

29  
PAGES OF  
EXPERT TUTORIALS

FREE 61 MINUTES  
OF JS VIDEOS

# web designer™

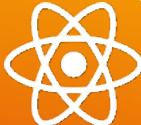
STREAMLINE  
**DESIGN**  
Speed up workflow  
with Sketch

**CHATBOTS**  
What do developers  
need to consider?

## HOT NEW

Top properties for today and tomorrow

CSS Grid 🔥 @supports 🔥 CSS Variables 🔥 CSS Filters



REAL-TIME  
**REACT**  
Create a feedback app



Build interactive 3D ▪ Get started with Three.JS  
Deploy to Google Cloud ▪ Convert Flash to HTML

Introducing **STORM**, a hosting platform that makes your agency and websites run smoother. 80% of our customers have seen a 25% improvement in site speeds.

# SUPER CHARGE YOUR HOSTING

**STORM**

Developed by  
 **NimbusHosting**

[nimbushosting.co.uk](http://nimbushosting.co.uk)

# 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**  
Editor

## Fire up your CSS



emember CSS back in the old days? Probably not, but it has come a long way and is a lot more powerful. Technologies and libraries that used to be outside helpers are slowly being merged into CSS as formal standards. Just think CSS Grid: how far away is that from using tables for layouts?

In our lead feature this issue we take a look at the latest, greatest and, more importantly, the most useful updates to CSS. Alongside CSS Grid, we take a closer look at @supports (AKA feature queries); how to apply CSS Filters; what you can do with CSS Variables; some lesser-known background-repeat properties; aspect ratio media

queries; how to use blend mode; how to apply object-fit to images; and, to finish, we look at five upcoming additions to the specification that deserve your attention.

What else do we have this issue? What are the best tools and techniques to streamline your design workflow? We show you how to go from wireframes right through to the development handover using Sketch and Zeplin.

Elsewhere we look at the cognitive chatbot revolution that is happening and why developers need to embrace chatbot tools. We also get answers to what accessibility issues could arise, and what the future may hold for conversational AI. And there's more: get the third part of our Getting started with Three.js series; how to build a real-time feedback app with React; how to deploy to Google Cloud; and much more. Enjoy.

When building components that use new and potentially unsupported features, it's important to include a fallback.

## Highlight

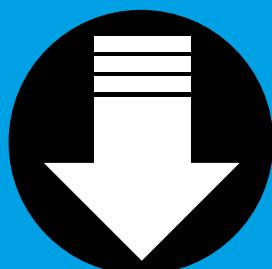


We are staunch advocates of Blair Enns' *The Win Without Pitching Manifesto* and have a policy of no pitching and no working for free.

Experience UX are on a mission. **Web Designer** finds out more. [Page 34](#)

Follow us on Twitter for all the news & conversation [@WebDesignerMag](#)

Visit our blog for opinion, freebies & more [www.creativebloq.com](#)



**FREE - exclusive with this issue**  
**153 Designer resources**

**Video Tuition** - Part 4 of Beginner's JavaScript video guides from Killersites ([shop.killervideostore.com](http://shop.killervideostore.com))

**Assets** - 121 Duotone PS filters and 12 Simple business card mockups from Sparklestock ([sparklestock.com](http://sparklestock.com))  
- Tutorial files and assets



[www.filesilo.co.uk/webdesigner](http://www.filesilo.co.uk/webdesigner)

# This issue's panel of experts

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



## Mirum

Andrew Collinson is a UI Team Lead at Mirum. He has 17 years of experience and is joined by Paul Sandwell (Interface engineer) and Dan Hughes (JavaScript engineer) for this article. Discover the latest and greatest CSS you need to be using in your projects and take a peek at what's coming to the spec very soon. **Page 42**

Things have become far more complicated for developers as people expect more, but now they are now slowly being merged into CSS as formal standards

### Alex Lee



Alex is a UX designer and minimalism advocate at Impression ([www.impression.co.uk](http://www.impression.co.uk)) and a Nottingham Forest supporter. This issue he focuses on streamlining web projects with Sketch and Zeplin. **Page 68**

### Mark Shufflebottom



Mark is a Professor of Interaction Design at Sheridan College. In this issue he uses Jeeliz Face Filter to track 3D models onto faces in WebGL using the device camera across desktop, mobile and tablets. **Page 54**

### Richard Mattka



Richard is an award-winning Interactive director, designer and developer. In this second tutorial in an ongoing 3D programming series, he shows you how to create lights and shadows, using the popular Three.js library. **Page 54**

### Vesko Kolev



Vesko is the VP of Product, Developer Tooling at Progress. He is responsible for a team of 150+ software engineers, product managers and UX designers. This issue he talks about cognitive chatbots and what the future holds. **Page 76**

### Daniel Crisp



Daniel Crisp is a senior frontend developer at a startup in London's Canary Wharf. This issue he shows you how to deploy a single page application to Google's global infrastructure using either Google Cloud or Firebase. **Page 86**

### David Howell



David is a journalist with over 20 years' experience in publishing and runs his own business, Nexus Publishing. This issue he takes time out to talk to the talented crew at Experience UX and discovers how they like to work. **Page 34**

### Matt Crouch



Matt is a frontend developer working with Vidsy, he uses WebSockets to communicate between products in real time. This issue, find out how to use WebSockets to power a React application using Node. **Page 80**

### Leon Brown



Leon is a freelance web developer and trainer who assists web developers in creating efficient code for projects. This issue he recreates a host of techniques as inspired by the top-class sites seen in Lightbox. **Page 14**

## Follow us!

**Facebook:** [www.facebook.com/  
WebDesignerUK](http://www.facebook.com/WebDesignerUK)  
**Twitter:** [https://twitter.com/  
webdesignermag](https://twitter.com/webdesignermag)

**Future PLC** Richmond House, 33 Richmond Hill, Bournemouth, Dorset, BH2 6EZ

### Editorial

Editor **Steven Jenkins**  
[steve.jenkins@futurenet.com](mailto:steve.jenkins@futurenet.com)  
01202 586233

Designer **Harriet Knight**  
Editor in Chief **Amy Hennessey**  
Senior Art Editor **Will Shum**

### Contributors

Louis Georgiou, Mark Billen, Leon Brown, David Howell, Mark Shufflebottom, Andrew Collinson, Paul Sandwell, Dan Hughes, Richard Mattka, Tomasz (Merixstudio), Alex Lee, Matt Crouch, Daniel Crisp, Philip Morris, Rob Mead-Green, Will Shum

### Photography

James Sheppard  
All copyrights and trademarks are recognised and respected

### Advertising

Media packs are available on request  
Commercial Director **Clare Dove**  
[clare.dove@futurenet.com](mailto:clare.dove@futurenet.com)  
Senior Advertising Manager **Mike Pyatt**  
[michael.pyatt@futurenet.com](mailto:michael.pyatt@futurenet.com)  
01225 687538

Account Director **George Lucas**  
[george.lucas@futurenet.com](mailto:george.lucas@futurenet.com)  
Account Manager **Chris Mitchell**  
[chris.mitchell@futurenet.com](mailto:chris.mitchell@futurenet.com)

### International

*Web Designer* is available for licensing.  
Contact the International department to discuss partnership opportunities  
International Licensing Director **Matt Ellis**  
[matt.ellis@futurenet.com](mailto:matt.ellis@futurenet.com)

### Subscriptions

Email enquiries [contact@myfavouritemagazines.co.uk](mailto:contact@myfavouritemagazines.co.uk)  
UK orderline & enquiries 0344 848 2852  
Overseas order line and enquiries +44 (0) 344 848 2852  
Online orders & enquiries [www.myfavouritemagazines.co.uk](http://www.myfavouritemagazines.co.uk)  
Head of subscriptions **Sharon Todd**

### Circulation

Head of Newtrade **Tim Mathers**

### Production

Head of Production **Mark Constance**  
Production Project Manager **Clare Scott**  
Advertising Production Manager **Joanne Crosby**  
Digital Editions Controller **Jason Hudson**  
Production Manager **Nola Cokely**

### Management

Managing Director **Aaron Asadi**  
Commercial Finance Director **Dan Jotcham**  
Editorial Director **Paul Newman**  
Head of Art & Design **Greg Whittaker**

**Printed by** William Gibbons, 28 Planetary Road, Willenhall, WV13, 3XT

**Distributed by** Marketforce, 5 Churchill Place, Canary Wharf, London, E14 5HU [www.marketforce.co.uk](http://www.marketforce.co.uk) Tel: 0203 787 9060

### ISSN 1745-3534

We are committed to only using magazine paper which is derived from responsibly managed, certified forestry and chlorine-free manufacture. The paper in this magazine was sourced and produced from sustainable managed forests, conforming to strict environmental and socioeconomic standards. The manufacturing mill holds full FSC (Forest Stewardship Council) certification and accreditation.

All contents © 2018 Future Publishing Limited or published under licence. All rights reserved. No part of this magazine may be used, stored, transmitted or reproduced in any way without the prior written permission of the publisher. Future Publishing Limited (company number 2008865) is registered in England and Wales. Registered office: Quay House, The Ambury, Bath BA1 1UA. All information contained in this publication is for information only and is, as far as we are aware, correct at the time of going to press. Future cannot accept any responsibility for errors or inaccuracies in such information. You are advised to contact manufacturers and retailers directly with regard to the price of products/services referred to in this publication. Apps and websites mentioned in this publication are not under our control. We are not responsible for their contents or any other changes or updates to them. This magazine is fully independent and not affiliated in any way with the companies mentioned herein.

If you submit material to us, you warrant that you own the material and/or have the necessary rights/permissions to supply the material and you automatically grant Future and its licensees a licence to publish your submission in whole or in part in any/all issues and/or editions of publications, in any format published worldwide and on associated websites, social media channels and associated products. Any material you submit is sent at your own risk and, although every care is taken, neither Future nor its employees, agents, subcontractors or licensees shall be liable for loss or damage. We assume all unsolicited material is for publication unless otherwise stated, and reserve the right to edit, amend, adapt all submissions.



Future plc is a public company quoted on the London Stock Exchange (symbol: FUTR)  
[www.futureplc.com](http://www.futureplc.com)

Chief executive **Zillah Byng-Thorne**  
Chairman **Richard Huntingford**  
Chief financial officer **Penny Ladkin-Brand**  
Tel +44 (0)1225 442 244

# Digital Camera **PHOTOGRAPHER** OF THE YEAR 2019

Sponsored by



NOW  
FREE TO  
ENTER!

[www.photocrowd.com/dcpoty](http://www.photocrowd.com/dcpoty)

# Digital Camera **YOUNG** **PHOTOGRAPHER** OF THE YEAR 2019

Sponsored by



AFFINITY  
PHOTO

# 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@futurenet.com](mailto:webdesigner@futurenet.com) • [@WebDesignerMag](https://twitter.com/WebDesignerMag) • [www.creativebloq.com](http://www.creativebloq.com)

## 08 What is the number one browser in the world?

No surprises, it's Google Chrome. But it isn't as dominant as you might expect

## 10 WebKit: The best must-try resources out there

Discover the libraries and frameworks you need

## 11 Voice assistants: approach with caution

Voice assistants are here to stay – but how are consumers engaging with brands?

## 14 Lightbox

A showcase of inspirational sites and the techniques used to create them

## 26 Chasing Rainbows

Find out how three entities came together to create the Universal Love website

## 34 Generation UX

Experience UX are on a mission to transform the experience of digital interface users

## 42 Hot new CSS

Discover the top properties you need to be using today, and what's coming tomorrow

## 68 Streamline your design

The pros reveal the tools and techniques that make a great design workflow

## 76 A new conversation

A look at the applications for intelligent chatbots and the considerations developers should have

## 92 Hosting listings

An extensive list of web hosting companies. Pick the perfect host for your needs

## 94 Course listings

Want to start learning online? Check out what courses are out there with this list

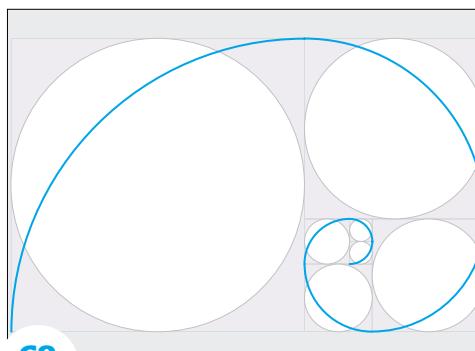
## 98 Next month

What's in the next issue of **Web Designer**?

## Cover focus



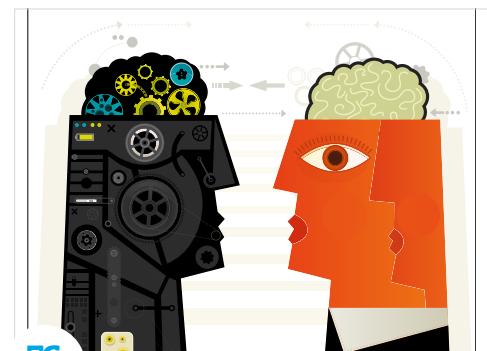
42



68

## Speed up design workflow

Find out what Sketch and Zeplin have to offer



76

## Chatbots

What do developers need to consider?

# FileSilo

96 Get the latest must-have resources and videos

- Part 4 of Beginner's JavaScript video guides from Killersites
- 121 Duotone Photoshop filters
- 12 Simple business card mockups



**Subscribe today and save 20%**

<https://bit.ly/2sGwB3h>



 Voice technology is in the same sort of space as the app store was ten years ago. It's all a bit Wild West 

COMMENT — Louis Georgiou - p11



**34**  
**ProFile:** Experience UX  
A deep understanding of the user



**14**  
**Lightbox:** Alessi  
Virtual fragrances you can almost smell

Visit the **WEB DESIGNER** online shop at  
**myfavouritemagazines.co.uk**  
for the latest issue, back issues and specials

## Tutorials

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



### 54 Get started with Three.js

The latest in the series demonstrates how to light a scene and introduce beautiful shadows

### 58 Convert from Flash to HTML5

How to use JavaScript, WebGL and the Phaser framework to get games working in all browsers

### 62 Code interactive 3D models

Create and build personal interactive experiences in 3D with the Jeeliz Face Filter head tracking library

## Web Workshop

### 52 Select character to switch titles

[hassyadai.com](http://hassyadai.com)

Introduce subtle effects that transform from image to text when selected by the user

### 66 Fade-in content navigation

[hinderlingvokart.com](http://hinderlingvokart.com)

Add simple but smart animations to add engagement to menus

## Web Developer

### 74 A new conversation

Learn what the applications for intelligent chatbots are, and the considerations that developers should have

### 80 Build a feedback app

Combine React's powerful state management with real-time messaging

### 86 Deploy your SPA to Google Cloud

Set up a simple Node.js server and deploy an app to Google App Engine or Firebase Hosting

# Header

The tools, trends and news to inspire your web projects

## What's the world's number-one browser?

No surprises, it's Google Chrome. Yet it's not as dominant as you might expect as **Web Designer** discovers



Browsers are a key component of the web experience and they are slowly supporting more and more technologies. The future is bright for designers and developers, but as we all know different browsers offer different support. One day, maybe, browsers will all be equal and designers and developers won't have to worry about who supports what. We live in hope.

Every so often *Web Designer* likes to look at how the browser market is evolving and what it might tell us. There may be no surprises in what takes top spot in the browser market. Yes, it's Google Chrome. Worldwide and across all platforms it has a 58.94 percent market share (as of June 2018), a rise of around five percent since last year. If we dig down into the figures we can see where Chrome is winning. It's worldwide share on mobile has seen a rise from 49 percent to 54 percent and its dominance on desktop continues with a

slightly smaller rise of three percent, going from 63 percent to 66 percent. But it's not a winner everywhere. Which device has a big winner? Tablet. Here, Safari is the browser of choice with 60 percent compared to Chrome, which has nearly 25 percent. Android has half of Chrome at 12 percent.

Back to all devices and what happens when we shift to different locations? Let's

percent market share but Safari is a relatively close second with 28 percent share. Swap to desktop only and the gap widens to 52 percent. But switch to mobile and the two are almost even with Google nudging ahead by just under one percent. Switch to tablet and Safari is miles in front with 68 percent compared to Chrome's 17 percent. Think this might be something to

**“Switch to tablet and Safari is miles in front with 68 percent”**

start with Europe. Again Chrome is out in front with nearly 59 percent market share, a rise of over six percent on the previous year. So where's Firefox? Second spot goes to Safari with just over 15 percent, down one percent on the previous year. Firefox doesn't even reach ten percent with 9.6 percent market share, a decline of nearly two percent. Heading to North America Chrome is once again out in front with a 52

do with Apple. Over to Asia and Chrome is the market leader with 58 percent, but the popular UC Browser comes in second with 16 percent. This is followed by Safari and Opera with 7.96 percent and 4.27 percent respectively. Over to Africa and it is Chrome again with 57 percent, but Opera slots into second with 15 percent, which suggests that low-cost, low-powered mobiles are a big part of the internet economy.

## STAT ATTACK

### DESKTOP VS MOBILE

How does mobile's market share compare with desktop?

#### Worldwide

52.52%



Mobile is nearly nine percent ahead of desktop

#### Europe

56.18%



Here desktop is king at 18 percent ahead

#### North America

51.48%



Mobile is closing – it's only ten percent behind

#### Africa

62%



Mobile is miles ahead by 27 percent

#### Asia

68.88%



Mobile is a whopping 35 percent ahead

**CB** CREATIVE BLOQ  
creativebloq.com

In-depth tutorials, expert tips, cutting-edge features, industry interviews, inspiration and opinion. Make sure to get your daily dose of creativity, design and development.



**Subscribe today and save**

**20%**

<https://bit.ly/2qLxVl4>

Source: gs.statcounter.com  
(correct as of June 2018)

# Sites of the month

**01.**

**GLOBAL VALUES WHERE DO YOU FIT?**

The ways we express ourselves are influenced by the things we believe are important - our values. But what do your values look like?

Take 2 minutes to answer some simple questions, find out whether you're an Explorer, Innovator, Guardian or Custodian then share & compare your results.

