your site loads and provides actionable recommendations on how to optimize it". Plus, they have a dedicated resources section and the option to go Pro for an even better site.

Get testing now. Remember, if your site isn't up to scratch, you will be missing out on visitors and ultimately missing out on building your brand and following the path to success.

## Get Web Designer digital editions

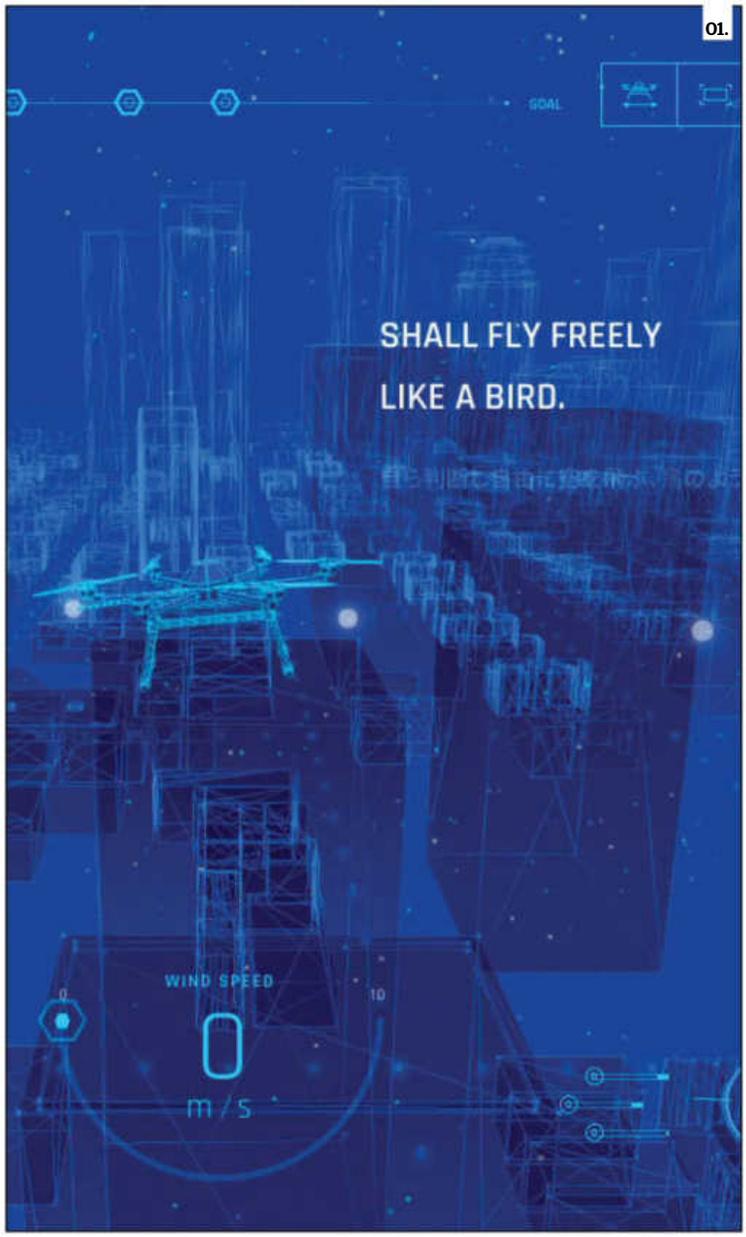
### Get your hands on a collection of the best contemporary techniques

**Check out the latest digital additions to Web Designer's burgeoning collection. Amazing Websites and How to Build Them Vol 04 brings together another collection of great-looking sites and picks out quick techniques to learn. The Web Apps Handbook Volume 2 and The Developer's Guide to JS select the best tools and methods to build apps with web technologies and show how to get started with ES6 and NPM.**

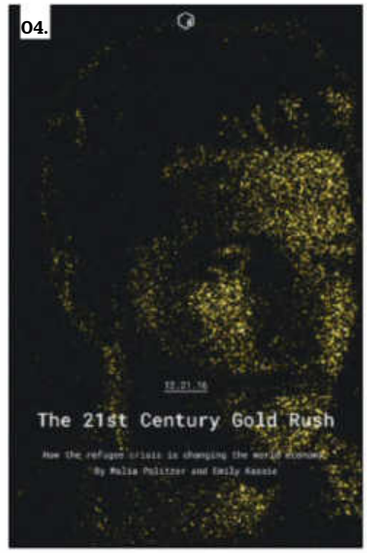
**Don't forget Animate with HTML & CSS and 3D and the Web. We cherry-pick the best animation and 3D-related techniques to add interest to any project.**

**Head to bit.ly/1hsGYgl, download the free Web Designer app and find them waiting for you as an in-app purchase.**

# Sites of the month






### 01. ACSL
acsl.co.jp
Get interactive and guide a drone through a digital landscape

### 02. UpToTab
uptotab.com
A simple illustrated platform game with a dislike for Flash

### 03. Wheelchair Dance
wheelchair-dance.com
Engaging interaction between the user and the on-screen video

### 04. 21st Century Gold Rush
huff.to/2ifeZUb
Animated golden flecks come together to create faces before being blown away

## Graphics
### Vector Animals
bit.ly/2jv3UIM
Design student Daniel Swainson brings together a collection of vector animals with impact. Choose from Wolf, Lion and Fox


## Colour picker
### Download Lassie Come
bit.ly/2iuP4IN

#00BCC7
#00BBD0
#00BFE3
#49C2C7
#62C4B0

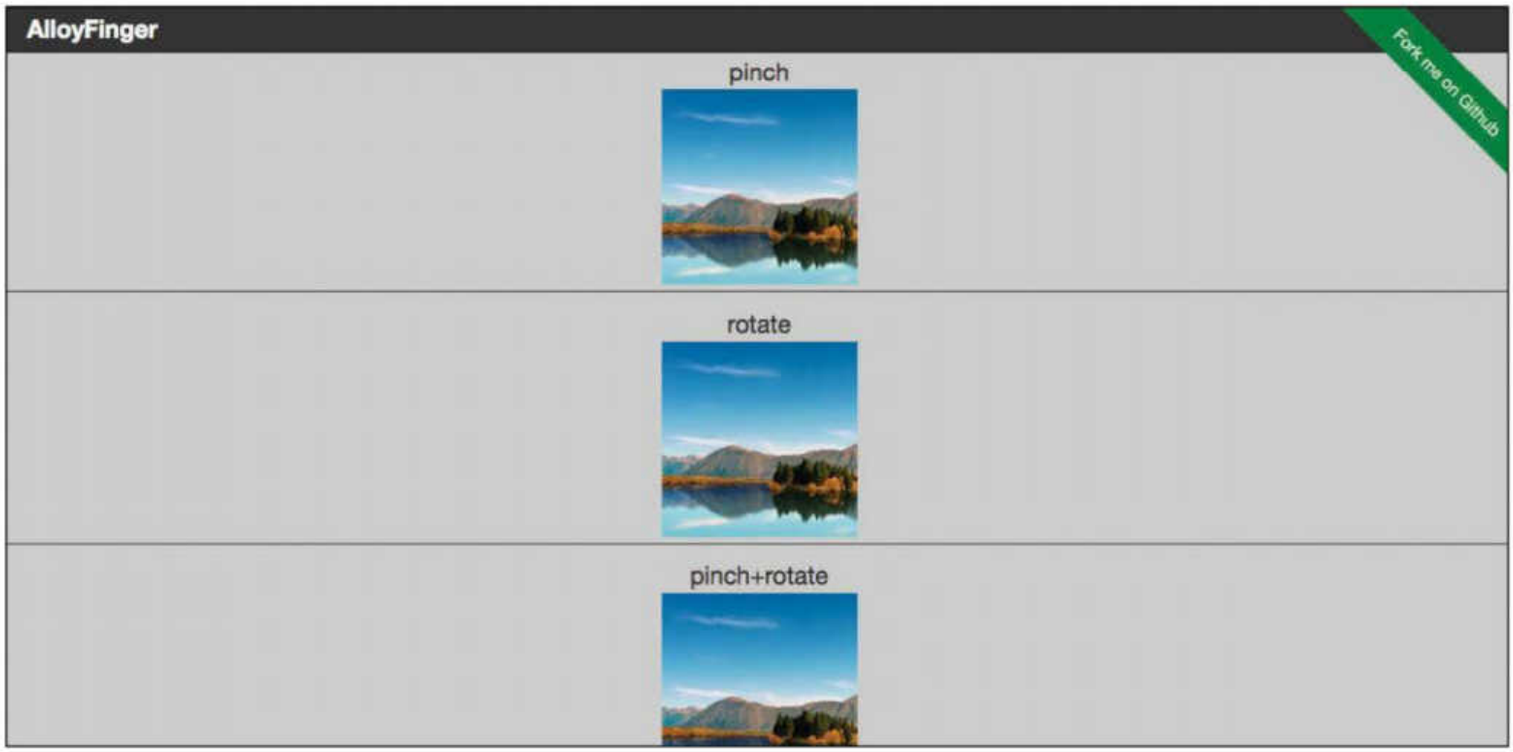
## Typesetter
### MTT Milano
bit.ly/2jUP1GF
Inspired by the Milanese typographic heritage and the Futurist movement that developed it, this font is fashionable and readable.

ABCabc
01234

## WordPress
### ForIT
forit.iondigi.com
Bright appealing palette and crisp graphics come together offering a contemporary theme for all screens

# webkit

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




## AlloyFinger

**alloyteam.github.io/AlloyFinger/**

Every mobile device has a touch screen, which makes interaction vital. A poor experience on a site will send a user off elsewhere. AlloyFinger describes itself as a 'super tiny size multi-touch gestures library for the web'. So what can it do? The demo shows off its many talents, including pinch, rotate, pinch+rotate, pressMove, doubleTap, swipe and more.

## Zoom.js

**github.com/nishanths/zoom.js**

A zoom effect comes in many guises and this option is a pure JavaScript image-zooming plug-in, as demonstrated on Medium.com.

## Lightgallery.js

**sachinchoolur.github.io/lightgallery.js**

This fully featured JavaScript lightbox gallery has no dependencies. It is responsive, modular, customisable and offers touch and YouTube support.


## Draft.js

**facebook.github.io/draft-js**

A framework for building rich text editors in React. All is customisable, enabling users to build exactly what they want. Be sure to check out the demo.

# TOP 5 WordPress themes - February 2017

Add contemporary style and functionality with this latest collection

## We are Maav

**bit.ly/2jQsavL**

Complete with a video header (among other options), this is minimalist, clean and classy. Great for showing off projects


## Muscat

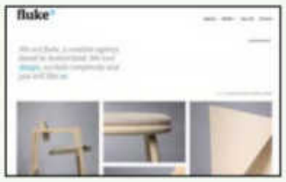
**asianitbd.com/wp/muscat**

A single-page theme focused on business. Neatly constructed sections are perfect for adding custom content


## Look

**webdesign-finder.com/look**

Full width sections, big bold images, contemporary fonts and fine layout and typography make this a fashionable theme




## Fluke

**bit.ly/2jrfeiN**

Simple but classic colour palette and serif fonts combine to provide the framework for a collection of classy imagery


## Piñata

**bit.ly/2jOoebT**

A collection of customisable homepages with simple but effective layouts, including Shop, Blog and Fullscreen

# Think technology think customer

With customer requirements constantly evolving, how can you exceed expectations?

**Louise Christiansen-Rolls** | Head of eCommerce, Fashion & Luxury | propeller.co.uk

ECommerce as a fast-paced, ever-changing industry is continuing to grow and develop across all sectors. With advancements in technology carving out the trends, eCommerce brands' future processes are being mapped out for them – and early adoption should be a priority.

2017 is no exception in this sense. We have seen progress of eCommerce capabilities via social media platforms, initial testing of delivery by drone and the use of chatbots – with rumours of AI in the works. However, this isn't something that the majority of brands can keep up with and in reality, the key takeaway from these developments is customer expectation.

We are privy to huge amounts of data recorded from on-site activity, which is key in identifying and analysing the pattern and traits of your customers. Using this data wisely will enable you to market your products to the right customers at the right time via multiple channels; from your website to social media to newsletters and a concierge service.

This is the beginning of creating an effective CRM strategy. As a starting point, those registered on your website can be divided into customers (those with a purchase history) and users (those yet to purchase). From there, decisions can be made on how to segment the data; for example, splitting your customers into preference groups based on cluster patterns. Groups based on size, colour, season, type, spend... will allow you to build profiles to which you can tailor your communication.

The use of CRM goes hand in hand with personalisation. By using a CRM and eMarketing platform, you will easily be able to implement relevant segmentation, helping you to identify different customer groups and maximize on your contact with them. By analysing customers' behavioural patterns, you can adapt your responses to them to be more helpful and more relevant; eg emailing them about certain product releases, recommending alternatives or upsells, or perhaps inviting them to an event. This all links to the idea of building relationships with your customers, something that is invaluable in regards to promoting loyalty and confidence in your brand. Research firm Access Development reported that 79% of customers would take their business to a competitor within a week of experiencing poor customer service. With this in mind, it's vital that each touch point has the customer experience at the forefront for any decision that is made. By encouraging repeat business through direct and personalised marketing, e-tailers have the opportunity to boost ROI and increase engagement and customer retention.

> **"It's vital that each touch point has the customer experience at the forefront for any decision that is made"**

This idea of personalisation also allows brands to be a bit more... well... personal. After a somewhat sombre 2016, 2017 is the year for brands to concentrate on personality. Don't be afraid to be more tongue in cheek, and show that your brand has character and that there are people behind it. This can come through in the tone of the content that you produce, the way you interact with your customers and the way you portray yourself on social media. A lot of clients understand their current customer base, but social media is a fantastic way to engage with a new demographic and audience pool on a global scale. Within reason, it's likely that anyone who comes across your brand is a potential customer and it's therefore your interactions with them that will encourage them to engage, consume and convert.

And now for the fun part: virtual reality. There was no running away from it in 2016; it made many appearances throughout fashion weeks and is already firmly planting its feet in the beauty market with brands like WAH Nails. Mixing fashion and beauty with cultural experiences, WAH has introduced a 'Salon of the Future' for its active millennial fanbase. Using digital technology to create a seamlesss customer journey both on and offline, after making a appointment with the upcoming online booking tool, customers then use in-salon virtual reality platforms to create their own nail art and designs which are then ported to the relevant nail technician's station, ready for action. Designs are subsequently pushed back across social channels, completing the link between digital and retail.

Far from being a gimmick, VR runs alongside the idea of the ultimate in personalised customer service and developing an enhanced customer experience, nurturing clients and encouraging confidence in their purchase decisions.

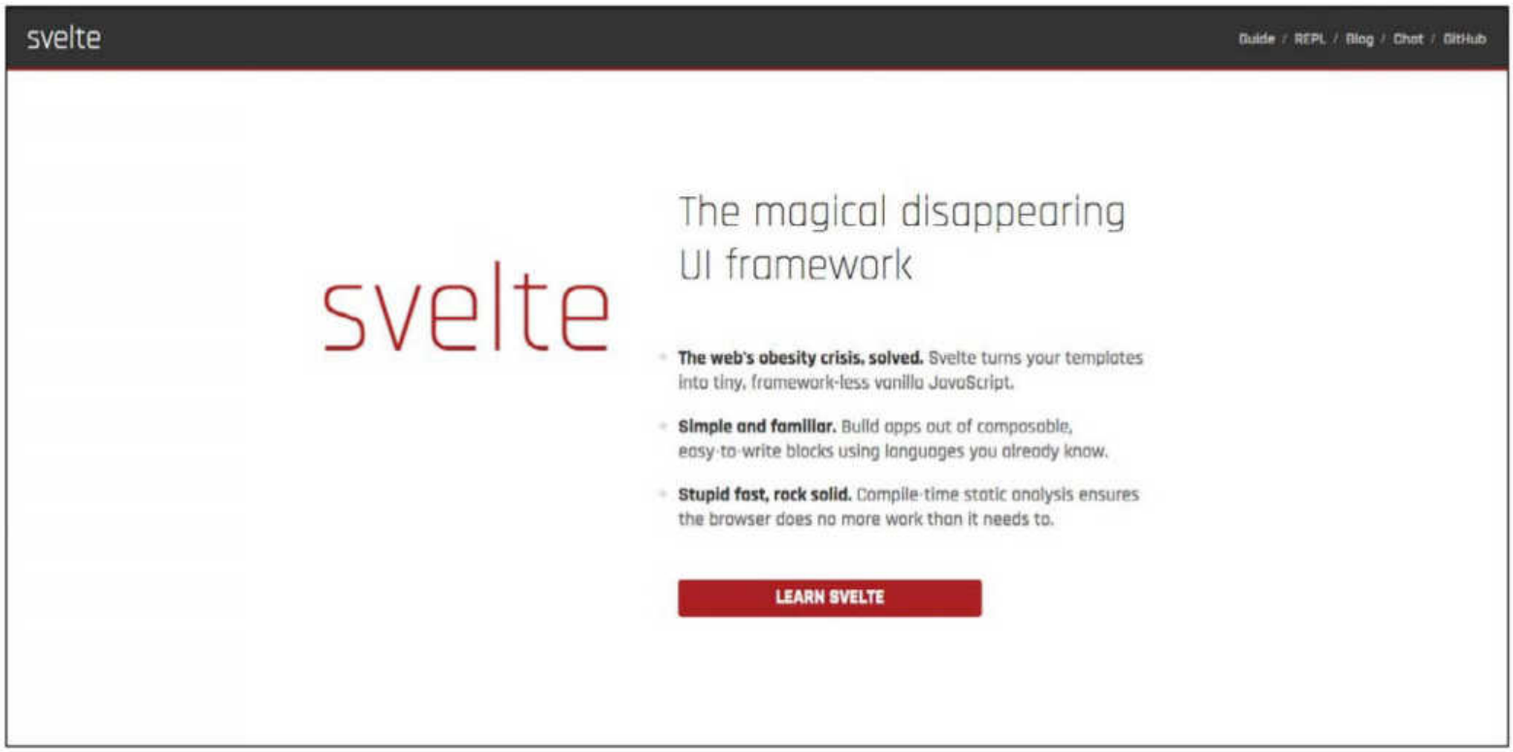
So – we have courted the user, encouraged them to engage and we're looking to convert them. VR systems such as the Virtual Mannequin from Metail give the added layer of confidence that will nudge users over the line. The cutting-edge 3D visualisation is a market leader and its algorithm-led fashion bots continually learn over time, giving accurate and personalised size and style advice. This feature has led to a 22% uplift in sales and guess what – returns have fallen by 5% too.

For 2017, this is your agenda: be disruptive, be more playful, send relevant content to your customers via segmentation and experiment with more engaging content, be it a GIF or via a full-on VR experience.

*CONTACT US AT:* webdesigner@imagine-publishing.co.uk | @WebDesignerMag

# webkit

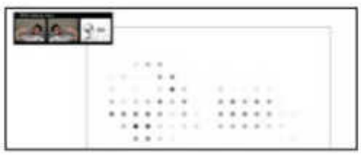
Discover the must-try resources that will make your site a better place




## Svelte

**svelte.technology**

JavaScript is a linchpin of building for the web, and Svelte is a new framework to help make the process easier. It shares similarities with React, Angular, Vue and Ractive, but it is different. It interprets application code at runtime, meaning an app is converted into ideal JavaScript at build time. Which means it performs better.



## Diffy.js

**github.com/maniart/diffyjs**

Diffy is a dependency-free motion detection library for the browser. It grabs snapshots from a webcam and combines them to find the difference.

## HTML Canvas Gauges

**github.com/Mikhus/canvas-gauges**

Need gauges to reinforce a point you're making? Try out these visually appealing and highly configurable gauges that use pure JavaScript and HTML5 canvas.

## Vocalizer

**vocalizer.io**

Is your name hard to pronounce? Want people top get it right? Add Vocalizer to your website and instantly make your name more accessible.

# TOP 5 Web conferences – February 2017

Get yourself a seat at the biggest and best conferences coming your way soon


## Digital Thinkers

**bit.ly/2iDE3qU**

Inspirational talks about UX/UI design, VR, Semiotics and wearables. Plus VR and data visualisation workshops.


## Reasons.London

**reasons.to**

Design, Code, Create & Come Together... Reasons gathers the best speakers from design and code to inspire attendees.


## FITC Amsterdam X

**fitc.ca/event/am17**

Designers, developers, motion graphics artists, digital artists and makers should attend.

## Mobile World Congress

**mobileworldcongress.com**

The world's largest gathering for the mobile industry with inspirational speakers.


## UX Camp

**bit.ly/2iDDUE2**

UX Camp will ensure that you get to learn about the hottest UX topics and learn something to make you even better.

# Web apps: which platform to build?

With the help of NativeScript, you can build for all platforms with one solution

**Dan Wilson** | Product Marketing Manager, Progress | progress.com

When it comes to building an app, it is often a case of 'which platform do we prioritise?' The problem with taking on more than one platform at once – Android and iOS, for example – is that you might as well be starting two projects at once. Budgets are rarely infinite, so a choice of 'either/or' often needs to be made, usually on the basis of what skills you and your team have.

The problem here is that as soon as you make this choice, the impact your application can have is dramatically reduced: Apple estimates that there are more than one billion iOS devices in use globally, while Android blew past that milestone way back in 2015. This doesn't just apply to consumer applications – it's unlikely that every employee in a company is running the same OS. While market share has shifted in Android's favour in recent years, hitching your wagon to one or the other won't cut it.

This means finding ways to make development as efficient as possible.

> **“ NativeScript provides direct access to each and every native platform API on both iOS and Android ”**

So what's the problem? Most developers are specialists in a handful of languages. They may be able to dabble in others, but not enough to deliver the quality of application they want or need. Look at this way: your friend may be able to order a bus ticket, but you wouldn't let him negotiate your ransom in Spanish. With Android being primarily based on Java and iOS on Objective-C, you're going to need high-level ability in both to create native apps for both. Even if you are, your workload is doubled.

In addition, if you want to develop for both iOS and Android using the official methods, there are various other problems to deal with. It will require familiarising yourself with new Software Developer Kits, installing new integrated development environments and learning new APIs. Although there will be similarities, each platform has its own specifics and affords certain flexibilities in others. Basically, it's twice as hard, twice as expensive or both.

Ah, but haven't we heard of web apps? It's certainly true that there is an obvious workaround for the problem of cross-platform development. Web apps can be built once and scale across any platform and any device. The benefits are obvious: development resources are maximised and so too is the reach of your application. However, without being able to access certain native device functions like the camera or geolocation, web apps have limited capabilities.

This is where hybrid apps come in. If you've developed a hybrid application before, you've probably used a framework like Apache Cordova/PhoneGap to make the most of your web knowledge to build what is nearly a native application. The key here though is 'nearly'.

Hybrid applications are good enough for a lot of scenarios, but at the same time not good enough for a lot of them. Hybrid application performance suffers as applications grow in size. In addition, it can be difficult to tell whether a mobile app is hybrid or native. A well-written hybrid app shouldn't look or behave any differently than its native equivalent, but they often do. A hybrid app needs to manipulate the DOM (which is an expensive task), and any noticeable lag can sour the user experience so that your app ends up in that unloved folder on your home screen.

So, you still want to go native? You want to make the best app possible, so that means you want a native app. Luckily, there is a solution out there. NativeScript is an open source framework which allows developers to write native mobile apps in JavaScript for Android and iOS with far less code than a hybrid mobile app, and in less time, without the need to open integrated development environments such as Xcode (iOS) or Android Studio.

NativeScript provides direct access to each and every native platform API on both iOS and Android. Using JavaScript, simply call the appropriate Java, Objective-C or Swift control and use it just like you would if it were written in JavaScript. You always have full control over the native platform, without writing native code. Further, NativeScript provides a comprehensive set of cross-platform abstractions to simplify development. Need a button? Just program in the NativeScript button in your application. Without you having to do anything else, NativeScript will communicate directly with the underlying Java on Android and Objective-C on iOS.

The runtimes give you the power of accessing the native APIs, SDKs and other frameworks, alleviating the previously mentioned problems when attempting to develop for both platforms using traditional methods. No need to reinvent the proverbial wheel, so to speak. You can consume thousands of pre-written packages on npm, CocoaPods, Android Arsenal and other popular open source repositories.

As a developer, you're well-accustomed to solving problems and finding your way around the insurmountable. NativeScript is the tool which can help you round your biggest challenge: cross-platform development. There's really no need to limit your potential audience any more; use the skills you have to make great applications that benefit your audience with NativeScript.


# Mozaik

**www.mozaik.com**

**Designer: Mozaik - mozaik.com**

**Development technologies -** WordPress, jQuery, SVG, Vimeo

*"This creative corporate site for Greek agency Mozaik visualises the tagline 'Vibrant Synergy' with pulsing organic animations"*

Sound

#D1D6FB

#5C42C4

#7183F6

#261A65

abcABC

**Above**
Campton Std font, by designer Rene Bieder, appears in both Light and Bold forms across the majority text.

abcABC

**Above**
Adobe's Letter Gothic Std font is used specifically to render the job titles within the People section.

# Add animation to a hamburger menu icon

Make use of CSS animation to create an animated hamburger icon for your website menus

## 1. Initiate document

The first step is to initiate the HTML document. This consists of the HTML document container, which includes sections for the head and body. The head section contains descriptive information and links to external resources – in this case the CSS stylesheet. The body section will be populated with the visible content in the next step.

```
<!DOCTYPE html>
<html>
<head>
<title>Hamburger Effect</title>
<link rel="stylesheet" type="text/css"
href="styles.css" />
</head>
<body>
*** STEP 2
</body>
</html>
```

## 2. Navigation content

The menu is made from a navigation container that stores a series of links. The first link in this container will be the hamburger icon; a link that will reference the navigation container. The hamburger icon has three span elements, which will be styled to show as the hamburger bars in a later step.

```
<nav id="example">
<a href="#example">
<span></span>
<span></span>
<span></span>
</a>
<a href="#">Alpha</a>
<a href="#">Bravo</a>
<a href="#">Charlie</a>
</nav>
```

## 3. Start the stylesheet

The HTML markup is now complete, so create a new file called 'styles.css' to start defining the CSS formatting. The first entry into this file will be to define the document and content body to display with a dark background without border spacing. We also set the font that all elements will inherit.

```
html,body{
display: block;
width: 100%;
height: 100%;
margin: 0;
padding: 0;
font-family: Helvetica, sans-serif;
background: rgb(21, 34, 47); }
```

## 4. Navigation container

The navigation container will be set to display at 25% of the document width without any margin. This is important if we don't want the navigation to display over the full page, hence defining the width controls how wide the menu is to be displayed at. A block display is also used to guarantee that width sizing will work regardless of how a browser treats navigation containers by default.

```
nav{
display: block;
width: 25%;
margin: 0; }
```

## 5. Default navigation links

Border and width settings are applied to the links within the navigation container so that they appear prominently at the same size defined for the navigation width. Navigation links are not visible by default, but become visible when their parent container is targeted; hence the reason why the HTML link of the hamburger icon references its parent navigation container.

```
nav a{
display: none;
width: 100%;
padding: 10%;
font-size: 1.5em;
border: 3px solid #fff;
background: #333;
color: #fff; }
nav:target a{
display: block; }
```

## 6. Hamburger style

The hamburger icon will always be the first element inside the navigation container, hence we reference it using the 'nth-child' selector. Firstly, we set this to display as a block of five character width. Secondly, the inner elements are styled as white bars in relation to the previously set size. Each bar has a specific margin and height set to appear visible and separate from each other.

```
nav > *:nth-child(1){
display: block;
width: 5em; }
nav > *:nth-child(1) span{
position: relative;
display: block;
width: 100%;
height: .5em;
margin-top: .25em;
background: #fff; }
```

## 7. Animation: preparation

The animation effect will be made from the 'before' element that CSS can apply to web page elements. This will add a bar with a red background applied to each of the span elements defined in the previous step. With their parent span using relative positioning, the position of these elements will be positioned in relation to their parent. These elements are initiated with no width, ready to be animated in the next steps.

```
nav > *:nth-child(1) span::before{
content: "";
position: absolute;
top: 0;
left: 0;
width: 0;
height: 100%;
background: red;
transition: width 1s;
}
```

## 8. Animation: bar settings

Firstly, the before elements of each of the hamburger bars need to have the 'open' animation applied to them when their parent link is being hovered. This animation will be played 'forwards' to ensure that it stops on the last frame. Secondly, the second and third bars need to have different delays applied to them for the full effect to work.

```
nav > *:nth-child(1):hover span::before{
animation: open 1s forwards; }
nav > *:nth-child(1):hover span:nth-
child(2)::before{
animation-delay: .15s; }
nav > *:nth-child(1):hover span:nth-
child(3)::before{
animation-delay: .3s;
}
```

## 9. Animation: definition

The 'open' animation applied to the before elements is a simple two frame animation that changes the width from zero to the full width of the parent burger bar. The browser will fill in all of the animation frames required between these two states to show a smooth transition lasting the time specified in the previous step.

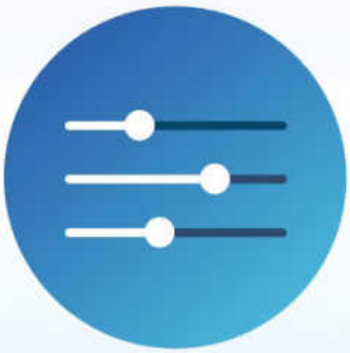
```
@keyframes open {
0%   {width: 0;}
100% {width: 100%;}
}
```

# NEW

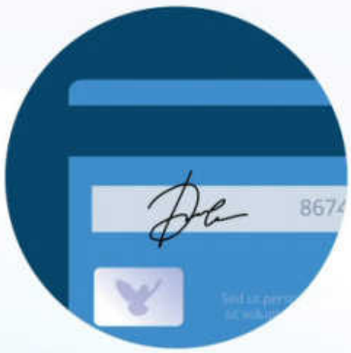
# CloudNX

## next generation Cloud Servers

**UK data centres, UK support**



**Flexible**



**Pay as you use pricing**

### Enterprise features

- Load balancing
- VPN
- External firewall

### Set up in seconds

- Easy to use control panel

**Cloud Servers** from:

**£15.00**

per month ex VAT charged at 20%

Call **0808 1686 777**

or visit **fasthosts.co.uk/cloud-servers**

SERVERS · WEB HOSTING · DOMAIN NAMES · EXCHANGE EMAIL

fasthosts


# UZUZ Corporation

**https://uzuz.jp**

**Designer: Bees&Honey Inc - http://beeshoney.jp**

**Development technologies** WordPress, jQuery, GSAP, JSON 3, VideoJS

*“UZUZ is an agency specialising in transitioning Japanese graduates into the best jobs and ‘entertaining careers’ via this energetic website”*


#FFE2AE

#E66700

#2C1F14

#D49B66

abcABC

**Above**
The Coalhand Luke font, by designer Jeroen 'Joebob' van der Ham, styles the distinctive handwritten heading

abcABC

**Above**
Hiragino Kaku Gothic Pro by JIYUKOBO Ltd is employed to display the predominant Japanese typography throughout

# Create a shivering text effect for navigation links

Use CSS animation features to apply a shivering text effect for navigation links on hover

## 1. Define the document

The first step of the project is to define the HTML document. This consists of the HTML document definition, which itself includes a head and body section. The head section is used to contain page meta descriptions and links to external resources, while the body is used to contain the visible page content.

```
<!DOCTYPE html>
<html>
<head>
<title>Shiver Effect</title>
<link rel="stylesheet" type="text/css"
href="styles.css" />
</head>
<body>
*** STEP 2
</body>
</html>
```

## 2. Shiver content

The shiver content will be a navigation container that has several links. This navigation container will have a class of 'shiver', which is what will be used to reference the items that will have the animation effect applied to them in the upcoming steps for styling the web page content.

```
<nav class="shiver">
<a href="#">Alpha</a>
<a href="#">Bravo</a>
<a href="#">Charlie</a>
</nav>
```

## 3. Stylesheet initiation

The next step is to create a file called 'styles.css', used to define how the web page content is to be presented. Our first step will be to define the page HTML document and body containers to have a dark background with no border spacing at all. Additionally, links are set to a shade of red.

```
html,body{
display: block;
width: 100%;
height: 100%;
margin: 0;
padding: 0;
font-family: Helvetica, sans-serif;
background: rgb(21, 34, 47);
}
a{
color: #c00;
}
```

## 4. Shiver: initiation

The animation effect is applied to all first level children of the container that has the shiver class. These elements are displayed as block elements with relative positioning. This is important because the animation positioning needs to be set relative to where the elements are originally located.

```
.shiver > *{
display: block;
position: relative;
}
```

## 5. Shiver: application

The shiver effect is activated when the user hovers the mouse pointer over one of the child (link) items. This in turn triggers an animation that lasts for a fraction of a second (.15) that repeats infinitely. We achieve this with the use of the hover selector and the animation attribute to specify the animation to call.

```
.shiver > *:hover{
animation: shiver .15s infinite;
}
```

## 6. Shiver: animation definition

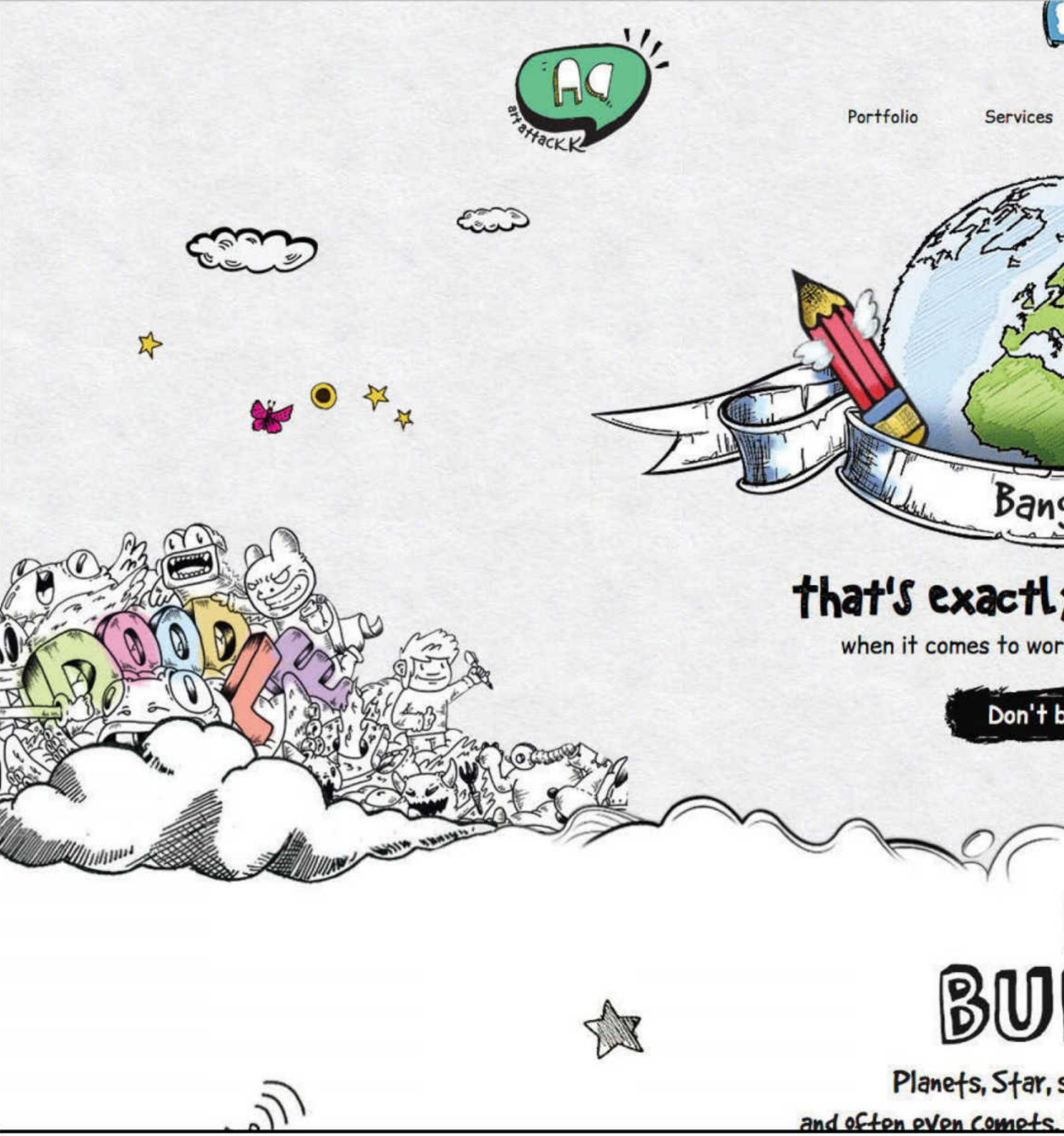
The shiver animation is defined as several keyframes. The first and last frames set the element to be positioned in their original location – ie zero pixels to the left. This allows the animation to repeat without appearing to jump back to the original state. The two other frames move the element backwards and forwards by 5 pixels; change this number if you want the animation to swing the text further.

```
@keyframes shiver {
0% {left: 0;}
25%   {left: -5px;}
75% {left: 5px;}
100% {left: 0;}
}
```


## Usability consideration

This effect is best for situations such as navigation hover where users are not focusing on trying to read the text.