#MyTomorrow



**03.**

Type Terms

[View project →](#)



## Graphics Look looks

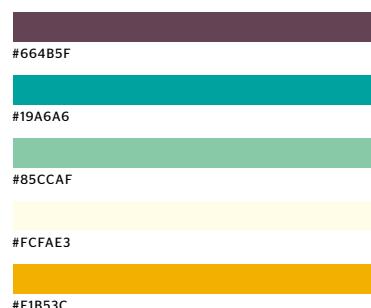
[bit.ly/2IU8DgW](http://bit.ly/2IU8DgW)

Very distinctive style and colour from the very talented Cathal Duane.



## Colour picker Little Did I Know

[bit.ly/2u8n2tz](http://bit.ly/2u8n2tz)



## Typesetter Walbaum

[bit.ly/2IYsWd4](http://bit.ly/2IYsWd4)

A huge family of fonts to choose from, with a host of stylish styles and weights.



## WordPress Ayojok

[bit.ly/2KUc8Pe](http://bit.ly/2KUc8Pe)

Fancy starting your own conference? This theme will give you the ideal starting point.



## Header

### Resources

CONTACT US AT: [webdesigner@futurenet.com](mailto:webdesigner@futurenet.com) | @WebDesignerMag

# webkit

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

The screenshot shows the Layoutit! interface. On the left, there are three sections: 'Grid Columns' with four columns set to 'fr', 'Grid Rows' with three rows set to 'fr', and 'Grid Gap' set to '1 px'. At the top right are buttons for 'Get the code!', 'Permalink', 'Save', and 'X'. The main area is a grid editor with a red border around a central column. A placeholder 'Area name...' is in the center. A cursor is shown over the grid.

## Layoutit

[layoutit.com/grid](https://layoutit.com/grid)

As we all know, grids are very much the basis of any good page layout and CSS Grid has given designers and developers plenty of power. But if you don't want to take the time out to learn CSS Grid, or just need something fast, then this tool is a great choice. Add your columns/grids, name the different components of the layout and then grab the code.



### Variable Fonts

[v-fonts.com](https://v-fonts.com)

Want to try out a variable font and see what it has to offer? Then this simple slider-powered font resource is just what you need.



### Essential Image Optimization

[images.guide](https://images.guide)

Image optimisation is an often forgotten, but essential art. This extensive e-book tells you pretty much everything that you need to know.



### Accessibility

[bit.ly/2BxyhPr](https://bit.ly/2BxyhPr)

Accessibility needs to be a key component of any design. This official document goes into detail about how it can be applied when using React.

## TOP 5 Web conferences - September 2018

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



### Reasons.to

[reasons.to](https://reasons.to)

A conference for multi-disciplined digital creatives who draw their inspiration from many different influences.



### The UX Conference

[theuxconf.com](https://theuxconf.com)

The theme of this conference is collaboration between designers and researchers.



### Circles Conference

[circlesconference.com](https://circlesconference.com)

Circles is a three-day conference that provides an engaging space to learn from industry leaders.



### Generate

[generateconf.com](https://generateconf.com)

The conference for web designers is back with a stellar line-up of speakers and must-learn workshops.



### Web Audio Conference

[webaudioconf.com](https://webaudioconf.com)

Dedicated to technologies including Web Audio API, Web MIDI and Web RTC / WebGL.

# Voice assistants: approach with caution

Voice assistants are here to stay, but how are consumers engaging with brands?



**Louis Georgiou**

Managing Director at Code Computerlove  
[codecomputerlove.com](http://codecomputerlove.com)

“ Voice technology is in the same sort of space as the app store was ten years ago. It’s all a bit Wild West ”

The latest figures from YouGov claim one in ten Brits now own at least one voice assistant and it's a trend that is continuing to rise as consumers look at more ways to make use of these handy devices in their day-to-day lives.

But a new survey of 1,000+ British consumers who currently own a voice assistant has revealed that what people are actually searching for, and using the likes of Alexa for, is still quite limited. Unsurprisingly, the most popular use was to play music or the radio (65 per cent) and half of owners use them for news and weather reports. Other common uses include travel updates (16.2 per cent) and playing audiobooks/podcasts (8.2 per cent). Interestingly, only seven per cent of users have ever bought anything via their smart speaker. When asked what they wish their home assistant could help them with, one in five Brits said they'd like their voice assistant to make them funnier and more attractive! 21 per cent want help with joke-telling, and almost ten per cent would like their voice assistant to help them with their flirting and dating skills. 12 per cent of parents also want their smart speakers to distract their children!

Looking at combined results, most smart-speaker users really want Alexa to act as their PA to help with day-to-day chores such as passing on messages, calendar reminders for major events, booking tickets, menu planning and paying bills. The survey also looked at where in the home virtual assistant usage is most prevalent. The lounge came in at number one (almost 60 per cent), followed by the kitchen (38 per cent) and then the bedroom (26 per cent). And almost one in ten have had a chat with their virtual assistant in the bathroom! Despite 40 per cent of consumers using their smart speaker daily, and a further 40 per cent at least once a week, 60 per cent said what puts them off using it more (or having additional conversations with their voice devices) is that it often doesn't understand them, gives wrong answers, doesn't do what they need it to do, or it bores them.

So how can brands and businesses encourage consumers to engage with them via these increasingly popular devices? To a certain extent, voice technology is in the same sort of space as the app store was ten years ago. It's all a bit Wild West. There's lots of gaming going on and experimentation. Code Computerlove's The Higher Lower Game, which is an App Store-topping game based on Google search data, is one such experimentation – developed using an Alexa Skill to help the team explore Voice UI. Developing The Higher Lower Game provided some useful insights into the challenges of moving into this space.

One area that is gathering pace is linking smart speakers with home automation systems. Alexa and Google Home can be used to control home heating, lighting and security technology; even window blinds fitted with a motorised operating system can be raised and lowered using voice technology. As voice assistant usage continues to rise, brands will need to look at their voice strategies and design digital experiences that are created to not only be found using this technology, but that are worth engaging with.

But approach with caution. As soon as a brand new shiny piece of tech comes along, lots of companies can't resist the urge to 'want one of those'. It doesn't matter if they need it or not – it's corporate FOMO.

So prior to investing hundreds of pounds in developing for voice technology, it is important to check that the technology is there to achieve what you're trying to deliver to the business. Role-playing and mapping out the potential UX – using the tried-and-tested Post-its on the wall technique – can quickly identify that the end product is achievable with the current technology and will actually provide a return on investment.

Don't get carried away with the idea of creating a conversation – a bit of playfulness can work, but remember this is still a simple command/response tool right now. Think about the space your user will be in and the task you're helping them accomplish – is voice genuinely a better way of them getting the job done than typing, swiping or clicking? If you're building an Alexa Skill, there's nowhere better to start than Amazon's own guide ([amzn.to/2AcVlPo](http://amzn.to/2AcVlPo)).

# webkit

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

MAKE A WEBSITE START A BLOG REVIEWS RESOURCES BLOG ABOUT

## HTML5 MOBILE FEATURE CHECKER

Are those cool new APIs you read about safe to use in the real world? Find out here.

Tested on real devices.

✓ SAFE

### CANVAS API

2D Drawing API

Score: 99/100 [Which browsers support this?](#)



### VIEWPORT DEFINITION

Meta tag support.

Score: 98/100 [Which browsers support this?](#)



### CSS3 BASICS

opacity, backgrounds, text effects, rounded corners

Score: 96/100 [Which browsers support this?](#)



✗ UNSUPPORTED

### SERVER-SENT EVENTS

EventSource pattern to maintain the connection to the server open

Score: 59/100 [Which browsers support this?](#)



### FULLSCREEN API

Allow the application to get a full screen mode

Score: 57/100 [Which browsers support this?](#)



### MEDIA CAPTURE

Taking pictures, record video and audio from an input file type

Score: 52/100 [Which browsers support this?](#)



## Mobile Feature Checker

[bit.ly/2IYUCaV](http://bit.ly/2IYUCaV)

Which of the hottest new APIs works on the small screen? Does the API you want to use have support across mobile browsers? This simple site is split into two: Safe and Unsupported. Find the API you want to use and click 'Which browsers support this?' for more info.



### Gio.js

[giojs.org](http://giojs.org)

Gio.js is a declarative 3D Globe data visualisation library built with Three.js. It's simple to use with fully interactive, rich 3D results.



### Selection.js

[bit.ly/2IV9xd3](http://bit.ly/2IV9xd3)

A simple, lightweight library to realise visual DOM Selections. Supports any CSS library. Plus it has support for Node.js and touchscreen devices.



### Angular Augury

[augury.angular.io](http://augury.angular.io)

In a nutshell, Angular Augury is a Google Chrome Dev Tools extension for debugging Angular 2 applications. Gives insights into app structure and more.

## TOP 5 WordPress themes

Need to get a good-looking website up and running quickly? Try one of these themes



### Nord

[nord.fwp.fastwp.net](http://nord.fwp.fastwp.net)

A clean, minimalistic theme that offers a touch of sophistication and breathing space to show off any work or projects.



### Watsonia

[bit.ly/2KO4FEI](http://bit.ly/2KO4FEI)

A simple, but stylish, lifestyle theme with a large header and a focus on bold imagery and smart typography.



### Magazilla

[bit.ly/2IYSvnl](http://bit.ly/2IYSvnl)

A news magazine-style theme with a big hero image/story to grab users' attention. Plus neat grids to offer plenty more stories.



### Gretna

[bit.ly/2KzpwfD](http://bit.ly/2KzpwfD)

A simple, blog-style theme, which combines big colourful sections with neat typography and different image styling.



### Prequelle

[bit.ly/2IZ7ogE](http://bit.ly/2IZ7ogE)

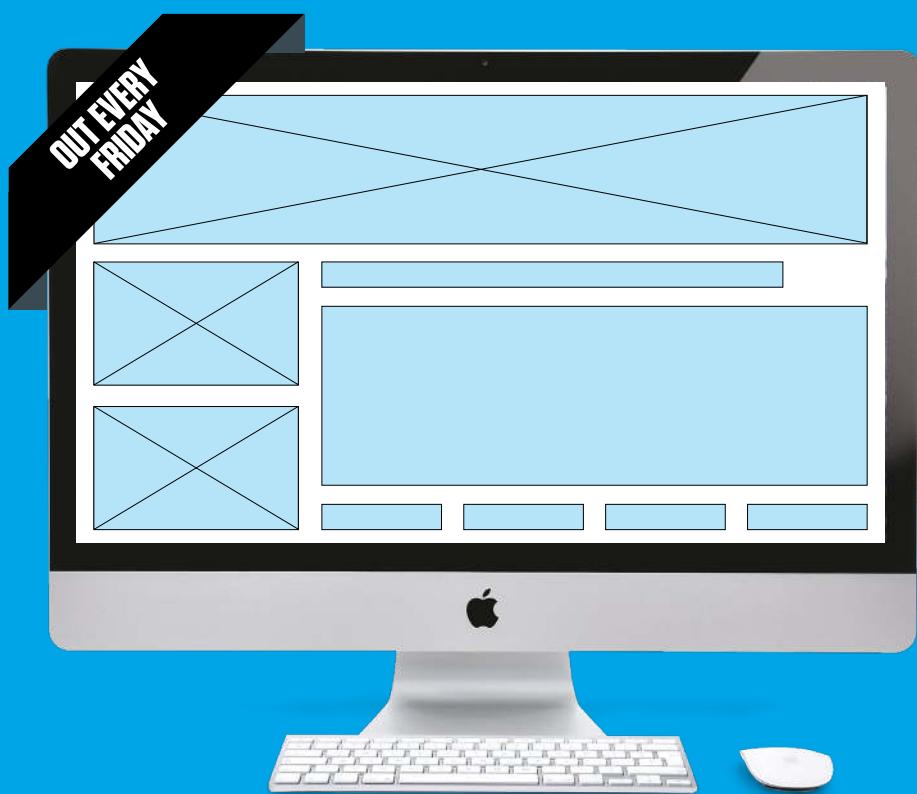
An eCommerce theme with a nod to fashion. With plenty of white space and a strong focus on the product.

**HEAR FROM THE PROS**

**SIGN UP TO THE**

**designer**<sup>TM</sup>

**NEWSLETTER TODAY!**



**Get weekly news, tips & inspiration**

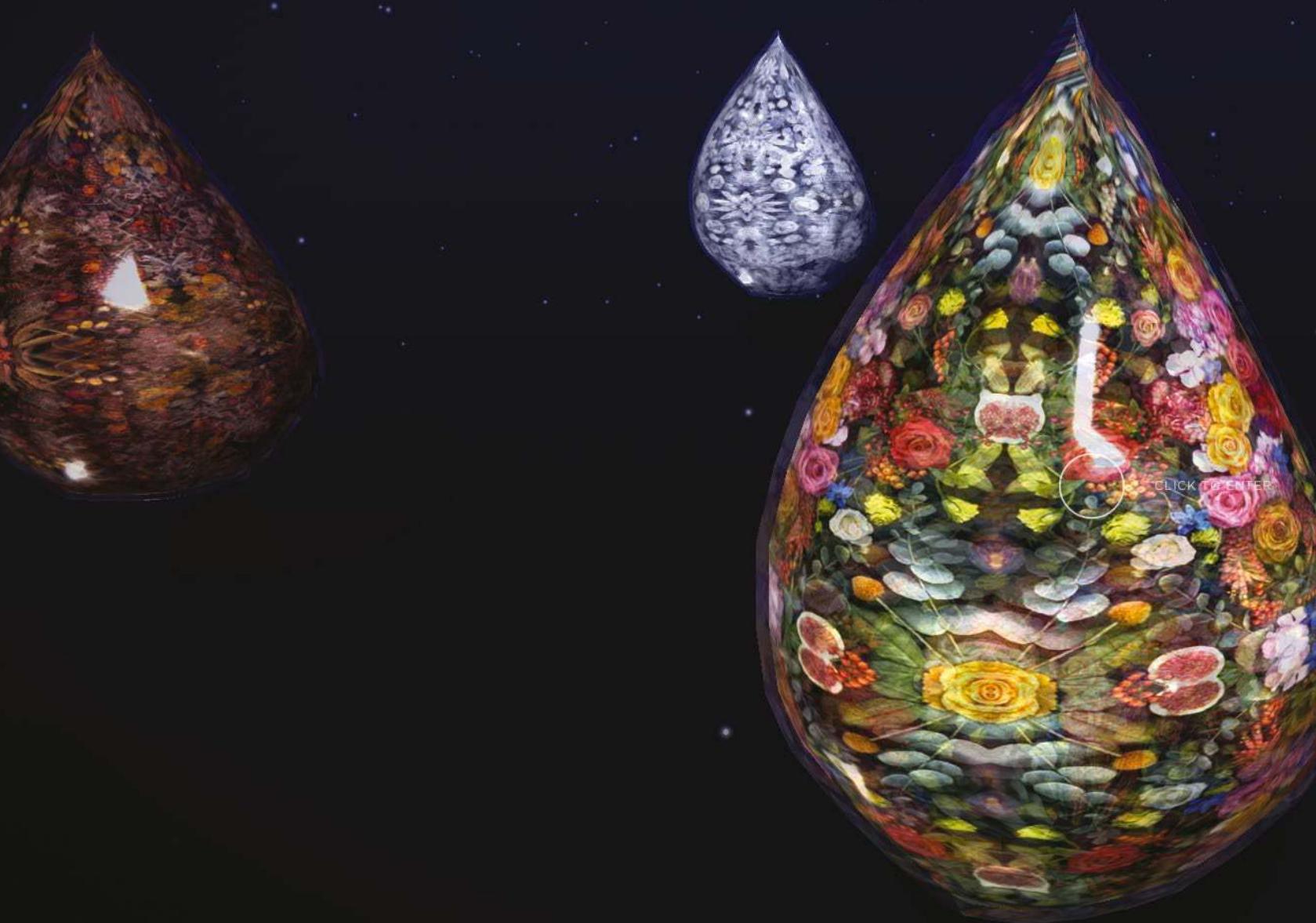
**SIGN UP NOW!**

[bit.ly/2KI5b7Y](https://bit.ly/2KI5b7Y)

# Alessi - The Five Seasons

[alessi.com/thefiveseasons](http://alessi.com/thefiveseasons)

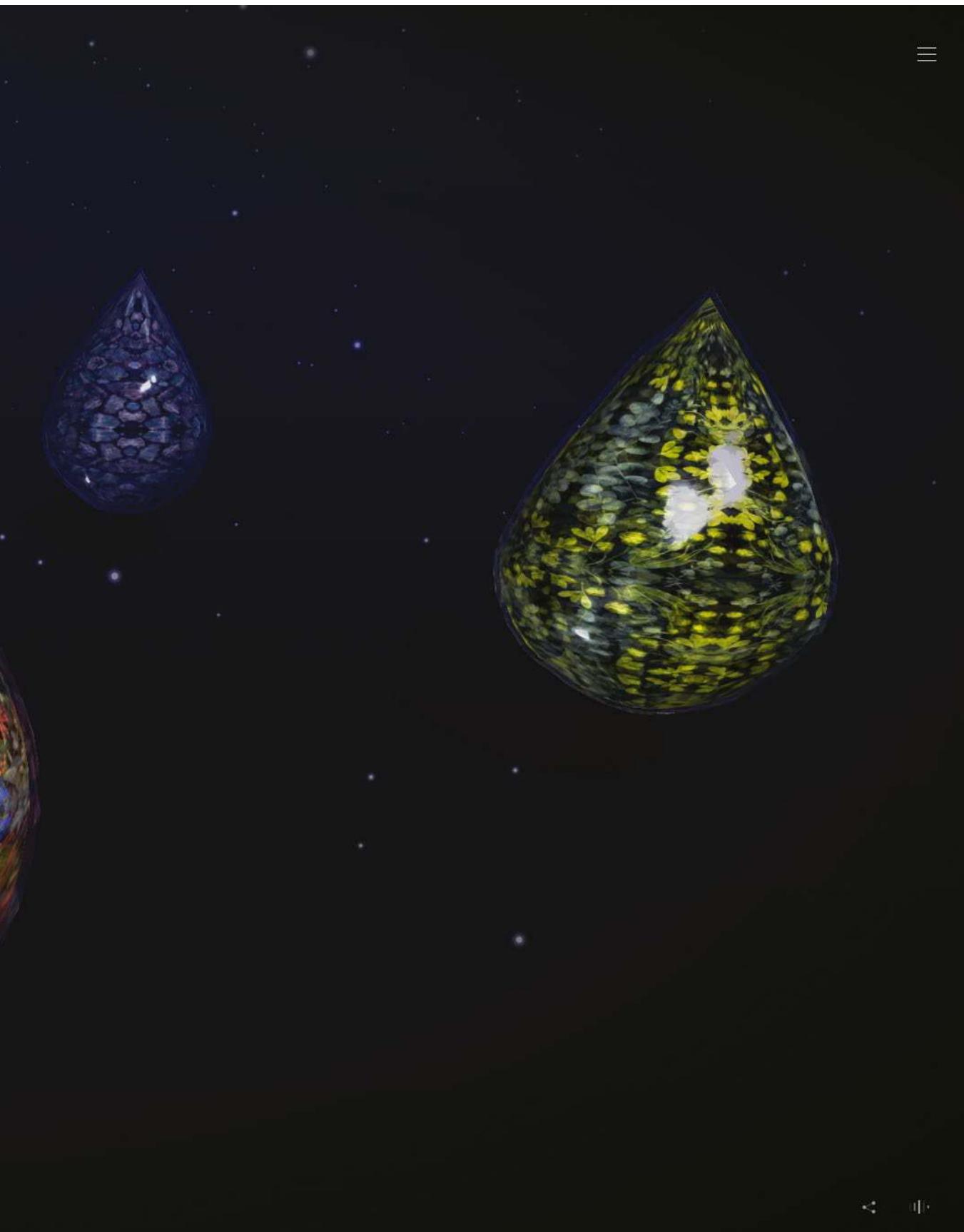
THE FIVE  
SEASONS  
HOME FRAGRANCE



## Designers:

Triboo <http://triboo.com>   Monogrid [www.mono-grid.com/en](http://www.mono-grid.com/en)

“This engaging, 360-degree interactive experience uses WebGL technologies to digitally reimagine Alessi's fragrances collaboration with Marcel Wanders”



## Colours



#1E1945



#7F5555



#ACAF4E



#D55452

## Tools

Three.js, GLSL Shaders,  
Vue.js, CSS3

## Fonts

abcABC  
1234567890

Gotham, by Tobias Frere-Jones and published by Hoefler & Co, is used exclusively in Book variety to style all site text.



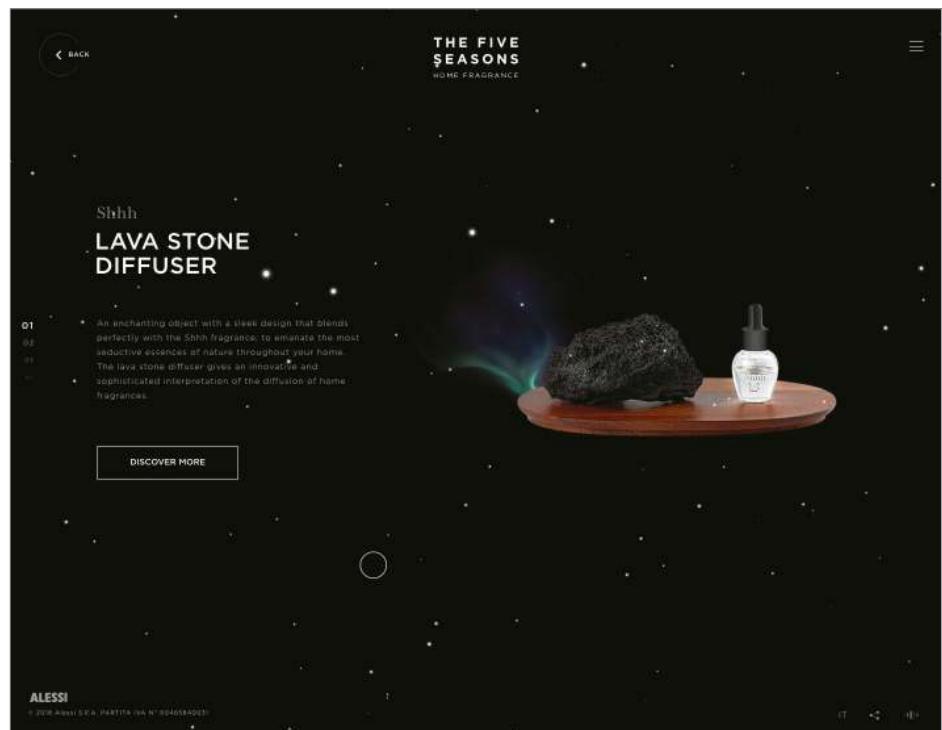
**Above**

Select each fragrance droplet from the main menu and unleash a sense of each scent by manipulating the virtual flowers



**Above**

Designed for consistency across desktop, tablet and mobile platforms, the various WebGL interactions react accordingly to touch and gyroscopic



**Above**

The audacious front-end experience is backed by Alessi shop links for discovering further details and purchasing products in the range

# Create an animated mouse hover text ripple effect

Use this animated ripple effect as a fancy design element to indicate interactivity

## 1. Document initiation

Initiate the webpage with the HTML document definition. This consists of the HTML document container, which stores the head and body sections. While the head section is used to load the external CSS and JavaScript resources, the body is used in step 2 for the visible page content elements.

```
<!DOCTYPE html>
<html>
<head>
<title>Ripple Effect</title>
<link rel="stylesheet" type="text/css"
      href="styles.css" />
<script src="code.js"></script>
</head>
<body>
*** STEP 2 HERE
</body>
</html>
```

## 2. HTML content

The webpage content consists of the text element that the ripple effect is to be applied to. This example uses article containers, but any element type can be used as long as the 'ripple' class is applied. This class will be used by JavaScript and CSS to apply the effect.

```
<article class="ripple">
  Some text goes here...
</article>
<article class="ripple">
  Another text goes here...
</article>
```

## 3. Text manipulation

Create a new file called 'code.js'. The code in this file waits for page loading to complete, upon which filling all containers using the 'ripple' class. This step changes the HTML inside these containers so that each letter of the text is wrapped with a 'span' element. It's an important feature to allow the individual letters to be animated via CSS!

```
window.addEventListener("load", function()
{
  var nodes = document.querySelectorAll(".ripple");
  for(var i=0; i<nodes.length; i++){
    var letters = nodes[i].innerText.
      split('').join('</span><span>');
    letters = letters.split(' ');
    join('&nbsp;');
    nodes[i].innerHTML =
```

```
"<span>" + letters + "</span>";
*** STEP 4 HERE
}
});
```

## 4. Animation delay

The effect requires each letter to have a slightly longer animation delay than the previous letter. Instead of manually writing this within the CSS, JavaScript is used to automatically calculate and apply this style to each letter based on their position. The calculation is performed as a 10th of each letter's index position.

```
var children = nodes[i].childNodes;
for(var n=0; n<children.length; n++){
  children[n].style.animationDelay =
  (n/10)+"s";
}
```

## 5. Stylesheet initiation

Create a new file called 'styles.css' - this is the CSS stylesheet. The first style applied is to set the page body background to black, with the text aligned centrally. The black background is used for this example to allow the blur effect to stand out; feel free to change these settings for your own project.

```
body {
  background: black;
  text-align: center;
  font-size: .5cm;
}
```

## 6. Default stylings

This step defines default styling of letters inside the ripple container. A clear text shadow is applied - ie zero blur and set at the desired white colour. The text colour is set to transparent because only the shadow is required for display. Items are set to use an 'inline-block' display so that their text flow can appear as normal text elements.

```
.ripple > *{
  display: inline-block;
  text-shadow: 0 0 0 #fff;
  color: transparent;
}
```

## 7. Hover effect

Each letter inside the ripple container is to have the 'rippleEffect' animation applied to them - but only when the ripple container is hovered by the mouse pointer. The animation is applied over a duration of five seconds and will repeat infinitely for the duration of the mouse hovering over the ripple container.

```
.ripple:hover > * {
  animation: rippleEffect 5s infinite;
}
```

## 8. Ripple effect animation

The ripple animation is made from three keyframes that the browser will manage transitions towards. The first occurs at 40 per cent of the way through the animation, which sets the individual letters to semi-transparent. The second keyframe at 50 per cent through the animation moves and resizes the letter element, while setting blurry visibility on the text.

```
@keyframes rippleEffect {
  40% {
    opacity: .35;
  }
  50% {
    transform: translate3d(.5em, 0, 0)
    scale(1.1);
    text-shadow: 0 0 20px #fff;
  }
  *** STEP 8 HERE
}
```

## 9. Animation finalisation

The final frame occurs 75 per cent through the animation. Its purpose is to animate the letters back into view. This means moving items back to their original position, eliminating the blurriness of the text shadow, and returning transparency to full visibility. The remaining time after this keyframe is used to pause the visibility of the text.

```
75% {
  transform: translate3d(0, 0, 0)
  scale(1);
  text-shadow: 0 0 0 #fff;
  opacity: 1;
}
***
```



# GDPR – The Grandmaster Challenge



[www.ogury-gdpr.com](http://www.ogury-gdpr.com)

Designers:

Ogury <https://www.ogury.com>

Mashvp <https://mashvp.com>

**“Built around a multiple-choice test, this distinctive site quizzes visitors on the EU’s General Data Protection Regulation (GDPR) laws”**

# The Grandmaster Challenge.

Colours



Tools

SVG, GSAP, PixiJS

Fonts

abcABC  
1234567890  
abcABC  
1234567890  
**abcABC**  
**1234567890**

*The Favorit font designed by Johannes Breyer and Fabian Harb at DINAMO Typefaces is used predominantly across Light, Regular and Bold weights.*

**abcABC**  
**1234567890**  
*Sharp Grotesk Bold 20 by Lucas Sharp makes an appearance within the GDPR logo and countdown sequences.*

# 01

COMPANY X HAS AN ANNUAL REVENUE OF €10M.  
WHAT IS THE MAXIMUM FINE THEY COULD  
RECEIVE, FOR NOT BEING GDPR COMPLIANT?

A.

€400,000

B.

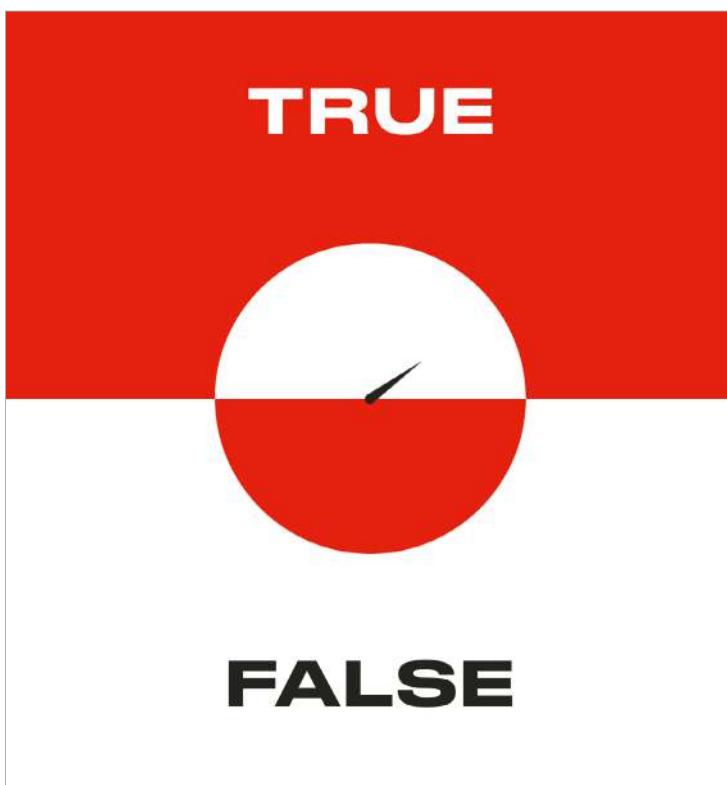
€1M

C.

€20M

**Above**

Not afraid of negative white space, the design poses a series of GDPR questions and answers to click through



**Above**

Answers are greeted by a set of amusing True or False screens utilising effective nuggets of GSAP animation



**Above**

A big red screen congratulates completion, inviting participants to sign up for email scores or share the quiz socially

# Create an animated sliding letter box effect

Introduce titles and headline text content with an animation effect that draws attention and focus

## 1. Page document initiation

The webpage is defined with HTML, which must define the document structure. This consists of the HTML container, which stores the head and body sections. While the head section is used to load the external CSS and JavaScript, the body section is used to store the HTML created in step 2.

```
<!DOCTYPE html>
<html>
<head>
<title>Slide Effect</title>
<link rel="stylesheet" type="text/css"
      href="styles.css" />
<script src="code.js"></script>
</head>
<body>
*** STEP 2 HERE
</body>
</html>
```

## 2. HTML content

The page content consists of text containers for each group of text that the animation is to be applied to. Any element can be used as the container; the most important feature is the application of the 'slide' class. This will be used by JavaScript and CSS to apply the effect.

```
<article class="slide">
  Witness
</article>
<article class="slide">
  The Power
</article>
<article class="slide">
  Of this Example...
</article>
```

## 3. JavaScript manipulation

Create a new file called 'code.js'. The JavaScript must find the containers that the 'slide' class has been applied to – then update their content so that each letter is wrapped inside a 'span' element. This functionality must be executed after the page has loaded – hence the code being placed inside a 'load' event listener attached to the browser window.

```
window.addEventListener("load", function()
{
  var nodes = document.querySelectorAll(".slide");
  for(var i=0; i<nodes.length; i++){
    var letters = nodes[i].innerText.
```

```
split('').join('</span><span>');
  letters = letters.split(' ');
  join('&nbsp;');
  nodes[i].innerHTML =
  "<span>" + letters + "</span>";
  *** STEP 5 HERE
});
```

## 4. Unique positioning

Each character inside the container needs to have unique styling so that they can move towards their target position. They also need to use a 'z-index' that positions them under the previous letter, which is calculated via the JavaScript. This code allows the effect to be applied without restriction to the number of characters used.

```
var children = nodes[i].childNodes;
var x = 0;
for(var n=0; n<children.length; n++){
  children[n].style.left = x+"px";
  children[n].style.zIndex = letters.
length-n;
  x += children[n].offsetWidth; }
```

## 5. CSS initiation

Create a new file called 'styles.css'. This step initiates the CSS stylesheet by defining the page body with a black background colour. The containers using the 'slide' class are also defined with relative positioning so that text elements can be positioned in relation to their location. A minimum height is applied to avoid obstruction between the different containers.

```
body {
  background: black; }
.slide{
  display: block;
```

```
position: relative;
font-size: 3em;
min-height: 1.5em; }
```

## 6. Slide letters

Each child inside the slide container is the 'span' element containing individual letters of the text; created in step 3's JavaScript. These elements are positioned in the top-left corner of their parent container – defined in step 5. A transition is applied to the 'left' attribute that will trigger an animation when change is applied via the JavaScript from step 4.

```
.slide > *{
  position: absolute;
  display: inline-block;
  top: 0;
  left: 0;
  background: red;
  color: #fff;
  border: .1cm solid #000;
  margin: 0;
  transition: left 3s; }
```

## 7. Letter colouring

A rule is applied using some algebra within the nth-child selector so that individual letters can be updated with unique colours. The 3n means select the rule for every 3rd element, while the +1, +2 and +3 will move the selector by 1, 2 or 3 elements to the right of the 3rd element.

```
.slide > *:nth-child(3n+1){ background:
red; }
.slide > *:nth-child(3n+2){ background:
green; }
.slide > *:nth-child(3n+3){ background:
blue; }
```



# #Art4GlobalGoals

<https://art4globalgoals.com/en>

A SPECIAL EXHIBITION WITH

LEON LÖVENTRAUT

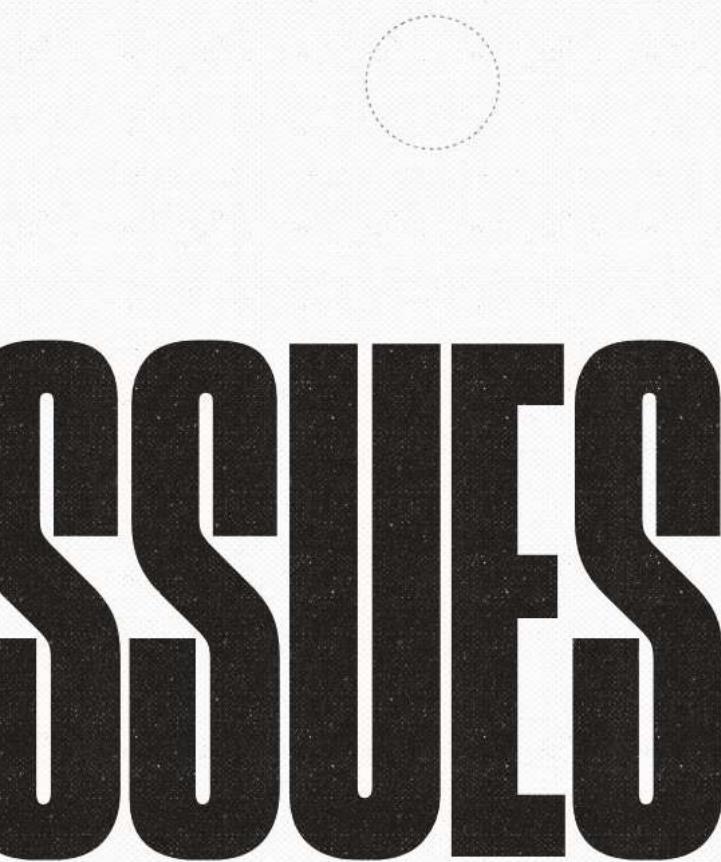


Brush the  
of these

Designers:

Denkwerk [www.denkwerk.com](http://www.denkwerk.com)

Polyclick <https://polyclick.io>



ISSUES

through and erase the 17 issues that lay at the heart  
goals.