LOSE YOURSELF IN A WORLD OF
Vinyl
FIND YOURSELF IN
OXFAM'S ONLINE SHOP
oxfam.org.uk/shop
TIME ON MY HANDS
OXFAM



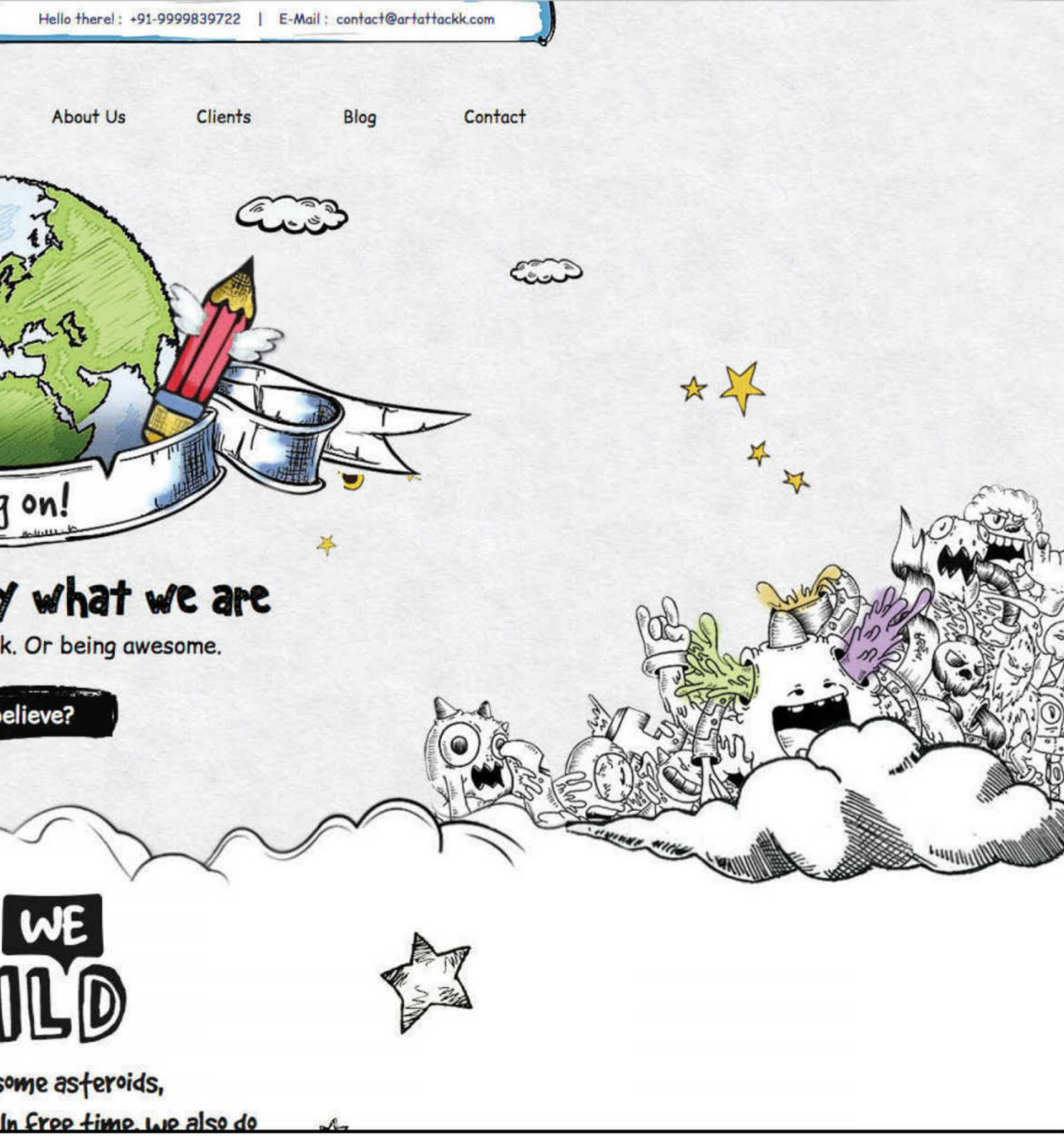

# Art Attackk

**https://www.artattackk.com**

**Designer: Art Attackk - www.artattackk.com**

**Development technologies** · HTML5/CSS3, jQuery, jQuery Colorbox, GSAP

*"Art Attackk is a web design agency based in Delhi, India and this charming portfolio site uses handdrawn visuals to convey a playful approach"*




#69C291

#EC6B81

#EEE777

#CDE7FC

**Above**
KhrysKreations' KBZipaDeeDooDah font is used in Medium, distinctively styling the paragraph headings

**Above**
Comic Zine OT by Blue Vinyl brings a sketchy graphical look to the typography across main section headings

# Add a dynamic animated effect to a static image or object

Use CSS animation and selectors to apply an overlay animation to web page content elements

## 1. Page document

The project is started with the HTML page template used to define the head and body section. The head section is used to define non content items; in this case the CSS stylesheet reference and descriptive title. The body section is used to store the visible content elements, which will be inserted in the next step.

```
<!DOCTYPE html>
<html>
<head>
<title>Animation Effect</title>
<link rel="stylesheet" type="text/css"
href="styles.css" />
</head>
<body>
*** STEP 2
</body>
</html>
```

## 2. Page content

Our example content consists of three letters that will have the animation effects applied to them. Each letter is contained within a span element, which is used to apply class attributes. These attributes define the presentation settings for the letters, as well as describing which animations to apply to them.

```
<span class="animation A big">A</span>
<span class="animation B big">B</span>
<span class="animation C big">C</span>
```

## 3. Stylesheet initiation

With the HTML markup now complete, the next step is to create a file called 'styles.css'. This file will be used to define the CSS presentation rules. Initiate this document with the general page styling settings of the main HTML and body containers. We will set this with dark background and bright text colours, along with the standard font and no border margin or padding.

```
html,body{
display: block;
width: 100%;
height: 100%;
margin: 0;
padding: 0;
font-family: Helvetica, sans-serif;
background: rgb(21, 34, 47);
color: #fff;
}
```

## 4. Big text

Our example uses big text items to present the surrounding animation effects. We get this by setting the size of the letters using the 'big' class, which will set the font size to be three times bigger than the default size. We also set the margin to be the size of three text elements on all sides so that we can clearly see each element individually.

```
.big{
font-size: 3em;
margin: 3em;
}
```

## 5. Default animation settings

Each element shares a set of default settings that apply to all of their animations. To avoid repetition, we define these defaults as part of the 'animation' class. These elements must use relative positioning so that the 'before' animation elements can be positioned in relation to them. We apply settings to the 'before' elements for their border, size, colour and animation that are to be their default.

```
.animation{
display: inline-block;
position: relative;
}
.animation::before{
content: "";
position: absolute;
top: 0;
left: 0;
width: 100%;
height: 100%;
border: solid #ccf;
box-sizing: border-box;
animation: 1s infinite;
}
```

## 6. Individual animation settings

With the default animation defined, the next step is to apply the individual animation settings. Each element has a class name A, B or C, which we use to reference in combination with the 'animation' class. We define the name of the animation to use, along with other unique settings. In this case, we use the 'ping' animation for all elements, with unique border styles to change their animation appearance.

```
.animation.A::before{
animation-name: ping;
}
.animation.B::before{
animation-name: ping;
border-style: dashed;
}
.animation.C::before{
animation-name: ping;
border-style: double;
}
```

## 7. Animation definition

The final step is to define the 'ping' animation. In this case, the animation has two frames. The first frame has the border size set to a quarter of the current font size with full opacity. The second frame has the border size set to zero with no opacity. The 'before' elements will animate between these two states to give a 'ping' effect.

```
@keyframes ping {
0% {border-width: .25em; opacity: 1;}
100% {border-width: 0; opacity: 0;}
}
```

## Drawing attention

Consider how you want to draw attention to content elements without being too much of a distraction or annoyance for your visitors.

# EXPLORE THE TECH INSIDE

www.gadgetdaily.xyz


Gadget™

Available from all good newsagents and supermarkets

## ON SALE NOW

■ SAVE £5,000 A YEAR ■ TECH TO SUPERCHARGE YOUR LIFE ■ PLAYSTATION 4 PRO TIPS


# BUY YOUR ISSUE TODAY

**Print edition available at www.imagineshop.co.uk**

**Digital edition available at www.greatdigitalmags.com**

Available on the following platforms

Print Tablet Phone Desktop

facebook.com/gadgetmagdaily twitter.com/gadget_magazine

# CREST OF


**PROJECT**
CLIPPER ROUND THE WORLD YACHT RACE

**WEB**
CLIPPERROUNDTHEWORLD.COM

**AGENCY**
Salad

**WEB**
saladcreative.com

**FOLLOW**
@Saladcreative

**PROJECT DURATION**
13 months, 2 weeks, 3 days

**NO. OF PEOPLE INVOLVED**
6

**TOTAL MAN HOURS**
1770

## PERSONNEL

**ADAM MORLAND**
Digital Director

**ARABELLA LEWIS-SMITH**
Managing Director/
Account Director

**DAN YOUNG**
Design Manager

**MATT ABREU**
Designer

**LAWRENCE DONALD**
Creative

**JAMES HILL**
Web Developer

# A WAVE

## FACING THE CHALLENGE OF COVERING ITS ICONIC ROUND THE WORLD YACHT RACE, WE HEAR HOW CLIPPER BROUGHT SOME SALAD DAYS TO A WEBSITE CELEBRATING SAILING'S MOST PRESTIGIOUS EVENT

### AWARDS

**LOVIE**
GOLD, Sports

**THE DRUM DADI**
Finalist, Sport Category

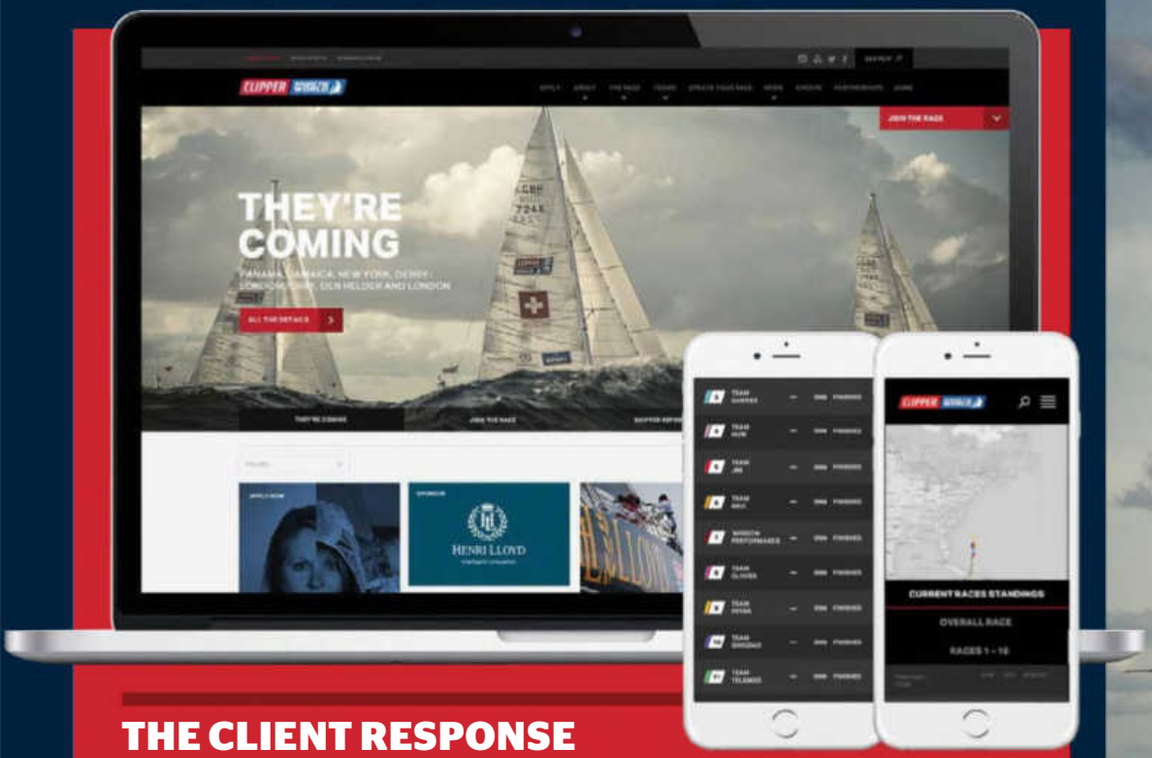
**THE DRUM DADI**
Finalist, Technical Innovation

**WIREHIVE100**
Finalist, Digital Innovation

**BIMA**
Finalist, Website of The Year, Sport Category

**DBA**
Finalist, Digital & Creative Impact 2016

**Without getting too philosophical, occasionally it's appropriate to stop and ponder what the internet has really done for us.** Aside from the obvious benefits for Mark Zuckerburg's bank manager and the ability to watch skating accidents on YouTube anytime, anywhere – some would argue that bringing us all closer together is a neat side effect. The ease of information exchange across the World Wide Web has made the world in general seem considerably smaller, something that can be especially useful in certain situations. Take for example, traversing the globe in a boat. Not just bobbing along leisurely either, sipping cocktails, but battling the elements at a competitive rate of knots. In those moments, the world can seem a very big and fairly lonely one, so our featured site this month is all about digitising that experience to share the thrill. The Clipper Round The World Yacht Race is a gruelling endurance sailing challenge over 40k nautical miles, comprising eight legs and at least 14 individual races. Uniquely, however, it is open to all and invites everyone of any background or ability to throw down a challenge and conquer volatile tropical seas in pursuit of glory. So it figures then that such an embracing event would want to leverage the web to make registration, participation, coverage and celebration of such a race more accessible. As official organisers, client Clipper recruited an agency already not just familiar with branding and communications, but also a taste for sea air. Salad, boasting a Poole office on England's south coast, was enlisted to head an expansive online initiative. "Clipper reached out to us because they needed a new digital campaign primarily focused on a new main website with two further websites and a web app waiting in the wings," explains Arabella Lewis-Smith, Salad's Managing Director & Founder. "These would be launched in three stages to generate crew recruitment, attract sponsorship and showcase the awesome story of the Clipper Round the World yacht race." Throwing around words like "vast" and "epic", the team knew straightaway that the project called for the shackles to be cast away to inspire future sailors.




## THE CLIENT RESPONSE

"Since the launch of the site designed and developed by Salad 24 months ago, we've seen a 45% increase in hits to the site," Katharine Schäfli, Global Head of Brand Clipper. "During this time, we've completed one business cycle and analytics show new visitors to the site visit 4.5 pages, staying on the site for 4 minutes. But perhaps more remarkably 78% return to the site meaning improved engagement with the brand, and this is born out in the ever greater number of Race Crew signing up to do the Clipper Race, around the world."

Buoyed by those encouraging stats post-launch, it's nice to hear from a satisfied client and for Salad to realise the deliverables are working. An important early stipulation was to better cater for a broader outreach to users across multiple devices and this has been particularly successful in terms of reaching new people. "The responsive design appeals to our target audience, people on-the-go from all walks of life and nationalities. The Clipper Race URL is widely publicised on all our marketing collateral and increasingly digital marketing spend is attracting new audiences to the site, especially Europe and the USA. Plus in the recent months we launched a Chinese language version of the site. As a result of the improvements to the website, we cut our print marketing by 25%."


## ON YOUR MARKS...

Having established that the project was so multi-faceted and would comprise a series of stages, Clipper supplied Salad with one of the most detailed briefs the team had ever seen. The three stages that Arabella mentioned would span pre-race, race warm-up and indeed a race mode to provide keen followers with real-time coverage of event progress. Initially, content would pitch participation to would-be entrants and sponsors while supporting the application process in engaging ways. "The details in the initial brief included the hosting spec, visual design requests, information architecture and timescales," begins Digital Director Adam Morland. "The specification document gave us a great basis to kick-off the project and move into the planning phase of the process. Once our planning was complete, we had a clear understanding of the project requirements and the team pulled around a table for a kick-off meeting. Tasks were assigned in sprints and we worked iteratively with the Clipper team from the beginning to ensure we were fulfilling their vision for the site." This typically collaborative approach also benefitted from applying an 'open-minded' rule of thumb that Salad has developed over 16 years, being mindful of a client requirements but never limiting project scope by a strict 'to-do' list. "We approach projects from our clients' key objectives – both in terms of the project we're working on and where it fits into the larger picture. For this website, that was the creation of the responsive Race Viewer. This feature allows for an engaging and user-friendly way for followers to track progress on the global ocean race from anywhere, and importantly, on any device. We had a fantastic archive of action photography from the races to draw from and a large body of content to present we adopted a clean, modular style to the website that allowed for the effective communication of strong messaging to inspire users to apply and experience the excitement for themselves. We ensured that the experience across desktop and mobile was smooth and easy to use with dynamic feeds, crew recruitment calls to action and a variety of content layouts. Whether the users are engaging with the race from their desk or from their own boat the experience and design is not compromised."

## RESCUE MISSION

Without being too dramatic, the team mentions the


*L-R: Adam Morland - digital director, Arabella Lewis-Smith - managing director/account director, Matt Abreu - designer, Dan Young - design manager*

notion of a "rescue" when recalling how initial contact was made between the two teams. It could be said that Clipper found itself a little lost at sea with its previous website, facing numerous issues at a crucial point of the 2013-14 race calendar. Salad answered the 'SOS' call quickly to initiate a relationship of trust and awareness, which would feed into such a collaborative effort again here. "Projects of this size can be extremely complex and working with Clipper as partners, keeping open channels of communication ensured the reduction of all sorts of typical delays," affirms Arabella. "We are extremely flexible when it comes to the preferred communication channels of clients. Like any business, we have a few key channels that we use internally but will talk to clients in the channel that best suits them. For example, some clients based in London or beyond prefer to use Skype to feel more 'in the

## SITE HIGHLIGHT

**IN THE WAKE OF THE SITE'S SUCCESS, DIGITAL DIRECTOR ADAM MORLAND DESCRIBES A HIGHLIGHT FOR THE CLIPPER PROJECT AND WHY IT MATTERS**

**"People have lost their lives in this incredibly dangerous race and creating an industry-first Race Viewer allowed family, friends and fans to keep up with the action from anywhere. Reflected in the stats, the viewer received 3.6m page views during the 2015-16 race and it's something we're extremely proud to have delivered as a team. We look forward to any future innovations we have the opportunity to create with Clipper"**

room'. Clipper were fairly traditional so our communications were generally done in face-to-face meetings which we prefer, on the phone or by email. We then make the most of any channel mix by following up calls and meetings with an email that outlines what was discussed and the key action points to focus on. This ensures we keep up the same momentum beyond being 'in the room'. As the timeline was so tight on this project, with much to deliver, we also sent a detailed weekly progress report that was a welcome addition to the process."

## BOW-END BUILDING

With Clipper defining homepage content already within the very detailed brief, the team was tasked with producing a new design that would present news, social media updates and race messages in a compelling way. Staying true to the client's established branding while also adding freshness, there was also the acknowledgment that fans increasingly access the site on a myriad of devices. "Our first task was to undertake a period of research and immersion," explains Design Manager Dan Wilson. "We spent plenty of time devouring the pages of various sport and news websites to become as knowledgeable as possible about what already existed. While we wanted to create something original and unique for Clipper, it was critical to learn from the mistakes made by others. Perhaps most importantly, we could see how other brands or sporting events were handling similar types and quantities of information. After that, we decided that some sort of grid would be most idyllic and much of the initial design resource was focused on establishing the perfect balance in terms of the size, quantity and hierarchy within the layout." An original homepage design had in fact been produced for the initial pitch, layering on a 'wow factor' that demanded an 'all-hands-on-deck' approach at odds with Salad's usual process. With project won and a final concept approved, the team reverted to its more conventional method, assigning a single designer sole responsibility of rolling out subsequent pages. "This included defining the font styles, image treatments and a significant amount of illustration work. Given the size of the project this was no small feat but the new website was released in three phases – making the task more manageable. The resulting designs were visually harmonious and totally consistent throughout the website journey. After the initial set of page templates had been approved, our web developers were able to help take some of the strain by rolling out the less critical pages using the pre-established style sheets. By this stage, the creative guidelines were so clearly defined that it was easy for other designers to join the project and work on subsequent phases, retaining continuity."

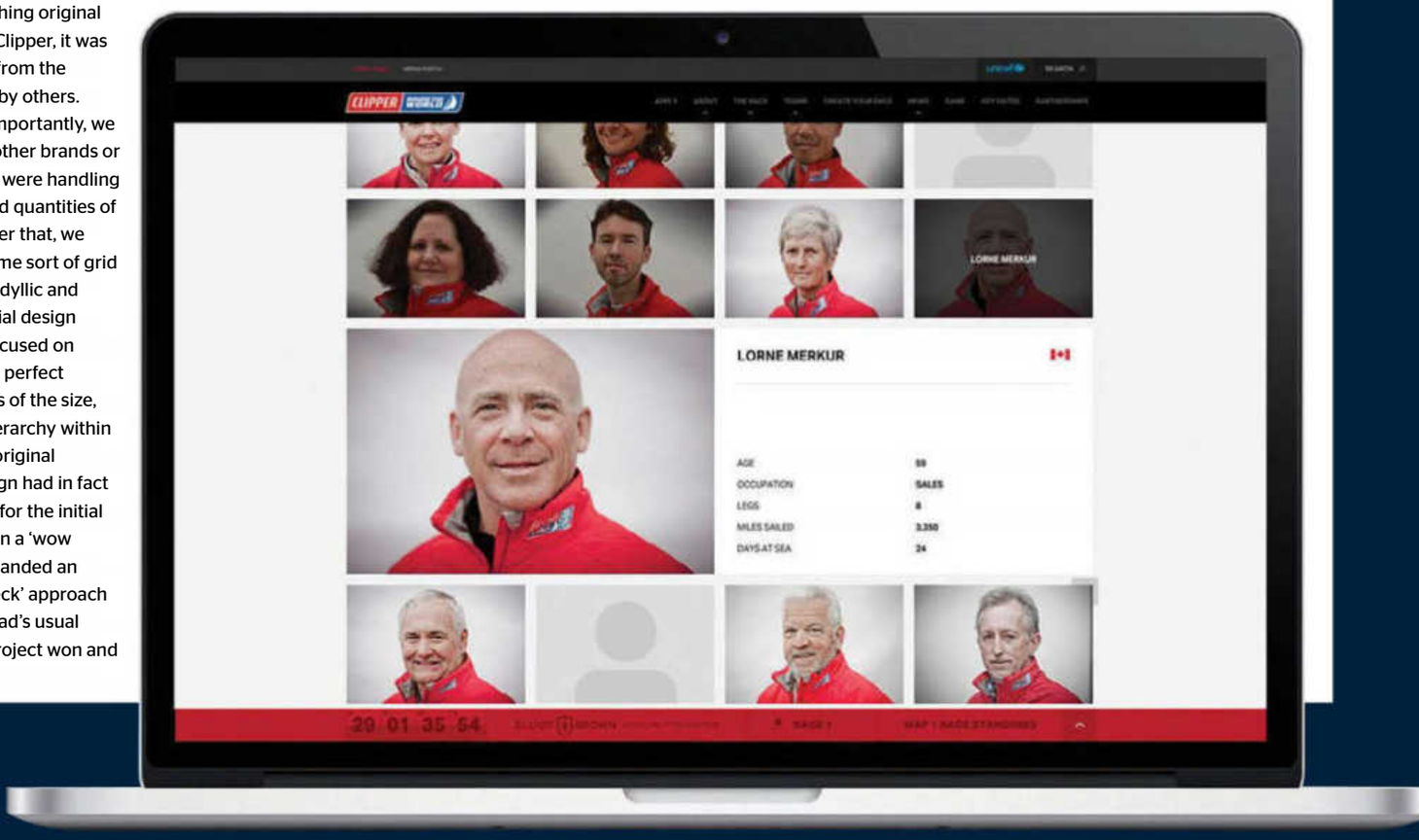
## FLOOD OF DATA

"The biggest coding challenge we faced in this project was the Race Viewer," admits Morland when quizzed on the back-end difficulties. "We had to work with a continuous transmission of raw GPS data from the yachts and process this to calculate all the key measurables the audience would expect. Then we were tasked with displaying this flood of data in as clean and interactive way as possible. Additionally, we created a suite of tools and overlays that enabled followers of the race to customise their viewing experience. Features included information about the current and future weather conditions and a ruler overlay to calculate the distance between boats. Behind the scenes, the GPS information is used by the Clipper Race Team to monitor team progress and provide an essential race safety tool." In terms of technologies, the developers decided to use a PHP framework over an off-the-shelf CMS to provide maximum flexibility. This was vital for such a multi-faceted project with three deliverable phases, including the main public website, Crew Hub, Race Viewer and Media Portal. The website would also be required to synchronise with Clipper's bespoke CRM, used to manage race applications and pre-race administration for crew members. "One challenge we addressed at the beginning, during the planning phase of the project was server specification and configuration. Due to the length of the race, a year long sporting event, we had to plan for significant surges in visitor traffic to ensure the website didn't struggle with concurrent users. Having shone light on this early on, it was a challenge we solved without any negative impact."

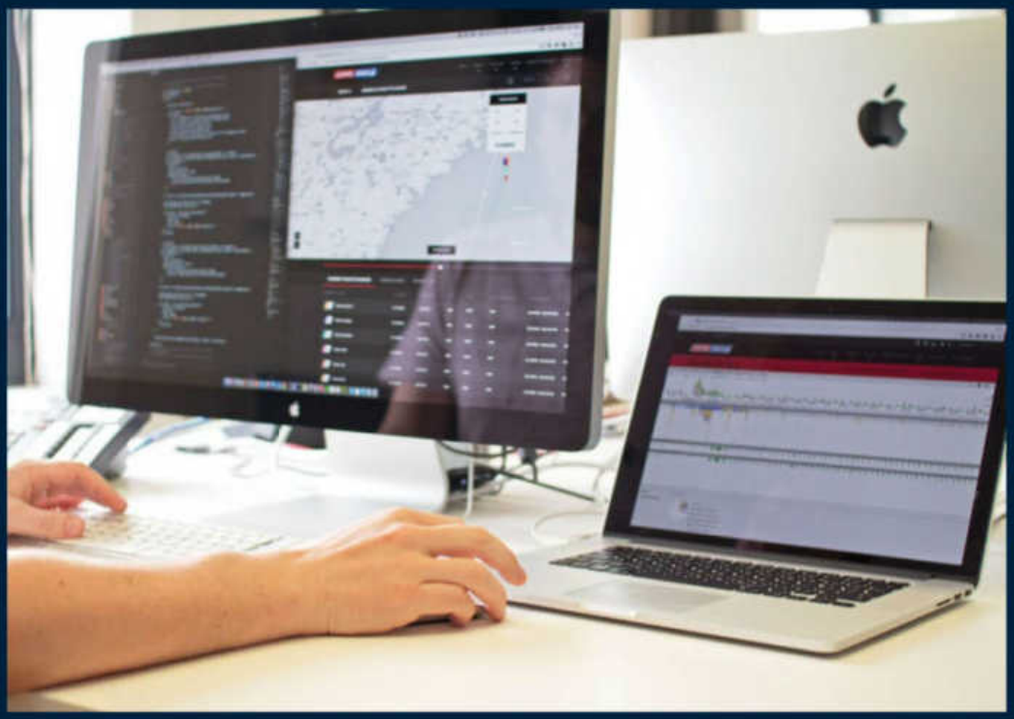
> **"The creative guidelines were so clearly defined that it was easy for other designers to join the project"**

## NAUTICAL BUT NICE

Those challenges in development would also impact on handover and site launch, when faced with communicating the finished project to Clipper. Perhaps like few others, this commission represented numerous integrated deliverables that depended on data support and sound administration going forward. "Before any

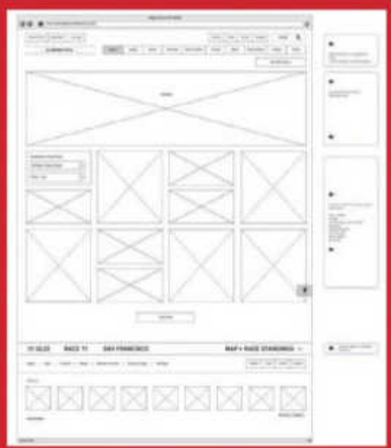


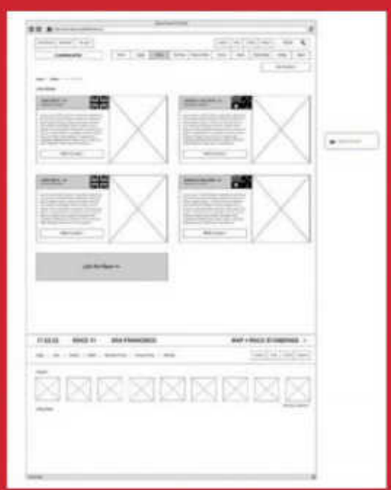
of our projects are put live we give all clients a CMS training session designed to help them navigate through the back end of their site," explains Sophia Taglialavore, Account & Studio Manager. "All and any questions, bugs, issue or amends can therefore be made well before the all-important live date. We find that investing this short but crucial time with clients at the beginning enables them to feel competent and gives them the independence to run their website without relying on us." Indeed with the site successfully launched, the continued work is about keeping it afloat by bringing in visitors to raise interest around compelling content and enlist new generations of race participants. "One of the main challenges for marketers continues to be keeping the website competitive, fresh, attractive and accessible," confirms Clipper's Global Head of Brand, Katharine Schäfli. "The feedback we receive of **clipperroundtheworld.com** is extremely positive which is encouraging. We continue to work with Salad to make the content effective and the functionality efficient."

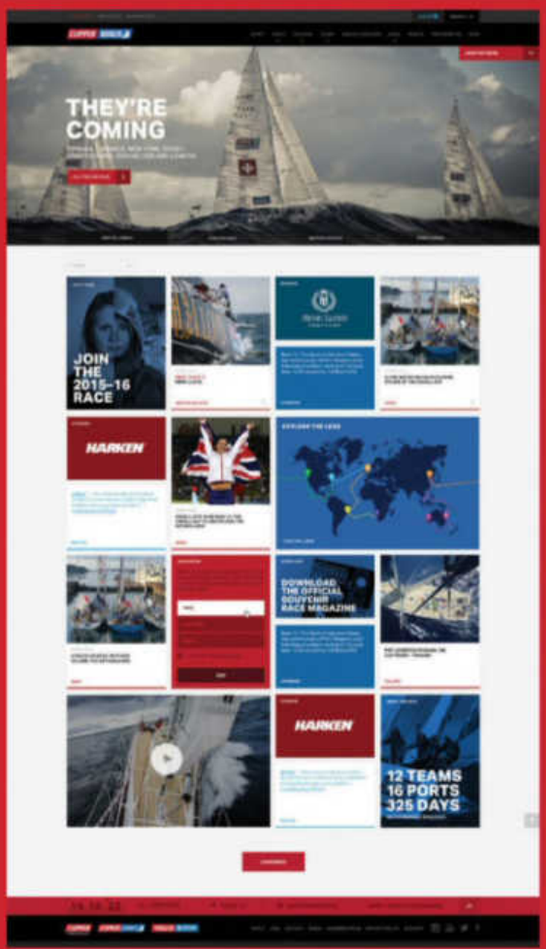


## CALMING CHOPPY WATERS

The Salad team are quick to stress that clients are never 'cast away' abruptly upon project completion. Quite the opposite, when asked about how much service support is provided at launch the first response is an extension of the strong communication policies employed throughout the process. "Upon go-live clients continue to receive regular communication from their designated Account Manager throughout each stage of the process," confirms Account & Studio manager Sophia Taglialavore. "This includes sensitive handling of propagation and why they might be able to see the new live site for 24 to 48hrs. There is then a small window of time where we are on-hand to answer any questions, make updates and generally be available should any additional support be required." In fact, all of Salad's website projects include a support and maintenance package of a minimum of three months but in many cases usually longer. Working like a bank of hours for clients to utilise however they wish, it provides some flexibility to best-fit unique requirements. "If they don't use the time, it rolls over or if they need less and less time, it is reduced so there's an obvious business justification. We're glad to report that while we maintain communication after every site launch, clients are busy handling their boost of new leads, enquiries and sales. Radio silence is usually the sign that we have another satisfied customer."







# Digital Denizens

**who** Clearleft

**what** Web design, user experience, front-end development, digital strategy, responsive design, typography, usability

**where** 68 Middle Street, Brighton, BN1 1AL

**web** clearleft.com

**With over a decade of experience, Clearleft has not only inhabited the digital space but shaped its landscape with innovations that impact on us all**

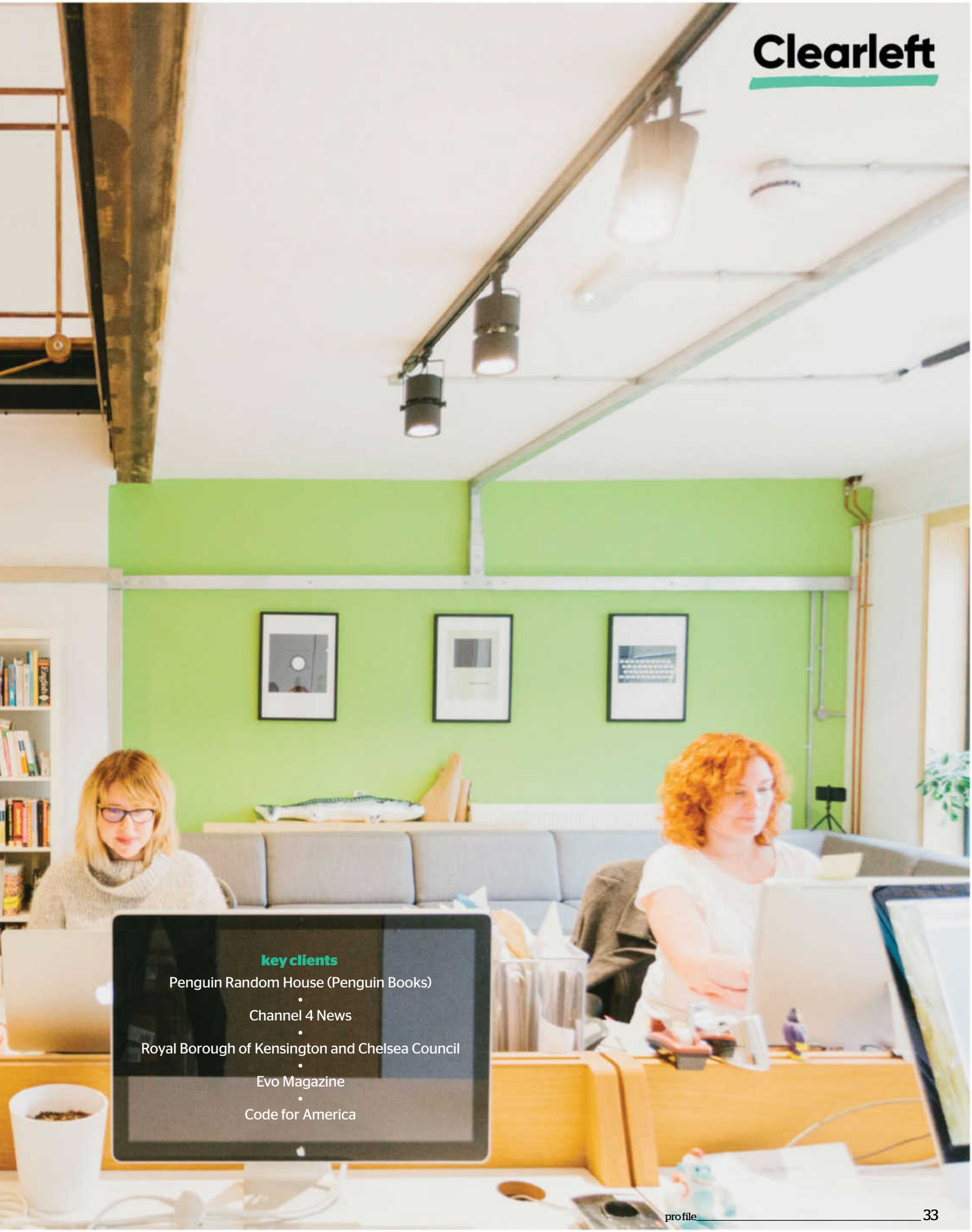
**Looking at the roster of clients that have come to Clearleft to help them develop and innovate within the digital space, it's clear that this agency defies the typecasting that many other agencies suffer from.** With a clear design sense and technical prowess to support their ambitions, Clearleft continue to work across many disciplines to form all our digital futures.

Clearleft started in 2005 as a meeting of minds between its three founders: Richard Rutter, Jeremy Keith and Andy Budd who were friends in Brighton, each working for different companies but sharing a love of design and of the web.

All three were avid bloggers and between them had a significant following of other designers keen to hear about the latest techniques. One evening early in 2005, Rich, Jeremy and Andy got together for a beer and decided that they should combine their skills – UX, front-end development and visual design respectively, and go into business together.

2005 was fairly early on in terms of the web as an 'industry', so all three founders had no formal training in web design or development as no such thing existed. Two of them – Richard and Andy – had backgrounds in chemical and aeronautical engineering, while Jeremy came from an art school education. What all three had in common – and this pervades through Clearleft now – is an autodidactic drive. A desire to self-teach and keep learning,

# Clearleft



**key clients**

Penguin Random House (Penguin Books)

•

Channel 4 News

•

Royal Borough of Kensington and Chelsea Council

•

Evo Magazine

•

Code for America

## Industry Insight

**Richard Rutter**
**Production Director and Co-founder**

"If you're looking to break into the industry, show that positive attitude. Show enthusiasm, that you're a self-starter. That you can apply critical judgement and talk through your process. Show that you can write well."

motivated by a love and fascination of digital design and the web as a medium.

There are differing schools of thought on the importance of an agency name and URL. Richard, director and co-founder explained their approach: "The agency name came first. It won't be lost on readers of this magazine that "Clearleft" is derived from the CSS 'clear:left.' We even used the colon for a while. There's no particular reason behind the name, but it does have a pleasing alliteration to it, and the 'clear' part retrospectively alludes to our design philosophy. It so happened that the .com was available too, but they were easier to come by 11 years ago."

With over a decade of experience, few other agencies have been able to witness and adapt to the massive changes to the internet. Is the online presence of its business important to maintain?

"An agency's website is extremely important," said Andy Budd, managing director and co-founder. "It is the one constant mouthpiece and showcase for what we do. That said, until just recently our website was rather dated and didn't reflect what we do all that well. We recently finished a long overdue redesign – complete with rebrand – which we soft launched around Christmas. We had to forego some interesting client work in order to keep a team focused on delivering the new site. Being perfectionists, and highly critical of our own work, these things take time, but by running some offsite hack-like design sessions we're pleased with what we've got and where we're heading."