**“**Creative humanitarian website that blends bold typographic work and brushstroke interactions with the vibrant work of artist Leo Lowentraut.  
**”**

Colours



Tools

HTML5, PHP, TypeKit

Fonts

**abcABC**  
**1234567890**

Druk by Berton Hasebe is used in its Condensed X Super variety to add a distinctive bold contrast to headings.

abcABC  
1234567890

abcABC  
1234567890

**abcABC**  
**1234567890**

Adrianna font, available via TypeKit is employed across a myriad of typefaces including Light, Regular and Extended Bold.

# Create an animated random marker dash effect

An effect to apply the effect of marker pen dashes to any defined area(s) of web pages

## 1. Page initiation

The first step is to initiate the web page definition. This is made from markup describing the HTML page container. There are two sections that the document container stores. The head is used to load the Javascript and CSS, while the body section stores visible content that will be defined in the following step.

```
<!DOCTYPE html>
<html>
<head>
<title>Marker Dash</title>
<link rel="stylesheet" type="text/css"
      href="styles.css" />
<script src="code.js"></script>
</head>
<body>
*** STEP 2 HERE
</body>
</html>
```

## 2. Body content

The body content consists of a container that the effect is applied to. This container stores any visible text and has the "dashEffect" class. It is this class that will be used to allow the Javascript and CSS to find where to apply the effect. You can apply the effect to multiple elements by applying the "dashEffect" class.

```
<article class="dashEffect">
  <h1>Hello...</h1>
</article>
```

## 3. Container search

Create a new file called "code.js". Page content is only available after the page has completed loading, hence code must be placed inside an event listener that waits for this. The focus of this step is to find all containers using the "dashEffect" class, upon which the creation of the "marker" effect elements are initiated. This is achieved by setting the loop to repeat up to the amount defined by the count variable - i.e. 5.

```
window.addEventListener("load", function()
{
  var parents = document.querySelectorAll(".dashEffect");
  var count = 5;
  for(var c=0; c<parents.length; c++){
    var delay = 0;
    for(var i=0; i<count; i++){
      *** STEP 4 HERE
    }
  }
});
```

## 4. Marker blocks

Each consists of the creation of a container element for the marker effect - created as a div element. A loop is used to define the HTML for 5 child span elements; each having an opacity calculated using their index "n" position from the loop. This HTML is applied to the parent node, along with the class name "dash" being applied so that the parent can be referenced from the CSS.

```
var node = document.createElement("div");
var html = "";
for(var n=0; n<5; n++){
  html += "<span style='opacity: "+(1/(n+1))+"></span>";
}
node.innerHTML = html;
node.className = "dash";
*** STEP 5 HERE
```

## 5. More dash node styling

Additional styles are applied to the parent dash node. These styles position the dash in unique positions and rotations, avoiding the need for the CSS or HTML to contain bloated markup. A unique animation delay is also applied; defined in step 4 and incremented by one second at the end of this step.

```
node.style.left = Math.floor(Math.random()*parents[c].offsetWidth)+"px";
node.style.top = Math.floor(Math.random()*parents[c].offsetHeight)+"px";
node.style.height = parents[c].childNodes.length+"em";
node.style.transform =
"rotate("+((i*60)+"deg"));
node.style.transformOrigin = "50% 0%";
node.style.animationDelay = delay+"s";
parents[c].appendChild(node);
delay++;
```

## 6. CSS initiation

Create a new file called "styles.css". This step starts the stylesheet by defining the page to have a black content and white applied as the default text colour. For this example's visibility of the effect, articles are set to cover the full screen height. You can set the size of the effect containers to any size required by your project.

```
body {
  background: black;
  color: #fff;
}
article{
  min-height: 100vh; }
```

## 7. Dash parents

Elements within the "dashEffect" containers that have the "dash" class are the parents of the visible dash marker elements. These need to display with absolute positioning so that they appear at the random positions defined in step 5. Every dash parent shares the same width and opacity settings - set to invisible by default. The "dashAppear" animation is applied to infinitely "introduce" the marker over a duration of five.

```
.dashEffect > .dash{
  position: absolute;
  display: block;
  width: 1.5em;
  z-index: -1;
  opacity: 0;
  overflow: hidden;
  animation: dashAppear 5s infinite; }
```

## 8. Dash children

The elements placed inside the dash parent are displayed with a red background colour and are set to appear at a specific height. They are set to display as a block with 100% width to guarantee that they appear stacked on top of each other within their parent dash container.

```
.dashEffect > .dash > *{
  display: block;
  width: 100%;
  height: 1em;
  background: red; }
```

## 9. Appearance animation

This step defines the animation applied to the dash parent in step 7. The animation introduces the dash with full visibility, but with no height. It is quickly animated to a visible height at 10% through the animation. Opacity is more slowly animated to invisible at the end of the animation.

```
@keyframes dashAppear{
  from{ height: 0; opacity: 1; }
  10%{ height: 10em; }
  to{ opacity: 0; } }
```



# generate

The conference for web designers

## LONDON

19-21 SEPTEMBER 2018

3 DAYS OF INSIGHT AND INSPIRATION

TICKETS ON SALE NOW

[www.generateconf.com](http://www.generateconf.com)  
**#generateconf**

BROUGHT TO YOU BY   The voice of web design 



# CHASING RAINBOWS

WITH HIGH-ROLLING  
CLIENT MGM RESORTS ON  
A MISSION TO SING OUT A  
MESSAGE OF INCLUSION  
TO THE LGBTQ COMMUNITY,  
COULD McCANN NEW YORK  
AND YOUR MAJESTY MAKE  
UNIVERSAL LOVE  
RESONATE ONLINE?

## Universal Love

[universal-love.com](http://universal-love.com)

by

### Your Majesty

[yourmajesty.co](http://yourmajesty.co)  
[@yourmajestyco](http://@yourmajestyco)

#### PROJECT DURATION

12 weeks

#### PEOPLE INVOLVED



**Georgios Athanasiadis**  
Project Director



**Sheldon Lotter**  
Art Director



**Lotte Peters**  
Art Director



**Kasper Kuijpers**  
Technical Lead



**Hugo Wiledal**  
Technical Lead



**Michael Hoang**  
Senior Producer



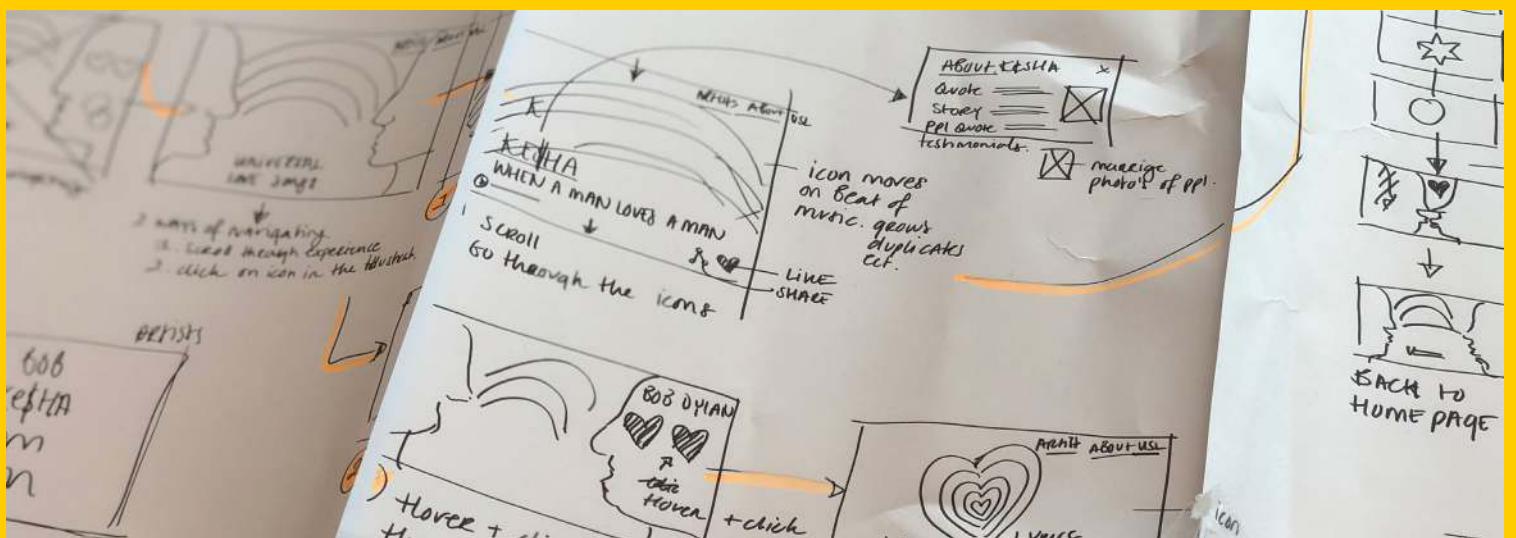
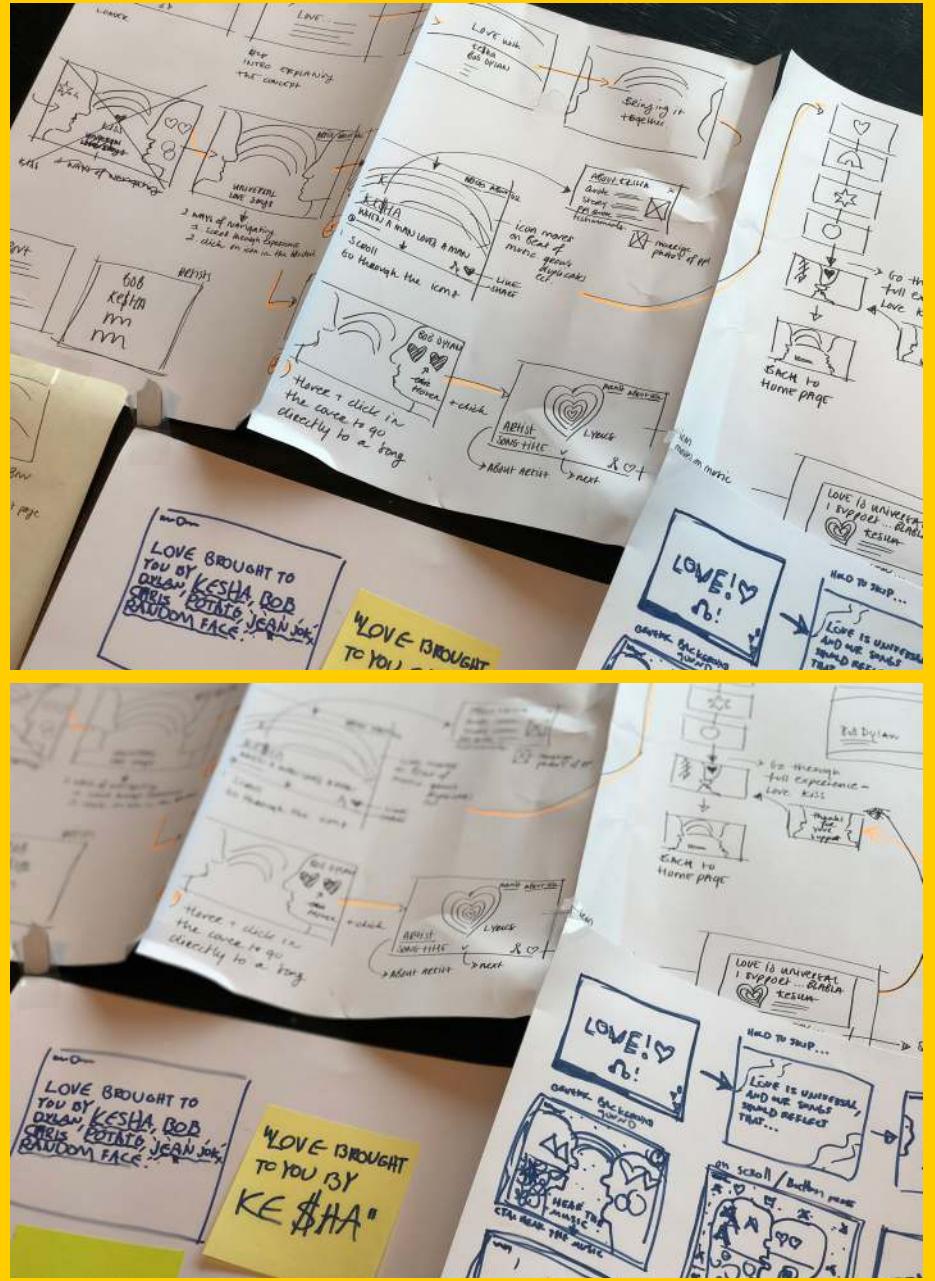
**Anna Benson**  
Associate Producer

FOR MANY WHO STILL BELIEVE in the founding conceptual pillars upon which the internet was built on, a big word is freedom. The 'superhighway' of inter-linked URLs and, at the time, pretty crude HTML documents would set a precedent for something open and diplomatic. Everyone could find himself or herself a presence, not just to visit but grow the rapidly expanding world with voices unrestricted by any kind of policing or formal restriction. Some of those

liberties may have been traded surreptitiously since, but what does remain is a strong platform for collectiveness that has the power to influence real-world thinking. As the politics of division threaten to keep us apart, the web proves furtive ground for acceptance, embracing change and certainly bringing people together. This is really the goal message behind Universal Love, a musical project celebrating the LGBTQ community with a collection of reimagined wedding songs.

# UNIVERSALLY SPEAKING

Perhaps the biggest initial hurdle for Universal Love was settling on a visual theme that would adequately live up to the vibrant music at the project's heart. An undoubted advantage here, however, was the ability to reference the album's already commissioned artwork, with the Your Majesty designers able to use it as a touchstone for setting their own aesthetic aims. "Given the vibrant cover artwork from Craig and Karl as inspiration, the possibilities for creative expression were endless," starts Senior Producer Michael Hoang. "In order to ensure the platform achieves its monumental mission, we established an execution strategy that aims to balance both communication and emotional objectives." This formulation of an 'execution strategy' would set out some rule-based goals for what the visual concept for the website should fulfil. Firstly here, the team decided that it had to be 'celebratory' to mark the fact that Universal Love was signposting a momentous time for perceptions around love. Secondly, it needed to be 'loud' enough and confidently bold in terms of supporting a revolution, while thirdly being 'exciting' in the way that love in all its forms is fun and sensational. With this framework in mind, the team could then undertake a conceptual phase during which every member sketched out various concepts for bringing the strategy to life. "After a few iterations and consolidations of ideas, we decided that a natural channel to transmit the music was through the album's artwork, particularly the rainbow at its centre, which reflects the colour sequence of the LGBTQ pride flag," explains Art Director Sheldon Lotter. "Our aim was to infuse feelings of festivity, joy and love into every aspect of the website."



Born out of collaboration between MGM Resorts – the largest hotel brand in Las Vegas, Nevada – and marketing agency McCann New York, the concept was to have leading artists put a new spin on iconic love songs in the name of equality. “Each of the six updated love songs gives same-sex couples a new soundtrack for their own love stories and feature pronouns changed to reflect the world of LGBTQ relationships,” explains Your Majesty, the stellar agency commissioned to deliver Universal Love online. “McCann New York and MGM Resorts partnered with us to design and build a digital stage for the Universal Love project.” And so they might, with the high-profile design and technology firm keeping a client book in keeping with the recording stars involved. Your Majesty’s award-winning work for the likes of Netflix, Samsung, Adidas, BMW, Spotify, Bentley, American Express and Red Bull now sits side-by-side with a beautiful website featuring Bob Dylan, Kesha and St. Vincent. A rainbow-themed jukebox with an inspiring message, we discover from the folks involved how UniversalLove.com received the creative care it deserved.

## UPENDING NORMS

Universal Love as a broader project really starts with MGM Resorts and, of course, its place in the global perception of its undisputed home. As a hugely successful leisure and events presence in Las Vegas, they play a significant role in establishing the city as “wedding capital of the world” for many. However, for whatever reason, this hasn’t always necessarily been the perception for those in the lesbian, gay, bisexual, transsexual or ‘questioning’ community. “With a long history of support for the LGBTQ community, MGM Resorts was on a mission to spread a message of inclusion, showing that [Las] Vegas was truly welcoming to people of any sexual orientation,” begins Senior Producer at Your Majesty, Michael Hoang. “In order to spread the message of inclusion, MGM Resorts teamed up with McCann New York to create an album that reimagines classic wedding songs, flipping the pronouns to upend norms and challenge conventional notions of love songs, which have traditionally been written from a heterosexual point of view. With the project’s grand purpose, the updated but timeless songs, and the cover artwork from Craig and Karl, McCann New York came to us to build a space to host and amplify this revolution.” This potential ‘space’ would, of



**ABOVE**  
Get more details of an artist via their bio

**RIGHT**  
The user experience can be enjoyed on all devices



course, need to fulfil a dual purpose of putting the Universal Love message up front, while providing a robust enough platform for serving the music appropriately. Songs such as Bob Dylan’s “He’s Funny That Way”, “I Need a Woman to Love Me” by Kesha and St. Vincent’s rendition of “And Then She Kissed Me” say so much in themselves, but the challenge was presenting the audio in a striking context.

## CIVIL PARTNERSHIP

The conceptual work to find the best direction for the site’s presentation undoubtedly demanded close collaboration. This project really was about fulfilling the overarching partnership between MGM Resorts and McCann New York, and the brief the latter had for marketing it. “In our case, we primarily

interfaced with McCann New York as the lead creative agency of the project,” Hoang continues. “At Your Majesty, we are not afraid of showing work in progress and prefer to work with our partners rather than for our partners. As a result, we and McCann roughly maintained a bi-weekly creative check-in process.” While the arrangement and scheduling of these kinds of meetings might change, it remains a common key thread to the most successful work this industry produces. In this instance, the Your Majesty and McCann guys would view regular consultations together as both informal brainstorming sessions, and as more structured chances to find common ground on the choices being made. “We treat these check-ins not as review points, but rather opportunities for both teams to bounce ideas off of each other and to ensure creative and communication alignment between what we are designing and building with the rest of the campaign’s touchpoints.” This strong communication work laid the foundations as work moved into a conceptual phase and an ‘execution strategy’ for realising a design.

**“MGM RESORTS WAS ON A MISSION TO SPREAD A MESSAGE OF INCLUSION”**

### COVER VERSION

Such an execution strategy would set out certain rules and provide a framework for inspiring visual sketches that referenced the album's vibrant cover artwork. Produced by artists and illustrators Craig Redman and Karl Maier, otherwise known as Craig & Karl ([craigandkarl.com](http://craigandkarl.com)), the cover itself features the familiar LGBTQ rainbow motif to arrive at a striking design that Your Majesty wanted to reflect. "The key challenge was then to bring the album cover art to life, making it feel vibrant and fun," confirms Technical Lead, Hugo Wiledal. "As with any project, it is really a dialogue back and forth between design and development, where each role can inform and challenge the other, especially when it comes to animation and interactivity. After some initial exploration we landed on the idea of starting with the album cover and then taking the user

of subtle effects and transitions. "The graphical elements of the site inspired lots of playfulness to be explored, and we wanted to take full advantage of that without losing typographical fidelity. With WebGL we can create wildly visually interesting things, but HTML and CSS has always been the best when it comes to native scrolling and making things responsive. We ended up with a combination, leveraging WebGL for the highly interactive parts — such as the elastic rainbow background that bends and bounces as you scroll — and HTML with CSS for all the text content and clickable areas."

### LOVE SPREADS

Jumping forwards to the climax of the project's 12-week production time, and the efforts would naturally shift to marketing Universal Love's campaign. This phase for the new website would see things shift back from Your Majesty

## "WE ENDED UP WITH A COMBINATION, LEVERAGING WEBGL FOR THE HIGHLY INTERACTIVE PARTS"

to a zoomed-in view of the rainbow element, an isolated part of the larger artwork. Thanks to this, we were allowed to bring the shapes to life without having to worry about the performance issues that come with rich graphics. The elastic motion you scroll really lends a feeling to the whole album artwork. We utilized specially prepared SVGs containing hidden elements that we could use for tracking positional data in the frontend. This allowed designers to influence code without having to go through developers."

### CONSTRUCTIVE COUPLING

Given that the Universal Love website would support a fairly straightforward function, the team really could allow the frontend work to be the focus. With any heavier backend development largely unnecessary, it was also decided that a more streamlined build would enable the performance to pop even during times of high demand. "Due to the static nature of the project, and the expectation of high traffic, we opted for a backend-less project structure," explains Wiledal. "Where a single-page app was hosted directly on a CDN for maximum capacity." The coding challenges would, therefore, revolve much more around getting the surface technology choices right for achieving the frontend look and feel. Visitors to the final live site will know that Universal Love's striking appearance and purposefully colourful theme also incorporates a myriad

and over to McCann New York, in terms of directing audiences to the album. In this respect, rather than creating a launch campaign consisting of traditional advertising, the songs on the album would become the centrepiece of an ecosystem as the music spread to social, digital and retail platforms. The project's eye and ear-catching new website would enjoy an important role across all three of course, providing an insightful listening booth for sampling the standout songs.

Recorded and then launched simultaneously on Universal-Love.com as well as on every major music streaming platform including Spotify, iTunes, Apple Music and Pandora, the album generated considerable buzz within local, national and international news. "The team hooked up with our artists to spread the message in as many touchpoints as possible. They all — minus Bob Dylan who doesn't use social media — posted it on their social media channels at launch. Week two then saw the release of a music video by Kesha, where she marries a lesbian couple at an MGM hotel in [Las] Vegas. In Week three, we hit late night, with Benjamin Gibbard performing "And I Love Him" on Conan. In Week four, 10,000 vinyl records were sold in stores across the country for Record Store Day. Lastly, She and Him wrote the first original song inspired by the project and released it in two versions to celebrate universal love."

With music by:

**Bob Dylan**  
**St. Vincent**  
**Kele Okereke**  
**Valerie June**  
**Benjamin Gibbard**  
**Kesha**

ABOVE  
Artists give their rendition of classic songs

BELOW  
Dynamic elastic rainbow bends add engagement

UNIVERSAL LOVE

MGM RESORTS

Ke

## SITE HIGHLIGHT

Art Director on the project Sheldon Lotter gives his honest heartfelt reaction to what aspect of the experience draws the most universal love from the Your Majesty crew



**“With small but thought-through interactive details like the ability to listen to love itself, an explosion of colour when browsing through the rainbow and the hearts bursting with elation, Universal-Love.com invites all visitors to take part in and celebrate this momentous moment of love for all”**



# SUBSCRIBE

TODAY AND RECEIVE AN  
**ECLIPSE CHARGER**



- **Charge multiple devices at maximum speed**  
3-port USB charging station, featuring Smart IC technology, ensuring safe, high-speed charging.

- **Stylish design**  
Has a capacitive touch sensor and emits an ambient light, to help you find your cables in the dark with ease.

- **Precise cable management**  
Holds up to 20 feet of cables, neatly wound around its centre for clutter free and untangled charging.

**IT'S EASY TO SUBSCRIBE**  
**myfavouritemagazines.co.uk/WEBE**

OR CALL: 0344 848 2852

**SUBSCRIBING FROM OUTSIDE THE UK? HEAD TO [BIT.LY/2UBG4NA](https://bit.ly/2UBG4NA)**

**PLUS** with your subscription  
you have access to  
**900+ FREE RESOURCES**  
from previous issues  
with FileSilo. See page 96



**GIFT  
WORTH  
£69.99**

## CHOOSE YOUR PACKAGE!\*

**PRINT EDITION ONLY**



6 months of **Web Designer** in print, plus **Eclipse Charger**

**DIGITAL EDITION**



6 months of **Web Designer** on your iOS or Android device

**PRINT & DIGITAL BUNDLE**



6 months of **Web Designer** in print and digital, plus **Eclipse Charger**

\* OFFER NOT AVAILABLE FOR DIGITAL EDITION ONLY PACKAGE

**Terms and conditions:** This offer entitles new UK Direct Debit subscribers to pay just £34 every 6 months plus receive a n Eclipse Charger worth £69.99. Gift is only available for new UK subscribers. Gift is subject to availability. Please allow up to 60 days for the delivery of your gift. In the event of stocks being exhausted we reserve the right to replace with items of similar value. Prices and savings quoted are compared to buying full-priced print issues. You will receive 13 issues in a year. Your subscription is for the minimum term specified and will expire at the end of the current term. You can write to us or call us to cancel your subscription within 14 days of purchase. Payment is non-refundable after the 14 day cancellation period unless exceptional circumstances apply. Your statutory rights are not affected. Prices correct at point of print and subject to change. UK calls will cost the same as other standard fixed line numbers (starting 01 or 02) or are included as part of any inclusive or free minutes allowances (if offered by your phone tariff). **For full terms and conditions please visit: [www.bit.ly/magterms](http://www.bit.ly/magterms). Offer ends October 2018.**

# Generation UX

On a mission to transform the experience of digital interface users, Experience UX have a deep understanding of how digital environments must speak to their users, offering unique experiences that place the user at the centre of every design





**xperience UX** is the brainchild of a psychologist and a project manager. Co-founder Damian Rees, graduated university with a

degree in Applied Psychology and spent his early years working in UX including a stint at the BBC and National Air Traffic Control.

Whilst at the BBC he stepped in for the Head of UX becoming the lynchpin co-ordinating all user research for the organisation, where he had a lightbulb moment. "I realised that none of these high profile UX agencies were asking the right questions or delivering any effective solutions to the problems they we're uncovering in their research," Damian explains.



### Who Experience UX

#### What Usability Testing, User Experience, Interaction Design, and User Research

#### Where 4th Floor, Avalon, 26-32 Oxford Road, Bournemouth, BH8 8EZ

Web [experienceux.co.uk](http://experienceux.co.uk)

#### Key Clients

Burberry

McCarthy & Stone

The Open University

The Body Shop

The Financial Times

"I knew there was far a better way to design great experiences, so I started developing a business plan. It wasn't until my path crossed with Ali (Carmichael, co-founder and Commercial Director) that the timing felt opportune to start a new business."

In the meantime, Ali had graduated with a degree in Tourism Management but discovered that his passion was actually in digital so he developed his early career in sales and project management at full-service agencies. It was at one such agency, with Ali as Head of Project Management and Damian as Head of Design, that they decided to make the leap into running their own business.

"We were both frequently frustrated that projects had key user insights missing from them, but were brought to us too far down the line to influence. It wasn't the culture back then to put user needs central to the design process and it was obvious to me that here was an opportunity. Following a team building day we went to the pub and, a few whiskies later, decided to set up a business that helped companies see the world from the perspective of their customers first and ultimately, to make life easier for people.

One area that the founders spent a lot of time on was deciding on the name of their new business, as Damian explains: "It took Ali and I ages to agree on an agency name and we found it quite a stressful process. Eventually we were born as Experience Solutions and then later changed to Experience UX, both domains (.co.uk) were bought afterwards and neither was available as .com. Clients refer to us as Experience so in our next iteration we may even drop UX altogether.

"Deciding on a name can be a difficult process but in hindsight it doesn't have to be a sticking point to launching a business. There are plenty of examples of great agencies with ordinary names and just as many with witty creative ones, so just pick something unique and run with it. Consistency of your identity and tone of voice that will help develop your brand, and the work that you deliver will speak the most."

As user experience developers their own website must be a showcase for their latest user thinking? "We've always found that our website is about 18 months behind our thinking because our strategy is constantly evolving," says Damian. "We've recently invested time in updating our website but to be honest, it's in constant evolution.

"Number one, an agency's site should be easy to use rather than a 'bells-and-whistles' example of what is possible in web development. We practice what we preach, when user research on our site led to a rebrand. You can tie yourself up in knots trying to be ▶

perfect when our research showed that, in reality, an agency website is there for credibility and that intangible, but oh-so-important 'gut feeling' of a visitor about your skills and credibility is more vital than an amazing piece of animation. We champion the role of the user in our approach to all projects, so we've worked hard to make our site more human.

"We're also passionate about sharing best practice so we publish interviews with world-class industry experts such as Jared Spool and Caroline Jarrett under our blog UX Insider. This alongside various How To Guides means we're a useful resource so we get plenty of traffic to the site."

The approach agencies take acquiring new clients can be manifold. For Experience UX they have been following a policy that enables them to gain not only the clients they want, but also to focus their business's future on a path that they have defined. Damian outlines their approach: "We are staunch advocates of Blair Enns' The Win Without Pitching Manifesto ([winwithoutpitching.com/the-manifesto](http://winwithoutpitching.com/the-manifesto)) and have a policy of no pitching and no working for free.

"We love talking to new clients about how we can help them, but we're not interested in being compared with other agencies on price and being reduced to procurement checklists. Our attitude is very much "let us show you what we can do" but we'll only invest in doing that when we feel we have good potential for a great working relationship. We get a high proportion of referrals via word of mouth and even higher percentage of work comes from existing clients who see the value of our processes and throw more of their problems at us."

Damian concludes: "In the early years we would take on small projects and/or just user research to get the foot in the door. What we quickly discovered is that companies often underestimate the importance of UX or it's a last-minute effort, so they try to rush it through. Now, we are much more selective and only get involved with clients who match our strong values and ethics. Once you have that in place it's a much more pleasurable experience and the effectiveness goes through the roof. Our clients these days tend to be large household or industry brands, with a passion for user-centred design so we're already reading off the same page."

With such a wide remit to improve the user experiences across all digital spaces, Damian outlines one project that distils their attitude to improving how everyone interacts with today's digital services.

"Our ethos has always been 'solving real problems for real people'. A recent project that epitomises that is McCarthy & Stone retirement living. They have a mature, under-represented

"We don't believe that a user journey can or should be looked at from the perspective of only one device. People interact with technology in complex ways with often a series of short interactions at different stages of their decision-making process. These journeys can take place on a variety of different devices in many different contexts"

Damian Rees, Co-founder, Director



The Experience UX team doing what they do best in their shiny new offices



audience and a customer journey that is incredibly emotional," says Damian. "We discovered so many powerful surprises through the user research and it was refreshing to have a client that embraced them without hesitation. We found that the biggest audience was in fact the influencers, namely younger family members, and

that retired people don't want to see patronising stock images of 'smiling old people'. Their customers demanded real representation and because the client listened, we were able to transform a journey that often starts with anxiety and fear into excitement and reassurance."

As a client brief can be diverse to say the least, Experience UX have developed their processes over time to enable them to work efficiently. "Every project is different but there are usually three core roles with additional support bolted on when required," Damian

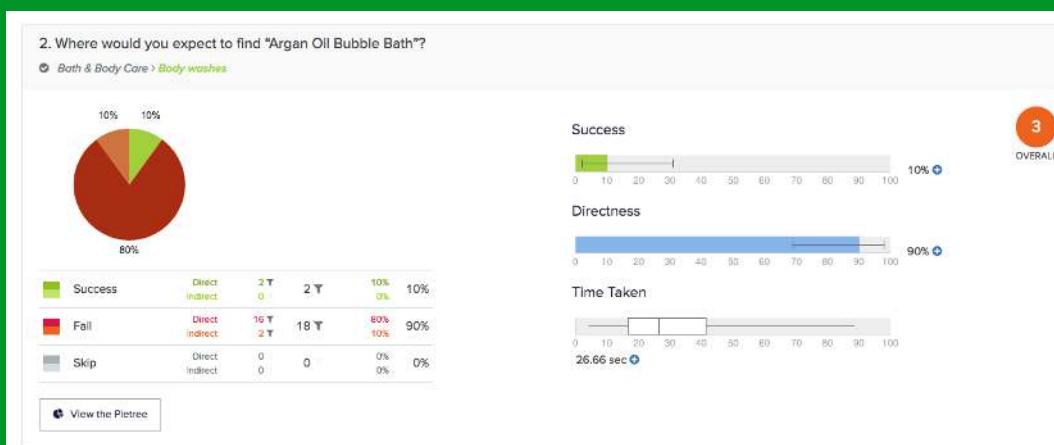
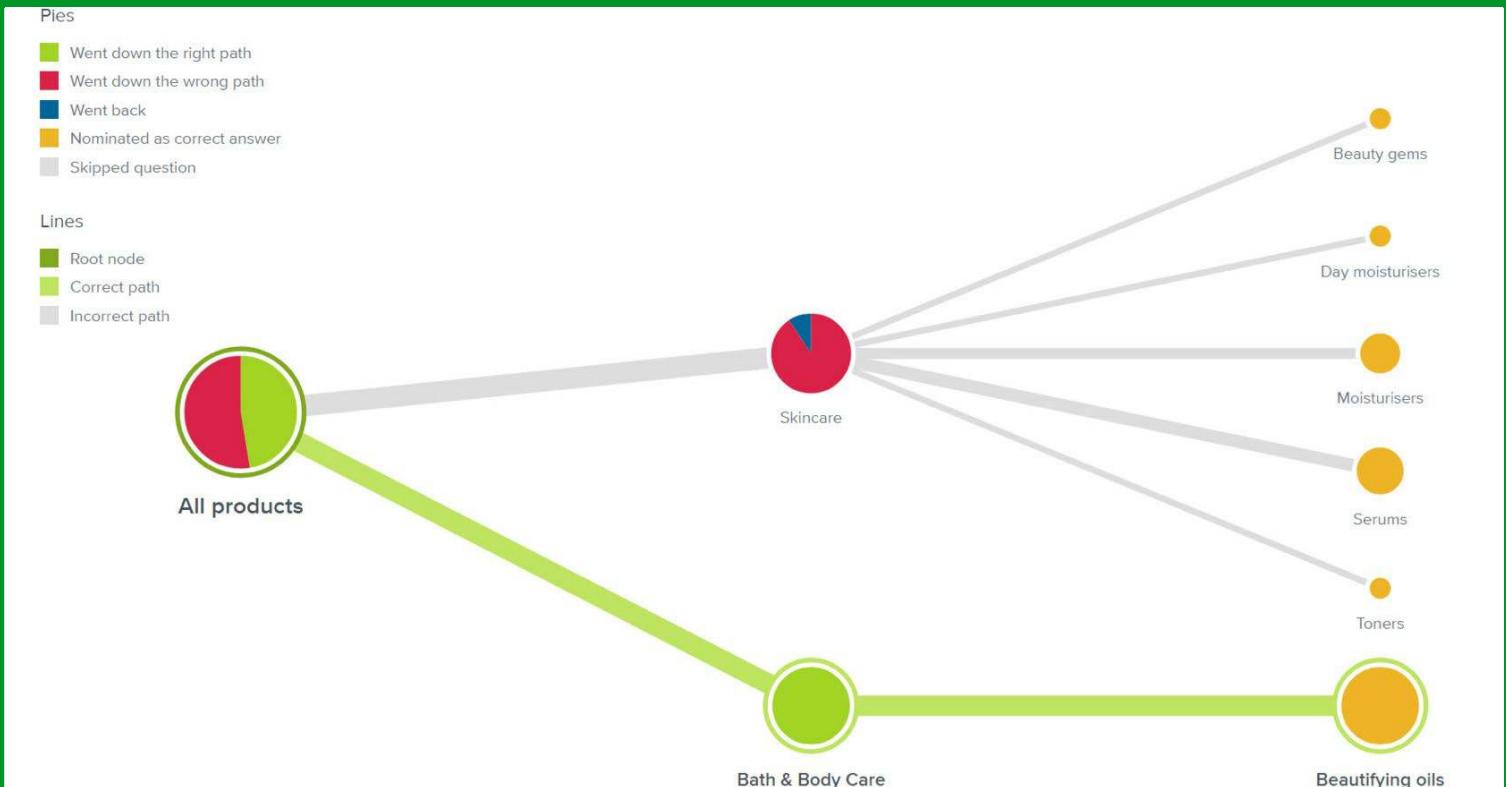
explains. "It's no surprise that the Account Director manages the relationship with the client and keeps us on track with what we are trying to achieve. The UX Lead owns the project and has the day-to-day communication with the client. They research, plan and design the project from start to finish, with support

from a UX Support who develops the user research plan, facilitates any workshops and provide sense checking. This is a critical role in the ideation phases."