**"Few other agencies have been able to witness and adapt to the massive changes to the internet"**

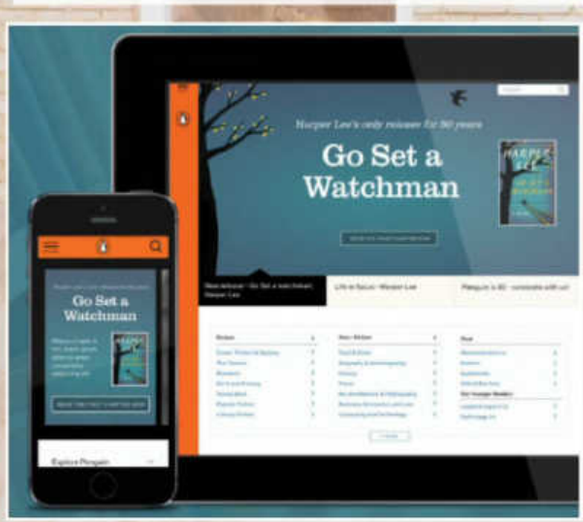
The roster of clients that Clearleft has worked with over the years is nothing if not impressive. From large media companies to high street names, Clearleft has attracted brands who needed not just

## Agency Breakdown

| | |
|---|---|
| 1 | Managing Director |
| 1 | Production Director |
| 1 | Technical Director |
| 1 | User Experience Director |
| 1 | Creative Director |
| 2 | Operations |
| 4 | Front-end developers |
| 4 | UX designers and content strategists |
| 3 | Visual designers |
| 2 | UX researchers |
| 2 | Project managers |
| 1 | Events manager |

## Penguin Random House

### penguin.co.uk




Over the course of 18 months of research, strategy and design, we led Penguin Random House (PRH) on a journey to their digital future. At the end, they had the design and the strategy to create digital destinations for thousands of authors and their plethora of publisher brands.

As formerly separate companies, Penguin and Random House each had their own websites. Collectively, they also managed more than 700 other websites focussed on authors, books, series, and characters. They knew this wasn't working. Few of those 700 sites had much traffic, and commissioning new sites with external agencies every time was getting expensive.

Penguin was to become the new consumer-facing brand; it has 100 per cent market recognition and the biggest catalogue in the world. PRH wanted to expand the brand beyond classics. People also don't know that it's related to two other strong brands, Puffin and Ladybird. How could we change the perception of Penguin and its related brands? This brief became one of our most challenging projects.

We spent a great deal of time researching with the many stakeholders and ultimately helping PRH form a digital strategy for their combined offering. Our team accompanied PRH's marketing, design, strategy, and C-level executives on their journey to prepare for significant shifts in their digital culture.

The new platform we devised avoids the problems of 700+ different websites. Our work pushed the brand into unchartered waters, places the brand guidelines didn't cover, enabling brands to work together under a single umbrella.

Also, the new consumer-facing website acts as a 'universe' of everything that Penguin offers. It houses the imprints in one place, with original and exclusive content focussed around the authors and their books. No longer will the organisation spend hundreds of thousands of pounds commissioning separate author websites.

experts within the digital space, but new thinkers that can push the boundaries of what digital design means today and speak directly to each brand's consumers.

More than that, Clearleft is frequently hired by its clients not just to provide user-centred design services, but to change and improve an organisation's design ethos and capability. "We like to work in close partnership with our clients," says Richard.

"And that means we get a really good understanding of what makes the company tick. This leaves us well placed to help a client form a vibrant design culture and introduce design thinking to a high level in the organisation."

Jeremy Keith, managing director and co-founder outlined how Clearleft approaches attracting new clients. "Fortunately our good reputation precedes us so most of our work comes to us," Jeremy said. "Or rather the leads come to us. Much of the time we still have to pitch, but word-of-mouth gets us a foot in the door at the very least.

"We work hard at getting our reputation out there. We like to give back to the community that supported us in the first place. Sharing knowledge is what drives the web along. Coming from a variety of more conventional professions we are very aware that this is not the case in all industries."

Agencies often strive to not become associated with one type of digital design. Also, agencies need to appreciate how each client they take on will have an impact on their business in the marketplace.

For Clearleft having the ability to turn down work is part of what makes their business different, as Richard outlined:

"Contrary to popular belief we don't get to pick and choose who we work with. We do however have the option to say 'no thanks' occasionally. We work on different scales of project, from self-

# Clearleft


## EVO magazine app

### evo.co.uk

Devoted to the 'thrill of driving', EVO magazine puts the reader behind the wheel of the most exciting cars in the world. We were approached by Dennis Media Factory to help them define the strategy and create a vision for the next iteration of the magazine's native apps. Our brief was to create a product that still delivered the rich editorial experience of page-turning, but would widen the audience and prolong reader engagement.

EVO readers are familiar with a finite set of content, delivered at the same time each month and displayed in roughly the same order. We spent time deconstructing and understanding the magazine and the content EVO produced across their various channels. Interesting opportunities quickly started to appear. There seemed to be no single place where you could consume the best of EVO's content. They had a large audience on YouTube (~230k users), who weren't necessarily engaging with the magazine or app, and Android users weren't currently being served.

The idea of the app as the central hub for all EVO content needed to be represented in a clear engaging way. That's where the "feed" came from: a mosaic of content presented as a grid. But without knowing the shape of future content, we couldn't define a pixel-perfect grid. Instead we created a system, using a series of art boards, that would adapt to the ever-changing content.

Under the hood, the app was using web technologies – HTML, CSS, and JavaScript – which allowed us to use media queries to adjust the presentation. We were doing responsive design in a native app!

contained week-long design sprints, through more conventional three- to six-month redesign work, to 12 months plus programmes which see us providing service design, multiple platform design and the digital culture change strategies that go with it."

A great example of this is the work Clearleft did with the Royal Borough of Kensington and Chelsea. Over a period of 18 months, Clearleft worked with the London council to completely redesign the website with a laser-focus on providing utility for its users – the local residents and businesses.

This meant working with departments across the council to understand their needs and fundamentally change the way (and what) they publish to the website. Punctuating the longer strategic work, was a series of smaller projects, such as redesigning specialist search functions, running online form design workshops and responsive design training.

How an agency functions is often fascinating to place under the microscope. With diverse clients needing very different approaches to their work, Clearleft has developed a number of working practices.

"A core team consists of UX design, visual design, front-end development and project management in differing proportions depending on need," Jeremy explained. "For large or particular projects we will supplement the team with specialists such as content strategists, creative technologists and dedicated researchers." In addition, Andy said: "All our large projects include a significant discovery

*Pen and paper are the most important tools Clearleft use. Here are sketches of early author page designs*


## TIMELINE

| Year 2005 | Year 2007 | Year 2012 | Year 2013 | Year 2014 | Year 2015 |
|---|---|---|---|---|---|
| No. of employees 3 | No. of employees 6 | No. of employees 11 | No. of employees 15 | No. of employees 17 | No. of employees 23 |
| Andy, Rich and Jeremy take the plunge and get Clearleft off the ground in 2005. | To help popularise usability testing, Clearleft create Silverback,. | Clearleft designs Channel 4 News' responsive website and iOS apps. | Clearleft renovates a derelict building in Brighton's famous Lanes. | Clearleft's graduate internship scheme concludes with the launch of Chüne. | Staff past and present celebrate the company's 10th birthday. |

phase which can sometimes be as much as 50 per cent of the project. The discovery phase is when most of the research occurs (with end-users, stakeholders, analytics, and so on), followed by strategy development (digital, content, technological), sometimes early design concepts and just enough planning."

With an array of tools at every digital designer's fingertips, Richard outlined the toolset that Clearleft use to create their rich digital environments: "Our primary tool is paper. Our studio walls are covered in stickies, sketches, whiteboard scribbles and print-outs. After that we work with whatever software does the job. Sketch, Illustrator and Invision all get used.

"Also, Axure, Omnigraffle, Keynote and all manner of text and code editors. We're not precious or prescriptive about tooling or software, and the same can be said of our general approach to the design process. We don't like leaning on process, we prefer to select from an ever-expanding toolbox into which we keep adding new stuff, leaving some of the less used tools to rattle around in the bottom."

Jeremy also noted: "HTML, CSS and JavaScript are the three fundamental languages with which the

## "New code won't break old browsers, at least not if coded accessibly and diligently"

Web is built, regardless of framework or back-end technology. That is not going to change in the foreseeable future, although those languages will continue to develop as needs changes in predictable and unimagined ways. The beauty of HTML and CSS is that old code will work in the future and new code won't break old browsers, at least not if coded accessibly and diligently."

Today there is clearly a focus on the mobile platforms that are now ubiquitous. For Clearleft, these are a necessary and accepted channel that needs to be supported.

"Responsive design is a fundamental aspect of modern web design – it should be the norm for any design professional," says Jeremy. "But responsive design is not a binary yes-or-no tick in a box. It is a spectrum, so what is required, desired and possible will vary."

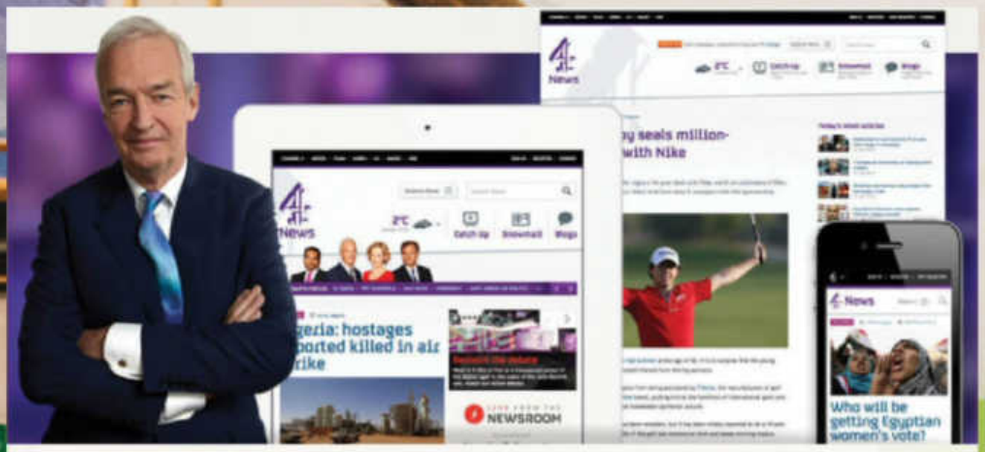
Andy added: "While we believe in the power of the web, we also understand the need for native apps, although the technological possibilities of web apps are closing the gap all the time. Ironically, a significant edge that native apps have is that the user goes through the process of installing them and this gives the user an attachment to that app. Again web apps are closing this through the addition

### Industry Insight

**Jeremy Keith**
**Technical Director and Co-founder**

"We like to give back to the community that supported us in the first place. Sharing knowledge is what drives the web along. Coming from a variety of professional backgrounds we are very aware this is not the case in all industries."

*A peek through the glass reveals the denizens at Clearleft creating the digital spaces we will all inhabit*




## Channel 4 News

### channel4.com/news

Channel 4 and ITN work together to deliver the news, which means there are lots of people involved at each stage. They have an enormous responsibility to the public but their online strategy didn't reflect the many ways in which news is captured and consumed.

The Channel 4 news team have a passion for the power of journalism. Together we set out to design a way to convey that passion on the web. We need to capture the excitement and speed of the Channel 4 news room and convey the unique personality of the Channel 4 news team on the web.

We had worked with Channel 4 News before, designing their mobile apps and refreshing their website. But this was the biggest challenge yet. Not only did the new website need to adapt to many devices and screen sizes, it needed to adapt to the fast-paced way the news itself gets produced.

We found ourselves in the news room, in the centre of all the activity, watching the team capture, edit, and share events from all over the world. Seeing it all come together is a rush. It's fast-moving, exciting, and iterative. In that moment, we realised that we could also design in a way that harnesses that same excitement.

At the end of our three-month collaboration, they had the first responsive news site in the UK, one that conveyed not just the news, but also the unique perspectives of the people behind the news.

## Clearleft

### www.clearleft.com

**Founders**
Andy Budd, Jeremy Keith, Richard Rutter

**Year founded**
2005

**Current employees**
23

**Location**
Brighton

**Services**
Digital strategy
User experience design
Responsive web design
User research
Content strategy
Product innovation



of service workers, and browser-based add-to-home screen functionality."

And what of the new technologies, frameworks and tools making an appearance? Jeremy continued: "We tend to shy away from front-end frameworks. They are usually solutions to someone else's problem. We get more excited by tools which improve our workflow or communication such as Sketch Measure (**https://github.com/utom/sketch-measure**) and our own Fractal open source pattern library tool (**http://fractal.build**)."

Richard also explained their approach to social media networks: "In many cases social media is the lifeblood of any digital and content strategy. This is particularly true of the charities we work with. In their context it's worth seeing social media not as a way of driving traffic to a website, but as reaching out to people where they are at that time and then pointing them to the thing you want them to do next, which may or may not be on their own website."

**"We have a genuine affection for the medium and we're constantly striving to do our best work"**

Even though Clearleft is over ten years old, it remains relatively small regarding head count. Richard explained their approach to recruitment: "Depending on the role, we'll be looking for experience and knowledge, but for every role we'll want the right attitude. Clearleft folk are enthusiastic. We have a genuine affection for the medium and we're constantly striving to do our best work. We are self-starters, good communicators, respectful and appreciative of other people's skills, always looking to learn, share and listen. We work hard but not to the detriment of family or personal life. Above all we are nice people."

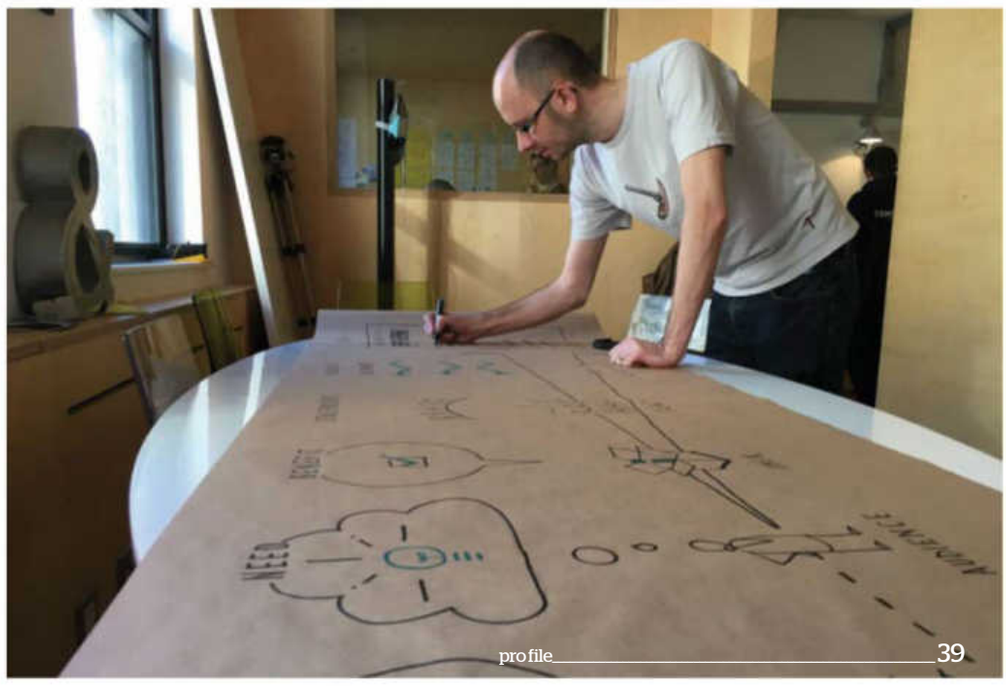
Richard also advised: "If you're looking to break into the industry, show that positive attitude. Show enthusiasm, that you're a self-starter. That you can apply critical judgement and talk through your workings. Show that you can write well."

Finally, what does the future hold for Clearleft? Andy concluded: "Since 2005, Clearleft has expanded slowly and steadily to be 25 people at the last count. That expansion will continue at a similar pace, enabling us to take on larger projects and add more specialists to our designers and consultants. For example, we have just created a dedicated research practice within the company.

"We've been getting much more under the skin of our clients recently, up-skilling and training their in-house design teams and helping them instil a culture of design thinking and leadership. Tied with broader service design and digital strategy, we see the consultation side of our business expanding, while still retaining our dedication to design."

Clearleft is much more than a digital design agency. The drive to create new digital experiences has pushed them to become innovators in its own right. Clients looking for forward thinkers who can deliver much more than simply code, gravitate to Clearleft, as they understand how to shape the future that their clients' customers will be inhabiting.

The fact that they are moving into more consultancy work and developing their own in-house research practice speaks volumes about their ambitions, which are only limited by their imaginations. If a tool doesn't exist to realise their vision, this agency will create it.




## EXPLORE THE LATEST CSS TECHNIQUES AND WHAT'S HAPPENING WITH HTML5

**In the past, new standards specs have been huge documents taking so long to compile that by they time they were released, things were already feeling dated due to the pace at which our industry moves.** Now that they are incremental releases, however, it means that new features can be adopted by browser vendors more quickly, so that we don't have to wait so long to use the latest feature.

With all major browsers now being 'evergreen' (yes, even Microsoft), the days of having to support age-old browsers should be a thing of the past. An evergreen browser constantly auto-updates itself to fix security flaws and maintain the latest standards. This means that a user would have to go to some lengths to have an out-of-date browser.

Keeping up to date with what's available to you as a developer is vital, as it means that you can improve your workflow, create features that weren't previously possible and aid the user with improved state highlighting and semantics.

"THE BIGGEST CHANGE THIS YEAR IS CSS4 SELECTORS. WE ARE NOW ABLE TO SELECT A GREAT NUMBER OF STATES THAT WE EITHER COULDN'T DO BEFORE OR RELIED ON JAVASCRIPT TO ACHIEVE. THE INPUT VALIDITY SELECTORS ARE GOING TO BE A GREAT TIME-SAVER."

PAUL THOMAS

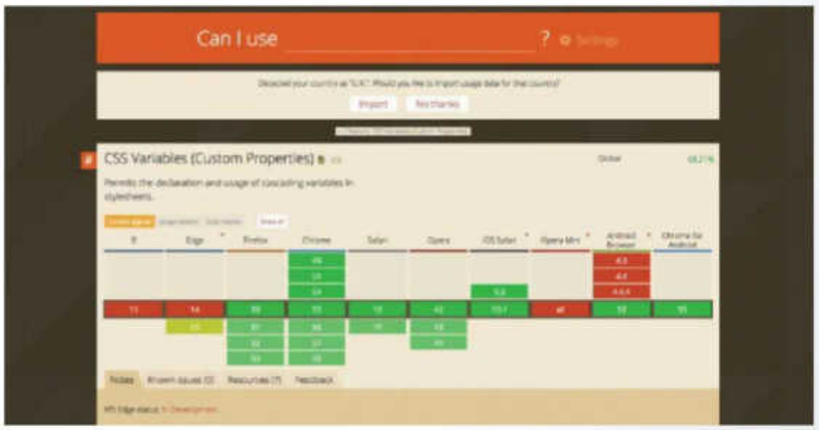
SENIOR DEVELOPER AT GREEN CHAMELEON, WWW.TWITTER.COM/MOTIONIMAGING

## DIFFERENT DRAFTS

A Working Draft is the latest version of the specification that the W3C feels is worth sharing with the wider community. It is unlikely to change much before making it to a Candidate Recommendation (likely to be implemented by browsers).

Before this stage is the Editor's Draft, the very latest thinking of the group. Items in this version of the specification are highly subject to change and should not be relied upon. This is useful if you are involved in standards compilation, but also of interest to the curious.

## STATE OF BROWSER SUPPORT




Browser support is different for just about every feature mentioned in this article. Some you can use today; others you may never be able to use. With all major browsers now being evergreen, you should be able to start using those that get implemented very quickly, although it is always worth keeping a progressive enhancement mentality and not relying on them for basic functionality.

Before using any of the latest CSS and HTML techniques in production, it is essential to check feature support against the browsers you intend on supporting. The go-to resource for compatibility is **caniuse.com**, where you'll find the latest support for most browsers as well as any known major bugs, links to the specs and other resources. **caniuse.com/#feat=css-variables**

# CSS4 SELECTORS YOU CAN USE NOW

Check out this selection of selectors that will add more power and flexibility to your workflow

## Negation

:not has been around since CSS3, but now it can accept multiple selectors as an argument, keeping our code more readable and concise.

```
Existing syntax: h1:not(.this-element):not
(.another-element) { font-size: 1.2em; }
New syntax: h1:not(.this-element, .another-
element) { font-size: 1.2em; }
```

## Matches

This is the opposite to the :not selector and can also take multiple selectors as an argument. It's not that different to the current selector syntax, but improves readability and maintainability.

```
Existing syntax: h1.this-element,
h1.another-element
{ font-weight: bold; }
New syntax: h1:matches(.this-element,
.another-element) { font-weight: bold; }
```

```
.element:drop(invalid)
{ border-color: red; }
```

## Indeterminate

Matches elements that are in an indeterminate state. Checkboxes, radio buttons and progress bars can all be indeterminate. For example, a parent checkbox with just one child checkbox selected may be indeterminate. This cannot be set in HTML, only in JavaScript.

```
input[type="checkbox"]:indeterminate
{ opacity: 0.75 }
```

## Default

Matches the default element among a set of similar elements, eg the default 'selected' <option> in a <select> element

```
option:default
{ font-weight: bold; }
```

## User Interaction

Style an element that is invalid (eg out-of-range, required) only after an element has been interacted with, not on page load (if the initial value was invalid).

```
input:user-error
{ border: 1px solid red; }
```

## Mutability

Style elements such as <input> or <textarea> based on read-only or read-write.

```
textarea:read-only { color: grey; }
textarea:read-write { color: black; }
```

## Placeholder Shown

Elements can show placeholder text as a hint to the input required. We can now style these elements that are showing placeholder text; ie they are waiting to be filled.

```
input:placeholder-shown {
border-color: blue; }
```

## Relational

There's a chance this might not make it in (currently in Editor's Draft) due to its complexity to implement. It's the long-awaited 'parent selector' that will allow us to style an element based on its content, rather than just the other way around.

```
.container:has(.this-element)
{ background: green; }
```

> "WE'VE NEARLY ALWAYS MANUALLY ADDED AN ASTERISK (*) TO THE LABEL OF AN INPUT FIELD TO SHOW A USER IF IT IS REQUIRED OR NOT. WHY NOT DO THIS AUTOMATICALLY WITH THE NEW :OPTIONAL AND :REQUIRED CSS SELECTORS"

## Case Sensitivity

You may find that due to developer inconsistencies, you have elements you want to style that have attributes with unknown or mixed case. You can now match attributes with case insensitivity.

```
input[name="email" i] /* Matches
name="name", name="Name", name="NAME" etc.
*/
```

## Drop

Add styling to HTML5 dropzone targets. Styling can be applied if the dropzone is valid, invalid or active.

```
.element:drop
{ border: 3px solid; background: grey; }
.element:drop(active)
{ border-color: blue; }
.element:drop(valid)
{ border-color: green; }
```

## Validity

Elements that are valid/invalid according to the elements rules, such as 'required' or 'email'.

```
input:invalid { border: 1px solid red; }
input:valid { border: 1px solid green; }
```

## Range

Applies to elements that have range limitations, such as a slider or number input with max/min values.

```
input:in-range { background: green; }
input:out-of-range { background: red; }
```

## Optionality

Style elements to clearly show which elements are optional and which are required in a form.

```
:required
{ border: 2px solid black }
:optional
{ border: 1px dashed grey; }
```

## Local Link

Style elements differently depending if they are on the same domain or not. Takes optional argument for hierarchical styling.

```
a:local-link
{ text-decoration: none; }
```

## Time Dimensional

Useful to link content to screen readers. Primary use would be to highlight text that is currently being read aloud, text that has already been read and text that will be read next.

```
p:current {
background-color: yellow; }
p:past,
p:future(p, span) {
font-style: italic;
}
```

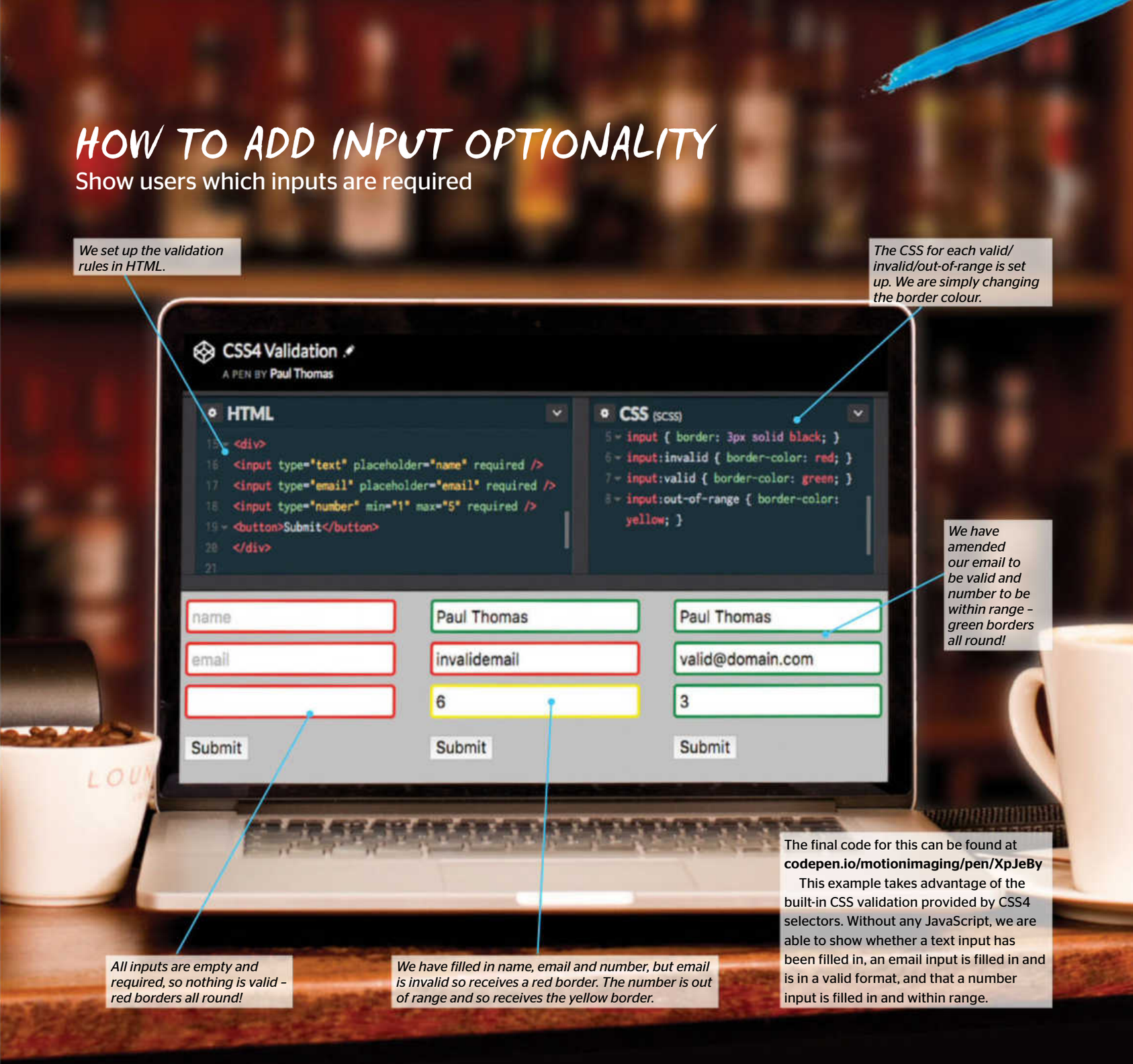
## Hyperlink

Match any element that is a link (has an href). It replaces a:link, a:visited.

```
:any-link { text-decoration: underline;
}
```

# HOW TO ADD INPUT OPTIONALITY

## Show users which inputs are required




**We've nearly always** manually added an asterisk (*) to the label of an input field to show a user if it is required input or not, but why not do this automatically and more clearly with the new :optional and :required CSS selectors?

The final code can be viewed at **codepen.io/motionimaging/pen/dNPZzO**

### 1. Requirement rules

We set up the 'required' rule on the HTML element itself, the HTML5 way.

```
<input type="text" required />
```

### 2. Highlight the requirement

We're going to show the user a dotted border if it is optional and a solid border if it is required. Unfortunately, as <input> elements don't support ::before, we can't automatically add an asterisk.

```
input:optional { border-style: dashed; }
input:required { border-style: solid; }
```

### 3. Placeholder indication

Showing placeholder text is useful to give hints as to the input expected, but it can be confused with actual input. We are going to fade out the element slightly until the user has filled it in.

```
input:placeholder-shown { opacity: 0.6; }
```

### 4. Validation

We are going to indicate that the input is valid using the :valid selector, but only once the placeholder text is no longer shown by combining it with the :not selector.

```
input:valid:not(:placeholder-shown) {
border-color: green;
}
```

# WHAT ARE CSS VARIABLES?

## Why CSS variables wield more power than preprocessor variables

Most of us have used variables in CSS preprocessors such as Sass/Less to store values such as colour hex codes or font sizes. However, by their very nature, once they have been set they cannot be updated on the client-side. Sure, we can alter a value and run the build process again, but once deployed, that value cannot be updated by the website itself.

This is where CSS variables come into play. They can be updated via the website itself using JavaScript, using getPropertyValue() and setProperty(), meaning they can be updated on the fly site-wide. The makes them far more powerful and flexible that their preprocessor counterparts.

You can also, of course, use them just as you would preprocessor variables – for storing colour values and the like. This means we will be less reliant on third parties and can return to writing native CSS code for this task.

Other possibilities lie in responsive design. For example, we may want to change the grid gutter at a certain screen size. This task becomes trivial with CSS variables and we can also base other properties off this value using mathematics. This is not possible with SASS as we would need to re-declare each property value.

They also cascade just like all other CSS. This means we can update values just as we would any CSS value.

It's not immediately obvious why CSS variables are better than their preprocessor counterparts. Here are some things you can't do in Sass. The final code (including the HTML used) can be viewed at **codepen.io/motionimaging/pen/qRaqBd**

### 1. Basic syntax

Variables are declared first in the :root element and are written as `--variable-name: value` and used as `property: var(--variable-name)`.

```
:root {
--primary-color: red;
--secondary-color: green;
--spacing: 20px;
}
.title {
color: var(--primary-color);
border-bottom: 1px solid var(--secondary-color);
}
.grid-item {
float: left;
width: 300px;
margin: var(--spacing);
padding: var(--spacing);
border: 1px solid var(--secondary-color);
}
```

### 2. Change variables with a media query

We can change the variable value at a particular screen size. We only need to change the value of the variable and it will be applied anywhere we've used it. We don't need to apply it to the elements as we would need to with a preprocessor. Here we change the --spacing value at larger screen sizes.

```
@media (min-width: 1000px) {
:root {
--spacing: 40px;
}
}
```

### 3. Cascading Variables

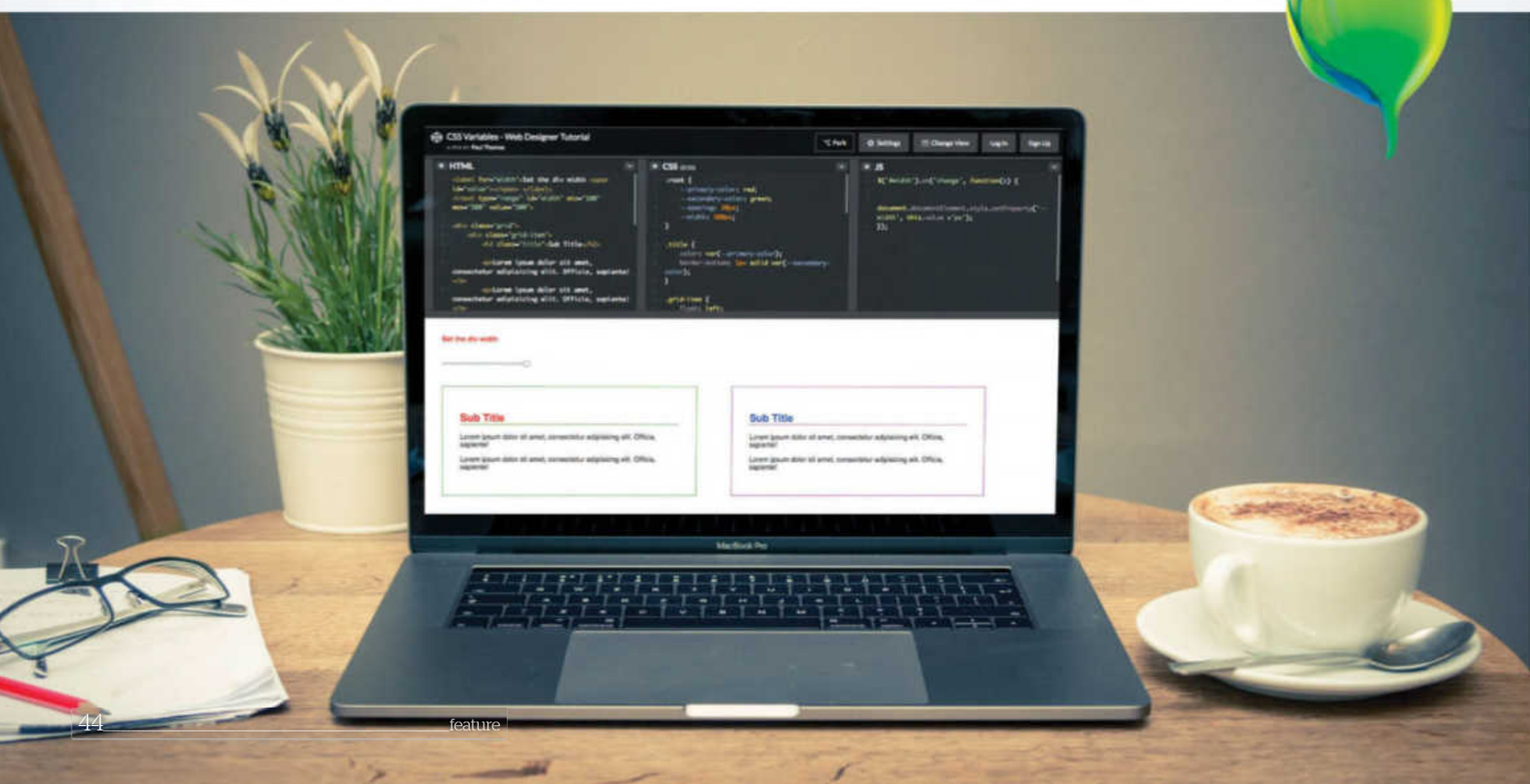
CSS Variables can cascade, unlike preprocessor variables. Perhaps we want to swap primary and secondary colours in the last div.

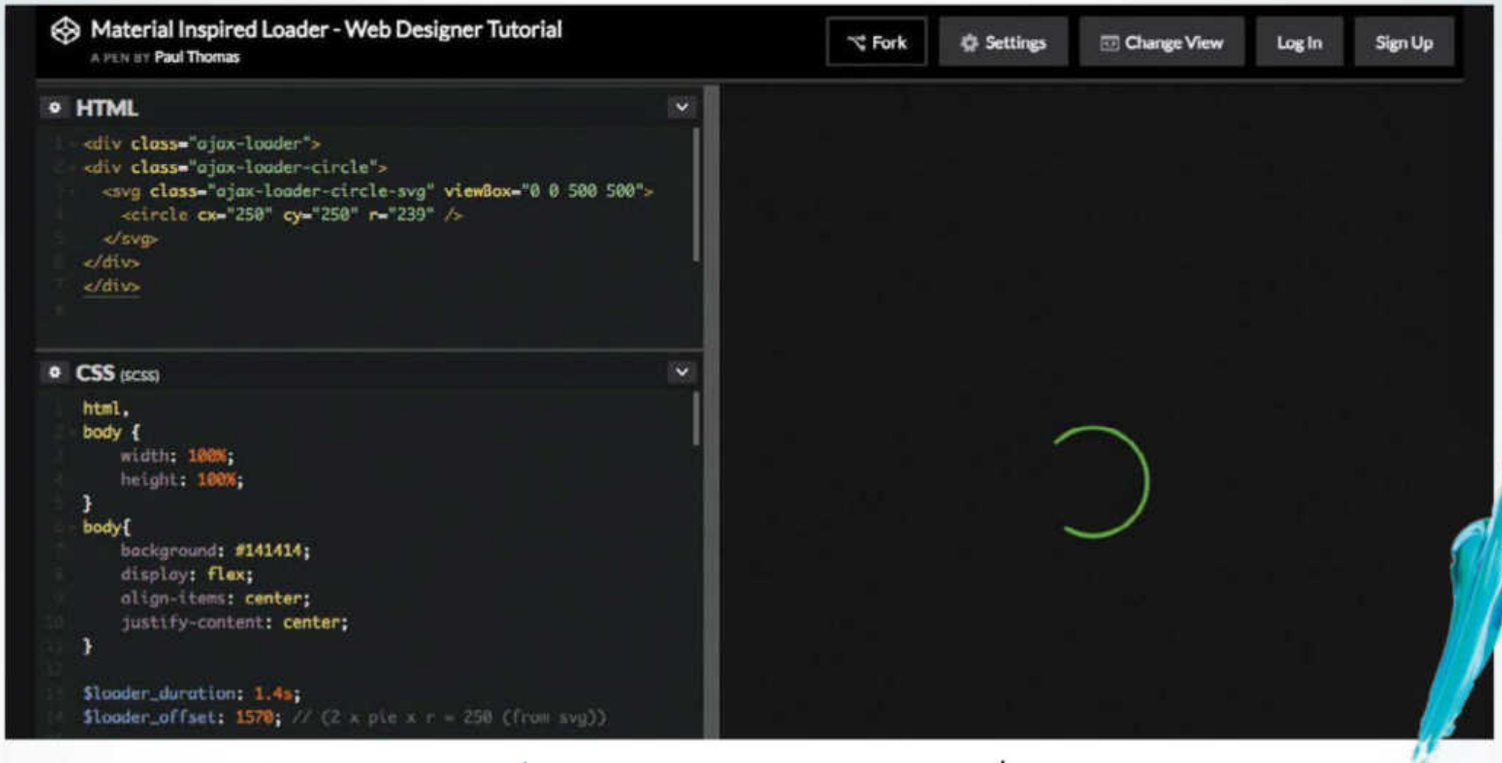
```
&:last-child {
--primary-color: blue;
--secondary-color: purple;
}
}
```

### 4. Access with JavaScript

We can access and apply variable values in JavaScript . Set up a HTML5 range input. We then access the value of this input and update the variable value.

```
<input type="range" id="width" min="100"
max="500" value="300">
$('#width').on('change', function(e) {
document.documentElement.style.
setProperty('--width', this.value +'px');
});
:root {
--width: 300px;
}
.grid-item {
width: var(--width); }
```






# CREATE A MATERIAL-INSPIRED CSS LOADER

## We can animate SVG paths using CSS to create impressive-looking animations

### 1. Create the SVG

To animate the CSS properties, the SVG needs to be inline. First, we create a basic SVG circle and wrap it in two divs to aid us with the animation. The CSS for the layout is up to you to size, but an example can be seen at **codepen.io/motionimaging/pen/EZgKdy**

```
<div class="ajax-loader">
<div class="ajax-loader-circle">
<svg class="ajax-loader-circle-svg"
viewBox="0 0 500 500">
<circle cx="250" cy="250" r="239" />
</svg>
</div>
</div>
```

### 2. Animate the colours

We are going to change the colour of the SVG by animating the stroke colour of the SVG. We loop through five colours. Try experimenting with different colours to get an effect you like.