Turning to the tools Experience UX use across their business, UX Consultant Laura Yarrow explains: "As we work on such a wide range of projects, we prefer to give autonomy to the UX Lead to use the tools that make sense for the project. Our go-to tools are Axure for prototyping, Float for project scheduling, and Asana for project and task management.

"We do a tonne of mind mapping to organise our thinking and research findings, and that ▶

### "Retired people don't want to see patronising stock images of 'smiling old people'"



## The Body Shop

This project was all about The Body Shop's mobile site and how it could be improved. The starting point for the mobile project was the need for The Body Shop to understand their customer better, the conversion funnels and the pain points. It was about understanding the full journey and how the 'why' complements the 'what' of analytics. What are the users' buying decisions? How can we optimise their purchasing journey?

We ran usability tests on the mobile site and worked on the company's navigation redesign project. One of the key aims of the navigation redesign was to improve the taxonomy of the mobile site.

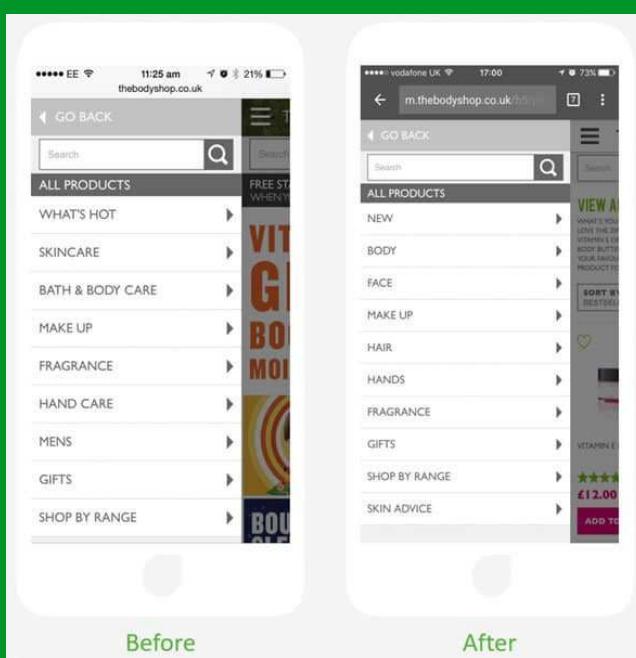
The research we conducted helped highlight which products were causing ambiguity, and where customers were getting confused or lost on their journey. The golden nugget was that skincare was not a massively searched term in Google and that people didn't seem to categorise by skincare – they went straight to body part. This was one of the key insights we took from the project that helped us rethink how we structure things. Among the biggest changes were rebranding 'Skincare' as 'Face' and adding a 'Skin Advice' category.

After the launch of the new navigation, The Body Shop saw positive uplifts across all categories. The most significant improvements were seen in the 'Face' category (rebranded from skincare) and through the introduction of the Skin Advice section. Even the early statistics were very positive, with year on year orders up by 28%.

**Top** This process showed Experience UX how people tried to find products and helped them identify issues both at the top and lower levels

**Middle** Experience UX's validation test showed their new navigation structure worked much better than the existing design

**Right** Before and after of the navigation. The new navigation was based on body parts and increased sales by 28%



## The Open University

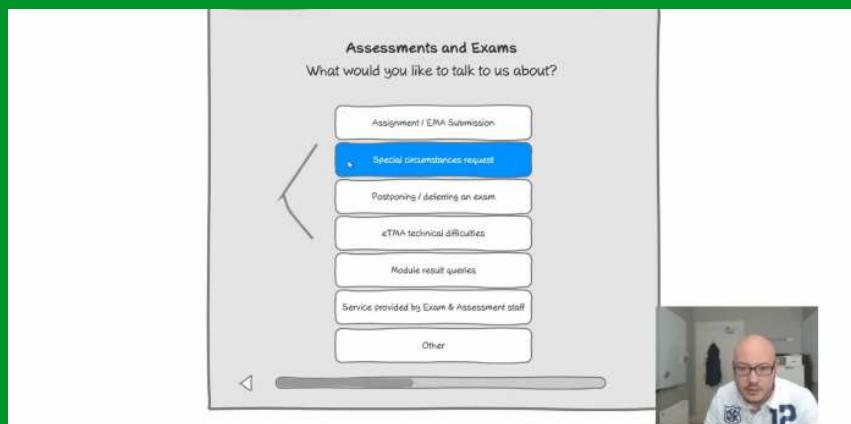
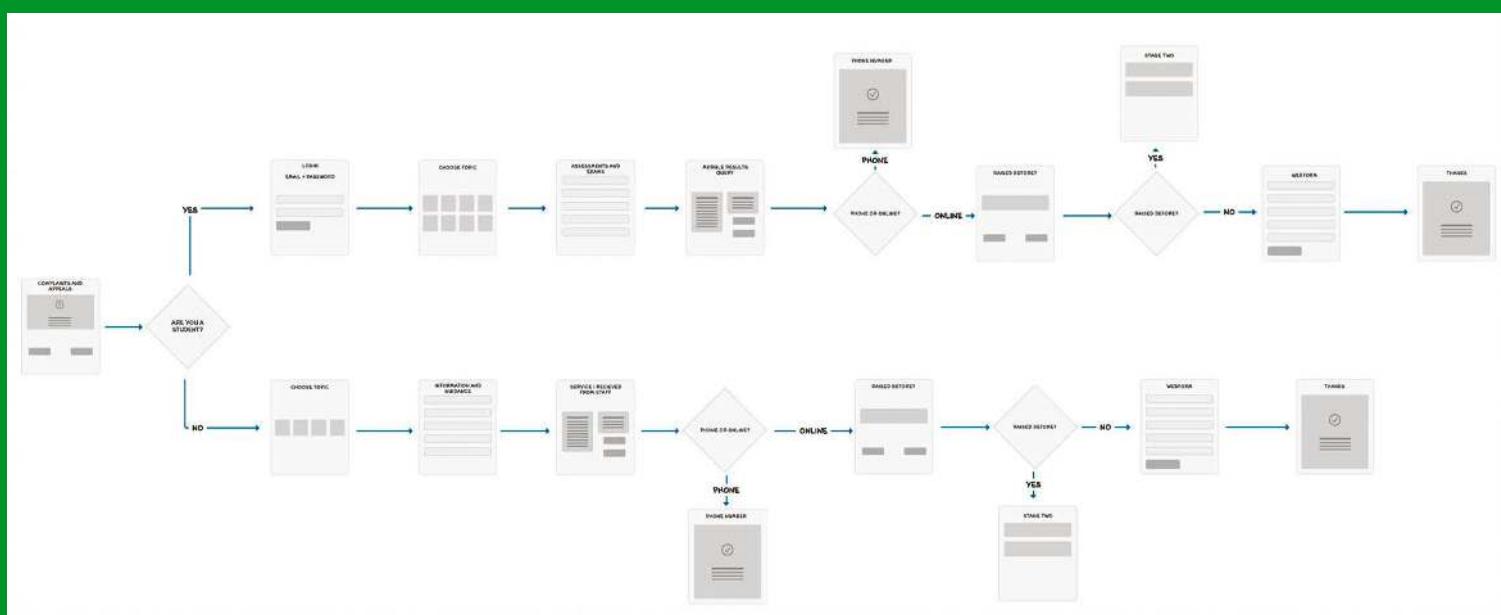
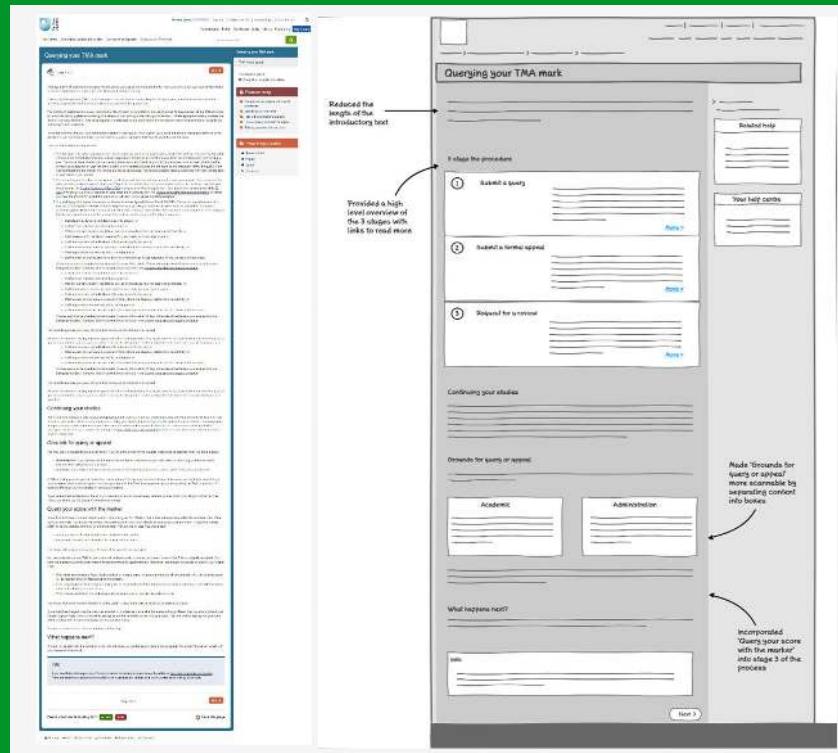
The Open University recognised that their complaints and appeals process was long-winded, confusing, and resulted in a high proportion of complaints being escalated to official complaints. We were asked to conduct a full user review of the complaints and appeals process to identify the issues and produce a clear plan for how the online system should be redesigned.

We adopted a user-centred design approach to the problem. We first understood who the users would be, what the context of use was, and then evaluated their experience before redesigning it.

We produced an audit report of best practice complaints processes by conducting a full audit of the existing process to map out every step in the customer journey. We then looked at 'best in class' complaints processes in alternative industries to establish a set of best practice principles and inspiration to draw from.

Taking on board the requirements from a variety of internal stakeholders, and the lessons from the audit in stage one, we designed an entirely new complaints process which included a fully working prototype of our proposed solution.

After presenting our proposed solution to stakeholders and taking on board their refinements, we conducted a series of usability tests with OU students (observed by the OU team). Following the tests, we refined the prototype and tested again until we were confident of the final solution. The finished prototype was handed over to internal OU design and development teams to implement.



*Top Experience UX took the long-winded information (left) and broke it down, keeping the key elements. They then looked at how they might redesign the page without graphics and within the restrictions of the CMS*

*Middle The mockup of the entire user journey using lo-fidelity prototyping (left) and the live service now (right)*

*Bottom Once the prototype was finalised in terms of content and structure, Experience UX did more research with OU students to further validate the process. They conducted 14 usability tests of the prototype before delivering their finalised version*

tends to be in XMind, although there are web-based alternatives such as Coggle if we need to collaborate with other team members. We also rely heavily on Slack for communications and Dropbox to organise and back up our work."

Laura also advises: "Sometimes sticking to a specific tool can be unnecessarily restrictive, so it's worth going back to basics with good old-fashioned pen, paper and Post-it Notes, as this can often be the most distraction-free way of working. If we're running user-testing we have a variety of tools and equipment to run the sessions. We use Morae Recorder to capture the session on camera and, if we're doing mobile testing, we like to use AirServer to record what's happening on the device screen, and a document camera to see how the device is being interacted with by the person we're testing."

The technologies that form the foundation of digital experiences today have continued to develop. For Experience UX, the technologies or the tools in use, will always be secondary to the user, as Laura explains: "First and foremost UX is about focusing on people, regardless of the technology used to build products and services for them. The minute technology is prioritised over the end user, the person will be the one to lose out. Every project we take on is grounded in solid user research first, which guides the rest of the project and the recommendations we make to our clients. Any other approach is just guesswork."

Laura concludes: "Technologies and the methods to create the end products will always evolve over time, but humans and their behaviours remain relatively unchanged. To be limited to a small set of technologies or even

looking at the future in terms of just 'The Web' is also short sighted. Agencies already have to design for a much wider scope of technologies and touch points including voice interfaces, VR, AR and other wearable devices. As an experience agency we have to consider the end-to-end experience a person might have with a company, considering all the touchpoints, both online and offline. To limit our work to such a small slice of technologies, web pages and screens would be to do our clients, and their end users, a huge disservice."

One of the most important digital spaces that brands are racing to support is mobile. Damian outlines their approach:

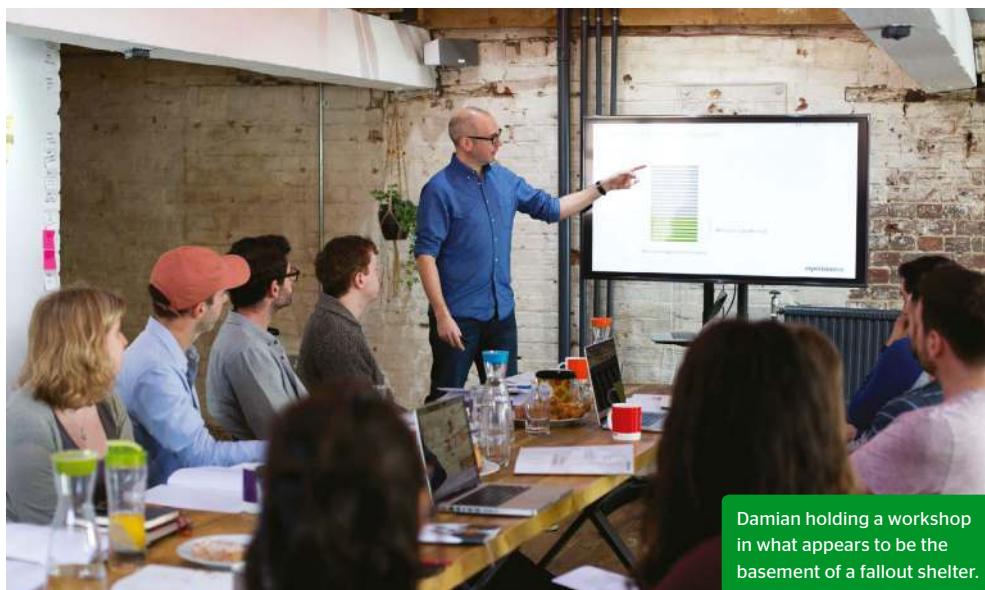
"All of our projects are multi-device. We don't believe that a user journey can or should be looked at from the perspective of only

one device. People interact with technology in complex ways with often a series of short interactions at different stages of their decision-making process. These journeys can take place on a variety of different devices in many different contexts. Although it is tempting to look at a discrete 'mobile journey', our research suggests that this can be a flawed approach, so we are careful to look at the whole user journey."

One of the most important spaces for brands to understand is social networks. Damian explains how concentrating too much on this aspect of a design can often derail its true objectives: "We concentrate on creating great customer journeys that convert, rather than solely looking at driving traffic from one source. Agencies can sometimes focus too much on traffic and not enough at conversion. We work with clients on a long-term conversion



## We also rely heavily on Slack for communications and Dropbox to organise and back up our work



Damian holding a workshop in what appears to be the basement of a fallout shelter.

## Timeline

### 2007

Experience Solutions set up winning their first major client The Financial Times.

*Employees: 2*

### 2009

First UX Training programme developed. Team expands with recruitment of UX Designer and Architect.

*Employees: 4*

### 2012

Experience Solutions move from incubation space to their own office.

### 2013

Experience secures a large on-going relationship with Siemens Financial Services, and Burberry.

### 2014

Experience lands the Open University as a client. The OU are still using their services today.

### 2015

Experience Solutions is rebranded to Experience UX.

### 2016

Launch of UX Bournemouth quarterly events. Experience UX wins their first two-year contract with McCarthy & Stone.

*Employees: 8*

### 2017

Experience outgrow their offices in Bournemouth and move to a bigger open-plan space.

### 2018

Launch of one day conference UX Insider Bournemouth in October 2018.

*Employees: 12*



- 3** UX Consultants
- 2** Research Consultants
- 1** UX Director
- 1** Commercial Director
- 1** Account Director
- 2** Finance/Admin
- 1** Graphic Designer
- 1** Marketing & PR

# Generation UX

UK £ Stockists 020 7433 1863  
FREE UK DELIVERY OVER £29\*

LILY'S KITCHEN PROPER FOOD FOR PETS

PET CARE LILYLAND

Register / Sign In Search

Warming hearts feeding souls

Menu FOR DOGS [SHOP NOW](#)

Menu FOR CATS [SHOP NOW](#)

## Lily's Kitchen

Our work for Lily's Kitchen started as all our projects do, with an initial client planning meeting to fully discover what their needs, objectives and requirements for the project were. It's important to us that we listen carefully to our clients, and really understand both their business and emotional needs for the project, to enable us to do a great job. We find that keeping communication lines open throughout the project are key to happy client relationships, so we regularly check in to make sure they are kept up to date with our progress.

We then spent time analysing and testing their current information architecture to understand where users would look for information and see where improvements could be made. With a new design of the website soon to be launched we created a high-fidelity prototype of the website to test with users before it went live and confirm that any design changes would not create any barriers to the users' buying process.

The testing of the prototype was conducted at research labs with in-person moderated sessions, to understand where further improvements could be made to the designs. The user-testing sessions were attended by the client, and it's always exciting to see them witness their customers using their product first-hand.

Our observation rooms are always abuzz with our customers having the important 'penny drop' moments, when a test participant struggles with part of the interface, or on the other hand enjoying an unexpected part of the experience. Those are the moments that make our job really exciting.

Once this user research was completed we analysed the observations and findings to report back our recommendations for improvements, and also created an updated fully working prototype.

May we recommend some of our  
**BESTSELLING RECIPES?**



UK £ Stockists 020 7433 1863  
FREE UK DELIVERY OVER £29\*

LILY'S KITCHEN PROPER FOOD FOR PETS

PET CARE LILYLAND

For Cat > Food For Cat > Shop By Product Type > Wet Food > **CLASSIC CHICKEN DINNER**

Classic Chicken Dinner

Lily's Kitchen Classic Chicken Dinner recipe is a wonderful, nutritionally complete meal with a very smooth pâté-like texture that your cat will love.  
read more

★★★★★ (5 reviews)

KEY BENEFITS:

- 65% MEAT & VEGETABLES
- GRAIN FREE
- OPTIMAL HUMECTANT LEVELS

PRICE: from £0.80

PACK SIZE:  QUANTITY:

LILY'S KITCHEN PROPER FOOD FOR PETS

UK £ Stockists 020 7433 1863  
FREE UK DELIVERY OVER £29\*

LILY'S KITCHEN PROPER FOOD FOR PETS

PET CARE LILYLAND

FOOD FOR CATS

Shop by AGE

- Kitten (up to 1 year)
- Adult (1-6 years)
- Senior (7+ years)

[SHOP ALL RECIPES](#)

Shop by PRODUCT TYPE

- Wet food
- Dry food
- Treats
- Selection boxes

Shop for SPECIAL DIETS

- Grain free
- Organic
- Hypoallergenic

BUY NOW

Experience UX undertook high-fidelity prototype testing with the target audience. Their aim was to understand what information users require throughout the purchase process. Finally, they tested a new IA structure to make sure users can achieve their purchase goals

"Sometimes it can be easy to quickly jump to conclusions about an end user's actions, but it takes a keen observer with lots of practice and the skills to interpret the actions of a person using a product, service or company in the way they do"

Laura Yarrow, UX Consultant



Experience UX clearly have a major case of Post-it-itus

optimisation plan which analyses user journeys triggering from multiple traffic sources through the conversion funnel.

"We analyse the data, highlight any areas we feel need more investigation, and then conduct tactical user research studies to understand why people are behaving the way they do. Once we fully understand the data, we then implement a series of iterative solutions that are fully researched with end users."

Laura also explains how new technologies are impacting the approach they take when working with their clients: "Recently we have been exploring Jobs to be Done ([jtbd.info](#)) as a research framework; it's a thought-provoking approach to helping clients understand why they should be investing in the type of research work we do.

"Sometimes it can be easy to quickly jump to conclusions about an end user's actions, but it takes a keen observer with lots of practice and the skills to interpret the actions of a person using a product, service or company in the way they do. Jobs to be Done (JTBD) can be a useful guide for what you are trying to achieve in your research and, can be used to check if you really have found the 'Job' your product or service is being hired to do.

"Lately, we've also been fascinated by some of the accessibility-related issues we've uncovered in our user testing and, have enjoyed using the Funkify Chrome browser extension to check if the designs we create are inclusive enough for those with a variety of visual, physical and cognitive needs.

"It's also been interesting to try out some of the newer online tools available such as [Overflow.io](#), [RealtimeBoard \(realtimeboard.com\)](#) and [Smaply.com](#) for rapid experience mapping online. The advantage of using online tools is that, as a team, we can all collaborate

on each other's work, with the output being a professional and consistent-looking end result."

As specialists, the people that work for Experience UX need to have a particular skill set but, as Damian outlines, a specific attitude is even more important: "The core attribute we look for is an authentic passion for UX. Someone who loves improving experiences, who enjoys learning more about good experience design and what makes a user journey poor. People who work here definitely have an empathetic nature, and a strong desire to keep learning and share their knowledge with one another.

"The other vital aspect is cultural fit with the team. We want people with a strong ethical compass and a set of values that match with our own. Ultimately, as a team we believe that life is stressful enough. If we can make experiences seamless and enjoyable instead, we can go home fulfilled at the

end of the day. The great part is that it's a win-win for our clients, their customers, and everyone who works here."

And looking to the future, as the digital space continues to diversify and touch even more of our lives, how will Experience UX shape these spaces and how we all interact with today's and tomorrow's technologies? "It's genuinely a fantastic time to be in UX at the moment," Damian concluded. "There are so many exciting changes in the industry and the constant evolution requires us to always learn and challenge ourselves.

"It's one of the reasons we have set up UX Bournemouth ([uxbournemouth.co.uk](#)); a series of evening talks with industry leaders, and have recently launched UX Insider ([uxinsider.net](#)), a one-day conference on 16th October. We want to lead the way in challenging the wider community to deliver great user experiences and we hope to do this

by growing a tribe of people interested in UX who are all keen to learn and adapt to the changes in the industry together.

"As a business, the direction we are heading in is one where we focus on the overall experience. As digital becomes more integrated in business as a whole, we believe that the customer journey is much less about a series of touchpoints, and instead more of a series of moments. We're working with companies who are keen to make those moments innovative and delightful."

The web, social media, mobile, VR and AR are all spaces that millions of people traverse every day. However, how users experience these channels is often overlooked. Experience UX, then, have a mission to not only understand how users move through these spaces, and how technology can support their needs and desires, but also to understand how digital channels and their enabling technologies will continue to evolve in future.

# experienceux

[experienceux.co.uk](#)

**Founders**  
Ali Carmichael and Damian Rees

**Year Founded**  
2007

**Current Employees**  
12

**Location**  
Bournemouth and London

**Services**  
UX Research  
UX Architecture  
UI Design  
Conversion Rate Optimisation  
UX Training & Mentoring

# HOT NEW

**CSS is changing fast. Here's our guide on the rules to use now, and what to keep an eye on**

“

**THINGS HAVE  
BECOME FAR MORE  
COMPLICATED FOR  
DEVELOPERS  
AS PEOPLE  
EXPECT MORE**

As the web has rapidly evolved, it almost seems that apologetic CSS has been left behind. Each week something happens, but seldom where it matters – the interface.

The internet follows an almost everlasting converge-then-diverge-then-converge model in applications, and now it's the turn of CSS. As more frameworks, ideas and standards have emerged, complexity has skyrocketed with more and more helper

technologies being invented from necessity; things have become far more complicated for developers as people expect more, but now they are slowly being merged into CSS as formal standards. CSS has become unrecognisable from its early inception in the 1990s.

And what to the future? We can probably say now that CSS of the future will see another divergence as different digital and physical devices and channels appear, even

to a brand-new technology. We can probably also guess that CSS will provide straightforward access to even richer interfaces and that the 'S' curve in development complexity and development features will narrow and steepen.

But what for now? These are our picks for the latest, greatest and most useful updates to CSS. Some might make you famous, some might simply make your code work on more browsers.

**EXAMPLE 5**

For example, the following rule

```
@supports ( display: flexbox ) {
  body, #navigation, #content { display: flexbox; }
  #navigation { background: blue; color: white; }
  #article { background: white; color: black; }
}
```

applies the rules inside the '@supports' rule only when 'display: flexbox' is supported.

**EXAMPLE 6**

The following example shows an additional '@supports' rule that can be used to provide an alternative for when 'display: flexbox' is not supported:

```
@supports not ( display: flexbox ) {
  body { width: 100%; height: 100%; background: white; color: black; }
  #navigation { width: 25%; }
  #article { width: 75%; }
}
```

Note that the 'width' declarations may be harmful to the flexbox-based layout, so it is important that they be present only in the non-flexbox styles.

**EXAMPLE 7**

The following example checks for support for the 'box-shadow' property, including checking for support for vendor-prefixed versions of it. When the support is present, it specifies both 'box-shadow' (with the prefixed versions) and 'color' in a way that would cause the text to become invisible were 'box-shadow' not supported.

```
@supports ( box-shadow: 2px 2px 2px black ) or
( -moz-box-shadow: 2px 2px 2px black ) or
( -webkit-box-shadow: 2px 2px 2px black ) or
( -o-box-shadow: 2px 2px 2px black ) {
.outline {
  color: white;
  -moz-box-shadow: 2px 2px 2px black;
  -webkit-box-shadow: 2px 2px 2px black;
  -o-box-shadow: 2px 2px 2px black;
}
```



**CSS @supports is a great way to build your project gracefully with the latest techniques while taking into consideration backward compatibility**

Andrew Collinson  
UI Team Lead - Mirum

# Feature queries (@supports)

## Check if a feature is supported before relying on it

When building components that use new and potentially unsupported features, it's important to include a fallback. This has been achieved using feature detection via JavaScript with plugins such as Modernizr. Now, this is possible directly inside CSS using the new @supports syntax.

With the addition of @supports, also known as a feature query, we can check whether the code we would like to use is or is not supported:

```
@supports (display: flex) {
  .hagrid { display: flex; }
}
```

When the page is loaded, a check will be performed to see if the browser in use supports the flex feature. If it does, the styling within the support braces will be applied by the browser.

If it is not, it will be ignored - it is worth having a fallback option for when the feature being queried is not supported.

The most appropriate way to include the fallback code will depend on the project and what is being changed as there may be

@supports code overriding default styling. In other cases, it may be preferable to include another @supports check to see if code is not supported. This is achieved with by negating the feature query:

```
@supports not (display: flex) {
  .dobby { display: block; }
}
```

Multiple checks can be combined to see if all the required features are supported at the same time, in order to apply either all or none of the style rules atomically.

The syntax for this is very similar to combining media queries, such as applying styles on devices between specific sizes. This is achieved by using the 'and' operator and multiple feature queries:

```
@supports (display: flex)
and (transform: scaleY(3)) {
  .fluffy {
    display: flex;
    transform: scaleY(3);
  }
}
```

The counterpart combination query for the 'and' operator is the 'or' operator, which checks if either of the feature queries matches

and applies the style if at least one is supported:

```
@supports (display: flex) or
(transform: scaleY(1.5)) {
  .gramp {
    display: flex;
    transform: scaleY(1.5);
  }
}
```

Something which is useful given that we have already covered CSS variables is that with @supports you can also check to see if a custom property is supported or not, which can be done with the following approach:

```
@supports (--ron: orange) {
  body {
    color: var(--ron);
  }
}
```

With the above example a check will be carried out to see if the custom property condition is supported and when it is the body will have a font colour of orange applied. @supports is available to use in all browsers with the exception of IE 11 and below, although some polyfills are available if you do need to support as far back as Internet Explorer 9.

## Graceful degradation

This is a great way to build in new styling rules, but it is also just as helpful at ensuring you build websites that degrade gracefully. If you use @supports then also use it to check if the browser does not support your CSS rules, this doubles work but creates something which downscale browsers more elegantly.



## CSS Grid

Build custom print-style layouts with less code than ever before

“

**STARTING OFF WITH THE GRID CAN BE DAUNTING AT FIRST, BUT IT'S SMOOTH SAILING ONCE YOU GET TO GRIPS WITH THE SYNTAX**

The arrival of Flexbox has allowed us to spend less time polluting our code with clear fixes, hacks and workarounds, and focus more on writing concise CSS and HTML. While Flexbox is essential for any UI dev's toolbox, it's best suited for working in one single direction at a time. This is usually enough, as we are usually only restricted on one axis at a time (for example, the width of the page is restricted, but not the height).

However, in the cases where we are restricted in both dimensions, say a dashboard style app, then CSS Grid is definitely one to shine. Starting off with the grid can be

daunting at first, but it's smooth sailing once you get to grips with the syntax. Say we have the following HTML structure:

```
<div class="container">
  <div class="gryffindor">Gryffindor</div>
  <div class="slytherin">Slytherin</div>
  <div class="ravenclaw">Ravenclaw</div>
  <div class="hufflepuff">Hufflepuff</div>
</div>
```

We can define the grid, set the dimensions, and then define

named areas for our grid cells:

```
.container {
  display: grid;
  width: 100vw;
  height: 100vh;
}
.container > div {
  border: 2px solid #000;
}
.gryffindor {
  grid-area: gryffindor;
  background-color: #C91018;
}
.slytherin {
  grid-area: slytherin;
  background-color: #26A147;
}
.ravenclaw {
```



```
grid-area: ravenclaw;
background-color: #005782;
}

.hufflepuff {
  grid-area: hufflepuff;
  background-color: #FFD63C;
}
```

Finally, on the .container we can make use of the grid-template-areas property to lay out what we want the grid to look like:

```
.container {
  display: grid;
  width: 100vw;
  height: 100vh;
  grid-template-areas:
    "gryffindor slytherin
    ravenclaw"
    "gryffindor
    hufflepuff
    ravenclaw"
}
```

```
"gryffindor
slytherin slytherin
ravenclaw"
"gryffindor
hufflepuff
hufflepuff
ravenclaw"
}
```

Additionally there are a whole host of additional properties to fine tune the presentation of the grid. To name a few:

```
grid-column-gap: <size>;
- gutter between columns
grid-row-gap: <size>;
- gutter between rows
```

```
grid-template-columns: <size>;
- width of columns, space separated for multiple
grid-template-rows: <size>;
- width of row, space separated for multiple
align-items: <center | end | start | stretch>; - where to vertically align the contents of each grid cell, where stretch is to vertically fill the cell entirely
justify-items: <center | end | start | stretch>;
- similar to align-items except along the horizontal axis instead
place-items: <align-items> / <justify-items>; - shorthand notation, combining align-items and justify-items
```

So in practice, we could update our previous grid:

```
.container {
  display: grid;
  width: 100vw;
  height: 100vh;
  grid-template-areas:
    "gryffindor slytherin
    slytherin ravenclaw"
    "gryffindor slytherin
    slytherin ravenclaw"
    "gryffindor hufflepuff
    hufflepuff ravenclaw";
  grid-column-gap: 10px;
  grid-row-gap: 15px;
  grid-template-columns: 200px
  auto auto 200px;
  align-items: stretch;
  justify-items: stretch;
}
```

This is all just barely scratching the surface of the possibilities of CSS Grid - there's so much more that can be done. Currently this

is supported in all modern browsers, with IE11 being the exception, whereas it supports the older syntax.

But, wait, there's more, with the use of CSS Grid we're finally able to centrally align content

without the stress!

```
.container {
  display: grid;
  height: 100vh;
  place-items: center center;
```

## Read-only selector

The 'read-only' pseudo selects elements which are not editable by the user. Combined with the ':read-write' selector, they form a simpler way to style disabled, read-only and content-editable HTML. This is especially useful when building forms that have some read-only inputs showing existing, non-editable data.

## ::playing and ::paused

The '::playing' or '::paused' pseudo elements can be applied to highlight a component when it is currently being played or is paused, which will be useful if you've ever got any videos in your build. Unfortunately this code is in draft so it is not fully supported yet.

### Handle breakpoints

Use Grid to move your content around your page when moving between breakpoints instead of hiding duplicated content in your mobile, tablet or desktop breakpoint.



**Filters is a great way to apply visual changes to the same old images whether this be through default states or changing a hover/active state of an image hyperlink.**

Andrew Collinson

## CSS Filters Creating visual effects with your CSS

CSS Filters are probably one of the most powerful aesthetic tools you can use in your code to update the appearance of your build as they are a method of editing images in your build using only CSS.

There are several effects you can apply with CSS filters but importantly you can chain some of these effects together so you can apply a contrast and grayscale effect to an image if you are hoping to make an expressive gallery of images.

The effects which can be applied are blur, brightness, contrast, drop-shadow, grayscale, hue-rotate, invert, opacity, saturate, sepia and also apply an SVG filter.

The good thing about the filter code is its simplicity. It is easy to remember and understand what is going on, as to declare the effect you want is often as simple as saying something like the following code:

```
.flitwick {
  filter: drop-shadow(10px 10px
10px rgba(0,0,0,.9));
}
.lockhart {
  filter: saturate(8);
}
.mcgonagall {
  filter: hue-rotate(90deg);
}
.snape {filter: blur(5px);}
```

What the example above would do is apply three chained filters to an image which are blur, hue-rotate and drop-shadow; admittedly drop shadows in CSS already exists so this one isn't that great of an addition.

### Chaining filters

Try applying more than one filter to an image to give it a more creative difference, something like applying grayscale and contrast changes to give your images that more striking look inside a gallery.

There are some known issues with filters, such as the ordering when chaining effects, which can sometimes have adverse effects - if you were to apply grayscale after sepia it would result in an entirely greyed out image.

Browser support is also a little problematic as there is no support in Opera Mini, IE 11 and below, and Edge only has partial support.

### 1. blur(5px)

This works by defining a blur size in px to your image. The higher the number the greater the blur, for instance, blur(10px) would create a strong blur effect.

### 2. grayscale(100%)

Applying this filter converts the image to grayscale. The value to change to complete grayscale would be 100% or 1, and to change to semi-grayscale you could use 50% or .5.

### 3. saturate(1)

The use of this completely changes the colouring of the image making everything more vibrant. Saturation works in a similar way to grayscale with 1 or 100% being complete saturation.

### 4. hue-rotate(90deg)

This is probably the most unpredictable filter as it works by rotating the colour balance of the image. Updating the rotation can be done using degrees, turn or radians.

## Resetting with 'all'

No doubt you have encountered a situation where you wished there was an easier way to revert several properties to their default, rather than having to juggle several properties at the same time. 'All' isn't as yet 100% browser compliant and does not work in IE, Edge or Opera Mini.

```
:root {
    --color-background-inverse: #aba49d;
    --color-background-main: white;
    --color-background-backdrop: #f3f2f2;
    --color-text-default: #5c606c;
    --color-text-heading: green;
    --font-family-heading: 'Open Sans', Arial, sans-serif;
    --font-family-monospace: Consolas, Menlo, Monaco, Courier, monospace;
    --font-family-text: 'Open Sans', Arial, sans-serif;
    --shadow-card: 0 19px 38px rgba(0, 0, 0, 0.3), 0 15px 12px rgba(0, 0, 0, 0.22);
    --spacing-large: 2rem;
    --spacing-medium: 1rem;
    --spacing-small: 0.5rem;
    --radius-large: 0.5rem;
    --radius-medium: 0.25rem;
    --radius-small: 0.125rem;
    --duration-instantly: 0s;
    --duration-immediately: 0.05s;
    --duration-quickly: 0.1s;
    --duration-promptly: 0.2s;
    --duration-slowly: 0.4s;
}
```

# CSS Variables Reduce repetition and dry out your code with variables

Within CSS, to reuse code throughout a build, you used to have to type it out manually over and over, then preprocessors came along and the world was good - except you had to install and compile them, which wasn't always as easy as it should be.