### 3. Animate dash offset

The trick to the loader look is in animating the SVG dash-offset from a small amount to nearly the full circumference of the circle. We store the full circumference of the circle as a variable and then animate from 25% to 90% of this value in our keyframe animation. The circumference is calculated as 2*3.14(pie)*250(radius) = 1570 for the circle we have here. The dash-array must be equal to the full circumference for this to work.

**LEARN SOME BASIC MATHS**

*With animations comes more maths. You don't need too much, but knowing some basic trigonometry. Pythagoras and geometry will come in useful.*

### 4. Rotate the containers

To give the loader the appearance of spinning, we must rotate the two container divs at the same rate as the dash-offset.

```
.ajax-loader{
position: relative;
width: 100px;
height: 100px;
transform-origin: (50%, 50%);
animation: ajaxLoaderSpin
$loader_duration linear
infinite; }
.ajax-loader-circle-svg{
position: absolute;
left: 0;
top: 0;
width: 100%;
height: 100%;
transform-origin: (50%, 50%);
animation: ajaxLoaderDashSpin $loader_
duration ease-in-out infinite; }
@keyframes ajaxLoaderSpin{
0% ...
```

***Remember, find code example on CodePen***

# CSS GRIDS

Layout is the foundation of the internet: without it we would be lost. Yet we still find ourselves struggling with the flow of elements at different screen sizes and maintaining a well-ordered document structure.

Flexbox has been around for a while and allows us to reorder elements and align them pretty much how we want. However, there is one large caveat – it's in one dimension only. This is very useful, but it leaves a lot of room for improvement.

There's been a lot of hype around CSS Grid Layout and it will finally be landing in browsers early in 2017, and can already be used behind experimental flags.

There are a lot of similarities with flexbox in terms of layout, but it works along two axes, meaning we have full control over the space taken up by an element, both in terms of columns and rows. Sometimes it feels a bit like laying out with <table>, but semantically correct and far more powerful and usable.

# READ MORE...

**CSS Tricks – A Complete Guide To Grid**
**css-tricks.com/snippets/css/complete-guide-grid**

**Rachel Andrew – Grid by Example**
**gridbyexample.com**

# WHAT HAPPENING WITH HTML5.1?

Updates to the specification means changes. Check out the key ones

## WHAT'S IN

### <picture>

Use entirely different image sources for a range of situations using the <source> element. Art direction via media queries. Different image formats depending on support (eg WebP for Chrome). Can be used in conjunction with srcset.

### <details> / <summary>

These two elements work in conjunction. The text within <summary> is always displayed and when clicked, the remaining text within the <details> block is shown.

### <menuitem>

Used in conjunction with <menu type="context"> it can add functionality to the browser's context menu. It can be one of theses types: command, checkbox, radio.

### requestAnimationFrame() API

This has been the go-to method of creating browser-optimised animations for quite some time, but it now makes its way into the spec, cementing its place as the best method to use.

### history.scrollRestoration

Traversing the History API is an essential feature of many web apps. This allows manual control over the scroll position, previously unavailable, allowing for a smoother user experience.

## WHAT'S OUT

### appCache

This is being ditched in favour of the newer Service Worker method of caching applications for offline usage. It will likely be supported by browsers for some time to come, but is not being improved upon so usage is discouraged.

### Media Controllers

This was used to synchronise media elements across a page. However, only Safari ever supported it and so it is being dropped due to lack of implementation.

### type="range"

This would have allowed for multiple sliders on one range slider, but was not implemented by any browsers and so is being dropped.

### Command API

The fact that <menuitem> is being introduced and allows for the use of a command means that the command API is no longer the preferred method to handle this functionality, so it's time for it to go.

### <object usemap="#mapname">

You can no longer use an image map with an object. A lack of support and lack of use means that this is no longer part of the spec. You can still use them on images, should you so wish.

# WHAT'S CHANGED IN THE LATEST VERSION?

### accesskey

The accesskey attribute's value is used to create a keyboard shortcut to focus an element. It can now only take a single character (exactly one Unicode in length).

### <header> nesting

<header> elements can now be nested within <header> elements, so long as they form part of a new <section> element.

### <footer> nesting

<footer> elements can now be nested within <footer> elements, so long as they form part of a new <section> element.

### Empty <option>

The <option> element can now be empty. This may be useful if you want the default option to be blank, rather than something like 'please select from...'

### Wheel event

The 'mousewheel' event is non-standard and deprecated. It is now called the 'wheel' event and is triggered whenever a wheel button of a pointing device (a mouse) is rotated.

### Translatable value on submit buttons

The value attribute of input type="submit" now respects the translate attribute to allow or prevent language translations. It seems this was missed from the spec and has now been added.

### <figcaption> location

<figcaption> can now appear anywhere within the <figure> element. Previously it had to be the first or last element.

### <img /> alt and title

Simply putting a title="" instead of alt="" does not mean the element is conformant any longer. Use those attributes the way they were intended!

### <img width="0" />

Setting a width of '0' on an <img> is now allowed.

### <title> in <svg>

If an <svg> element has multiple <title> elements, the first is now used, rather than the last.

"THE NEW CSS MODULES ENHANCE THE LANGUAGE WITH CAPABILITIES THAT DEVELOPERS HAVE BEEN PREVIOUSLY LACKING. VARIABLES IN PARTICULAR ARE SO POWERFUL AND FLEXIBLE THAT YOU MIGHT ALREADY BE USING THEM WITH ONE OF THE NUMBER OF FRAMEWORKS – SUCH AS SASS AND LESS – THAT FEATURE THEM AS A MAIN SELLING POINT."

MIKE BABB
WEB DEVELOPER
MIKEBABB.COM

# RESPONSIVE IMAGES WITH <PICTURE>

## Use the best possible image available depending on screen size and image support

### 1. Get yourself some images

We are going to be need small and large versions of an image, in both PNG and WebP formats; PNG should be straightforward, but if your graphics editor doesn't export WebP, you can use a grunt task to convert JPEG to WebP: grunt-cwebp from **www.npmjs.com/package/grunt-cwebp** (also available as a gulp package).

```
$ npm install --save-dev grunt-cwebp
```

### 2. Mobile browsers

<picture> gives us the option to use an entirely different image based on screen size. This code allows us to set one image for screens over 750px, with a fallback if <picture> is not supported or the screen is below this size.

```
<picture>
<source srcset="images/img_original/
raspberry.png" media="(min-width: 750px)">
<img src="images/img_original/raspberry_
small.png" alt="An illustration of some
raspberries">
</picture>
```

### 3. Supported image formats

We are able to get the <picture> element to use a certain image if it is supported. Here we are asking for a WebP image if it is supported (Blink/Chrome browsers). Again, the fallback is used if <picture> or WebP are unsupported.

```
<picture>
<source srcset="images/img_webp/raspberry.
webp" type="image/webp">
<img src="images/img_original/raspberry.
png" alt="An illustration of some
raspberries">
</picture>
```

### 4. Combine the two

To get the best possible saving, we would like to be able to show the correctly sized image depending on screen size and the correct image format depending on support. This code mixes both requirements to give us the best possible image.

```
<picture>
<source srcset="images/img_webp/raspberry.
webp" media="(min-width: 750px)"
type="image/webp">
<source srcset="images/img_original/
raspberry.png" media="(min-width: 750px)">
<source srcset="images/img_webp/raspberry_
small.webp" type="image/webp">
<img src="images/img_original/raspberry_
small.png" alt="An illustration of some
raspberries">
</picture>
```


# CURRENT STATE OF HTML 5.2

Rather than work on major overhauls of the spec, the specs are now written in a more incremental way. This means that new features are implemented more smoothly and more quickly.

HTML 5.2 is in its infancy. The good news is that you can get involved! The W3C is developing the specs in an open source manner via its GitHub page. Anyone can suggest, make a pull request and all the normal things an open source project would entail, before it moves on to the next stage of the spec lifecycle.

Get involved via GitHub at **github.com/w3c/html**

## 3 TO FOLLOW

### Šime Vidas
**twitter.com/simevidas**
Curator of Web Platform Daily and always up to date with the latest features.

### Chris Coyier
**twitter.com/chriscoyier**
This guy lives and breathes the web and is also behind css-tricks.com and CodePen. It's worth hearing what he has to say.

### Jen Simmons
**twitter.com/jensimmons**
Host of The Web Ahead podcast and designer advocate at Mozilla, Jen covers a large amount on standards, CSS, HTML and the web in general.

# Create masked changing background images

As seen on **fres.lt/en/**


**SVG Mask**
The main title is an SVG file with transparent letters allowing the background image to show through. Being an SVG, this keeps the letters crisp at any resolution, with minimal file size.

**Animating background colours**
The main colour of the background, cycles through different colours so that it displays a full range of hues. This continually shifts every few seconds.

**Cycling images**
Not only does the background colour change, but the background image cycles through different images, this is timed differently to the coloured background animation.

**Preloading the content**
When the content loads, an overlay container scrolls up and off the top of the page, revealing the content of the site below.

**Single-page site**
The FREŠ website is a single-page scrolling site. The whole site background changes colour, just as it does here in the header section.

# Create masked changing background images

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

**EXPERT ADVICE**
**Consistent site brand**
As the user scrolls down the page, the logo in the header section leaves the top of the page as expected. In order to keep some site branding on the page, a smaller logo slides in from the right, fixing itself to the bottom right of the page. This also doubles up as a scroll-to-top function.

**FREŠ is an events company based in Lithuania and, like its name, have a very fresh, vibrant website.** The appeal of this site is that whereas many others might opt for a minimal white glossy look, the FREŠ site is focused on bright colours and conveying a sense of vitality. The background is a continually changing range of vibrant colours, but because of the subtle transition to each colour and the length of time on the screen, it actually works so much better than it sounds. The team have to work hard to ensure that the events go off with precision, so it is no surprise that the enthusiasm of the team is represented with the bold use of colour on the page.

Because the page features such strong colours, this might be difficult to include the use of photography, but the header section of the site – featured here – actually has colour photography as part of it and this is integrated thoughtfully with the masking of the letters. The remainder of the site uses black and white photography, keeping the neutral colours that will always work with whatever colour the cycle is moving through. The result is a refreshing change to many neutral-coloured sites.

**<comment>**
What our experts think of the site

## Vivid colours lighten the mood

"The site for FREŠ is aimed to highlight a young company in the entertainment sector. Vivid powerful colours and rich white space provide a sense of an open-air beach party. Slow, unobtrusive animation keeps the viewer focused on content, and fast hover animation makes them press the right button."

**Aleksej Tisionok, lead designer at Bitsens**

# Technique

## 1. Changing background

The changing background found on **fres.lt** is created with an SVG image set as a background mask over the top of photographs. This is the HTML structure you will need to recreate the design in the browser, consisting of two divs with background images and a div that will hold the mask.

## 2. Full screen ahead

In order to get the content to work, the `html` and `body` are set to the full width and height of the browser with the regular padding and margin removed. The `hide` class will be used to hide one of the images and will be toggled using JavaScript later.

```
html, body {
margin: 0;
padding: 0;
width: 100%;
height: 100%; }
.hide {
opacity: 0; }
```

## 3. Full screen SVG

The SVG is set as a background mask in the SVG div and is set to fill the screen with the background sizing set to cover. The SVG is coloured using the `color` property and this could easily be transitioned like in the **fres.it** site.

```
#svg {
width: 100%;
height: 100%;
color: rgb(2, 200, 242);
background-color: currentColor;
-webkit-mask: url(img/img-hero1.svg) no-
repeat 50% 50%;
-webkit-mask-size: cover; }
```

## 4. Positioning the back

The background is set to a lower z-index so that it will sit behind the SVG content. It's fixed in position so it will not scroll and, again, set to fill the browser's width and height. This is the container that holds the background images.

```
.bg {
position: fixed;
top: 0;
left: 0;
z-index: -1;
width: 100%;
height: 100%; }
```

## 5. Background images

Each background image is given a transition of 1.5 seconds and then the background of the first child is given the image **1.jpg**. Copy the same for the second child of the background and this time set the background image to **2.jpg**.

```
.bg_item {
position: absolute;
top: 0; left: 0;
width: 100%; height: 100%;
background-position: 50% 50%;
background-repeat: no-repeat;
transition: opacity 1500ms ease; }
.bg_item:nth-child(1) {
background-image: url(img/1.jpg);
background-size: cover; }
```

## 6. Switching between the two

Here the element with the ID of fader is targeted and every three seconds the `switcher` function is called. All this function does is toggle the images class to be hidden or visible and this therefore turns the image off and on with the transition effect on the opacity.

```
<script>
var fader = document.getElementById("fader");
var timer = setInterval(switcher, 3000);
function switcher() {
fader.classList.toggle("hide"); }
</script>
```

# Create a stunning watercolour effect with WebGL

Add beautiful organic movement to your transitions, perfect for watery and painterly effects with the amazing Aquarelle library


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

**Most transition effects on the web involve sliding content onto the screen, fading content and scaling content, but at the heart of all of these transitions is the basic underlying page division.** Every designer can see that the rectangular div or image is present in the effect. The Aquarelle library changes that as there is a true rippling distortion to the content as well as a very organic spread with the mask as the content is revealed. Aquarelle itself refers to a style of painting that uses very thin, transparent layers of watercolour and is also the Italian word for watercolour.

This makes it the perfect effect to use if your client's site has the subject of watercolour painting, or you can turn down the rippling distortion so that you just have a very organic spreading to reveal your content.

In this tutorial, the theme will be a website that has a sign-up tutorial for creating watercolour effects in Photoshop. This is exactly the kind of site that would benefit from this effect. The beauty of the Aquarelle effect is that it is also possible to use it to trigger other animation; so as the effect is a few seconds in, you can use JavaScript to start other animations such as the menu, branding and page title transitioning into place.

## 1. Setting up the project

From the tutorial files, open the **start** folder in your code editor and start editing the **index.html** file. This is a simple bare-bones HTML page ready for inserting the design that will be used. The first part is to connect the CSS file in the head section of the page. This will enable the page design; the CSS code will be added later in the tutorial.

```
<link rel="stylesheet" href="css/style.css">
```

## 2. Connecting the libraries

The Aquarelle library needs the three.js library to work and the associated post-processing effects composer. We've assembled all the files from the relevant libraries for you in the **JS** folder, with the exception of the contour library from the D3 framework which will be linked to an online source.

```
<script src="js/three.min.js"></script>
<script src="js/Aquarelle.js"></script>
<script src="js/EffectComposer.js"></script>
<script src="js/ClearPass.js"></script>
<script src="js/AquarellePass.js"></script>
<script src="js/ShaderPass.js"></script>
<script src="js/CopyShader.js"></script>
<script src="https://bl.ocks.org/mbostock/
raw/4241134/d3.geom.contour.min.js"></
script>
```

## 3. Adding the page content

Now move down to the body section of the page and here the watercolour image is added; this is the image that will be painted in later with the Aquarelle effect. A logo is added to the page and the HTML code for the three-line, burger icon menu; these are all wrapped in their own div tag.

```
<div class="fade">
<img id="watercolor" src="img/painting.jpg">
<div class="top">
<div class="logo"></div>
<div class="menu">&#9776; </div>
</div>
```

## 4. Adding text

The content here will be displayed above the watercolour image and will fade in over the top. The icon will be an image that will be added in the CSS, and the text is the heading that gives the page its context so visitors can immediately relate the image with the purpose of the site.

```
<div class="button">
<div class="icon"></div>
</div>
<div class="title">
<h2>Digital Watercolor Masterclass</h2>
</div>
</div>
```

## 5. Below the fold

The image is positioned to fill the browser window, so the rest of the content is 'below the fold', an old newspaper term for content not visible on the news stand. The content will be a simple image, heading and sign-up button to the site's service. Below this is the page footer.

```
<div id="content">
<container>
<h2>Simple step by step guide</h2>
<img src="img/steps.png" alt="easy as 1, 2,
3">
<a href="#">SIGN UP</a>
</container>
</div>
<footer>
<p>&copy; 2017 Web Designer Magazine</p>
</footer>
```

## 6. Switch to the CSS

Save the **index.html** page now and open the **style.css** for editing. The body has already been styled up, so just add this new code into the CSS. The content with the class 'fade' is set to fill the full screen of the browser but made invisible. Later, when the libraries have loaded, it will be made visible to the viewer. The canvas will replace the image and the effect will be rendered here.

```
.fade {
width: 100%;
height: 100%;
```

### HTML5 Canvas

The image in the page cannot have the Aquarelle effect applied to it as it is. The image is taken and added into WebGL, which is rendered in the HTML5 Canvas element.

**Left**
The tutorial uses the Aquarelle library, which is a shader extension for the three.js WebGL library, but this displays 2D images rather than 3D models

**Top left**
All the content has been added to the page, but the design is still to be added through CSS; however, the basic page structure is still clearly visible

**Top right**
The CSS styles up most of the content on the page. Here the opacity of '0' in step 6 has been removed, just to see the final design. The opacity is there so that the image can fade in

```
opacity: 0; }
canvas {
left: 50%;
max-height: 100%;
max-width: 100%;
position: absolute;
top: 50%; }
```

## 7. Top left, logo

The CSS for the logo places the logo element using absolute positioning into the top left of the page. The image is set as the background and is set to be contained within the shape of the div, so it will be scaled proportionally to fill that div. The relevant height and width are therefore added.

```
.logo {
background: url(../img/eg-logo.png) center
no-repeat;
background-size: contain;
height: 45px;
width: 35px;
left: 30px;
top: 12px;
position: absolute; }
```

## 8. Menu in the top right

Just as the logo is positioned in the top left of the page, the menu icon is placed in the top right. Again, this is absolutely positioned. As this is just a demo, the menu won't really serve any function except to position elements, so you can use this to start your project.

```
.menu {
position: absolute;
font-size: 1.6em;
right: 36px;
top: 16px;
width: 39px; }
```

## 9. The main heading

The title is displayed over the top of the image, so it is centred horizontally and vertically. Normally if text needs to be centred vertically, a negative 50% of the height is removed from the top, but instead an image will be placed there with the text just sitting underneath that image.

```
.title {
text-align: center;
height: 60px;
left: 50%;
margin-left: -400px;
margin-top: 70px;
position: absolute;
top: 50%;
width: 800px; }
```

## 10. Heading style

The generic heading two tag is styled up here, so that all text appears in upper case, the bold is taken off and the letters kerned together slightly. The next styling is for the text in the main title, which is given a white background shadow, just to make it readable against the different watercolours of the image.

```
h2 {
font-weight: 200;
text-transform: uppercase;
letter-spacing: 0.1em; }
.title h2 {
text-shadow: 0px 0px 12px #fff; }
```

## 11. Buttoning up

The button is the image that will be placed over the top of the watercolour and above the main title, so the positioning here resolves that. The content section is below this and uses a patterned background to differentiate the content in this section from the watercolour image above.

```
.button {
left: 50%;
margin-left: -58px;
margin-top: -60px;
position: absolute;
top: 50%;
width: 117px; }
#content {
width: 100%;
background: #333 url(../img/ps_bg.png); }
```

## 12. Icon inside

Inside the button section, the icon is placed on the screen. This is an image and in this case is a Photoshop icon, as that is the subject of this page content. You could replace this with whatever is relevant to your project and however you intend to use the watery effect.

```
.icon {
background-position: center;
background-repeat: no-repeat;
background-size: contain;
height: 120px;
width: 117px;
background-image: url(../img/ps.png);
}
```

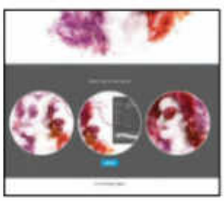
## 13. The lower section

The container section is just set up to display the content centrally on the screen, so here the container is given 1200px width and a margin of auto to centre it. There's a little padding on the top and bottom, just to give space to the content. The heading is given a slightly different colour to make it stand out in this section.

```
container {
display: block;
max-width: 1200px;
margin: 0 auto;
padding: 80px 0;
text-align: center;
}
```

### A call to action

This term might seem strange, but this is often a way that designers and marketing people describe an important task you might be trying to get your user to do, such as sign up for a service.



**Top left**
With all the CSS in place, the lower part of the screen – often known as 'below the fold' – finishes the design off for the page

**Top right**
With the default settings, the watercolour distortion is quite strong and this might be an effect that you need, but it is a little too distorted for this particular project

**Right**
Amending the amplitude and frequency of the Aquarelle effect really helps to give this a watery paint wash effect

### Syncing animation

In the final project, you can see that it is possible to sync up other animation in time with the effect as it plays, giving you full control over the way your page builds. This is the code found in steps 19 and 20.

The way this works is through Aquarelle's 'changed' event listener. Every frame that the Aquarelle effect is running, the 'changed' event fires, meaning that other events can be triggered based on the duration of how long the effect has been running. Let's examine the code that fires one of these events:

```
icon.style.opacity =
event.target.
transitionInRange(0, 1,
3500, 4500);
```

Here the Photoshop icon's CSS opacity is changed from 0, fully transparent, to 1, fully visible, and this transitions between 3.5 seconds and 4.5 seconds of the Aquarelle effect running. Armed with this simple piece of code, you can fully build your page around simple transitions that are all timed with the overall watercolour effect of Aquarelle.

```
container h2 {
color: #ddd;
font-size: 1em; }
```

## 14. Call to action

A larger button is placed on the screen so that people could sign up for this service. Here the CSS styles the 'a' tag into a button so that it stands out and can be easily clicked. There's a light blue background colour added to help it stand out on the design, which is a darker background at this point on the page.

```
a {
padding: 10px 20px;
color: #fff;
background-color: #19aae0;
text-decoration: none;
border-radius: 4px; }
```

## 15. Finishing off the CSS

The footer in the CSS rounds out the design styling of all elements now, so save the CSS content and switch back over to the **index.html** page. Scroll down to the closing body tag and add script tags that will contain the JavaScript functionality.

```
footer {
height: 90px;
padding-top: 15px;
text-align: center;
font-size: .8em; }
```

## 16. Bringing it all together

The first JavaScript will get references to various elements on the page so that they are cached inside their own variables. This makes changing their content much faster as when they are needed, the DOM has already been traversed and the tag found, so storing in a variable saves looking that up each time it's used.

```
var fade = document.querySelector('.fade');
var topNav = document.querySelector('.top');
var title = document.querySelector('.
title');
var icon = document.querySelector('.icon');
var image = document.
getElementById('watercolor');
```

## 17. Adding the water effect

To make the water effect that Aquarelle gives, we simply call for a new Aquarelle object and pass in the two images we need to use. The first is the image to show and the second the mask. It takes 8 seconds to reveal the content by default. You can't test it just yet as there are still some other parts to add.

```
var aquarelle = new Aquarelle(image, 'img/
mask.png', {
autoplay: true,
loop: false
});
```

## 18. A quick swap

When the aquarelle element is created, the fade content is given an opacity of 1 – it was set to 0 in CSS; this stops it flickering on then off again before the animation begins. The canvas that will render the image is added in front of it, and the image is taken away out of the page.

```
aquarelle.addEventListener('created',
function() {
fade.style.opacity = '1';
var canvas = this.getCanvas();
canvas.removeAttribute('style');
image.parentNode.insertBefore(canvas, image.
nextSibling);
image.parentNode.removeChild(image);
});
```

## 19. Animating content

As the watercolour effect appears, other animations will take place. The logo and menu icon will fade in from 5 to 8 seconds of the Aquarelle effect playing. The canvas will scale up from the centre from 75% to 100%, over the whole 8 second duration of the effect.

```
aquarelle.addEventListener('changed',
function(event) {
topNav.style.opacity = this.
transitionInRange(0, 1, 5000, 8000);
var canvas = this.getCanvas();
canvas.style.webkitTransform = canvas.style.
transform = 'translate(-50%, -50%) scale('
+ this.transitionInRange(.75, 1) + ')';
```

## 20. Final animations

The title is faded in from the start up until 2.5 seconds into the animation. It is also scaled up. The icon is faded in and also blurred, with the blur being removed over time, from 3.5 seconds to 4.5 seconds in. This is also given a scaling effect. Test this in the browser to see it in action.

```
title.style.opacity = this.
transitionInRange(0, 1, 0, 2500);
title.style.webkitTransform = title.style.
transform = 'scale(' + this.
transitionInRange(.8, 1, 0, 4500) + ')';
icon.style.webkitFilter = 'blur(' + event.
target.transitionInRange(4, 0, 3500, 4500) +
'px)';
icon.style.opacity = event.target.
transitionInRange(0, 1, 3500, 4500);
icon.style.webkitTransform = icon.style.
transform = 'scale(' + event.target.
transitionInRange(1.5, 1, 3500, 5000) + ')';
});
```

## 21. Final tweeks

The Aquarelle effect is good but can be improved by changing some of the default values. Add the amplitude values and frequency values to see the difference in the code added from step 17. The amplitude reduces from 5 to 0 so that it isn't distorted at the end. Save and give this a final test to see the watery, ripple effect grow onto the screen.

```
var aquarelle = new Aquarelle(image, 'img/
mask.png', {
fromAmplitude: 5,
toAmplitude: 0,
fromFrequency: 8,
toFrequency: 3,
autoplay: true,
loop: false,
toOffset: 50
});
```

# Learn how to get started with Sass

Discover the power and flexibility that CSS preprocessors can bring to your web design projects.


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

As our webpages and web apps become more complex, our style sheets also become larger and harder to maintain. One of the many tools we can use to help us maintain and manage our CSS is a preprocessor called Sass (Syntactically Awesome Style Sheets). Sass files cannot be interpreted by the browser, so they need compiling to standard CSS before they are ready to hit the web. That's why you need some sort of tool to help you translate .scss files into .css, and we'll take a look at one of those throughout the tutorial.

There are a few CSS preprocessors available; these include Less, PostCSS and Stylus. However, in this tutorial we will be focusing on learning the basics of Sass and we'll see how Sass can help you work faster and more efficiently. By using Sass we'll be able to streamline our CSS coding by giving us an arsenal of methods at our disposal, and in some ways it's a step into the world of 'object-oriented' CSS. We can nest styles, create functions, set variables, do maths, and a host of other goodies that only a programming language could do.

In this tutorial we will take things slowly and introduce you to the basic syntax of Sass and what tools to use. Then we will build a simple responsive navigation menu using as many Sass features as needed.

## 1. Downloading Koala

When using Sass we need a way to compile our .scss code to normal CSS. There are a few options to choose from and they all have their pros and cons. So to keep things nice and simple, we will use a free one called Koala – **koala-app.com**.

## 2. Using Koala

Once you have downloaded Koala and opened it up, create a new project folder called 'learn Sass' and then within the folder create a new index.html file, a styles.css file and a styles.scss file, and then drag the project folder into Koala and also open the .scss file within your text editor as we'll be doing all our coding in there.

## 3. Setting the output path

The very last thing we need to do before we can begin writing Sass is to set the output path. So, while in Koala, right-click on the green button (styles.scss) and choose 'set output path'. Then you'll be given an option to choose a file path. Choose 'styles.css' that's also located in your project folder. Now the .scss file will map to your .css file.

## 4. Saas variables

Variables in Sass work in a similar fashion to those in any programming language. When defining a variable we store inside it a certain value, which usually is something that will often reoccur in the CSS, like a palette colour or a font family for example. The way you declare a variable is with the $ sign.

```
// .scss
$font-family: 'Oxygen', sans-serif;
body {
font-family: $font-family; }
```

## 5. Global variables

In the styles.scss file, we can declare all our variables at the very top. By doing this we can easily access them and these will be considered global variables as they can be accessed from anywhere in our stylesheets (.scss). We will keep our variables simple and useful and follow best practices with regards to naming convention.

```
$white: #fff;
$light-grey: #c3c3c3;
$black: #000;
$nav-background: $light-grey;
$nav-font-color: $white;
$link-hover-color: $black;
```

## 6. Introduce comments

Comments are a great way to keep your stylesheets organised and there are two different types of comments to consider in Sass. Single-line comments (//) will be removed by the .scss preprocessor and wont appear in your .css file. Then there's the normal multiline comment (/*) that is valid CSS and will be preserved and appear in the .css file.

```
/* This is a comment that can be seen in
.css */
// This comment will be hidden in the .css
file
```

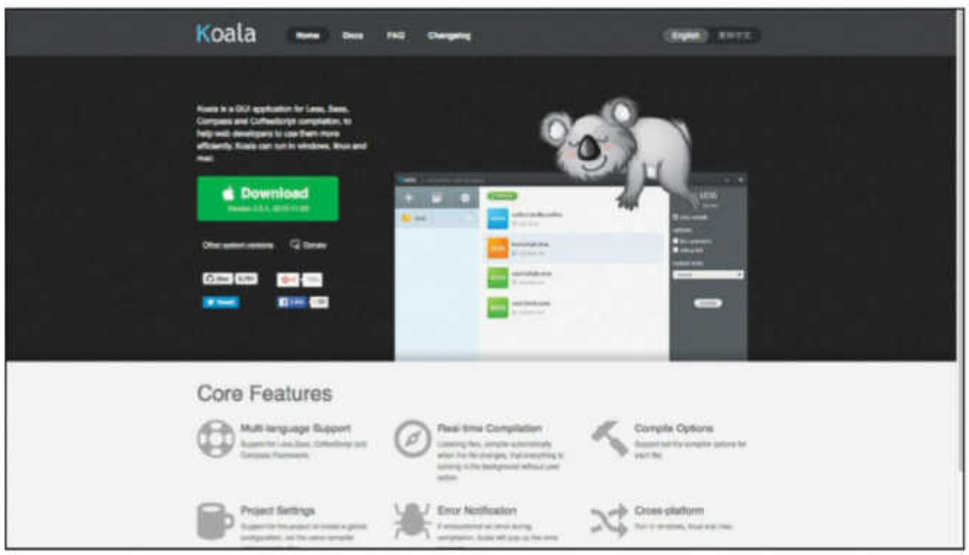
## 7. Nesting structure

HTML follows a strict nesting structure whereas in CSS it's usually a total mess. With Sass nesting you can organise your stylesheet in a way that resembles the HTML more closely, thereby reducing the chance of CSS conflicts. However, only nest if it's necessary and just indent the rest.

```
// The SCSS
ul {
list-style: none;
li {
padding: 15px;
a {
text-decoration: none;
font-size: 14px;
color: $white;
}
}
}
/* The CSS output
ul {
```

### Comment your code

It's good practice to leave comments within your CSS code, especially when using Sass, as it helps to keep things organised.




**Left**
The first step to writing Sass is to download and install the Koala app for either Windows or Mac

**Top left**
The HTML has been added and we can now begin to add some CSS

**Top right**
We have now added our first bit of CSS, by adding a background image and background colour to the navigation bar

```
list-style: none; }
ul li {
padding: 15px;
display: inline-block; }
ul li a {
text-decoration: none;
font-size: 16px;
color: #444; }
```

## 8. Mathematical operations

With Sass you can do basic mathematical operations right in the stylesheet and it is as simple as applying the appropriate arithmetic symbol. So, what if we wanted to split a column in quarters? We can do so by specifying the width using a variable and then using the division operator as shown.

```
// SCSS
$width: 800px;
.container {
width: $width; }
.column-half {
width: $width / 2; }
.column-forth {
width: $width / 4; }
/* Normal CSS
.container {
width: 800px; }
.column-half {
width: 400px; }
.column-quart {
width: 200px; }
```

## 9. Using mixins

Mixins let you define common properties once and then reuse them. You can think of mixins as a simplified version of constructor classes in programming languages. CSS3 styles that require vendor prefixes are a perfect example of when to use a mixin. Instead of typing the same property over and over, we write (declare) a mixin once, then call that mixin any time we want to use it.

```
//declare mixin(now being passed an
argument)
@mixin box-sizing($boxSize) {
-webkit-box-sizing: $boxSize;
-moz-box-sizing: $boxSize;
box-sizing: $boxSize; }
//use mixin
.mySelector {
@include box-sizing(border-box); }
/* Normal CSS
mySelector {
-webkit-box-sizing: border-box;
-moz-box-sizing: border-box;
box-sizing: border-box; }
```

## 10. Building a responsive nav bar

Now that we have learned the basics of Sass, we can put that newfound knowledge to the test and begin developing a simple navigation bar. So, open up your index.html file (make sure the styles.css file is linked within the head) and begin by adding the HTML.

## 11. Nav list

Our navigation bar will be responsive and be controlled by a bit of jQuery (which you can see in the section at the top of the next page). So we need to add an ID name of 'nav-toggle' that we can target using jQuery. Then we will add in our navigation list for our menu buttons.

## 12. Links to scripts

As pointed out in the last step, we will be using a touch of jQuery to control the toggle effect of our nav bar. So first we will need to include the CDN link to jQuery and then, if you haven't already done so, create a new file and call it scripts.js. Then we can link to that straight after the jQuery has loaded.

## 13. Global variables.

Create a new '.scss' file and open it up in a text editor. Then add in some global variables for us to use throughout the project. These will need to be added to the very top of the .scss file and only include ones that would more than likely change in the future. So there's no need to add variables for the sake of adding them.

```
// Navigation Variables
$white: #fff;
$dark-blue: #002a3d;
$dark-green: #004952;
$nav-background: $dark-green;
$nav-font-color: $white;
$link-hover-color: $dark-blue;
$content-width: 1000px;
$breakpoint: 800px;
$nav-height: 70px;
$font-family: 'Oxygen', sans-serif;
```

## 14. Adding some styling

To keep things more manageable, we will use the @import option to link to Google fonts. The font we will be using is called Oxygen. Then we can add a nice background image and then use our variables for the first time by setting the height and background of the nav bar.

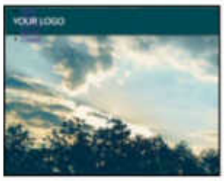
```
// Import fonts
@import url('https://fonts.googleapis.com/
css?family=Oxygen');
body {
background: url('imgs/lake-bg.jpg')
no-repeat center; }
.navigation {
height: $nav-height;
background: $nav-background; }
```

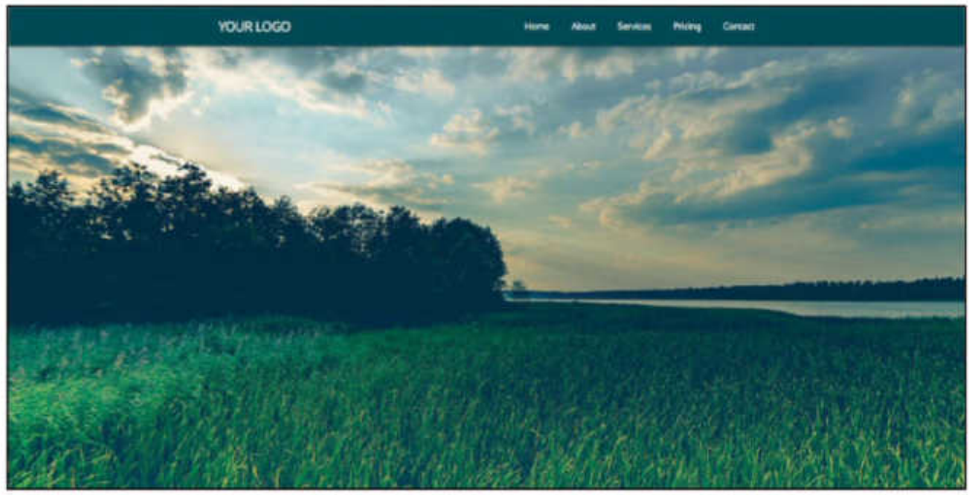
## 15. Logo and nesting

Twitter Bootstrap uses a class called 'brand' that sets the

### Creating variables

It's easy to go mad with variables in Sass, but a good tip is to only create ones that you will more than likely need to change in the future, such as colours and fonts.

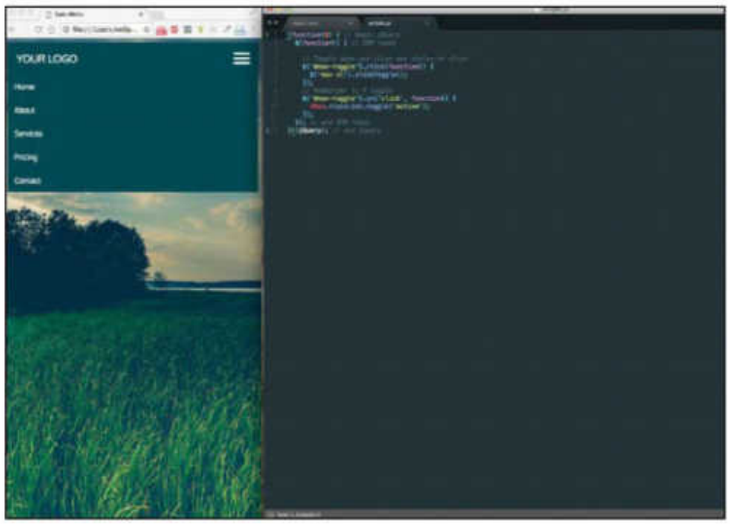



**Top left**
Even though we're not using Twitter Bootstrap, we will still use the 'brand' class for our logo

**Top right**
The navigation menu buttons are now nicely spaced out and styled accordingly

**Right**
To finish up the navigation bar, we will make sure all the menu buttons and logo are centred




### The navigation jQuery

A responsive navigation menu that displays the hamburger icon button when resized will most likely have some type of animation occur when clicked on. Therefore in this tutorial we will use some custom jQuery to create a nice drop-down animation. So if you haven't already done so, add a link to the jQuery CDN at the bottom of your HTML document that's just above the closing </body> tag, and a custom 'script.js' file.

```
<script src="https://
code.jquery.com/jquery-
3.1.1.min.js"></script>
<script src="scripts.
js"></script>
</body>
</html>
```

Then we can open the scripts.js file and add in our jQuery.

```
(function($) { // Begin
jQuery
$(function() { // DOM
ready
// Toggle open and close
nav styles on click
$('#nav-toggle').
click(function() {
$('nav ul').
slideToggle();
});
// Hamburger to X toggle
$('#nav-toggle').
on('click', function() {
this.classList.
toggle('active');
});
}); // end DOM ready
})(jQuery); // end jQuery
```

styles for the nav bar logo. Even though we're not using Bootstrap, we will still use that class name just in case we decide to use Bootstrap later on. Then we can nest the styles for the anchor links.

## 16. Styling the menu buttons

Styling the list items for our menu buttons will give us a good opportunity to use a bit more nesting. We can also make good use of our global variables.

```
nav {
float: right;
font-family: $font-family;
ul {
list-style: none;
margin: 0;
padding: 0;
li {
float: left;
position: relative;
a, a:visited {
display: block;
padding: 0 20px;
line-height: $nav-height;
background: $nav-background;
color: $nav-font-color;
text-decoration: none;
&:hover {
background: $link-hover-color;
color: $nav-font-color;
}  }
...
```

## 17. Centre menu buttons

Now we need to centre the navigation buttons. By using the $content-width variable value, which is 1000px, we can centre the logo and navigation buttons using 'margin: 0 auto'.

```
.nav-container {
max-width: $content-width;
margin: 0 auto; }
```

## 18. Mobile navigation

Before we start adding the media queries for the responsiveness of our navigation, we first need to specify the styles and positioning of the hamburger menu icon, which will be positioned to the right-hand side.

```
.nav-mob {
display: none;
position: absolute;
top: 0;
right: 0;
height: $nav-height;
width: $nav-height;
margin: 8px; }
```

## 19. Media Queries

To make our navigation responsive, we need to add in some media queries with a few breakpoints added. Again, you can see how useful variables prove by how we can use the $breakpoint variable on the 'max-width' property.

```
@media only screen and (max-width:
$breakpoint) {
// Hamburger nav visible on mobile only
.nav-mob {
display: block; }
nav {
width: 100%;
padding: $nav-height 0 15px;
ul {
display: none;
li {
float: none;
a {
padding: 15px;
line-height: 20px; }
ul li a {
padding-left: 30px;
} } } } }
@media screen and (min-width: $breakpoint)
{
.nav-list {
display: block !important;
} }
```

## 20. Display the hamburger icon

Up to this point, we haven't been able to see the hamburger icon in our navigation bar when we resize the browser. That's because we haven't yet created it. To create this we will use and style three span tags and use the 'before' and 'after' pseudo classes to position the horizontal white lines.

```
#nav-toggle {
position: absolute;
left: 18px;
top: 22px;
cursor: pointer;
padding: 10px 35px 16px 0px;
span,
span:before,
span:after {
cursor: pointer;
border-radius: 1px;
height: 5px;
width: 35px;
background: $nav-font-color;
position: absolute;
display: block;
content: ''; }
span:before {
top: -10px; }
span:after {
bottom: -10px;
}
}
```

## 21. Sass workflow

This tutorial only scratched the surface of what Sass can do. As a web designer or front-end developer, using Sass (or any other preprocessor) will surely help speed up your workflow and keep your projects manageable.

# Create custom page layouts in WordPress

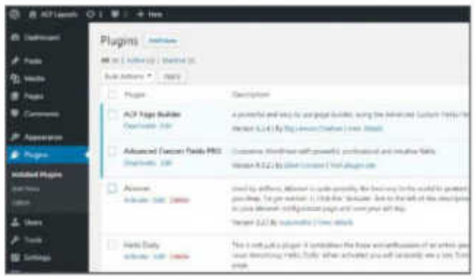
Make a drag-and-drop edit panel for your WordPress pages using ACF Pro and ACF Page Builder plugins


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

**Sometimes it can be difficult to make a dynamic page on the frontend while maintaining an easy-to-use edit screen, particularly if you're splitting your content into HTML blocks for use with Bootstrap or other gridding tools.** Putting HTML directly into your edit screen can get unwieldy, and if non-developer clients need to edit it they'll be mired in confusing code. Using shortcodes to take the place of your HTML can tidy it up a little, but there's always the danger one tag will go missing, or the client will want to reorder the blocks. There are many plugins on the market that help customise WordPress that will blend dynamic front-end display with a sleek and accessible admin panel. One good solution comes bundled with ACF (Advanced Custom Fields) Pro – the Flexible Content Field add-on. Using the power of this Flexible Content Field add-on, the ACF Page Builder plugin will enable the developer to define layouts and block types with various input fields that can be utilised on the frontend while giving the client an accessible and simple-to-use interface for when they are editing.

## 1. Install ACF Pro

Get started by purchasing and installing the ACF Pro Plugin from **advancedcustomfields.com** and install the free Advanced Custom Fields Page Builder plugin from **WordPress.org**. Activate both, and you should now have a new Custom Fields tab in your admin panel.



## 2. Create a field group

Go to Custom Fields tab in the admin panel and Add New Field Group. This group will hold the fields you will need later. Name it and set the rules for where the fields will display, eg Page Template equal to Default Template will make them available on pages with the Default Template.

## 3. Add a field

Use the + Add Field button to create a field which will hold the layouts you will need later. Name it and most importantly set Field Type to Flexible Content. The other options can be left for now, but will be useful for customising your display later.

## 4. Set the layout options

Define the layout block by giving it a name and a layout type. This tutorial makes use of Row for the layouts, but Table and Block will still be useful for other tile-like designs. Setting Minimum and Maximum rows will help to define how many of that layout type can be utilised on each page.

## 5. Add layout fields

Each layout can have multiple fields with different field types, eg title, content and thumbnail. Add and name the fields that you'll need to display in the frontend. You can also use fields for content that won't be displayed, like settings and IDs.

## 6. Add additional layouts

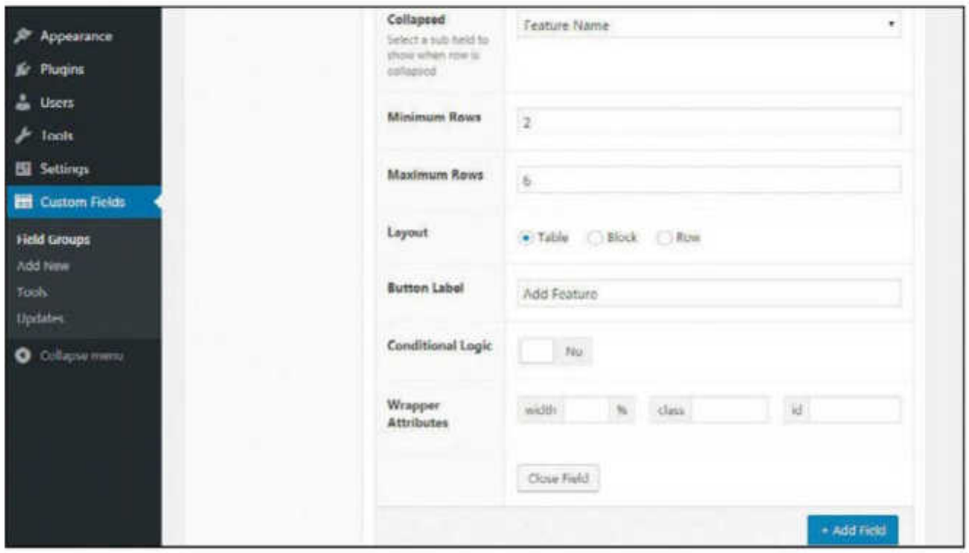
Now let's repeat the previous step for as many different layouts sections as you will be requiring. You can add additional layout sections by hovering over layout and selecting Add New. Every layout that you add will be available later in the edit panel, and can be reordered on the fly.

## 7. Save the layouts

Once you've finished adding your layouts, save the Custom Fields by scrolling to the top and clicking Update.

## Using wrapper attribute settings

Use the wrapper attribute settings to set classes and IDs for the back-end display, making it possible to further customise the layout of the admin panel. These wrapper attributes are not used in the frontend.




**Left**
Set a Minimum and Maximum Rows to impose a limit on the number of items the user can add. Change the button label to customise the backend display

**Top left**
When a new layout of that type is added, the admin panel will automatically add the minimum number of fields and return an error if required fields are not completed

**Top right**
Similarly, when the maximum number of rows is reached, an error will occur if the user attempts to add more rows than the layout allows

You can edit the layouts later, but be careful changing field IDs as they may conflict with the front-end code.

## 8. Edit the page

The newly saved fields can now be populated via the editing screen for the page you are customising. With default settings, there will be a new section below the content editing screen with the option to Add Row. You can customise the buttons and titles back on the Custom Fields admin panel.

## 9. Add a layout row

Add a block from the options under Add Row Populate the fields with your custom content. Now is the time to edit your Custom Fields if you've missed anything, before the frontend is coded and you start relying on field IDs.

## Keep things simple

If the layouts are being used by clients rather than developers, try and limit the amount of inputs the user needs to complete and name them all appropriately to reduce confusion.

## 10. Additional rows and reorder

Use the + and – buttons to add or remove another layout, or drag the layout bar to change the layout order. Use the drop-down arrow to collapse each row for a cleaner list. When you're finished, don't forget to save!

## 11. Start the frontend

Until the coding is in place, the edits to the admin panel won't display on the frontend. Currently the page template is only set to show the page title and main content. The blocks will now be coded to display below the main content.


## 12. Check for layouts

Edit your page template; start by adding an if statement that checks if the user has saved any ACF entries on the current page. If no, nothing will display. If yes, this is where the code to display the fields will go. 'Content' would be the ID/name of the Field Group from Step 2.

```
<?php
// check if ACF layouts are set
if( have_rows('content') ):
// loop through the layouts here
else :
// no layouts found
endif;
?>
```

All of the code is saved after the loop in content-page.php, using a prestyled child theme of Flat Bootstrap (**Byxtremelysocial.com**) – no CSS has been included with this tutorial.

## 13. Loop through the layouts

If there are layouts set, the code needs to loop through each layout in turn ready to display them one by one. Have_rows() is an ACF function that returns a Boolean if there are rows set on that page for the field group specified. The_rows() calls the array of values ready for use in the next step.

```
while ( have_rows('content') ) : the_row();
//define the layout type here
endwhile;
```

## 14. Define layout type

For each layout type, insert an if statement that checks which layout is currently being looped through. This defines a different structure for each of your layouts. Add an if statement for each layout type you defined. 'Block' would be the name given to the layout in Step 4.

```
// BLOCK layout
if( get_row_layout() == 'block' ):
//echo layout content here
endif;
// FEATURE layout
if( get_row_layout() == 'feature' ):
//echo layout content here
endif;
```

## 15. Echo layout content

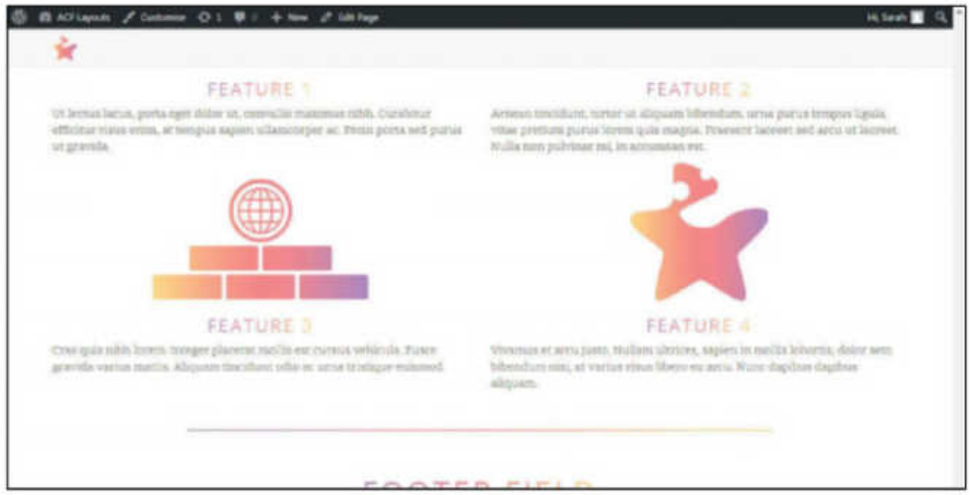
Using ACF functions, echo each of the sub-fields for the layout and wrap them in your desired HTML. Get_sub_field() will retrieve the value for use in a string, whereas the_sub_field() will echo the value directly. Parameters such as block_id are the field names we set in Step 5.

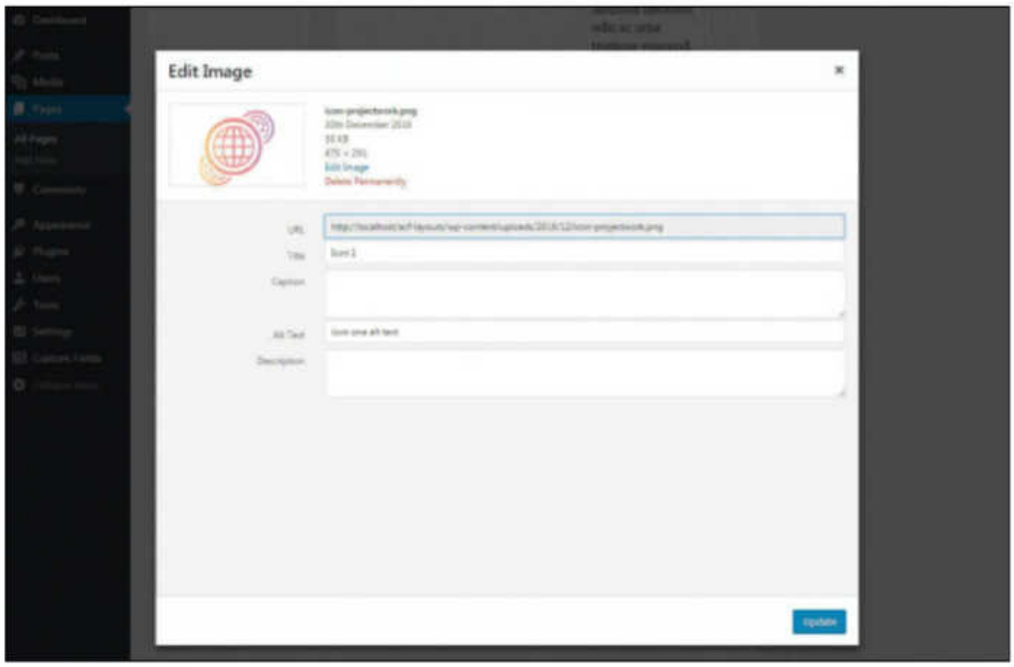
```
echo '<div id="'.get_sub_field( 'block_id'
).'">';
echo "<h2>" . get_sub_field( 'block_name' )
```

**Top left**
Add non-flexible content boxes for content that has a set order or placement, like a content footer. Add it as a new field like in Step 3

**Top right**
The new fields will display on the admin edit panel in the order you set. Unlike the flexible content fields, they can't be moved or repeated

**Right**
Add code to display the new content. The function used instead is get_field() rather than get_sub_field()as it is a top-level field






### Displaying images in detail

There are a few different ways to display images using ACF fields and WordPress functions, one way is to utilise the wp_get_attachment_image() function.

```
$image = get_sub_
field('feature_image');
echo wp_get_attachment_
image( $image['ID'],
array('300','184'), true,
array('class' =>
'aligncenter') );
```

First set the array of image properties returned from ACF to a variable. Use $image['ID'] to insert the ID into the first parameter of the attachment function. Set the rest of the parameters as defined by the wp_get_attachment_image function. The resulting echoed output will be an image tag that automatically includes the image URL in the defined size, with the alt text set in the edit screen.

```
. "</h2>";
the_sub_field( 'block_content' );
echo '</div>';
```


## 16. Add classes and styles

The block layout should now be displaying on your page. Insert classes into your code to fit the layout to your theme. This example uses Bootstrap classes to align the display with the main page content.

```
echo '<div class="center-text col-md-8
col-md-offset-2 col-sm-12">';
echo "<h2>" . get_sub_field( 'block_name' )
. "</h2>";
echo '</div>';
echo '<div class="col-md-8 col-md-offset-2
col-sm-12">';
the_sub_field( 'block_content' );
echo '</div>';
```

## 17. Validate optional fields

For fields that aren't mandatory, don't forget to include an if statement to check if they are set before displaying them. Use the if statements to change the display format if certain fields are available, eg adjusting the columns to include the thumbnail on the side.

```
if( get_sub_field('block_thumbnail') ):
//display thumbnail if set
endif;
```


## 18. Add repeater loops

For layouts that utilise the repeater field, check that they are set and then use another while loop to display each of the items for that layout. Next, place the code inside the if statement for the layout that it's contained in, as outlined in Step 14.

```
if( have_rows('feature_block') ):
// loop through the rows of data
while ( have_rows('feature_block') ) :
the_row();
//echo repeater content
endwhile;
endif;
```

## 19. Echo repeater content

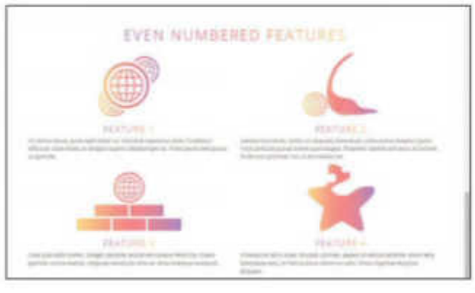
Echo the values inside the while loop, as in Step 15. Using the Minimum and Maximum Rows when setting up the fields can limit the number of repeated fields allowed, making it easier to define the limitations of your layout.

```
$image = get_sub_field('feature_image');
echo wp_get_attachment_image( $image['ID'],
'medium', true, array('class' =>
'aligncenter') );
echo "<h3>" . get_sub_field( 'feature_name'
) . "</h3>";
echo get_sub_field( 'feature_content' );
```

## 20. Detect row count

By using if statements in combination with the count() function you can display different block configurations depending on how many rows there are. For example, you can show features in columns of two or three if they are an even or odd number.

```
$rowcount = count( get_sub_field('feature_
block') );
if( have_rows('feature_block') ):
while ( have_rows('feature_block') ) :
the_row();
if ($rowcount % 2 == 0):
//display even rows layout
else:
//display odd rows layout
endif;
endwhile;
endif;
```




## 21. Putting it together

With the flexible nature of layouts, you can create any configuration you need to fit your theme. By experimenting with different field types and conditional statements you can make a multitude of layouts that meet both your, and the client's, needs.

# DELIVER FIRST-CLASS JAVASCRIPT

*The latest tools and techniques for keeping projects on track*

**JAVASCRIPT IS ONE OF THE MOST FLEXIBLE LANGUAGES OUT THERE. WHILE THAT MAKES IT GREAT FOR A BEGINNER, OVER TIME LARGER PROJECTS WILL NEED DIFFERENT STRUCTURES TO STAY MANAGEABLE. KEEPING CODE AS SIMPLE AND MODULAR AS POSSIBLE HELPS TO USE THAT FLEXIBILITY AS AN ADVANTAGE.**

***Matt Crouch***
Developer at Cinolla
***mattcrouch.net***

# 5 REASONS TO LOOK AFTER YOUR CODE

*HELP FUTURE MAINTAINERS OUT BY KEEPING YOUR CODE IN CHECK*

### 1. Smoother refactoring

The ecosystem on the web is constantly changing. When a framework or third-party plug-in needs to be updated, a monolithic codebase can cause these changes to become cumbersome to implement and risky to deploy. With modularisation, issues can be isolated and fixed in place rather than bringing down an entire site.

### 2. More portability

When a complex feature gets developed, it's a great idea to reuse that code where necessary to avoid slowing development. Modularising code can allow developers to drop in functionality where needed instead of copying and pasting into place. Equally, if a bug is found, only one change is needed instead of many.

### 3. Easier to bring in new developers

Websites change hands a few times during their lifespan. If the code is written in a clean and predictable pattern, new developers will find it easier to contribute later down the road. Even code written by you will be quicker to pick up again if changes are necessary a few years on.

### 4. Faster debugging

Code editors are becoming wise to common development patterns. For example, Visual Studio Code can bring up a description for a method as you use it and identify any missing or incorrect arguments. By using the right structure in your code, your editor can help identify syntax errors as you go.

### 5. Less technical debt

While it may be tempting to write or copy a quick fix in for a small feature, things have a tendency to grow over time. What saves you five minutes now can actually cost you tenfold down the line. Treating every feature the same – big or small – will stop too many issues flaring up in the future.

# 3 TOOLS TO IMPROVE YOUR CODE

### ESLint

**eslint.org**

Identifies and suggests corrections for poorly written JavaScript as code is created. It can be used as an extension to a code editor or plug-in as part of a build process.

### Lighthouse

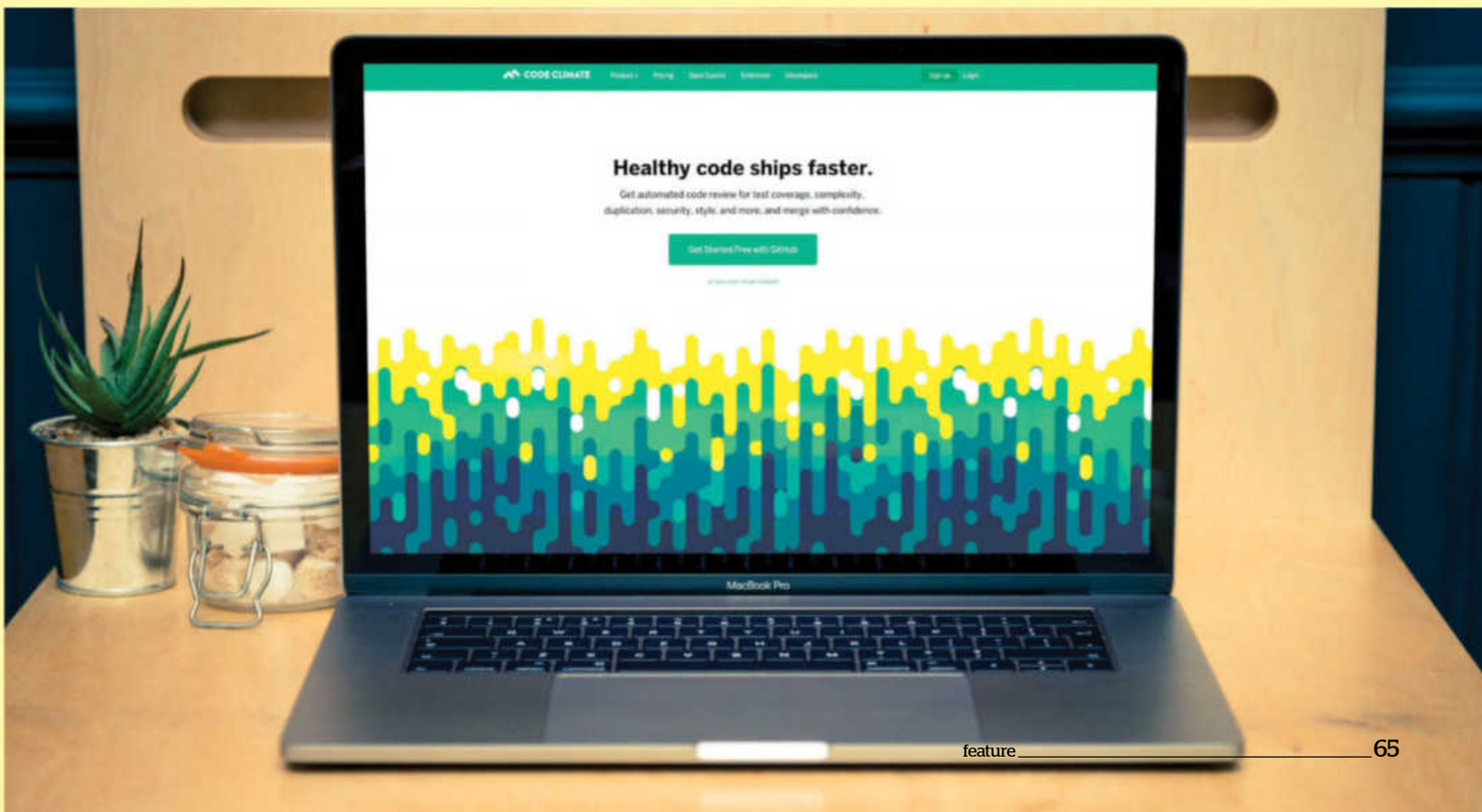
**bit.ly/2jnLfpG**

Geared towards PWAs, Lighthouse will audit a site for best practices in regards to performance, new features and removal of anti-patterns. Can run as a CLI or a Chrome extension.

### Code Climate

**codeclimate.com**

Identify issues with projects as a whole as they occur. Code Climate can detect a wide range of issues, from duplicated code to a decrease in test coverage. Not free for private projects.



# KEEPING CODE MAINTAINABLE

*DO FUTURE DEVELOPERS A FAVOUR BY SKIPPING THE SPAGHETTI CODE*

**Things move fast in this industry.** Finding a bug in a legacy project can involve jumping through hoops: following the trail of code in one giant file, only to replace a couple of lines. This headache can be avoided with a bit of forward thinking from the beginning.

First of all, think modular. A few years ago, best practice was to place all logic in one file to reduce downloads. Now, in the age of build processes, we can condense many files into a single, minified JavaScript file. Splitting code up into modules saves duplication, makes things more readable for developers, and tools like Webpack can optimise their loading to make things quicker for the user. Unit tests also become easier to write if the project does not have them already.

**EASY COMMENTS WITH PLUG-INS**

*Tools like JSDoc can help document code as you go. Download a plug-in to suggest comments and take out some of the repetitiveness.*

Once functionality is broken up, keep those files organised. Create separate directories for code that deals with data storage, computes data or displays it. That way, if an issue arises it becomes easier to find the source. The best structure will vary between projects, so use one that works best in each case. A React application, for example, might choose to separate by component role.

Code should also be kept documented. While it may be obvious at the time of creation what a set of function calls do, at some point down the line someone else will need to make sense of what that code is doing. Often this is done through commenting in the code itself. Code editors can pick up these comments and present them as hints when that function or variable gets used later on.

However, comments can be overkill for simple operations. A sensible name for a function or variable is sometimes all that's needed to describe its purpose. Longer names can be reduced through minification, so will not impact on file size for the end user.

```
//Avoid ambiguous names
let arr = [];
arr.push(first.result);
//Be descriptive
let testResults = [];
testScores.push(firstTest.result);
```

*Code editors, such as Visual Studio Code, can pick up on JSDoc-style formatted comments to provide hints when the Workspace class gets used*

## KEEPING CODE READABLE

While the right JavaScript will make your code work, unless it's presented correctly it becomes difficult for teams to maintain.

A few small changes can make all the difference. Naming variables and functions clearly can save time finding the right one later on. Stick to only using either single or double quotes throughout a file. Avoid creating code that tries to do too much at once.

Keep an eye on the latest updates to the specification. New syntax such as arrow functions and generators can create much clearer code without the workarounds that have been used in the past.

Style guides within larger groups can be beneficial. These can be combined using code linters to flag issues as they arise, or even stop poorly constructed code from being committed altogether. Some companies like Google and Airbnb share their style guides publicly, which can serve as a great learning resource to get started.

# USE ES2015 RIGHT AWAY

*HOW THE LATEST FEATURES CAN SAVE YOU TIME AND EFFORT ACROSS PROJECTS*

**It's always worth keeping an eye on the latest advances in JavaScript.** A new process means great, time-saving features are added to the specification all the time. Code transpilers like Babel allow these features to be used straight away, so there is no need to worry about browser support.

Arrow functions are a great place to start streamlining older code. While they are great for small, anonymous functions, they are flexible enough to work in all situations. Assumptions can be made about the number of parameters and return values to reduce boilerplate code. It also has a lexical 'this', meaning it takes the value of 'this' from the surrounding context rather than creating its own, so binding it is unnecessary.

```
var numbers = [1,2,3];
var doubled = numbers.map(number => number
* 2);
console.log(doubled); //[2, 4, 6]
```

ES2015 also brings the concept of promises, which revolutionises the way asynchronous JavaScript is written. Function callbacks can get out of hand when dealing with multiple layers. With promises, these calls become more linear and easier to read. They represent a pending value that may arrive – or 'resolve' – at some point in the future. They can be returned from functions like regular values, which allows other code to pick them up and act on their eventual value themselves. A lot of new browser APIs use promises, so it's best to get to know them now.

```
var promise = new Promise(function(resolve,
reject) {
setTimeout(() => resolve(true), 1000);
});
promise.then(value => console.log(value)); //
true, after 1s
```

Template literals are strings that allow computations inside of them. Instead of concatenating strings with JavaScript values, template literals use backtick notation. These can contain placeholders that hold small expressions to be evaluated when that string is read. If you use a

library for templating like Handlebars, these might work as a suitable replacement and reduce the size and dependencies of your project.

```
var person = {
name: “Matt”,
role: “front-end developer”
}
console.log(`This is
${person.name}. He is a
${person.role}`);
//”This is Matt. He is a
front-end developer”
```

## USE ES2015 MODULES & WEBPACK 2

*ES2015 modules can now be used with Webpack 2 without the need to convert to CommonJS/ AMD. Helps keep modules fast and portable.*


Finally, the ES2015 specification brought a native implementation of modules to JavaScript. While third parties have been managing their own module systems for a while now, going forward it is a good idea to use the platform where possible. These modules work in a similar manner to the existing systems. The ‘export’ keyword can define the functions and values you want your module to support, and ‘import’ can specify other modules you want to consume within it.

```
//doubler.js
export function double(value) {
return value * 2;
}
//file.js
import double from “doubler”;
console.log(double(3)); //6
```

ES2015 modules will eventually be supported natively by browsers, but they aren’t quite there yet. For now, they need to be compiled into existing module formats first, but with the adoption of HTTP/2 the prospect of native module importing is not far away.

## 5 VITAL PACKAGES FOR YOUR NEXT PROJECT

### Yeoman
**yeoman.io**

Generate starting points for different types of projects such as React or Ionic that use recognised best practices, or create custom ones for your own uses.

### Babel
**babeljs.io**

Write code using the most up-to-date standards out there and have them compiled to previous versions that older browsers can understand.

### Gulp
**gulpjs.com**

A project build system that can be used to automate long-running and tedious tasks, such as running tests, minifying code and optimising images.

### Express
**expressjs.com**

Quickly set up web servers for Node.js projects to serve websites or provide an API for local development. It can be extended for a variety of different uses.

### Redux
**redux.js.org**

A powerful and flexible state management system for JavaScript-heavy applications. Combined with development tools with ‘time travel’ capabilities that makes debugging easier.

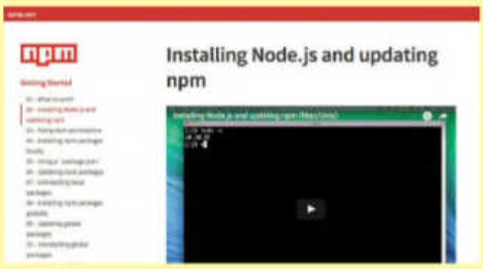
# GET NPM UP AND RUNNING

*MANAGE CODE WITH NODE'S PACKAGE MANAGER*

### 1. Download Node.js and npm

If you don't have Node.js on your system, it should be installed before continuing. Instructions for each operating system can be found on the Node.js website at **nodejs.org/en/download**. npm comes as part of Node.js, but it will need updating separately before continuing. Run the following on the command line.

```
npm install npm@latest -g
```




### 2. Initialise a project

Your project must be set up to accept packages before using them. After answering a few questions, the following command will create a package.json file in the root of your project. This keeps a reference to all packages the project uses. Navigate to the project folder on the command line and run this command.

```
npm init
```

### 3. Download a package

You can find many packages already created on the npm website at **www.npmjs.com**. Use the following command to include them in your project. This will download the 'webpack' package into your project, while the '--save-dev' flag means that other developers who download your module will also get this package to help develop this project further.

```
npm install webpack --save-dev
```

# AMD AND COMMONJS

*MODULE STANDARDS YOU CAN USE TODAY*

**If you have ever looked into modularising JavaScript, you have most likely heard of AMD and CommonJS.** They have been the go-to formats to modularise code and import dependencies for a while now. While ES2015 modules will eventually replace their role, projects can still make use of them today.

Although AMD and CommonJS are very similar in their intentions, they are separate in their approaches. These differences lend themselves better to certain scenarios.

AMD stands for Asynchronous Module Definition, which, as its name suggests, allows developers to define modules and pull them in as required. This is of great benefit to the frontend, as this means smaller files are downloaded that include only what is needed at the time.

A module is created using define(). This call will include a list of other modules it depends on and a function that gets called when all of these dependencies are loaded in. To conditionally import other modules that won't need to be exported itself, you can also use require().

```
define(
'myModule',
['moduleA', 'moduleB'],
function(moduleA, moduleB) {
var myModule = {
getAllNames: function() {
return moduleA.getNames().concat(moduleB.
getNames());
}
};
return myModule;
}
);
```

**WRITING FOR BOTH FORMATS**

*Check for 'module.exports' object or 'define' function to detect format being used and provide functionality regardless of setup.*

CommonJS is a format more suited to be used server-side. Unlike AMD, these modules will be loaded synchronously. While this may make things easier to read, it comes at a cost of speed and as such is not suitable for the client side.

Modules are shared using the module.exports object. Anything that is included in that object then becomes accessible when another module calls require() and assigns it to an internal variable. As things are kept synchronous, modules are fetched only when they are needed.

```
var moduleA = require("./moduleA");
var moduleB = require("./moduleB");
module.exports.getAllNames = function() {
return moduleA.getNames().concat(moduleB.
getNames()); }
```

While ES2015 modules may replace them in the future, it's important to know how these systems work, since a lot of existing modules will still use these formats. Tools like Webpack and Browserify will allow you to keep using these tools together as preference shifts.

***"ALTHOUGH AMD AND COMMONJS ARE VERY SIMILAR IN THEIR INTENTIONS, THEY ARE SEPARATE IN THEIR APPROACHES"***

## 10 ESSENTIAL TOOLS TO KEEP CODE AT ITS BEST

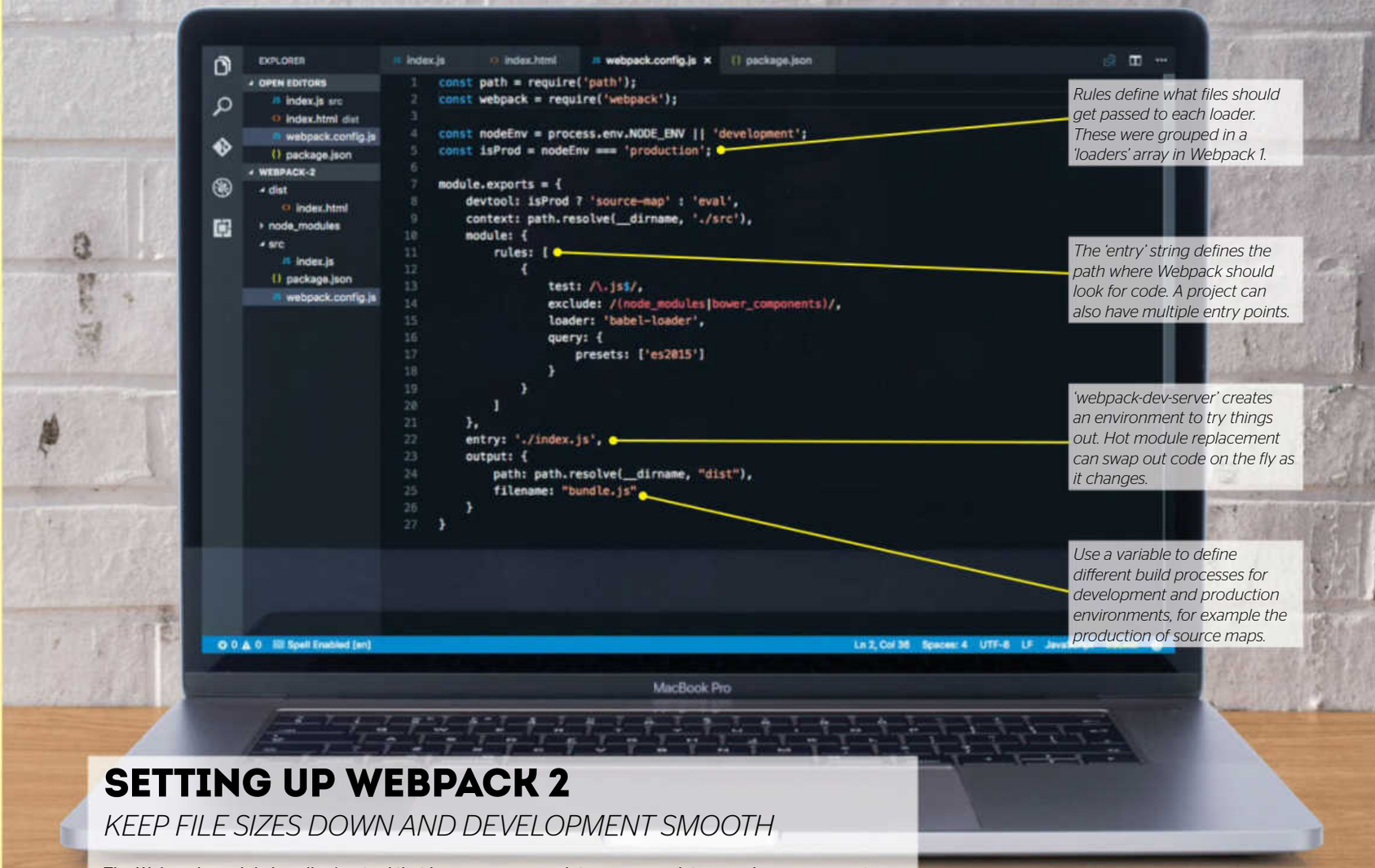
**JSHint**
**jshint.com**
JSHint can be viewed as a like-for-like alternative to ESLint. While much of the behaviour is the same, it comes with a reduced set of default rules. As a result, this allows developers more flexibility depending on the language being evaluated.

**TypeScript**
**typescriptlang.org**
Avoid the pitfalls of loose typing by explicitly defining variables to make code run as intended. Code editors pick up this information and make sure functions are not called incorrectly. All code is then compiled to JavaScript for browsers to understand.

**JSDoc**
**usejsdoc.org**
Automatically generate comments for functions and modules to quickly describe what they do. JSDoc can be used as a plug-in for most code editors, to generate comments as you go and save other developers time later on.

**David**
**david-dm.org**
David is a dependency management tool for npm packages. It keeps an eye on which ones are being used in a project and if one becomes out of date or has vulnerabilities, it will flag them for attention. A very useful tool.

**Closure Linter**
**github.com/google/closure-linter**
Closure Linter will check code against Google's own style guide to keep things good. It can be fixed automatically through the included 'fixjsstyle' application, which will update smaller issues.




# SETTING UP WEBPACK 2

## *KEEP FILE SIZES DOWN AND DEVELOPMENT SMOOTH*

The Webpack module bundler is a tool that has become indispensable in application development over the past couple of years. With the help of loaders, it can reduce the size of files that the end user needs to receive at a given point in an application. While this example is a relatively simple one, Webpack is flexible enough to accommodate complex, component-heavy applications.

Webpack 2 provides a few improvements – most notably the native handling of ES2015 modules and the conditional loading of modules within them. While it isn't ready for production use yet, it is a good idea to be familiar with the syntax changes.

### Sidekick

**sidekickcode.com**

Check updated code against company standards and identify security issues as you go. Sidekick checks either as it gets pushed or on an ad-hoc basis through the command line. It can also provide hints on how to fix some issues directly.

### Notes

**github.com/stephenb/node-notes**

A common problem with projects as they age is things get marked as 'TODO' and never get looked at again. Notes is a simple command-line tool that analyses a project and collects them into a handy list.

### Plato

**github.com/es-analysis/plato**

Large files and multiple methods in an object often point to a module trying to do too much by itself. Plato allows you to visualise file complexity and identify what files should be split into smaller chunks.

### Codacy

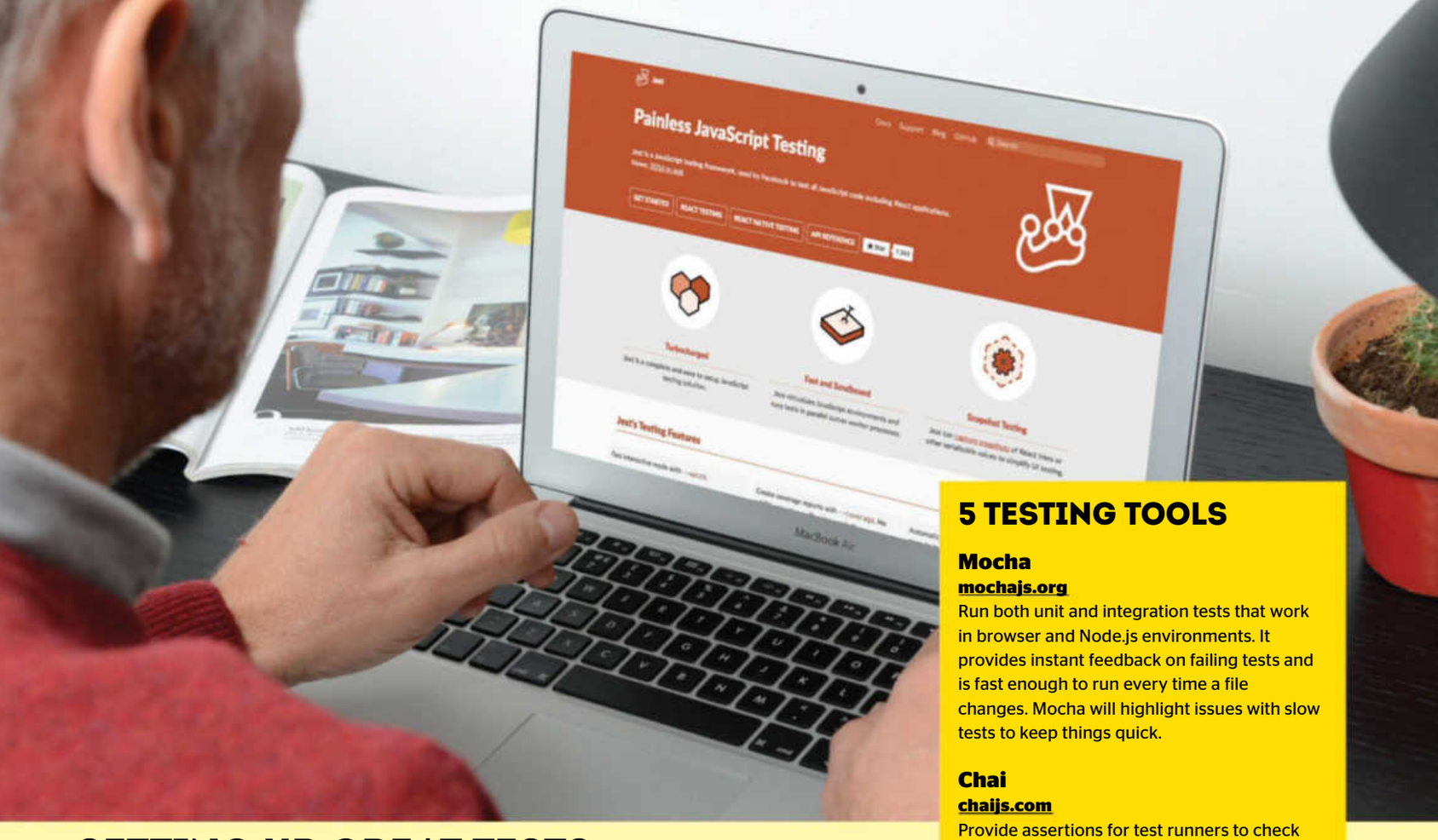
**codacy.com**

Codacy is a complete tool for project management with a focus on code quality. Analyse each pull request for performance and security issues. Each file is given an overall grade that can be tracked over time. It's free for open source projects.

### Debug

**github.com/visionmedia/debug**

Debug is a package that can be included in any Node.js project. Similar to console logging, calling debug() in modules where it's defined will colour-code the logs to make them easier to identify on the command line.




# SETTING UP GREAT TESTS

*HAVE CONFIDENCE YOUR CODE KEEPS WORKING AS EXPECTED*

**Code can break in unexpected ways.** Even if it gets double-checked, there are always edge cases that can cause unpredictable behaviour or, in worst cases, stop working altogether. If there's just one unhandled error in JavaScript, that can knock out an entire application.

Fortunately, there are plenty of methods to automate testing to make sure that doesn't happen. Each tests a different aspect of the code to make sure all parts do what they should. While there is a lot to proficient testing, there are three key stages at which to check code to be confident with any changes.

Unit tests are a way of making sure each part of a project works independently. When changes are made to one section, tests can automatically run to make sure everything continues to work as expected. They should be basic enough to run quickly and be plentiful to cover all situations. As more unit tests are written and more code is covered by a test case, less manual testing is needed. While they take time to set up, time is saved overall by not hunting down bugs manually.

There are two ways to approach unit tests. TDD – test-driven development – makes sure that tests define the functionality of a particular unit. By writing tests before code and making sure that they pass at the end of it, you know the code is doing the right thing. BDD – behaviour-driven development – takes that approach but focuses more on what the result is rather than how it got there.

Each unit test should focus on one component at a time. If other factors come into play, such as network requests or using browser APIs, they shouldn't form part of that test. Where their behaviour is necessary, JavaScript is flexible enough to allow dummy objects to emulate their behaviour. These are known as 'test doubles' and are available as part of other testing packages.

While unit tests check individual parts of a site work correctly, there is no guarantee they play nicely together. Integration tests take that role by checking, for example, a submitted form saves correctly server-side. These would typically run once all individual units are built and passing.

Finally, functional tests check that everything works correctly from a user's point of view. These higher-level tests make sure a visitor can perform actions such as using the sign-up form without issue. Commands that simulate text entry and button clicks can run in browsers to make sure those that are supported work as they should.

### TEST EARLY, TEST OFTEN

*While it's tempting to get on with writing code straight away, setting up unit tests beforehand ensures that code will behave as intended.*

## 5 TESTING TOOLS

### Mocha

**mochajs.org**

Run both unit and integration tests that work in browser and Node.js environments. It provides instant feedback on failing tests and is fast enough to run every time a file changes. Mocha will highlight issues with slow tests to keep things quick.

### Chai

**chaijs.com**

Provide assertions for test runners to check for correct behaviour. Mocha itself does not come with an assertion library, so tools like Chai are required to compare actual results with expected ones. It provides interfaces to write tests in both TDD and BDD styles.

### Sinon

**sinonjs.org**

Create doubles to simplify unit tests. Sinon generates spies to see how code uses function calls and stubs to shorten complex external APIs. By using these, tests can focus on how that unit behaves with no uncontrolled external influences.

### Jest

**facebook.github.io/jest**

Jest is Facebook's flavour of test framework. While flexible enough to work with all JavaScript projects, Jest is commonly used for React applications. It includes useful features such as snapshot testing, which compares the output of a component to a previous version.

### Nightwatch

**nightwatchjs.org**

Write functional tests in Node.js that run through Selenium, which can perform browser actions based on commands. A setup file defines which tests to run against which browsers. Tests are then written in a human-readable language and can be as simple or complex as necessary.

# FOR THE GNU GENERATION

## www.linuxuser.co.uk


EVERY ISSUE • FREE RESOURCE DOWNLOAD

LinuxUser & Developer

Available from all good newsagents & supermarkets today

## ON SALE NOW:

**Lock down your system » Power up your Pi » Best browser revealed**






# BUY YOUR ISSUE TODAY

Print edition available at www.imagineshop.co.uk

Digital edition available at www.greatdigitalmags.com

Available on the following platforms

Print Tablet Phone Desktop

facebook.com/LinuxUserUK twitter.com/LinuxUserMag

# WebGL

## TO INFINITY AND BEYOND

**Carl Bateman outlines the current state of WebGL, what's happening with the upcoming WebGL 2.0 spec and the best tools and tips to get building**

As you really should know by now, with universal support since way back in early 2014 (practically the age of dinosaurs in web development terms), WebGL is the incredibly powerful plug-in-free, native JavaScript API that gives the developer access to fast, in-browser 2D and 3D rendering, and general-purpose programming on the GPU (GPGPU).

Use cases abound: games, experiences, apps, webpage header, data visualisation, product visualisation, music videos and much more.

The power of WebGL comes from shaders, small(-ish) programs written in GLSL (a C-like language) which are compiled on the user's computer, then run on their GPU. Since shaders are compiled and not interpreted, they are as fast as a native app. Sadly, WebGL still relies on JavaScript for setup, control, passing data, user interaction, animation and, surprisingly, some matrix maths – this rather puts a brake on things.

To get some idea of the speed improvement possible, compare the two fractal rendering sites:

**Carl Bateman**
**Founder of WebGL Workshop**
webglworkshop.com

> "The future of WebGL is looking look good, with the growing popularity of libraries like three.js and Babylon.js, along with the game engine PlayCanvas, which are regularly releasing new versions with ever more features. The next iteration of WebGL (WebGL 2.0) is imminent and will add many new powerful core features"

**liamness.co.uk/fractal-viewer** (WebGL) and **ondras.github.io/fractal** (Web Workers and JavaScript only). Even without Web Workers, the WebGL-based page is fast enough for real-time zoom of the fractal.

WebGL may be complex at the lowest level, but APIs such as three.js, Babylon.js and their ilk make development several orders of magnitude easier and continue to lower the barrier to entry, while libraries like X3DOM make using WebGL as simple as adding markup to your page.

Shaders are low-level and require a reasonable understanding of maths, particularly vectors, matrices and geometry. Also, vanilla WebGL requires a lot of code (much of it repetitious) to achieve anything. Using boilerplate can reduce this repetition while keeping close to the underlying WebGL. Using a library or API adds layers of abstraction to simplify coding, avoids some of the maths and provides useful features, such as objects, meshes, lights and scene graphs.

Generally, WebGL workflows tend to follow the same path: asset generation, then code to render the assets and provide user interaction.

Depending on the complexity of a project's requirements, assets can be generated offline in third-party editors such as Blender, wholly generated in code in browser, or a hybrid approach can be taken with simplified assets generated offline which are then 'tweaked' in code. Programming can be in vanilla JavaScript or can use one of the many available APIs.

*This animation shows what can be done with shaders, albeit slowly. Courtesy shadertoy.com*

## WORKFLOW: WHICH IS BEST?

The workflow adopted depends on the complexity, requirements and time constraints of the project.

### VANILLA WEBGL VS LIBRARY

**PROS**

- With no bulky library file to download, vanilla WebGL will be smaller with little, if any, superfluous code while being faster.
- Vanilla WebGL is not tied to a library's way of doing things, making it easier to tweak code to improve performance.

**CONS**

- The developer must have a good understanding of geometry, vectors and matrices as well as the intricacies of GLSL.
- Everything is set up 'by hand' so can be time-consuming and fiddly with a steep learning curve and hard to maintain.

**PROS**

- Development is easier and faster thanks to complexity being hidden and common features (vectors, matrices, lights, cameras, etc) being available 'out-of-the-box'.

**CONS**

- The developer is locked in to the way the library developer does things; it can be impossible to customise the library.
- Libraries can be big with yet more syntax to be learned and many superfluous features that are never needed.

### OBJECT MODELLING VS PROCEDURAL GENERATION

**PROS**

- Particularly for complex objects, modelling an object will be much easier and be far more likely to meet requirements.
- The workload is shared between the modeller and the programmer; additionally, any required changes can be made in parallel to development.

**CONS**

- Complex objects, in particular, are likely to generate very large sizes and adversely affect download speed.
- Depending on the format, models exported can lose attributes like lighting or textures when porting to WebGL.

**PROS**

- Since objects are generated in code, there is no associated download. Generating in code will almost be certainly be much quicker than downloading.
- Changes to the model and its attributes can be easily and quickly made by adjusting a few parameters.

**CONS**

- All but the simplest objects will be difficult to generate; the code required may be complex and difficult to maintain.
- It can require a somewhat rare combination of skills, as the programmer is also working as the modeller.

## WHAT CAN WEBGL 2.0 DO FOR YOU?

WebGL 2.0 (based on OpenGL ES 3.0 and 3.1 for embedded systems) will be the next iteration of WebGL. Although well defined, WebGL 2.0 is still very much in the pipeline and highly experimental. When it does arrive, only programmers developing vanilla or boilerplate WebGL will see any significant changes, but others will still feel the benefits as it's adopted by libraries and APIs.

Due to security concerns and other issues, not all OpenGL ES features will be exposed. So, while compute shaders (which allow developers to perform GPGPU) may be available as an extension in some web browsers, sadly there will be no geometry shaders (which would have allowed the creation of geometry on the GPU rather than passing it).

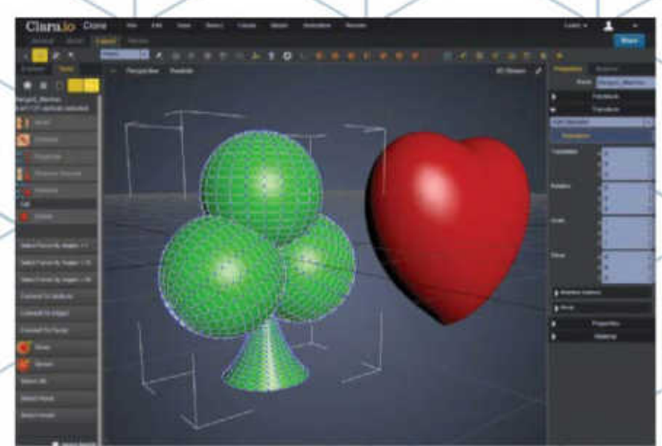
WebGL 2.0 has over 26 enhancements including 3D textures, matrix functions, the removal of loop restrictions, direct texel lookup, compressed texture support and geometry instancing – features which will give developers access to more powerful and advanced graphics capabilities. Some example applications: 3D textures simplify volumetric rendering for medical imaging; geometry instancing allows geometry to be reused, reducing calls to the GPU; compressed textures also reduce GPU calls; and matrix functions remove the need to perform matrix manipulation within JavaScript, markedly improving performance.

WebGL 2.0 examples are still few and far between with no examples in production.

To view the following demos, your browser will need to be set up appropriately. Take a visit to the How to use WebGL2 site, which offers more information (**bit.ly/2iPvGq7**).

## 4 TOOLS & RESOURCES TO GET STARTED

Here are just some of the many tools and resources available




### 1. Clara.io

If you're doing 3D, you need 3D assets – even if it's only as placeholders, it's slightly more interesting than a box while waiting for something production-ready to be delivered.

Clara.io is a WebGL-based online 3D modelling tool with a lot of features, hosting a range of free 3D models ready for use, and makes an excellent starting point for acquiring 3D assets.

The interface is simpler and more stable than Blender; existing models can be imported and edited, or new models can be built up from primitives.

It can import a wide range of file types from .3d to .zgl, while export formats including .babylon, glTF and three.js (json).

*Brandon Jones's WebGL 2.0 particle demo*

### WebGL 2.0 Samples
**webglsamples.org/WebGL2Samples**
A repository for about 30 WebGL 2.0 samples

### WebGL 2.0 Particles
**toji.github.io/webgl2-particles**
A 1 million particle demo using transform feedback

### WebGL 2.0 Crowds
**toji.github.io/webgl2-crowd**
Animation 'baking' using transform feedback

## WEBGL AND THE BROWSER

Adding WebGL to a webpage is as simple as adding a <canvas> element (which is where WebGL renders to). The three.js JavaScript library will even do it for you if you wish.

Since <canvas> is a standard HTML element, WebGL elements can be freely mixed with other HTML elements and have a lot of flexibility with styling. Teapot Test (**bit.ly/2jEnqt5**) is a simple yet excellent example displaying a teapot against background text.

Safari in Safari (**bit.ly/2jEhJey**) provides an introduction to 3D content on the web with the help of the Safari compass. It demonstrates using CSS styling to resize on mouse-over.

Making life even easier, SketchFab facilitates the embedding of 3D models into Facebook, while the glTF WordPress Plugin (**bit.ly/2ikpytW**) allows the embedding of 3D models into WordPress. Check out the examples on the website.

## 2. Performance

Unlike OpenGL, there are no performance analysis tools specifically for WebGL. Unfortunately, this leaves the developer to find bottlenecks with a mixed bag of tools.

The general performance of a browser can be checked using tools such as OORTONLINE (**oortonline.gl**) or WebGL Performance Benchmark (**luic.github.io/WebGL-Performance-Benchmark**).

Render performance can be monitored with Stas.js (**github.com/mrdoob/stats.js**).

And finally there's the nuclear option – chrome://tracing. Chrome's inline profiling subsystem provides more info than you could be possibly want about performance, recording all of Chrome's activities across every thread, tab and process.

## 3. WebGL usage and browser support

It is very useful to be able to quickly review WebGL usage and a browser's WebGL feature support, the following sites let you do just that.

Can I use (**caniuse.com/#search=webgl**) lists (with history) browser support for versions of WebGL.

Browser Leaks (**browserleaks.com/webgl**) fully details the WebGL feature support including extensions of the visiting browser.

WebGL Report (**webglreport.com**) is similar to Browser Leaks but covers both versions of WebGL and is presented in more concise format.

The website WebGL Stats (**webglstats.com**) has undergone a bit of a rework recently. It provides a CDN link to monitor WebGL usage stats as well as detailing.

## 4. Shader inspectors / editors

When debugging, a shader inspector is invaluable for identifying the cause of problems. They can also be a great educational tool for learning by examining existing shaders. A shader editor enables tweaks to be made in real-time.

While Firefox provides its own built-in shader editor, alas Chrome does not. The venerable WebGL Inspector (**benvanik.github.io/WebGL-Inspector**) hasn't been updated since 2012, but is still one of the better editors available. This advanced WebGL debugging toolkit provides much needed functionality: Trace, Timeline, State, etc.

A major shortcoming of WebGL-Inspector is its lack of an editor. The newer Shader Editor allows this, but since the two tools interfere with each other, only one can be used at a time.

# WEBGL

## ESSENTIAL QUICK HINTS

### Huge assets

Textures and 3D models, in particular, can be huge, leaving the user staring a blank screen or loading icon for some time.

Where possible, use a library that supports glTF, and streaming or progressive loading.

glTF will minimise the file size, whereas streaming will allow assets to be partially rendered as they are received.

Alternatively, render models first and textures later, load models in sections, or use simplified models as placeholders while waiting for the more detailed model to load.

### Fallback

Some WebGL features are browser specific, so if certain capabilities or extended features are critical, it is important to check for them (with getParameter and getSupportedExtensions) and provide a fallback if they're not working.

If WebGL is likely to fail, three.js is a good option as it provides CanvasRenderer as an alternative.

In the unlikely event of no support, provide an image (static or animated) or at least a warning.

WebGL support is currently fairly consistent; with WebGL 2.0 imminent, this will become more of a problem.

### Textures

When using textures (images), ensure each side is a power of two (ie one of 1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, or 2048 pixels) even if this means changing the aspect ratio. While WebGL has limited support for non powers-of-two textures, some functionality (ie mipmapping and repeating) will not work with them and will be rendered as black. Frequently altered images can be resized using HTML5 canvas, while finalised images can be resized static offline for simplicity and speed.

### Learning WebGL

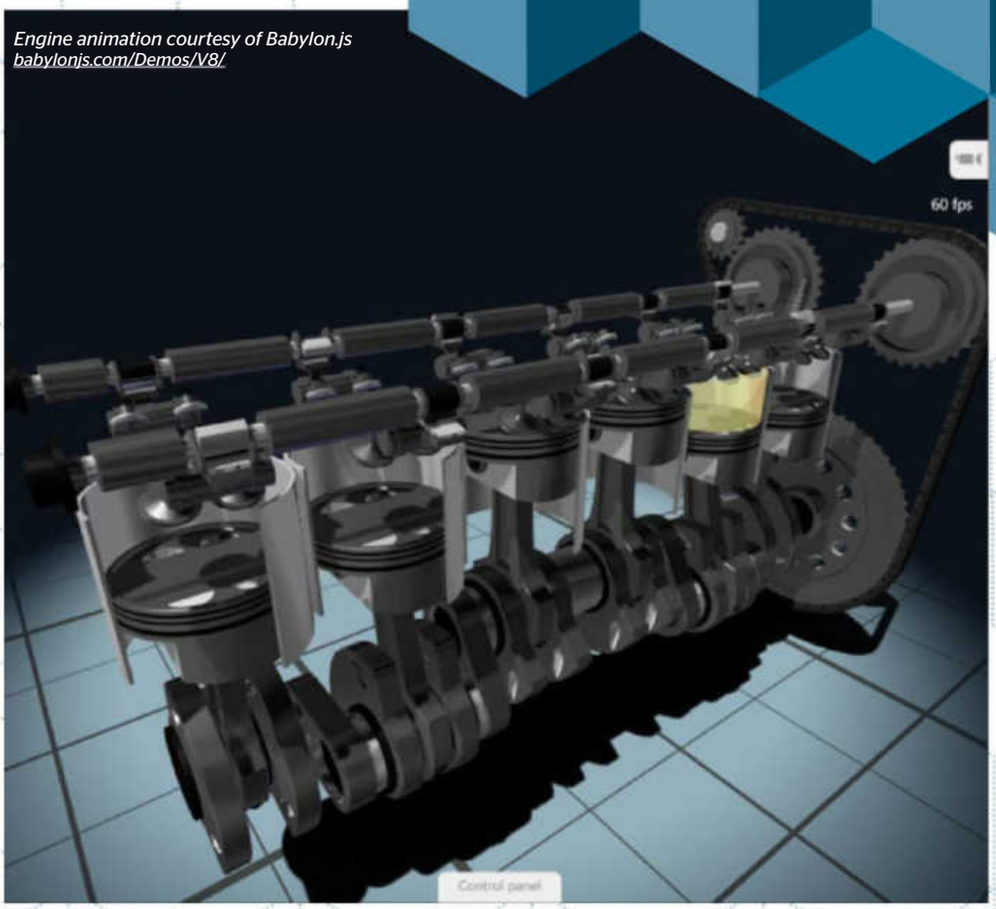
There are many single-page introductions to WebGL. Here are a few sites dedicated to helping you to learn WebGL...

The eponymous Learning WebGL (**learningwebgl.com/blog/?page_id=1217**) and its associated WebGL Cookbook wiki (**bit.ly/2isbTPV**).

WebGL and WebGL 2.0 Fundamentals at **webglfundamentals.org** and **webgl2fundamentals.org** respectively.

Oh, and of course there's the WebGL Workshop (**www.meetup.com/WebGL-Workshop-London**), which appears to have the only free, hands-on, face-to-face lessons in the UK.

Three.js enjoys a dedicated Learning Three.js site at **learningthreejs.com**. While Microsoft Virtual Academy hosts Babylon.js video course at **bit.ly/1YvAu2A**.




*Engine animation courtesy of Babylon.js babylonjs.com/Demos/V8/*

## BUILDING FOR THE WEB

Here, we create a simple scene with Babylon.js, then take a look at writing our own custom shader.

First, create the Babylon engine using a canvas.

```
var canvas = document.
getElementById('renderCanvas');
var engine = new BABYLON.Engine(canvas,
true);
```

Then create a new scene to hold objects.

```
var scene = new BABYLON.Scene(engine);
```

Add a camera so that we can see things, pointing it at the origin of the scene (0, 0, 0). The camera provides some basic control to the user, allowing panning and zooming of the scene.

```
var camera = new BABYLON.ArcRotateCamera("Ar
cRotateCamera", 0, 0, 0, new BABYLON.
Vector3(20, 10, 0), scene);
camera.setTarget(BABYLON.Vector3.Zero());
camera.attachControl(canvas, false);
```

Add a light to the scene so that everything's not dark and the objects are lit.

```
var light = new BABYLON.
HemisphericLight('light1', new BABYLON.
Vector3(0, 1, 0),
scene);
```

Add a sphere and some ground for it to sit on.

```
sphere = BABYLON.Mesh.CreateSphere
('sphere', 16, 2, scene);
sphere.position.y = 1;
var ground = BABYLON.Mesh.
CreateGround('ground1', 10, 10, 2, scene);
```

Finally, add a render loop to ensure the scene is updated correctly.

```
engine.runRenderLoop(function ()
{
scene.render();
});
```

WebGL has two sides. In the browser and on the CPU there is JavaScript, used for general control, user interaction, passing data and some basic matrix manipulation that is otherwise missing. The power of WebGL comes from its shaders, running on the GPU and written in GLSL, a strongly typed C-like language focused on geometry.

Much can be achieved with little to no understanding of the underlying maths used in modelling and rendering, particularly when using a library. However, applying a little mathematical muscle can yield intriguing results.

Libraries usually provide a convenient way to use your own shaders. Babylon is no exception, having the ShaderMaterial.

The process of adding your own shader may appear daunting, but is quite straightforward if perhaps lengthy.

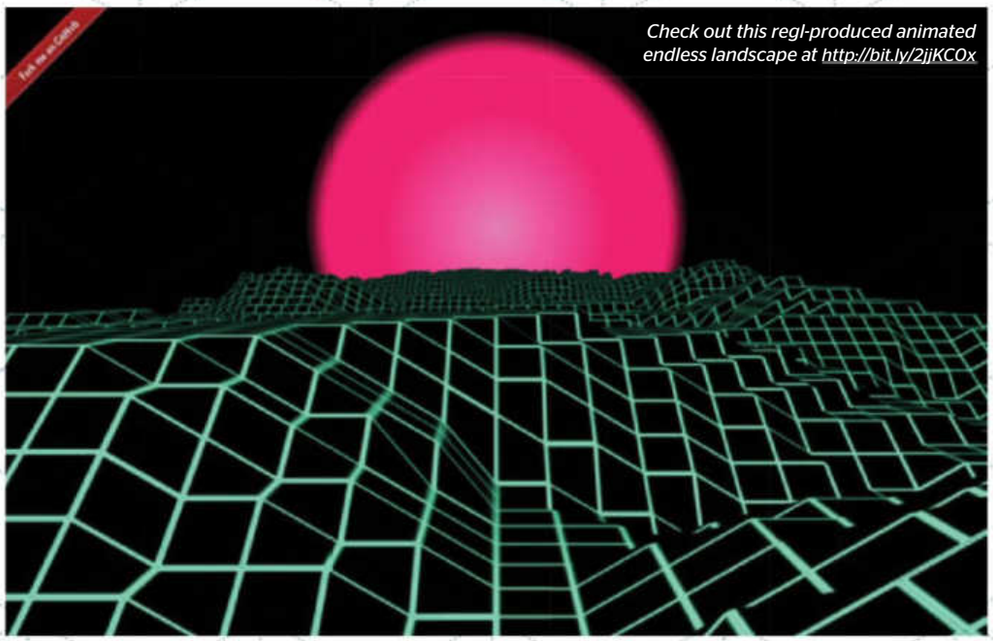
Add the vertex and fragment shader as scripts. This is a requirement of Babylon and a convenient place to put them. Scripts can also be placed in external files.

Note: These shaders are taken from Jos Dirksen's learning three.js GitHub repository (**bit.ly/2jre2tk**).

Sadly, shaders can frequently quickly become complex when attempting to get something interesting on screen.

Head to FileSilo (**www.filesilo.co.uk**) for this issue to get the rest of the code for this section.

*The updated version of Tanx from PlayCanvas*
*tanx.io*



*Check out this regl-produced animated endless landscape at http://bit.ly/2jjKCOx*

## 3 BEST TOOLS

### Babylon.js

**babylonjs.com**
**github.com/BabylonJS/Babylon.js**

While primarily a fully featured open source game library, Babylon.js is easily as capable as three.js and flexible enough for pretty much any 3D project. The original TypeScript is available on GitHub, or customised JavaScript builds are available via its online generator. It can be packaged with one or both of the oimo.js or cannon.js physics engines.

### PlayCanvas

**playcanvas.com**
**github.com/playcanvas**

PlayCanvas is a Unity-like online editor with hosting and comes in paid and free versions. The game engine is available as open source.

The editor makes it very easy work to with WebGL, reducing the process to drag and drop. The editor is intended for whole-page web apps/games and it's not possible to embed projects as part of a webpage.

### regl

**github.com/regl-project/regl**
**regl.party**

Regl is a relatively new kid on the block and takes a modular, functional approach to WebGL. It also features a very JSON-like syntax. While regl abstracts the underlying WebGL context, its more minimal API provides the developer with a lower-level approach than three.js or Babylon.js (and of course this means there's more work for the developer to do).

## ESSENTIAL QUICK TIPS

### Performance desktop vs mobile

Vendors implement WebGL differently on mobile and desktop. So, apparently correct code may render differently over platforms or, worse, cause a device to run hot and drain its battery.

For instance, Windows devices use ANGLE to translate WebGL shaders to HLSL. Consequently, valid code may appear buggy due to the differences between HLSL and GLSL. For instance, HLSL only supports line widths of 1.

Therefore, it's important to write conformant code and test on multiple physical platforms to ensure that code is rendering correctly.

### Beware SIMD in DIY shaders

When writing shaders, bear in mind that GLSL uses the Single Instruction, Multiple Data (SIMD) model: a single instruction is performed on multiple data simultaneously. Hence, a branching 'if' statement will cause the shader to be rerun for each state, potentially doubling the execution time with every 'if'. Avoid using 'if' by using GLSL functions – mix, max, blend, ceil, etc.

For instance,

```
if(length(position) < uRadius)
{
gl_Position = normalize(position) * uradius;
} else
{
gl_Position = position;
}
```

becomes

```
gl_Position = min(length(position), uRadius)
* normalize(position);
```

### WebGL error messages

Even when using a library, it's important to learn (or know where to look up and have a general idea) what the various WebGL error messages mean (debugging can be a pain).

When things go wrong (and they always do), a good understanding of error messages helps to locate where they are likely to be.

Where possible, make sure full error reporting is enabled and ensure that the webpage loads and runs with no WebGL errors.

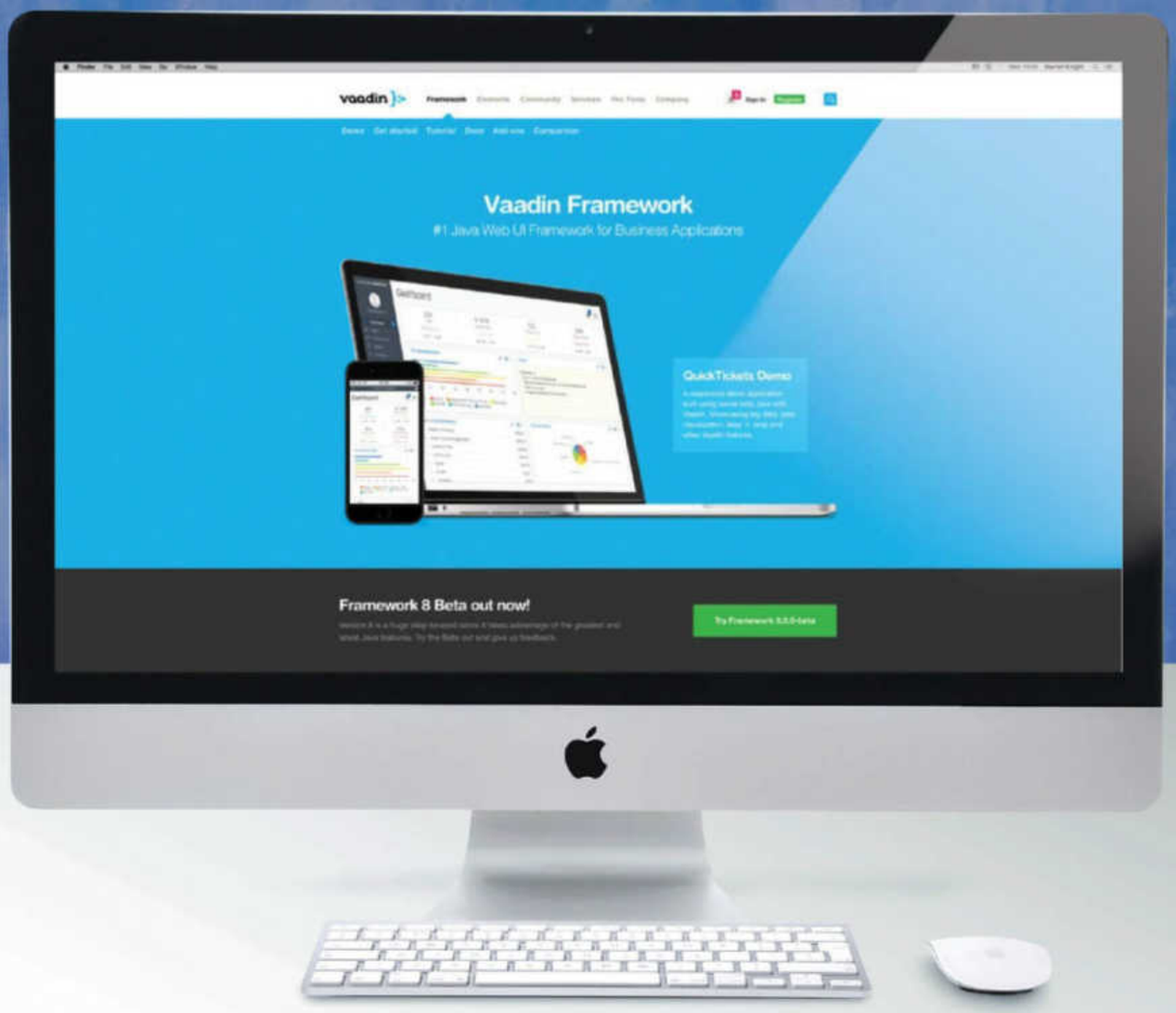
### Beware the CDN

Be wary of using CDN with some libraries (such as three.js). You may be keeping bang up-to-date, but sometimes changes can break functionality and it can be time-consuming to fix. It's better to download a version and develop against that.

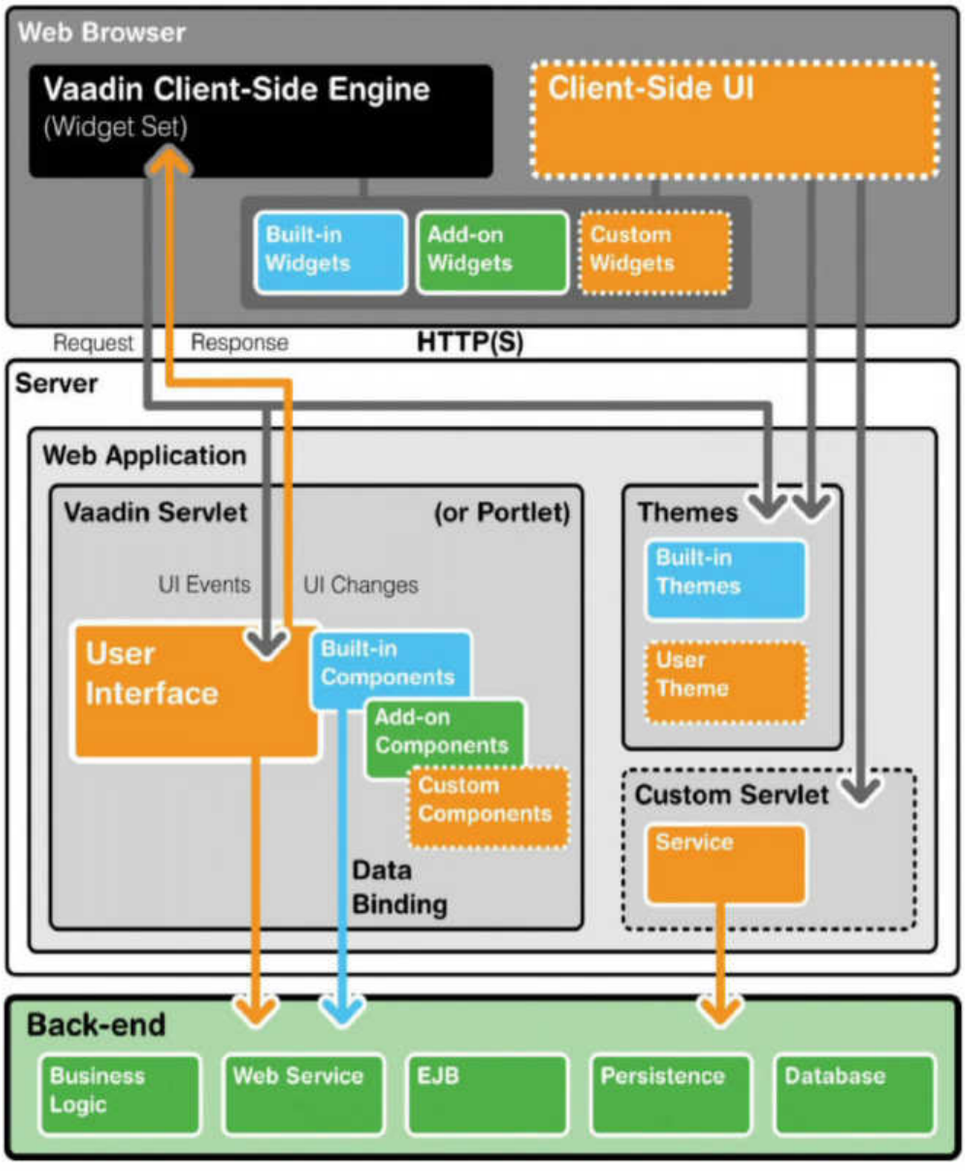
If time and inclination permit, come deployment time, get a later version and test against that and use it if everything works, but always keep a copy as a fallback against future API breakage.

# Build web apps with the Vaadin framework

Reuse Java IP in web applications with Vaadin, a combination of a JavaScript client with a Java backend




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




**Vaadin's architecture looks complex at first glance: the best way to think about it is that it involves a two-tier system.** The user's browser runs a traditional AJAX client, which is connected to a backend running on the developer's server – the similarities to VNC and the X Window System protocol found in various Unix workstations are intended.

Developers who own a significant amount of Java code or who have a strong dislike for JavaScript can profit greatly from this situation. The server is a Java applet which gets run inside an application server such as Apache Tomcat – when done right, engine code feels no difference to a traditional execution environment.

Vaadin's power lies in the high-performance GUI stack. As shown in the architecture overview on this page, developers have access to an expansive selection of widgets backed by a powerful data binding engine which makes connecting controls and data sources a breeze.

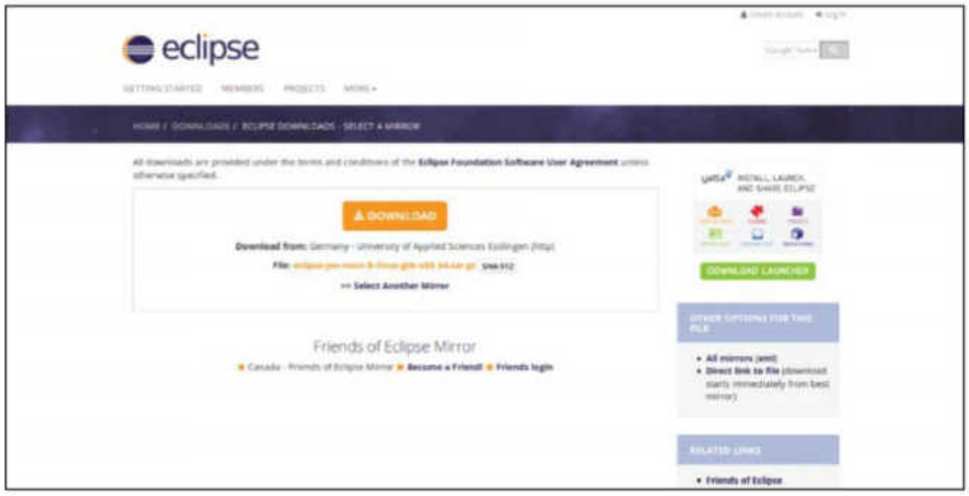
In short: friends of the Java programming language will find Vaadin their home in the depths of the mania of web technologies. Other developers should still keep the existence of the product in the back of their mind – if a large amount of Java IP pops up, knowing Vaadin's name can be a godsend!

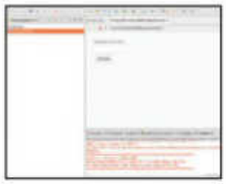
## 1. Download Eclipse

Around 50 per cent of Java developers love Eclipse. Vaadin's developer team is a member of the 'We Love Eclipse' camp, which is why Eclipse EE must be downloaded from **eclipse.org/downloads/packages/eclipse-ide-java-ee-developers/neonr**. Proceed to extracting the file to a pleasant location. The following steps assume Ubuntu 14.04LTS.

### Design by hand

Creating user interfaces in code can get tedious quickly. Vaadin Designer is a paid-for solution to this problem: its layout files can be loaded via a parser at runtime.



**Left**
Eclipse EE is offered in a wide variety of builds – pick one which suits your operating system

**Top left**
The Vaadin plugin can be deployed via the Add-on system found in Eclipse EE

**Top right**
Vaadin projects start out as a combination of button and label: running the code yields a web browser window inside Eclipse

## 2. Deploy the Vaadin plugin

Open Eclipse and click Help>Add New Software to open the Add-in installation wizard. Enter the URL **vaadin.com/eclipse** and select Vaadin. Finally, proceed through the rest of the wizard to kick off the deployment process, which can take some time, requires an internet connection and throws unneeded security errors.

## 3. Create a new project

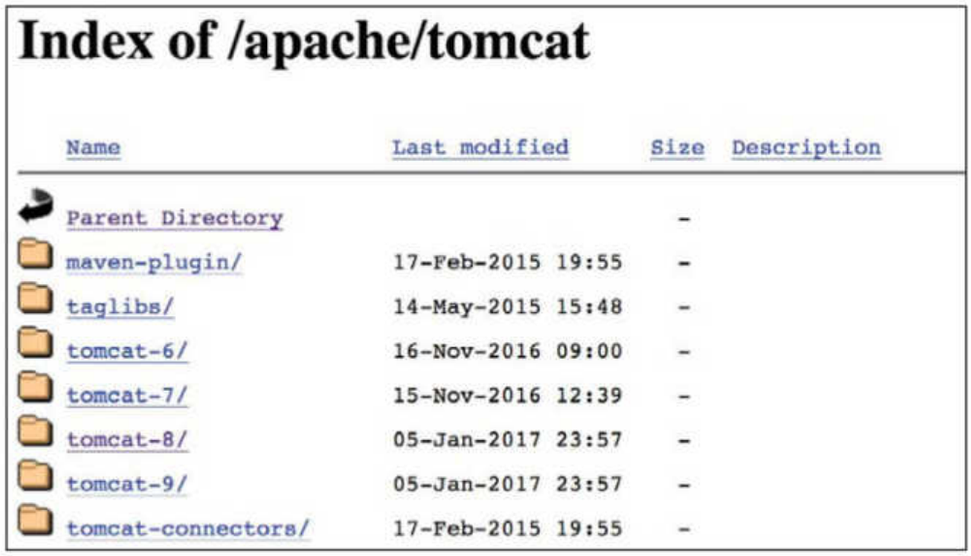
Click File>New>Project, and select the project template Vaadin>Vaadin 7 Project (Maven). Skip the Archetype selection by clicking Next and click Finish to start the Maven deployment process, which will download dependencies. Subsequent invocations of the New Project wizard will offer various archetypes.

## 4. Download an application server

Vaadin applications cannot live outside of an application server. Apache's Tomcat 8.0.x is by far the most popular choice – you can download it by visiting **tux.rainside.sk/apache/tomcat/tomcat-8/v8.0.39/bin/apache-tomcat-8.0.39.zip** and extracting the contents of the folder to a convenient location.

## 5. Introduce Eclipse EE

Open the Servers tab and click the link to open the server configuration wizard. Select the Apache Tomcat 8.0 template and point the wizard to the path where you extracted the contents of the Tomcat archive. Finally, click Finish to complete the configuration – nothing needs to be uploaded at this point.




## 6. Analyse our code

Open MyUI.java to reveal the core of our application: it implements the UI object, which represents a user interface shown in a browser window. The init() method is responsible for creating the GUI structure in a process not dissimilar to Android applications.

```
public class MyUI extends UI {
@Override
protected void init(VaadinRequest
vaadinRequest) {
final VerticalLayout layout = new
VerticalLayout();
final TextField name = new TextField();
name.setCaption("Type your name here:");
Button button = new Button("Click Me");
button.addClickListener( e -> {. . .});
```

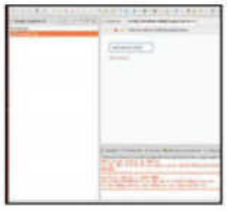
## 7. Add more code

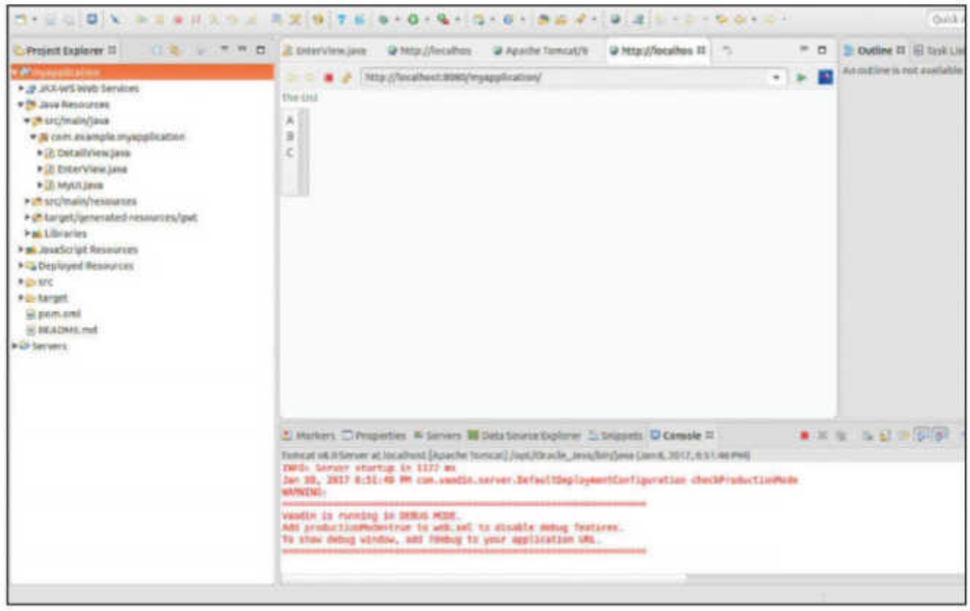
The rest of the init function is responsible for adding the widgets to the container. With that function now out of the way, the Servlet declaration acts as the actual connection between incoming HTTP requests and the user interface implementations.

```
layout.addComponents(name, button);
layout.setMargin(true);
layout.setSpacing(true);
setContent(layout);
}
@WebServlet(urlPatterns = "/*", name =
"MyUIServlet", asyncSupported = true)
```

### Don't go code-crazy!

Even though coding custom data classes is fun: in many, if not most cases, developers can make do with the implementations that are provided as part of the framework.

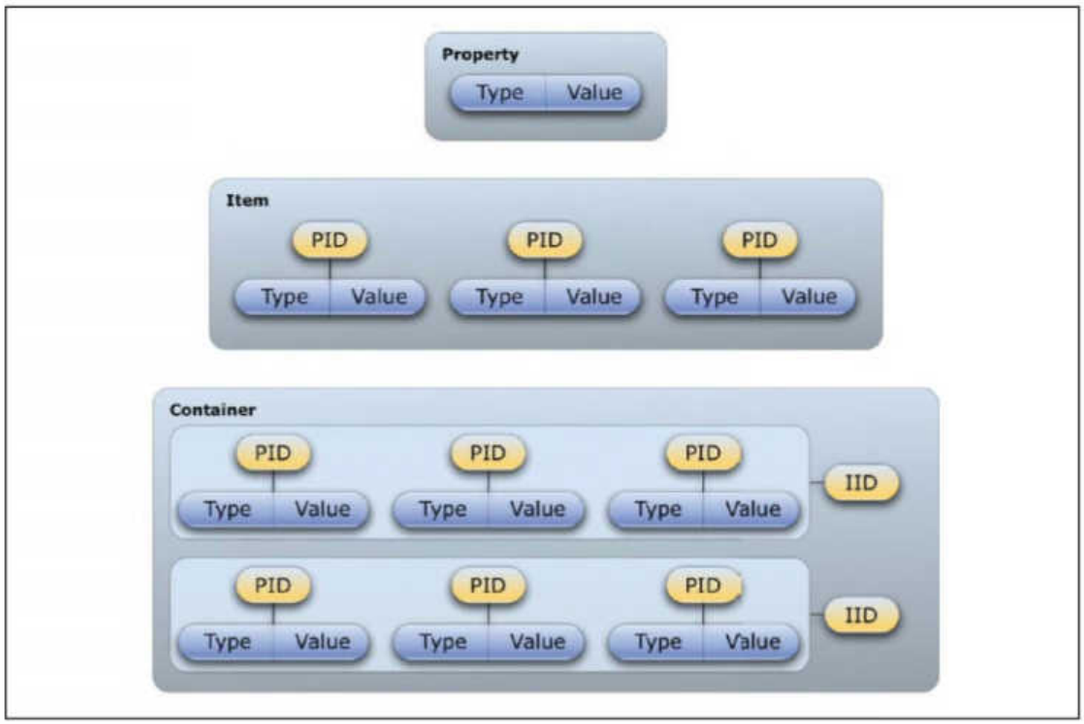




**Left**
The label's content is not updated immediately due to various delays in the Vaadin backend

**Top left**
Apache Tomcat can also be accessed from a browser running on the local machine, thereby allowing you to bypass Eclipse's weird browser window

**Top right**
Work on the list is complete: the contents of the backing store are copied automagically




**The spreadsheet of Vaadin**

Vaadin's data-binding model is best understood by following a spreadsheet-like concept. Containers should be thought of as spreadsheets, which are made up of a series of items (aka rows). Each of the rows then contains one or more fields carrying the actual payload – they are modelled as property implementations. Vaadin differs from spreadsheets in that each item can have its own set of properties if this is supported by backing store and data consumer. Keep in mind that all of the elements shown in the figure are but interfaces: in theory, a completely custom data binding layer can be created. In practice, developers will usually strive to avoid this tedious bit of work by using ready-made implementations provided as part of Vaadin.

```
@VaadinServletConfiguration(ui = MyUI.class,
productionMode = false)
public static class MyUIServlet extends
VaadinServlet { }
```

## 8. Run the example

Execute our program by right-clicking it in the Project Explorer. Next, select Run As>Run on Server to open the server selection dialog. Pick the Apache Tomcat issue we configured above and click finish. When done, Eclipse's web browser will open along with a view of the form created by our example.

## 9. Go single-window

Vaadin allows developers to add and remove widgets to forms at runtime. The little snippet of code accompanying this step demonstrates a primitive implementation – a more involved version would use dedicated instances of the View class.

```
@Override
protected void init(VaadinRequest
vaadinRequest) {
final VerticalLayout layout = new
VerticalLayout();
Button button = new Button("Stage two");
button.addClickListener( e -> {
layout.removeAllComponents();
Label myLabel=new Label("Stage two welcomes
you!");
layout.addComponent(myLabel);
});
layout.addComponent(button);
```

## 10. Bind some data

Vaadin 7's data structure is explained in detail in the boxout above. For now, start by creating a member which is able to host textual information that can be bound to other user interface components.

```
public class MyUI extends UI {
ObjectProperty<String> myDS;
@Override
protected void init(VaadinRequest
vaadinRequest) {
myDS=new ObjectProperty<String>("Hello
World");
```

## 11. Create the relationship

The second step to Vaadin goodness involves setting up a binding relationship between a control and the information found in the backing store. Our example binds an editable and a non-editable text field together for fun and profit.

```
@Override
protected void init(VaadinRequest
vaadinRequest) {  myDS=. . .
TextField myField=new TextField();
myField.setImmediate(true);
myField.setPropertyDataSource(myDS);
layout.addComponent(myField);
Label myLabel=new Label();
myLabel.setPropertyDataSource(myDS);
layout.addComponent(myLabel);
```

## 12. Beware of delayed updating

Run the program once again in accordance to the steps laid out above. Then, proceed to clicking the editable text box. Changes are not propagated immediately in all cases – some controls wait until the user stops interacting with them to save CPU time and transfer volume.

## 13. Navigate around town

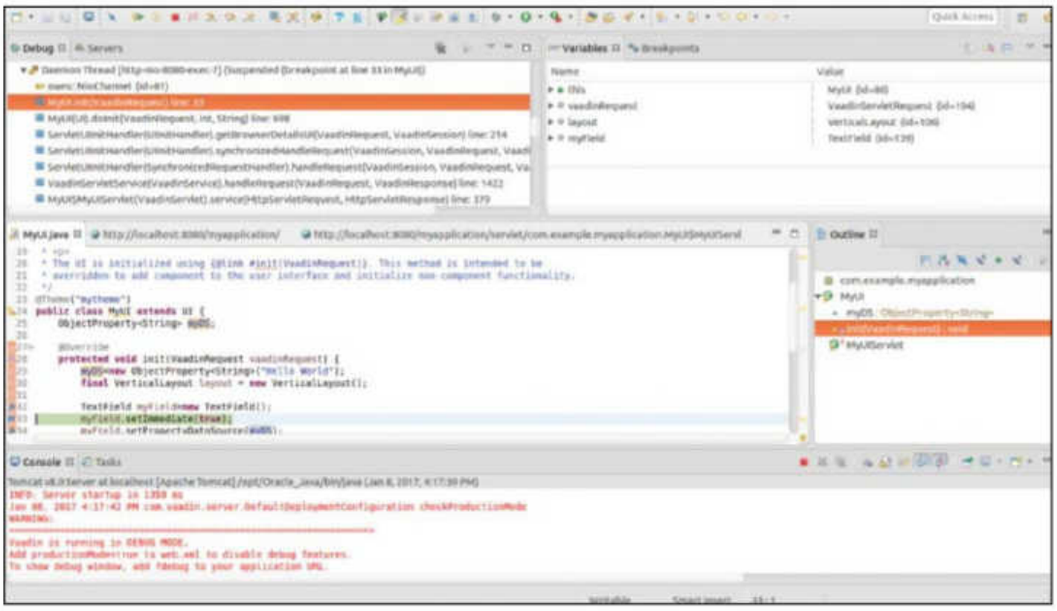
Dynamic user interfaces can be counter-intuitive: in some cases, users will expect to be able to bookmark internal pages. This kind of behaviour is best realised via the Navigator component, and this must of course be added to the application.

```
public class MyUI extends UI {
Navigator myNavigator;
@Override
protected void init(VaadinRequest
vaadinRequest) {
final VerticalLayout layout = new
VerticalLayout();
myNavigator = new Navigator(this, this);
myNavigator.addView("", new EnterView());
myNavigator.addView("detail", new
DetailView()); }
```

## 14. Add some views

Adding the navigator addresses is but part of the problem. We must now create the implementations for both EnterView and DetailView. Start out by creating a new empty Java class, which must be derived from the framework class View. This leads to the skeleton shown in the code accompanying this step.

```
public class EnterView extends VerticalLayout
implements View {
@Override
```



### Understand the power of Eclipse's debugger

Click the white bar to the left of a line to add a breakpoint: when program execution passes by this mark, Eclipse will halt and switch into the debugger perspective. In this mode, the window on the top right-hand side of the screen provides further information about any variables which are active in the current stack frame. Furthermore, the arrow buttons allow you to step over single lines of code to watch changes in program state.

When debugging is complete, click Stop to stop program execution. Afterwards, click the perspective changer tool in the top-right corner of the screen to leave the debugging perspective. Note that this is not well-suited to normal program development process.

```
public void enter(ViewChangeEvent event) {
// TODO Auto-generated method stub
} }
```

## 15. Create a UI

As our view is derived from a layout, adding content to it is not particularly difficult. Simply create the user interface in response to an invocation of the enter() method – the GUI widgets will behave just as if they were hosted directly inside of UI.

```
@Override
public void enter(ViewChangeEvent event) {
Label myView=new Label("This is the
EnterView!");
addComponent(myView); }
```

## 16. Switch between views

With that, it's now time to add another class implementing the DetailsView. Its code is quite similar to the one of the EnterView created before. Finally, open the program and navigate to the URLs **localhost:8080/myapplication/** and **localhost:8080/myapplication/#!detail** in an external browser to see the navigator's view switching in action.

```
public class DetailView extends
VerticalLayout implements View {
@Override
public void enter(ViewChangeEvent event) {
Label myView=new Label("This is the
DetailView!");
addComponent(myView);
}
}
```

## 17. Switch views programmatically

One interesting approach involves changing the currently-active view at runtime. This is accomplished via the navigateTo method – change enterView as shown in the code accompanying this step to be able to change the view by clicking a button.

```
public void enter(ViewChangeEvent event) {
Button button = new Button("Stage two");
button.addClickListener( e -> {
getUI().getNavigator().navigateTo("detail");
});
addComponent(button);
}
```

## 18. Create a list model

The next step involves creating a list of items to show in a list-like window. As discussed in the boxout on the data model, start out by creating a container member during the initialisation of the EnterView.

```
public class EnterView extends VerticalLayout
implements View {
final BeanItemContainer<String>  mySource =
new BeanItemContainer<String>(String.class);
@Override
public void enter(ViewChangeEvent event) {
mySource.addBean("A");
mySource.addBean("B");
mySource.addBean("C");
```

## 19. Bind to list!

Bringing the data created on-screen is best accomplished via one of the list widgets already contained in Vaadin. Lists get created just like any other control – the code accompanying this step shows you how to set one up inside of Enter(). Finally, you can now run the program to see the list in action.

```
@Override
public void enter(ViewChangeEvent event) {
ListSelect select = new ListSelect("The
List");
select.setContainerDataSource(mySource);
select.setNullSelectionAllowed(false);
select.setRows(5);
addComponent(select);
}
```

## 20. Use the debugger

Java IDEs look back at a ten-year history of debugger refinements. When working on Vaadin, most if not all of these features are at your command. Running a Vaadin application with the connected debugger can be accomplished by right-clicking it in the Project Explorer. Next, select Debug as>Run on Server and proceed according to the previous step.

## 21. Learn more!

Discussing Vaadin completely in a single tutorial is impossible. Fortunately, the framework comes with an excellent set of documentation commonly known as the Book of Vaadin. Simply open **vaadin.com/book** in a browser of choice to find out more.

Special offer for readers in **North America**

# 6 issues FREE

When you subscribe*

**FREE** resource downloads in every issue


*"The only magazine you need to design and develop stunning websites"*





Order hotline **+44 (0)1795 592862**

Online at **www.imaginesubs.co.uk/wed**

Quote **USA3** for this exclusive offer!

***Terms and conditions** This is a US subscription offer. You will actually be charged £80 sterling for an annual subscription. This is equivalent to $105 at the time of writing, exchange rate may vary. 6 free issues refers to the USA newsstand price of $14.99 for 13 issues being $194.87, compared with $105 for a subscription. Your subscription starts from the next available issue and will run for 13 issues. This offer expires 30th April 2017.

# Reduce image sizes for quicker page loads

Avoid slow pages by learning how to optimise your images with a number of methods

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

**Making the most of the following steps in this tutorial requires an understanding of some basic tenets of computer science: by default, an image is but a selection of points containing colour information.** When this kind of representation is stored in a file, this is called a bitmap.

Bitmaps can be processed using a method called compression: a compression process c(bmp) usually, but not in all cases, returns a smaller version of the file passed in. The difference between the processes shows up when the process is reversed during a step called decompression: d(c(bmp)) always returns image data. We will call this bmp' in the steps for this tutorial.

Lossless compression formats satisfy the condition bmp==bmp' – when the image is decompressed, its contents are exactly the same as they were before the compression. Lossy compression formats differ in that insignificant information is thrown away – this leads to spectacular savings in some cases, but can cause significant artifacts in others.

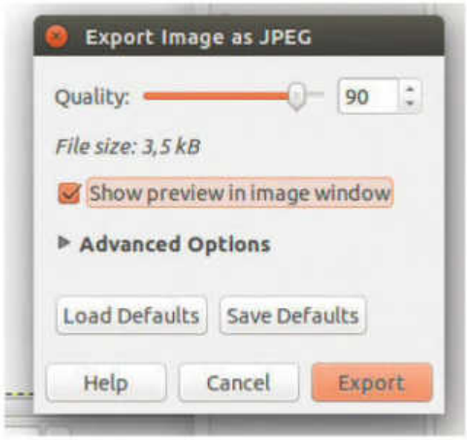
Lossy and lossless formats can play out their strengths in different situations. For example, logos with large single-colour areas usually are handled better by lossless formats, while lossy formats tend to excel when they are dealing with photographs of persons, landscapes or natural objects.

Finally, be aware that the following story touches programs for Unix and Windows. Throughout the tutorial we will tell you which operating system is affected.

Fortunately, the development of virtual machines provides a comfortable way around the problem: both Windows 10 and Ubuntu can be virtualised effortlessly.

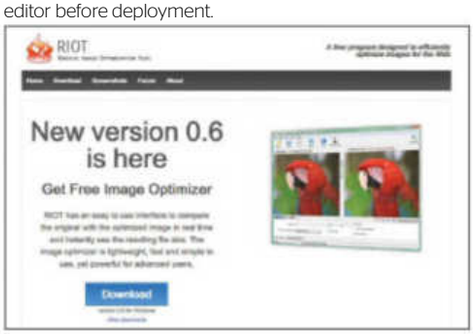
## 1. Experiment with JPG options

When images are handled in GIMP or PhotoShop, JPEG export usually stands at the end of the processing chain. GIMP's export dialog window provides us with a slider which allows you to set the quality level required – checking the Show Preview checkbox even provides an estimate of the resulting file size.




## 2. Go even further

Picking out the best file format – GIMP does offer a bunch – by hand becomes tedious quickly. A small application called RIOT can be integrated into GIMP: visit the RIOT download page at **luci.criosweb.ro/riot/download/**, and click the Download plugin button. Make sure that the path to GIMP is set up correctly, and close the image editor before deployment.

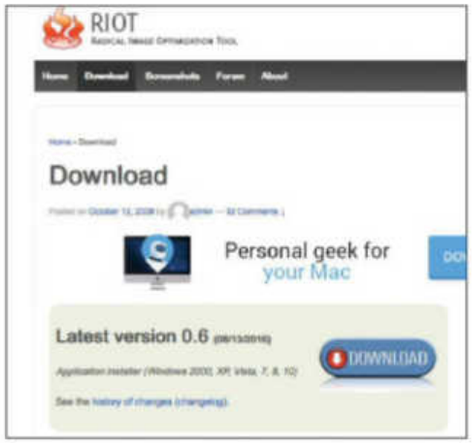



## 3. Invoke RIOT

After successful deployment, the File option of the GIMP is enriched by an option to invoke RIOT. Simply click it to open the side-by-side comparison view that has made the program famous.

## 4. Use RIOT standalone

RIOT can also be used outside of GIMP by deploying the standalone version found at **luci.criosweb.ro/riot/download**. Click the Open button to load a file, and use the mode toggle at the top-right corner of the screen to enable automatic mode. RIOT will then suggest a file format leading to optimal space savings – be aware that this is not always the optimal choice, and can be modified using the tabs below the image swatches.

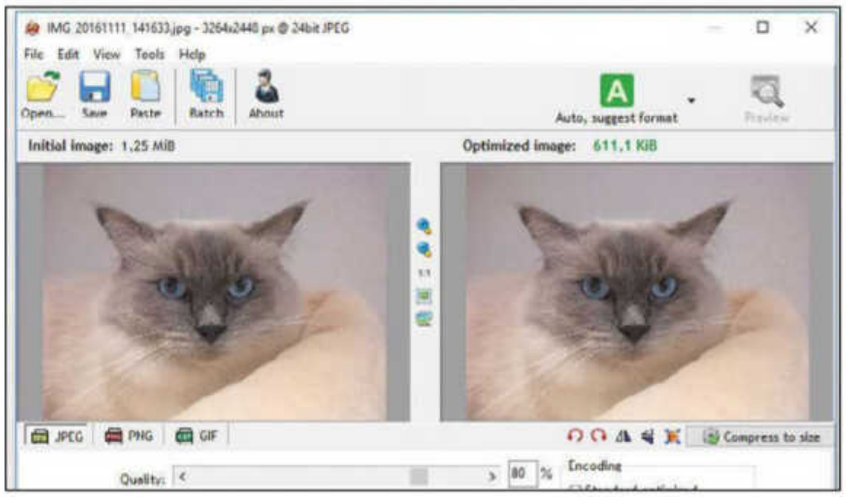



## 5. Multiple images

RIOT can also be applied to multiple images simultaneously. This is best accomplished via the Batch option. Simply click one of the Add options to add pictures to the to-do list of the program. Then, specify an output folder, click Start and brew some coffee while the product gets to work minimising them.

### Processor intensive!

Be aware that more effective compression processes take significantly more CPU time: a ten per cent reduction in file size can cause a tenfold increase in CPU usage.




**Left**
If the optimisation algorithm works well, then RIOT's picture results can be stellar

**Top left**
Don't forget to set the output path – if it is not specified, clicking the Start button leads to an error message

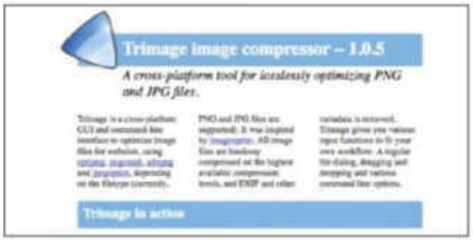
**Top right**
Drop an image into the Trimage window – the program will get to work right away!

## 6. Use Trimage for Unix

Developers working under Unixoid operating systems don't need to miss out – for them, a product called Trimage offers very similar features. Deploy it by entering 'sudo apt-get install trimage' into a command line, and start it from the command line or dock.

## 7. Start processing

Trimage is strictly limited to batch processing and does not change the file formats. Thus, its usage is really easy: simply drop one or more images into the Trimage window. The program will get to work immediately, overwriting the original files in the process. Be aware that large PNG files can easily take a minute or two of processing time.




## 8. Use Zopfli to optimise images

Google engineers recently developed an optimised version of the well-known DEFLATE compression algorithm. One of its side effects was a significant optimisation of the PNG format, which uses DEFLATE internally. PNG files can be optimised using a small application – the code accompanying this step shows how to compile it.

```
tamhan@tamhan-thinkpad:~/Desktop/zopfli$ git
clone https://github.com/google/zopfli.git
tamhan@tamhan-thinkpad:~/Desktop/zopfli$ cd
zopfli/
tamhan@tamhan-thinkpad:~/Desktop/zopfli/
zopfli$ make zopflipng
```

## 9. Work with Zopflipng

When the compile process is complete, the program can be run by entering './zopflipng'. Let us start out by running './zopflipng -help' to get the help output that is shown in the figure accompanying this step on FileSilo.

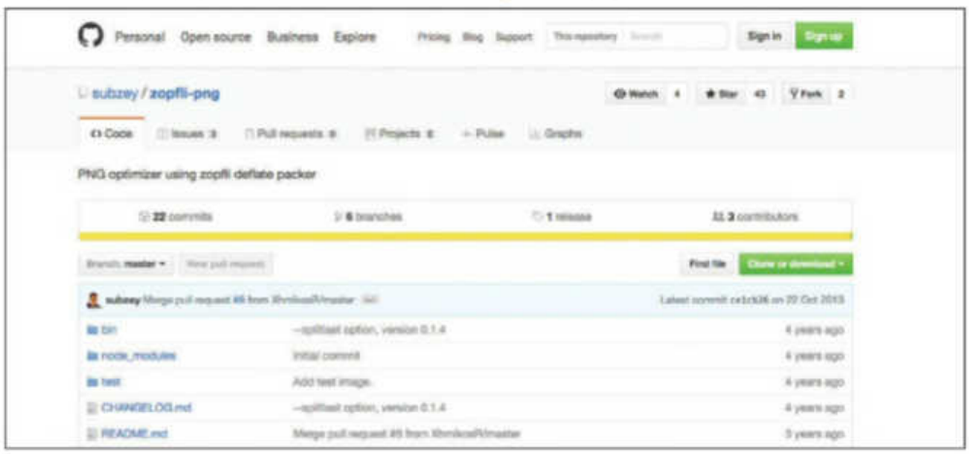
## 10. Compress a single image

The simplest way to use zopflipng is by setting the program loose on a single PNG file. This can be accomplished by entering './zopflipng -m BoesesDiagramm.png KleinesDia.png' – the first of the two parameters is the old file name, while the second of the parameters is the new one. Do take note though that the use of this process can take quite a bit of time – utilising -m can lead to an even more aggressive approach to compression.

## 11. Compress files simultaneously

Even though the most efficient way to run Zopfli most definitely involves custom shell scripts, you should know that zopflipng can also be applied on multiple files directly. In this situation, the –prefix operator will take a string, and this then needs to be appended to each of the files that can be found in the sequence of files delineated with a space.

```
zopflipng [options]... --prefix=[fileprefix]
[files.png]...
```




## Move zopflipng to bin

Should you feel like using zopflipng a lot, consider moving it to one of the binary directories of your workstation – it can then be run without having to type in the whole path every time.

```
tamhan@tamhan-thinkpad:~/Desktop/zop
Optimizing BoesesDiagramm.png
Input size: 326142 (318K)
Result size: 315358 (307K). Percenta
Result is smaller


real    534m58.181s
user    21m16.348s
sys     0m1.400s
```

**Above**
Be careful: optimising some images can literally take hours to complete

**Right**
Google's DNA is clearly visible due to the length of the parameter list permitting detailed optimisations

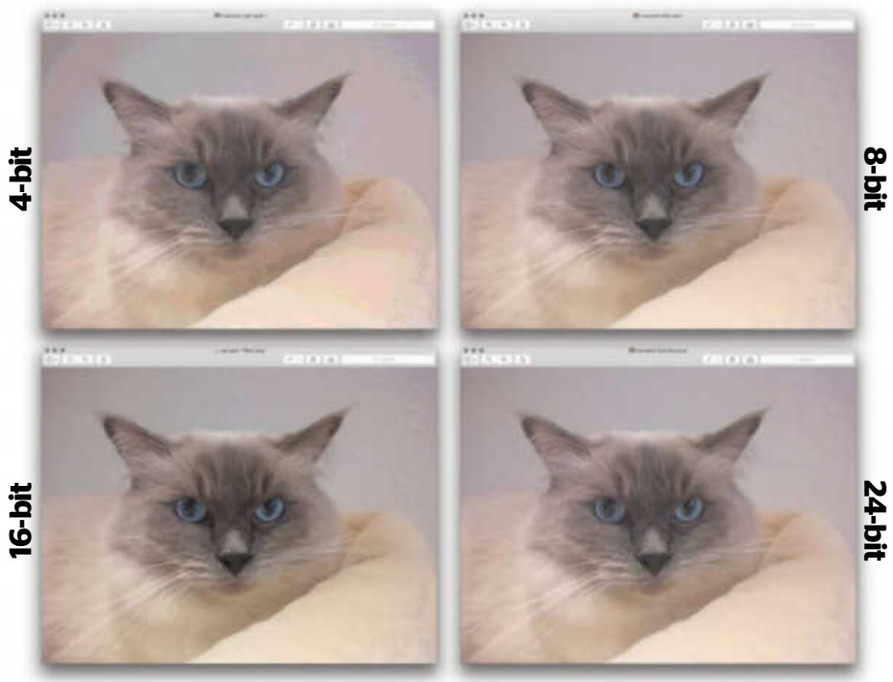
```
tamhan@tamhan-thinkpad:~/Desktop/zopfli/zopfli$ ./zopflipng -help
ZopfliPNG, a Portable Network Graphics (PNG) image optimizer.

Usage: zopflipng [options]... infile.png outfile.png
       zopflipng [options]... --prefix=[fileprefix] [files.png]...

If the output file exists, it is considered a result from a previous run and not overwritten if t
 is smaller.

Options:
-m: compress more: use more iterations (depending on file size)
--prefix=[fileprefix]: Adds a prefix to output filenames. May also contain a directory path. Wher
efix, multiple input files can be given and the output filenames are generated with the prefix
 If --prefix is specified without value, 'zopfli_' is used.
 If input file names contain the prefix, they are not processed but considered as output from pr
 This is handy when using *.png wildcard expansion with multiple runs.
-y: do not ask about overwriting files.
--lossy_transparent: remove colors behind alpha channel 0. No visual difference, removes hidden i
--lossy_8bit: convert 16-bit per channel image to 8-bit per channel.
 -d: dry run: don't save any files, just see the console output (e.g. for benchmarking)
--always_zopflify: always output the image encoded by Zopfli, even if it's bigger than the origir
chmarking the algorithm. Not good for real optimization.
-q: use quick, but not very good, compression (e.g. for only trying the PNG filter and color type
--iterations=[number]: number of iterations, more iterations makes it slower but provides slightl
mpression. Default: 15 for small files, 5 for large files.
--splitting=[0-3]: ignored, left for backwards compatibility
--filters=[types]: filter strategies to try:
 0-4: give all scanlines PNG filter type 0-4
 m: minimum sum
 e: entropy
```


**Script by hand!**
When dealing with user-uploaded images, creating a custom compression toolchain out of some of the command-line utilities mentioned in this tutorial can be a highly efficient choice. An ideal implementation would consider not only the amount of storage space saved, but also the amount of times that the image will be accessed.

For example, bringing out heavy compression algorithms such as Google's Zopfli is unlikely to make a lot of sense when dealing with a picture uploaded by an extremely unpopular user who celebrates himself on having a grand total of five followers. In addition to that, a less efficient compression algorithm could also be used in times where there are large amounts of images are being transferred.

## 12. Reduce colours

Throwing away unneeded things is the easiest way to downsize an estate. With image files, a similar approach can be taken by reducing the amount of colour information that is being stored. As an example, you can find in the figure below three versions of the same image which were treated to subsequent colour reductions. In many cases, reducing the amount of colour shades does not necessarily lead to significant deterioration of image quality (Base image © abctexte / Dr. Doris Maria KOHRS, used with permission).




## 13. Decolourise PNGs

While PNG is best known as a lossless format, nifty developers have since combined it with a variety of algorithms for colour depth reduction. Pngquant is one of the best examples of the breed – on Ubuntu, first check if the version of the program downloaded from the repository is recent as the quality of the output has increased a lot after the release of version 2.0.

```
tamhan@tamhan-thinkpad:~$ pngquant -version
warning: option '-version' has been replaced
with '--version'.
2.0.1 (September 2013)
```

## 14. Reduce file size

When the program is installed, creating a size-reduced version of a file is a question of entering a single command line command. Setting the speed value to 1 is important as it ensures that the program uses more CPU time in order to yield a higher quality output.

```
pngquant --speed 1 image.png
```

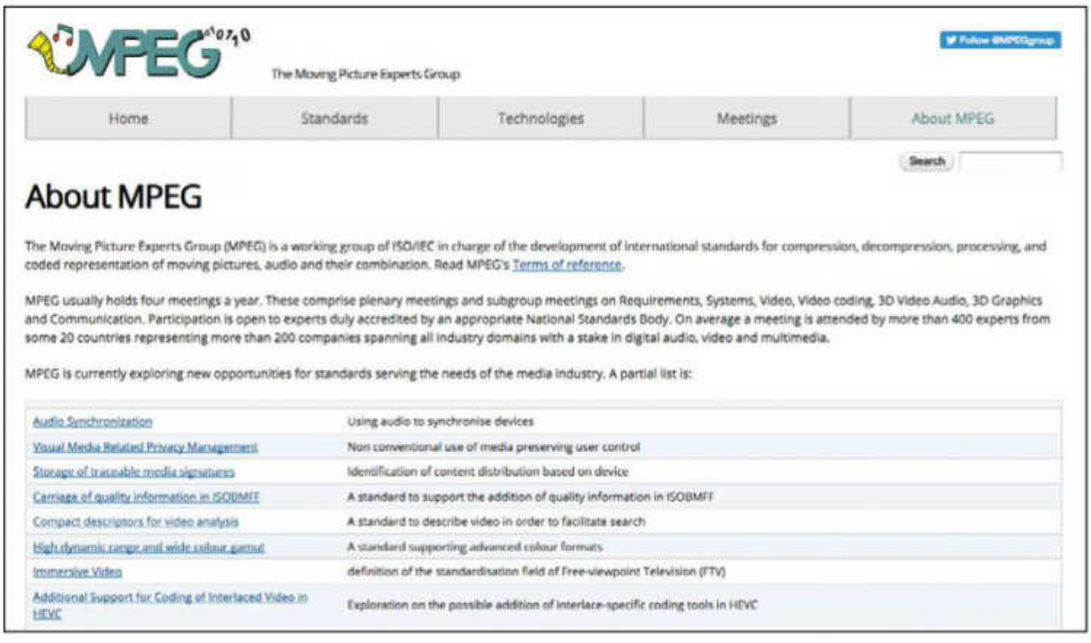
## 15. PNGQuante for Windows

Windows users can rejoice: PNGQuant is also available for all recent versions of Microsoft's operating system. Simply visit **pngquant.org/pngquant-windows.zip**, extract the archive to a comfortable location and kick off according to the instructions in Step 14 above.

## 16. Boost your JPGs

Mozilla engineers can be considered the inspiration behind Zopfli, as they found a way to create a more efficient encoder.

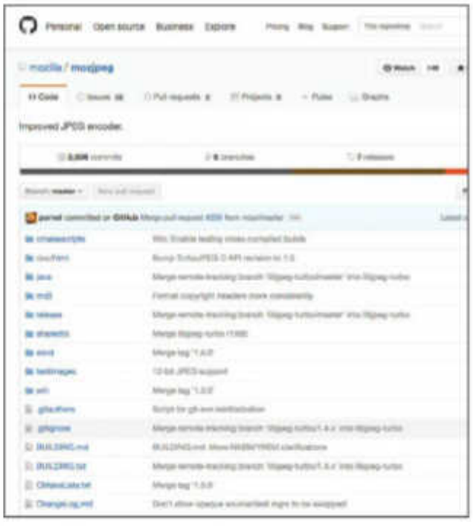
```
git clone https://github.com/mozilla/mozjpeg
cd mozjpeg
sudo apt-get install autoconf automake
libtool nasm make pkg-config git
autoreconf -fiv
./configure -prefix=/usr make
```




### Decipher decompressor-specified image file formats

Much of the magic shown in this article is enabled by a trick on the side of the specifiers of image and video formats: instead of specifying a format as whole, they specify the decoder. If an encoder is included, it is explicitly referred to as an 'example implementation'.

This allows developers to set their imagination loose at finding new ways to create smaller and smaller files: the rule is that everything goes, as long as the reference decoder provided in the specification is able to process and decode it without any modifications.



## 17. Compress more files!

With that, we are now ready to compress some files. Simply enter './jpgtran' followed by the name of the file to be processed to set the file crushing process in motion – be aware that the re-encoding process can take quite a bit of time.

```
tamhan@tamhan-thinkpad:~/Desktop/mozjpg/
mozjpeg$ ./jpegtran
```

## 18. Download Exif

The EXIF file lets digital cameras and photo editing programs embed metadata into JPEG files: while these bits of information usually aren't too big, removing them can still save a few hundred bytes each time. Under Windows, this is best accomplished by utilising the ExifTool which can be downloaded at **sno.phy.queensu.ca/~phil/exiftool/**.


## 19. Run ExifTool

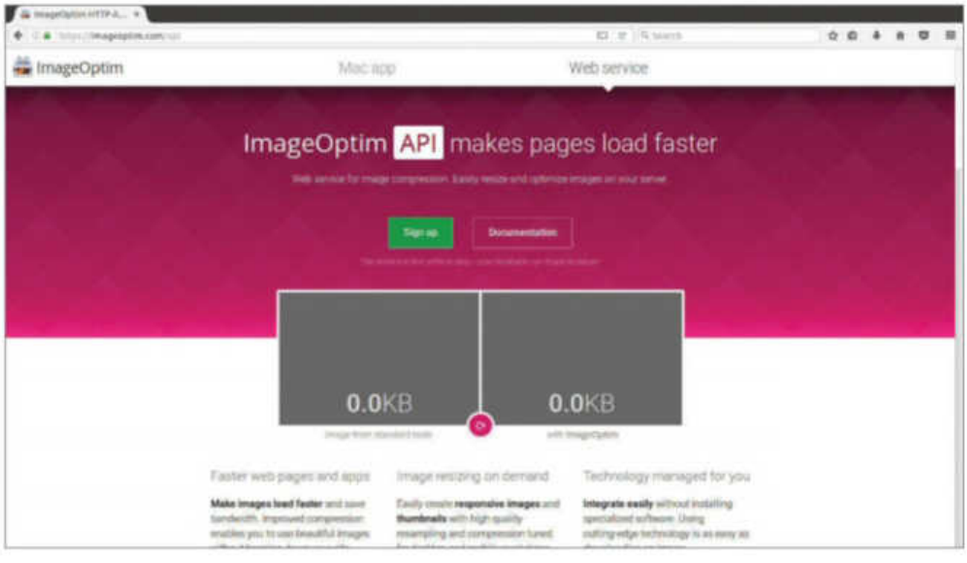
Once the program is installed, run ExifTool to make it output a detailed list of all command contained in the executable. Removing the Exif data in a single file can be accomplishing by invoking 'exiftool -all= file.jpg' – if multiple files are to be processed, a batch or PowerShell script will come in handy.

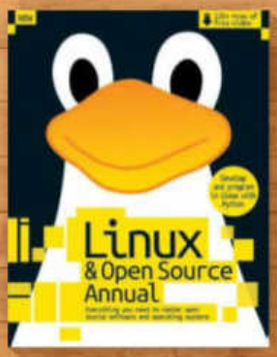
## 20. Use a web service

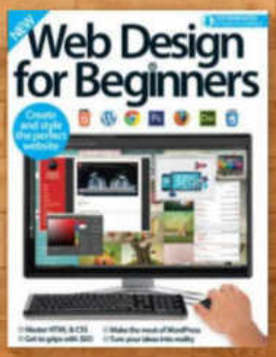
Various companies offer web services where developers and end users can upload images which are to be optimised. Mac users know ImageOptim well – its online derivate, found at **imageoptim.com/api**, is less well known. Be aware that web services are a two-sided sword: while easy to deploy, the usage charges can add up over time.

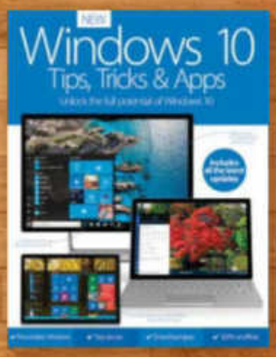
## 21. Avoid GIFs

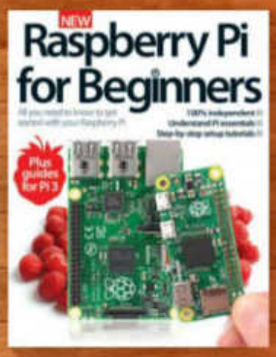
Even though the GIF patent wars are but a distant memory for the average developer: GIFs should still burn. A highly optimised PNG is more efficient than a GIF in almost all cases – if animations are not needed, using GIFs is a waste of bandwidth. This is also valid for automatic encoders, where the GIF run can be skipped.





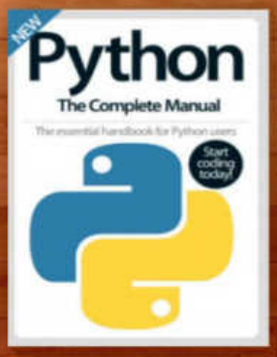

# Discover one of our great bookazines

## And save 20% on your first order when you buy direct from us

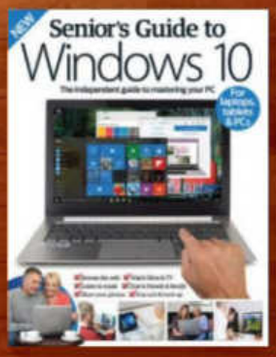

BRAND NEW!

### Raspberry Pi for Kids

Raspberry Pi is a great learning tool, helping to introduce your kids to the world of coding and programming. This book is packed with fun projects!







Get great savings when you buy direct from us

1000s of great titles, many not available anywhere else

World-wide delivery and super-safe ordering

Future

**www.imagineshop.co.uk**

Magazines, back issues & bookazines.

Use the code "WELCOME20" when you make your first order to receive your discount

## Get your listing in our directory

*To advertise here contact Chris*

**chris.mitchell@futurenet.com**
**+44 (0)1225 687832**

# COURSE LISTINGS

## Featured: Makers Academy

www.makersacademy.com
*Twitter: @makersacademy*
*Facebook: MakersAcademy*

### *About us*

**Makers Academy is a fully immersive, 12-week computer programming boot camp.** With their help, you will learn the principles of software craftsmanship and they'll also help you get your first job. They're Europe's number-one developer boot camp, running highly selective classes of the offline course every six weeks, and a remote course every 12 weeks. They take a 'learn by doing' approach, through project-based work. Students are encouraged to work in pairs on coding challenges, with weekly tests, culminating in a final project. They help set up job interviews via their network of hiring partners including ThoughtWorks and Deloitte Digital.

### *What we offer*

- **On-site:** 12 week full-time coding course from the on-site campus in London
- **Remote:** 12 week full-time coding course remotely from home

### *5 tips from the pros*

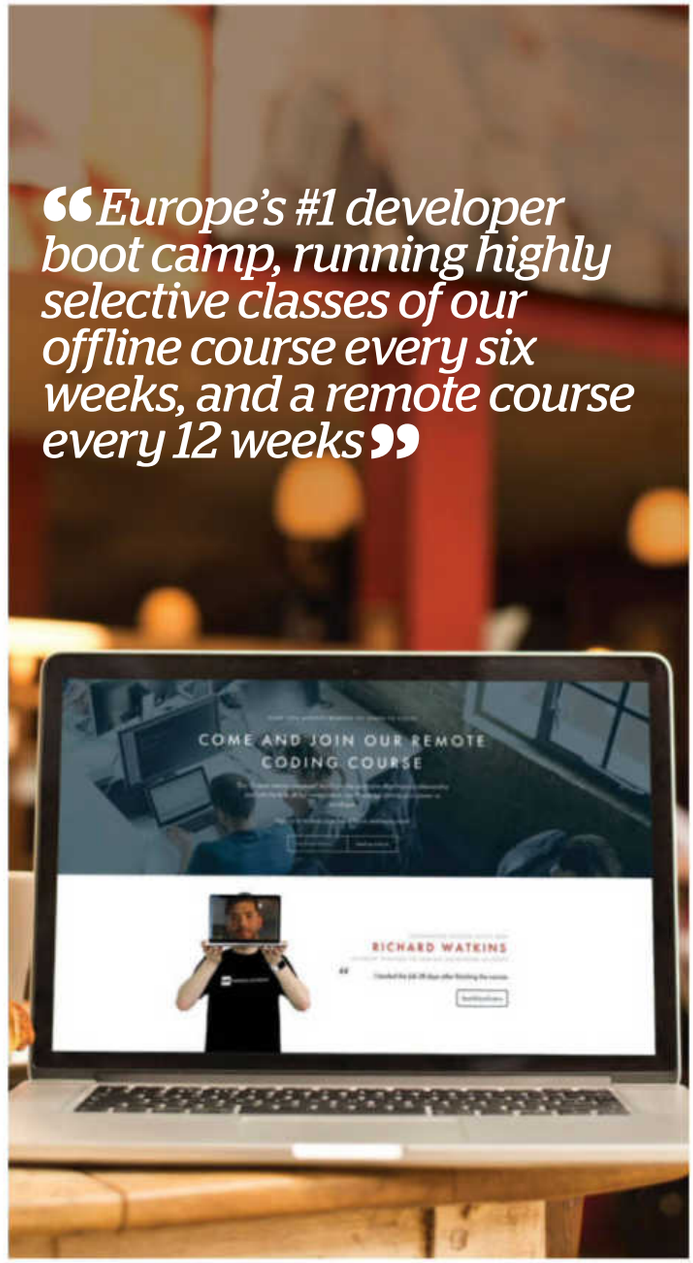
**1.Research all your options**
Do your research into lots of different boot camps, read the reviews, read the student blogs and reach out to previous graduates and speak to them.

**2.Dabble in code**
Although the course is for beginners, it's important that you've started to at least try to learn to code on your own.

**3.Prepare for the interview**
We send you everything to prepare for the interview. Make sure you go through all the resources and give yourself two weeks to prepare.

**4.Budget**
The course is full-time for three months and it can take up to three months after to secure a job. It's important to financially plan for the period you won't be working.

**5.Visit us!**
Book a visit and come visit us! Come see the Makers Academy HQ in person and learn more.




**Richard Watkins**
***Science teacher to junior developer at Shift***
Makers Academy was frustrating and daunting but amazing and I wouldn't change any of it. I landed the job 28 days after finishing the course.

**Ina Tsetsova:**
***Email campaign manager to graduate software developer at ThoughtWorks***
I found a really nice community and I've met really cool people. I got a job quicker than I expected.

**Hannah Carney**
***3D designer to junior developer at Play Consulting***
Makers Academy not only focuses on your learning for code, but they also focus on your well-being. Work feels like fun and I've finally found a job I love.

# Get your listing in our directory

*To advertise here contact Chris*

**chris.mitchell@futurenet.com**
**+44 (0)1225 687832**

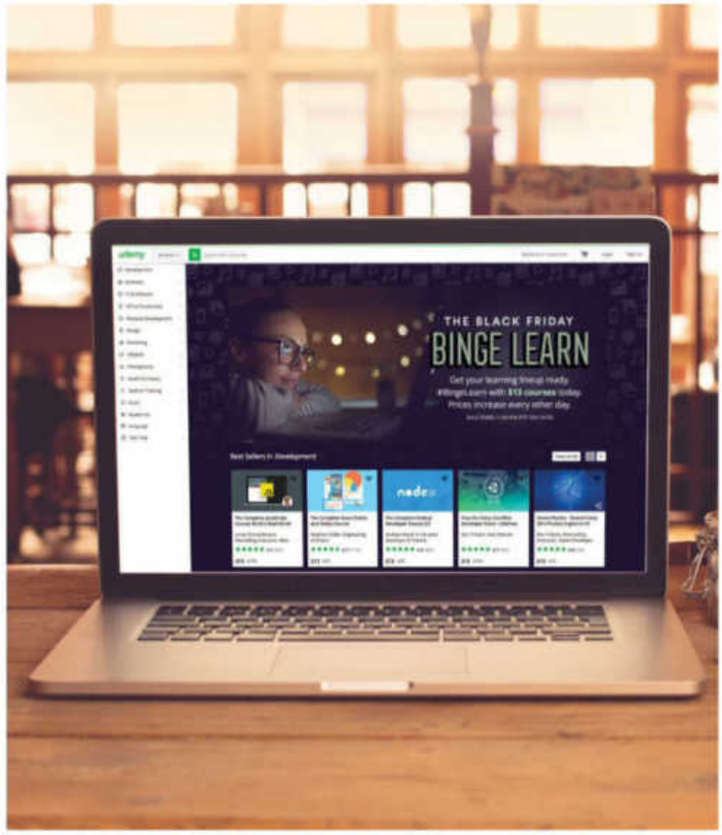
## UDEMY

www.udemy.com

*Twitter: @udemy*
*Facebook: udemy*

The inspiration for Udemy began in a small village in Turkey, where founder Eren Bali grew up frustrated by the limitations of being taught in a one-room school house. Realising the potential of learning on the internet, he set out to make quality education more accessible. Udemy is now a global marketplace for learning and teaching online. Students can master new skills by choosing from an extensive library of over 40,000 courses including HTML, CSS, UX, JavaScript and web development.

**40,000+ courses:** There is a course for every designer and dev
**Self-paced learning:** Learn how to code at your own pace



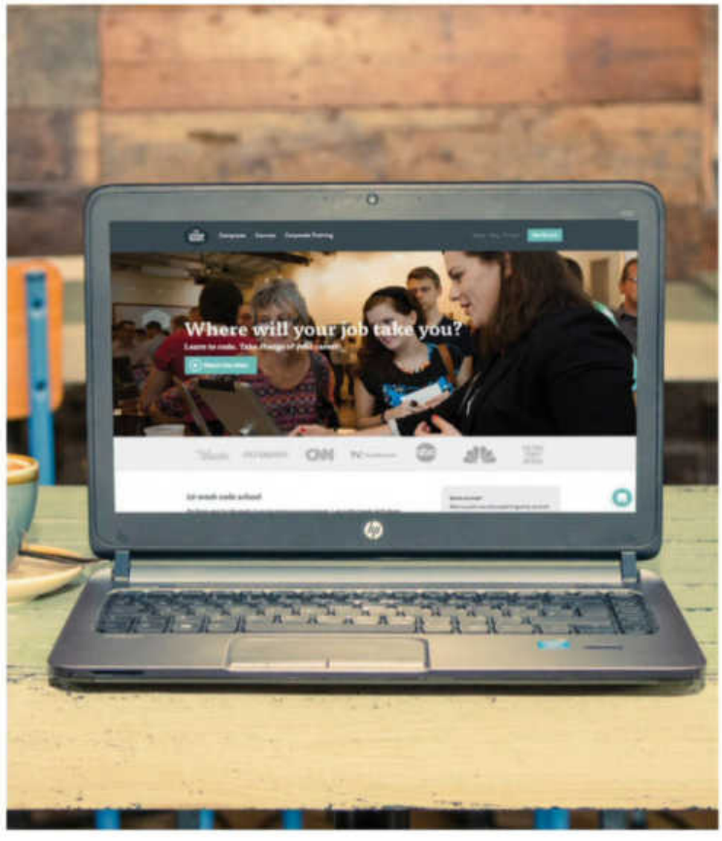

## THE IRON YARD

www.theironyard.com

*Twitter: @TheIronYard*
*Facebook: TheIronYard*

The Iron Yard is one of the world's largest and fastest-growing in-person code schools. It offers full-time and part-time programs in Back-End Engineering, Front-End Engineering, Mobile Engineering and Design. The Iron Yard exists to create real, lasting change for people, their companies and communities through technology education. The in-person, immersive format of The Iron Yard's 12-week courses helps people learn to code and be prepared with the skills needed to start a career as junior-level software developers.

**12-week code school:** Learn the latest skills from industry pros
**Free crash courses:** One-night courses, the perfect way to learn




## WE GOT CODERS

www.wegotcoders.com
*hello@wegotcoders.com*

We Got Coders is a consultancy that provides experts in agile web development, working with startups, agencies and government. Take one of their 12-week training course that covers all that is required to become a web developer, with highly marketable full-stack web development skills.

- Classroom-based training
- Real-world work experience
- Employment opportunities

## FUTURELEARN

www.futurelearn.com
*feedback@futurelearn.com*

Choose from hundreds of free online courses: from Language & Culture to Business & Management; Science & Technology to Health & Psychology. Learn from the experts. Meet educators from top universities who'll share their experience through videos, articles, quizzes and discussions.

- Learn from experts
- Free courses
- All-device access

## GYMNASIUM

www.thegymnasium.com
*help@thegymnasium.com*

Gymnasium offers free online courses designed to teach creative professionals in-demand skills. Courses are all self-paced and taught by experienced practitioners with a passion for sharing practical lessons from the design trenches.

- Gain real-world skills
- Get expert instruction
- Career opportunities

## Get your listing in our directory

*To advertise here contact Chris*

**chris.mitchell@futurenet.com**
**+44 (0)1225 687832**

# HOSTING LISTINGS

## Featured host: tidyhosts

www.tidyhosts.com / 05603 674610

### *About us*

**Tidyhosts has become a leading cloud hosting provider throughout the world.** It boasts a feature-rich selection of services including domain registration, shared web hosting, WordPress hosting, hosted exchange email, cloud virtual servers and media streaming. Its passion and drive for success has gained a highly reliable and trusted reputation from its customers, making us number one for hosting services. Founded in 2004 by three developers who wanted to build hosting infrastructure for developers, years on, the company has expanded to offer much more.

### *What we offer*

- **Domain names** – Simple domain registration with a large choice of TLDs.
- **Shared web hosting** – Includes one-click application installers and a choice of Windows or Linux.
- **Cloud virtual servers** - Take full control over your hosting, install the software you need on your server.
- **SHOUTcast hosting** - Start your own radio station with our SHOUTcast hosting service, easy to set up and use.

### *5 Tips from the pros*

**1. Ensure that you choose the right domain**

**When choosing your domain, make sure it's easy to remember and resembles what you are offering.** Customers are more likely to come back if they can remember your URL.

**2. Create clear, concise website content**

**Keep the content on your website clear, informative and, more importantly, relative!** Avoid any duplication of content on different pages as this can affect your rankings on search engines.

**3. Utilise SSL certificates to stay secure**

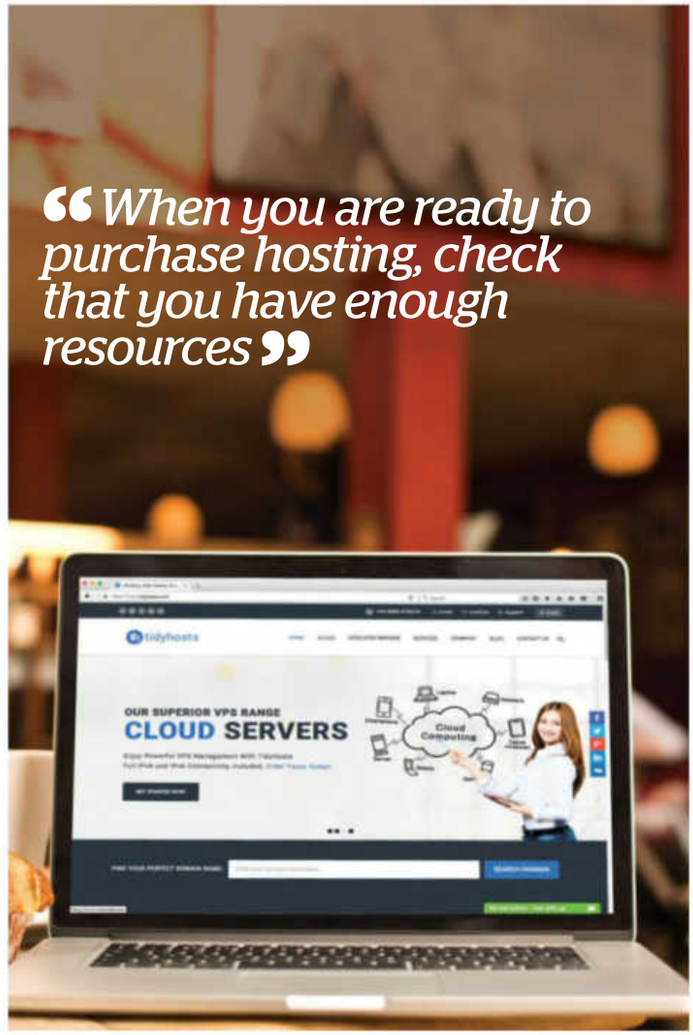
**More sites are moving to SSL security to protect their customers from the ever-increasing threats on the web.** Search engines are now ranking sites with an SSL higher than those without.

**4. Make sure you choose the right plan**

**When you are ready to purchase hosting, check that you have enough resources, especially if you expect your website to grow quickly.** Seek advice from tidyhosts if you are unsure.

**5. Use one-click installers when building a website**

**If you are new to building a website then we have a number of useful one-click application installers.** These help you get up and running, including the popular WordPress system.




### *Testimonials*

**Kelly Underwood**

"I've been with a number of web host providers in the past which have at some point let me down. I now realise the importance of using a host that is well known."

**John Corey**

"We have our email hosted with tidyhosts, which serves a small number of users in our office. We have found this solution is much more cost-effective."

**Jenny Brice**

"A great host with fantastic knowledge. I have only had to use the support channels a few times, but the replies I have had helped me instantly, so well worth it."

# Get your listing in our directory

*To advertise here contact Chris*

**chris.mitchell@futurenet.com**
**+44 (0)1225 68s7832**

## UK-based hosting

CYBERHOSTPRO

www.cyberhostpro.com
0845 5279 345

**Cyber Host Pro are committed to providing the best cloud server hosting in the UK; they are obsessed with automation.** If you're looking for a hosting provider who will provide you with the quality you need to help your business grow then look no further than Cyber Host Pro.

- Cloud VPS servers
- Reseller hosting
- Dedicated servers

## Cluster web hosting

fasthosts

www.fasthosts.co.uk
0808 1686 777

**UK based and operating 24/7 from dedicated UK data centres.** Fasthosts keep over one million domains running smoothly and safely each day.
Their services can be self-managed through the award-winning Fasthosts Control Panel. They also maintain a highly successful reseller channel.

- Dedicated servers
- Cloud servers
- Hosted email

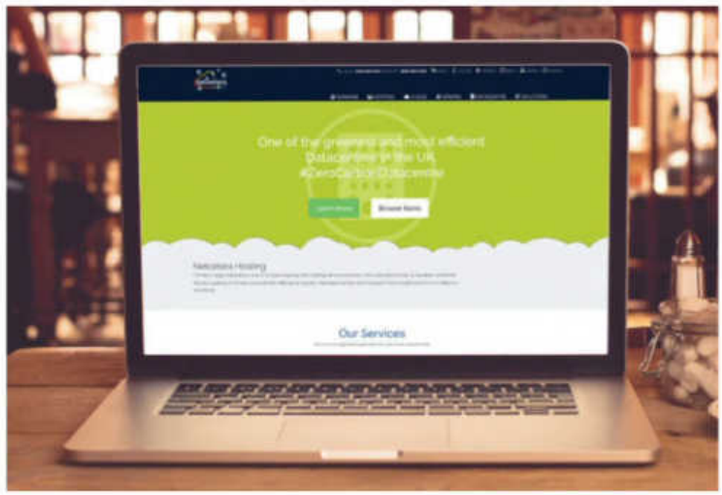
## Supreme hosting

CWCS MANAGED HOSTING — SUPREME HOSTING. SUPREME SUPPORT.

www.cwcs.co.uk
0800 1777 000

**CWCS Managed Hosting is the UK's leading hosting specialist.** They offer a fully comprehensive range of hosting products, services and support. Their highly trained staff are not only hosting experts, they're also committed to delivering a great customer experience and are passionate about what they do.

- Colocation hosting
- VPS
- 100% Network uptime



## Zero carbon hosting

netcetera

www.netcetera.co.uk
0800 808 5450

**Formed in 1996, Netcetera is one of Europe's leading web hosting service providers.** It has customers in over 75 countries worldwide. Netcetera is a leading IT infrastructure provider offering co-location, dedicated servers and managed infrastructure services to businesses worldwide. It has one of the greenest, most efficient, cost-effective state-of-the art data centres operating in the UK today.

- Domain names
- Cloud hosting
- Managed hosting


## Budget hosting

HETZNER ONLINE

www.hetzner.com
+49 (0)9831 505-0

**Hetzner Online is a professional web hosting provider and experienced data centre operator.** Since 1997, the company has provided private and business clients with high-performance hosting products as well as the infrastructure for the efficient operation of sites. A combination of stable technology, attractive pricing, flexible support and services has enabled Hetzner Online to strengthen its market position both nationally & internationally.

- Dedicated/shared hosting
- Colocation racks
- SSL certificates

## SSD Web hosting

bargainhost

www.bargainhost.co.uk
0843 289 2681

**Since 2001, Bargain Host have campaigned to offer the lowest possible priced hosting in the UK.** They have achieved this goal successfully and built up a large client database which includes many repeat customers. They have also won several awards for providing an outstanding hosting service.

- Shared hosting
- Cloud servers
- Domain names

## All-inclusive hosting

1&1

www.1and1.co.uk
0333 336 5509

**1&1 Internet is a leading hosting provider that enables online success.** Established in 1988, 1&1 now operates across ten countries. With a comprehensive range of high-performance, affordable products, 1&1 offers everything from domain names to eCommerce and powerful cloud servers.

- Easy domain registration
- Professional eShops
- High-performance servers

## Fast, reliable hosting

:BYTEMARK

www.bytemark.co.uk
01904 890 890

**Founded in 2002, Bytemark are "the UK experts in cloud & dedicated hosting".** Their manifesto includes in-house expertise, transparent pricing, free software support, keeping promises made by support staff and offering top-quality hosting hardware at fair prices.

- Managed hosting
- UK cloud hosting
- Linux hosting


Did you know that European forests, which provide wood for making paper and many other products, have grown by 44,000km² over the past 10 years? That's more than 1,500 football pitches every day![†]

Love magazines? You'll love them even more knowing they're made from natural, renewable and recyclable wood

[†]UNFAO, Global Forest Resources Assessment 2005-2015.

Two Sides is a global initiative promoting the responsible use of print and paper which, when sourced from certified or sustainably managed forests, is a uniquely powerful and natural communications medium.

There are some great reasons to #LovePaper
Discover them now,
**twosides.info**

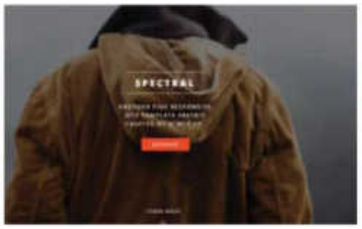
TWO SIDES

# Free with your magazine

## Instant access to these creative resources…

### Essential assets and templates

Get textures, fonts, backgrounds and more



### Exclusive video tutorials

Learn to code/create with HTML, CSS, JS & PHP

### Tutorial project files

All the assets you'll need to follow our tutorials

### Plus, all of this is yours too…

- All-new tutorial files to help you master this issue's HTML, CSS and JavaScript techniques
- 84 minutes of expert responsive typography videos from Pluralsight (www.pluralsight.com)
- 60 Retro effects and frames for Photoshop from SparkleStock (www.sparklestock.com)
- 3 responsive HTML/CSS templates

## Log in to filesilo.co.uk/webdesigner

Register to get **instant access** to this pack of must-have creative resources, how-to videos and tutorial assets

**Free for digital readers too!**
Read on your tablet, download on your computer

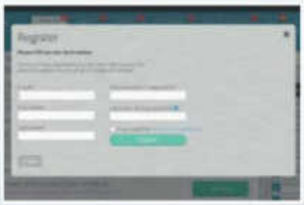
FileSilo

The home of great downloads – exclusive to your favourite magazines from Future PLC

- **Secure and safe online access, from anywhere**
- **Free access for every reader, print and digital**
- **Download only the files you want, when you want**
- **All your gifts, from all your issues, in one place**

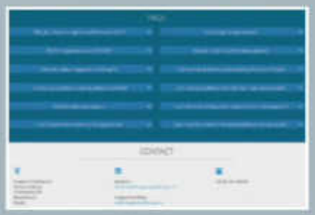
## Get started

Everything you need to know about accessing **your FileSilo account**



**01** Follow the instructions on screen to create an account with our secure FileSilo system. Log in and unlock the issue by answering a simple question about the magazine.

**02** You can access FileSilo on any computer, tablet or smartphone device using any popular browser. However, we recommend that you use a computer to download content, as you may not be able to download files to other devices.



**03** If you have any problems with accessing content on FileSilo, take a look at the FAQs online or email our team at the address below:
*filesilohelp@imagine-publishing.co.uk*

# An incredible gift for subscribers


## Subscribe today & unlock the free gifts from more than 45 issues

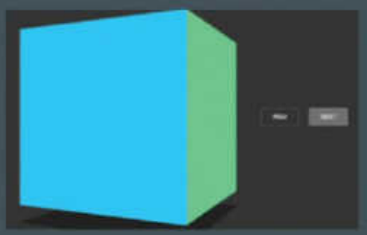
Access our entire library of resources with a money-saving subscription to the magazine – that's more than 900 free resources

### Over 45 hours of video guides

Let the experts teach you to create and code


### More than 400 tutorials

Get the code you need to get creative



### Over 210 creative assets

Templates, fonts, textures and backgrounds

## Head to page 62 to subscribe now

**Already a print subscriber?**
**Here's how to unlock FileSilo today…**

Unlock the entire Web Designer FileSilo library with your unique Web ID – the eight-digit alphanumeric code printed above your address details on the mailing label of your subscription copies – also found on any renewal letters.

**More than 900 reasons to subscribe**

**+ More added every issue**

# GET **SUPER FAST** SITES

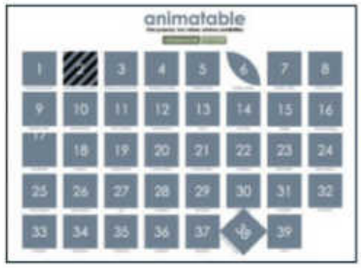
**Get your hands on the tools and techniques to speed up your sites, get more hits, and retain customers and visitors**

## UNIQUE LAYOUTS WITH CSS SHAPES

Step outside the standard and create page designs that impress

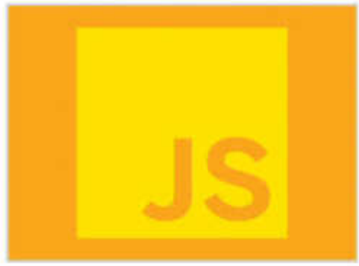
## GET STARTED WITH ANGULAR 2

Implement a parallax view using the Angular 2 framework




## CREATE INSTANT ANIMATION

Introduce multiple animations to page design with Animatable




## UNIVERSAL JS APPLICATIONS

Build a universal JavaScript app using the Next.js library



ALL IN YOUR NEXT

**WEB DESIGNER**

Issue 259 on sale

**Thursday 2nd March 2017**

Authorised Reseller


**UPGRADE & KEEP THE OLD KIT**

**NO QUIBBLE 3 YEAR WARRANTY**

**THE UNIQUE OPTION TO OWN THE KIT**

**LEASING IS 100% TAX DEDUCTABLE**

## APPLE iMAC 21.5"

FROM **£9.40**

iMac

## MACBOOK PRO 13" TOUCH BAR

FROM **£15.45**

Mac Pro

*PRICES PER WEEK + VAT

HARDSOFT

Established for over 25 years

020 7111 1643

WWW.HARDSOFT.CO.UK