Now within CSS you are able to use custom variables that allow you to pass through a string of content into your code such as a colour or size etc. Yet, unlike preprocessors, you can update the variable to be something different when used within a media query, which removes the necessity of declaring a specific variable to be used in each media query.

```
:root {
    --body-color: red;
    --primary-font-size: 12px;
    --paragraph-margin: 0 0 8px; }
```

## Variables for fonts

Try applying all your fonts as variables as this will allow you to create a standard list of font variants throughout your build by using variables for items such heading 1, 2, 3 etc.

The variable is declared inside ':root' and is then applied to your tag via the 'var(--variable-name)' code, but as you can see by the example above we have then overwritten those variables for screen sizes larger than 768px so we no longer need to create an

```
p {
    color: var(--body-color);
    font-size: var(--primary-font-size);
    margin: var(--primary-margin);
}
@media screen and (min-width: 768px) {
    :root {
        --primary-font-size: 14px;
        --paragraph-margin: 0 0 16px;
    }
    p {
        font-size: var(--primary-font-size);
        margin: var(--primary-margin);
    }
}
```

individual variable for each breakpoint we support.

The overall simplicity and cascading element of CSS Variables makes it easier and cleaner to use in your build than variables within preprocessors. This allows you to create much cleaner default styling throughout your build while still using the same variable.

## Nesting

CSS Variables can also be nested within one another, which can be helpful when working with large builds that are changing frequently. For example, you could do the following:

```
:root {
    --parent: 1rem;
    --child: calc(var(--parent) * 2); }
```

## Browser support

Overall, it is debatable whether you should use CSS Variables or not. Ultimately it all depends on what you have to support for the

build you're doing, but as far as browser support goes, variables are supported in all browsers except for Opera Mini, Samsung Internet 4 and old faithful Internet Explorer 11 as always.

One thing worth mentioning is that Microsoft Edge does support variables but with a few known issues in release 15 which are:

- You can not apply a CSS Variable to an item that is a pseudo element such as ':before' and ':after'. These issues unfortunately vary from applying background or border colours, or to applying animations to pseudo elements.
- Animations that use CSS Variables have been known to cause the browser to crash entirely - not ideal.
- Nesting variables within variables causes an error in Edge 15 that results in the browser not reading the variable at all; therefore no value is assigned to your component. Fixes are imminent, hopefully.



# Background repeat Taking control of how your background repeats



## Round

'Background-repeat' has been a stable part of CSS for what seems like an eternity, but there are two lesser known options available - Round and Space. Both of these options work by ensuring that

100 per cent of the background image declared is repeated and visible at all times, but they work in slightly different ways and will make a subtle difference to how images appear.

What 'background-repeat: round;' does is to repeat your background image without ever cropping the image, but will resize the image evenly until there is enough space to fit in an extra tile.



“

**Background repeat round is especially useful if your tiled background image looks strange if you can only see a part of it. For example, if you have a chessboard background, it makes more sense to show it slightly stretched than having a cropped tile.**

Paul Sandwell  
UI Engineer – Mirum



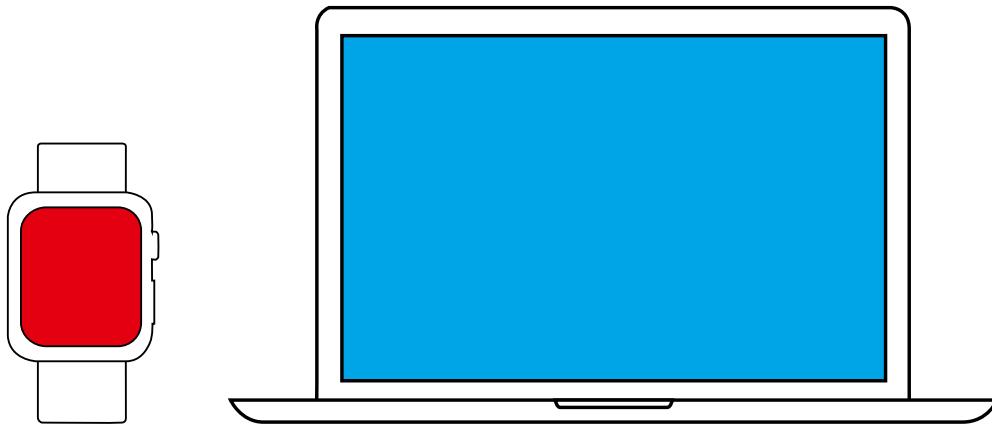
## Space

'Background-repeat: space;' tiles the image in both directions, similar to how Round works, except instead of resizing the image until there is more space to display another tile, it will instead create white space between the

tiles until another row or column can be shown.

The use of either of these options isn't relatively new or massively beneficial but it will give you that little extra control of the repetition of images in your layout

as you can control the way the tiling effect takes place over various viewport sizes. What is also very useful about these two options is that they are supported by every browser so they can be used without concerns.



## Max 16/9

```
@media (max-aspect-ratio: 16/9) {
    background: red;
}
```

## Min 16/9

```
@media (min-aspect-ratio: 16/9) {
    background: blue;
}
```

## Computed styles

## CSS 'will-change'

This relatively new feature of CSS has little impact on the appearance of your page – it is more of a processing tool than anything.

The idea with 'will-change' is to help the browser understand and prepare for elements of your code that will change when interacted with.

For example, when interactions are performed with CSS, the browser will do the best it can to plan ahead for what changes may occur. And with the addition of 'will-change' you can notify the browser in advance of what may change so it can prepare better. This will result in any interactions you need being more responsive.

## Aspect ratio

### Detect viewport ratio instead of width

The introduction of media queries changed the way we build websites. We were given the opportunity to build our layouts depending on the way the user viewed the page. More often than not this would be based purely on the viewport width or height but recently there has been a new option added.

Now you are able to apply CSS based on the page's viewport via a media query that detects the aspect ratio of the window you're browsing with. You are also able to detect whether the window matches a maximum or minimum aspect ratio too.

The code to use for this is fairly simple, as it matches the convention used with any other media query declaration, which would be something like:

```
@media screen and (aspect-ratio:
```

```
16/9) {
    Your CSS
}
```

```
@media screen and (min-aspect-
ratio: 16/9) {
    Your CSS }
```

While useful there are only a few scenarios in which aspect-ratio media queries would be useful. If you are building something specifically for mobile devices you could use the min/max aspect-ratio to target whether the user is browsing in landscape or portrait mode, but this is also covered with the orientation aspect ratio. Fortunately, in terms of support, aspect-ratio is accepted widely across the browsers. If this is really something you could use then there is little standing in your way – all major browsers support it, with Internet Explorer support going as far back as version 9.

## CSS Shapes (shape-outside)

Something that has previously been lacking from CSS was the ability to wrap content cleanly around images that do not have squared edges. This doesn't create visually pleasing layouts, but now with the introduction of 'shape-outside' you can create more adaptive layouts that resemble something closer to a magazine layout than our old standard floated block images. To use 'shape-outside' you should first apply it to a floated element such as an image, and then in your CSS you declare which shape method you would like to use. The options available are:

- circle: create circular shapes
- ellipse: create elliptical shapes
- inset: create rectangular shapes
- polygon: create any shape with 3 or more vertices
- url: identifies which image should be used to wrap text around. This works by placing text within the transparent areas of the image.

The support for 'shape-outside' isn't particularly good yet, as Internet Explorer, Edge and Opera Mini all offer no support, but Firefox has now begun supporting it from release 62 onwards. The support of 'shape-outside' in Microsoft Edge is currently under consideration, but there is no official word on when it could be implemented.



## Blend-mode A new way to combine images and colours for unique visuals

Applying background images is typically a stable part of every project build. But sometimes there is the need to be a little more creative with what can be done with the background image. This is where 'blend-mode' comes in to it.

By using 'blend-mode' you've got a little more freedom over how a background image works with the background colour instead of the usual background on top of the background colour. 'Blend-mode' works by combining the

background image with the background colour with a defined style. The options you can use are normal, multiply, screen, overlay, darken, lighten, colour-dodge, saturation, colour and luminosity.

The code to use for blending your background is fairly simple – all you need is one line of CSS to merge the background image with the background colour.

```
background-blend-mode: *blending-option*;
```

Blend modes are available to use in almost all browsers with the only exceptions being all versions of Internet Explorer and Opera Mini. Note that blend modes work in all version of Microsoft Edge.

### 1. Overlay

Using a blend mode of overlay works by mixing both the background image and colour together to echo the darkness or lightness of the backdrop area.

### 2. Difference

Using a blend mode of difference results in a more creative outcome. This works by subtracting the darker colours from the background image/colour from the lightest colours, which results in the image having a very high contrast effect.

### Be creative

Try something different and instead of applying 'blend-mode' to your background image try placing it on text that overlaps your image instead for a translucent effect.

### 3. Luminosity

Luminosity is one of our favourite visual effects. It kind of creates that washed out, old photo look. Luminosity works by preserving the top colour while fading hue and saturation of the background.

### 4. Chaining blends

When it comes to applying blends to your background image you needn't use just one mode. You can chain multiple blending effects together to create a more unique effect by using something similar to the code below:

```
background-blend-mode: multiply, darken;
```

## currentColour

currentColor is something that has crept into our daily code usage, however, the support for `color` isn't something new having been supported in all browsers from Internet Explorer 11 and onwards.

The idea behind currentColor is to declare your colour of choice via `'color: #000'` and then when you want to apply the colour to something else like a border, for example, you would use `'border: 1px solid currentcolor'`. This will result in the border colour matching the declared text colour. Additionally, the colour then cascades down as your styling changes. Pretty cool, eh?

If you were to declare a colour on a link and then apply `currentColor` to a border and SVG icon within that link, these would also change when you declare your hover and focus states like all responsible developers should be declaring.

“

**THE CODE TO USE FOR BLENDING YOUR BACKGROUND IS FAIRLY SIMPLE - ALL YOU NEED IS ONE LINE OF CSS**

# Object-fit

## 'Background-cover' works for backgrounds, but what about inline images?

Getting an image to fill a content box is often done by making use of the 'background-size: cover|contain' styles, however, previous attempts to recreate this behaviour for images or videos has been somewhat clunky. Thankfully the 'object-fit' property has emerged, seemingly out of nowhere, to help. This can be used to achieve the same result but on media assets such as images and videos. 'Object-fit' can take five values:

- 'Fill': stretches the asset to fit the parent container
- 'Contain': scales the asset to fit inside the parent container, maintaining aspect ratio (similar to 'background-size: contain')
- 'Cover': scales the asset to completely fill the parent container, cropping from the sides where necessary (similar to 'background-size: cover')
- 'Scale-down': similar to contain, however, this will only scale the asset down, and not make it larger than it is natively

- 'None': Render the asset as it would normally.

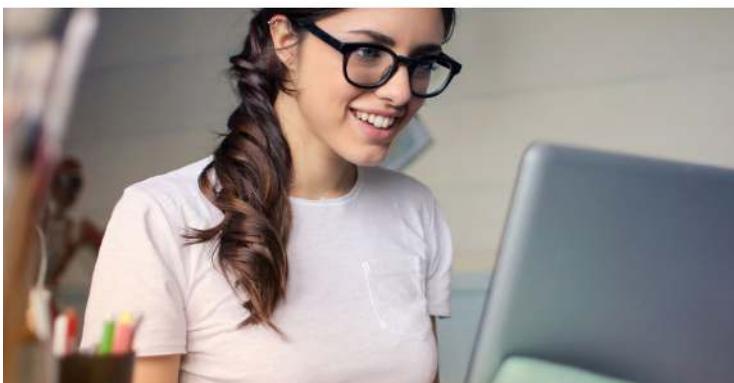
This property can also be combined with 'object-position' to determine where the asset should be scaled from.

To apply 'object-fit' to your images it is fairly simple in that all you need is something similar to the code below:

```
.lupin {
  height: 200px;
  object-fit: cover;
  width: 400px; }
```

It is important to remember height and width of the area you'd like to cover otherwise the image will take on its default aspect ratio. This will most likely result in the image spanning 100% width of your available area.

In terms of browser support, 'object-fit' is quite compliant as it is supported in all browsers with two exceptions. There is no support at all in Internet Explorer, while Microsoft Edge supports object-fit on images only. So don't use if you need to support either.



# 5 TO WATCH

## Here's our list of the top rules coming soon to CSS

### 1. Environment variables

<https://drafts.csswg.org/css-env-1>

Environment variables function similarly to the CSS Variables, but do not allow for overwriting. An environment variable defined once is guaranteed to have the same value throughout the document, whereas CSS Variables (note, these are actually called custom properties) can be overwritten on a component level and changes cascade down to child declarations.

### 2. :focus-within

<https://mzl.la/2gVA5A>

This is probably one of the most eagerly anticipated newcomers, this selector is active when the user has focused a child element.

This is especially helpful for exposing elements for keyboard accessibility.

### 3. Shape media-query

Most web devices are designed and implemented for rectangular devices, however, with the rising popularity of wearables, we are seeing more and more round devices show up. The shape media-query can be used to target rectangular and round devices. This is specifically useful for ensuring that everything fits onto the appropriate device's screen.

### 4. Scroll anchoring

<https://drafts.csswg.org/css-scroll-anchoring>  
The introduction of scroll anchoring solves one of the most frustrating problems that happens on the web when you unintentionally click on a re-positioned element after late loading content in your page finally displays. Scroll anchoring aims to solve this problem by forcing the browser to maintain your viewport position when content is loaded to the page.

### 5. Regions

<https://drafts.csswg.org/css-regions-1>

The core concept behind regions is being able to say, "Display this content over there instead." By setting an area of the document as a named region, content from other parts of the page can be moved into a different part of the document flow. Unfortunately, the draft for CSS Regions has not been updated in some time and may be dropped.

# web workshop

## Select characters to switch titles

Inspired by <https://hassyadai.com>

### Rotating geometry

The geometric Tetrahedrons spin in the background and look a little like constellations. The line thickness around them animates on the edges.

### Fading back

The other character fades further back and becomes slightly transparent to give more emphasis on the character that's selected.

### Bringing to front

When the user hovers over one of the characters their image moves forward and enlarges to bring emphasis to the selection.



### Main menu

The main menu is available over on the left of the screen. This brings an overlay with the various options available.

### Changing titles

As the character scales up and moves forward, the title in the centre of the screen changes and fades in to view giving context to the user's selection.

### Smokey background

The background has a particle system that uses a cloud texture to make a smoke effect rise and change colour at different parts of the screen.

### EXPERT ADVICE

#### Theming the site

The effects that are used on the homepage of the site set the site's theme up so that content that follows on other pages follows the same graphical styling. Effects such as transitions and scrolling take advantage of the 3D nature of the homepage, with one whole section scrolling through content using the z-axis in 3D.



#### <comment>

What our experts think of the site

#### Bringing it all together

The main reason for using Three.js to render the scene into WebGL is to give a consistent feel to the graphics. Transparency effects, blending and the like are all displayed with the same lighting and effects in the scene. For a simpler option this scene could be recreated inside div sections of a page.

**Mark Shufflebottom, Professor of interaction design**

## Technique

### 1. Text switching

To recreate the text switching effect as seen on the homepage of Hassyadai, there will be a basic HTML layout that has been created in the 'start' folder of the tutorial files. To make the images get larger on rollover, two classes will create that transition.

```
.grow {  
    transition: all .7s ease-in-out;  
}  
  
.makeGrow {  
    transform: scale(1.3);  
}
```

### 2. Showing the text

To show the text the 'hidden' class will be dynamically switched in and out on the text elements. This just has no opacity. Because of the transition on the text elements, the opacity will fade in and out.

```
.hidden {  
    opacity: 0;  
}  
  
#main,  
#leftText,  
#rightText {  
    transition: all .7s ease-in-out;  
}
```

### 3. Making it work

The functionality of this will all be controlled through JavaScript. Here the code is added to script tags at the bottom of the page. The code is referencing all the elements on the page that need to be controlled so that they are cached in variables.

```
var over = false;  
var middle = window.innerWidth / 2;  
var grid = document.getElementById("grid");  
var main = document.getElementById("main");  
var leftImg = document.  
getElementById("leftImg");  
var rightImg = document.  
getElementById("rightImg");  
var leftText = document.  
getElementById("leftText");  
var rightText = document.  
getElementById("rightText");
```

### 4. Grabbing the mouse

In order to make this work, the mouse position is needed. The overall container object is grabbed just to see if the mouse is over it or not. As there are many sub elements

that are visible or not, these will interfere with just doing a rollover test on individual elements.

```
grid.onmouseover = function() {  
    main.classList.add("hidden");  
    over = true;  
}  
  
grid.onmouseout = function() {  
    main.classList.remove("hidden");  
    over = false;  
}  
  
document.onmousemove = function() {  
    var x = event.clientX;
```

### 5. Left or right

Once over the container, the mouse is checked to see which side it's on. If it's the left, then the text over there is faded onto the screen and the image is scaled up to make this the most obvious.

```
if (over == true) {  
    if (x < middle) {  
        leftImg.classList.add("makeGrow");  
        leftText.classList.remove("hidden");  
        rightImg.classList.remove("makeGrow");  
        rightText.classList.add("hidden");  
    } else {  
        leftImg.classList.remove("makeGrow");  
        leftText.classList.add("hidden");  
    }
```

### 6. Over and out

The last part of the code is to restore everything to normal in the 'else' statement shown here. This restores the text to be invisible and the images to be their regular size if the mouse is not over the container.

```
rightImg.classList.add("makeGrow");  
rightText.classList.remove("hidden");  
}  
}  
} else {  
    leftImg.classList.remove("makeGrow");  
    leftText.classList.add("hidden");  
    rightImg.classList.remove("makeGrow");  
    rightText.classList.add("hidden");  
}
```



# Get started with Three.js - part 2

Learn how to light a scene and create beautiful shadows



 **DOWNLOAD TUTORIAL FILES**  
[www.filesilo.co.uk/webdesigner](http://www.filesilo.co.uk/webdesigner)



**WebGL** 3D enables browser-based experiences, based on the powerful OpenGL language. It taps into the graphics pipeline, for highly optimised, interactive experiences. Wide browser and device support for WebGL makes it a perfect approach for real-time rendering. No plugins are required and you can start learning these technologies right away.

Complex models with high levels of detail, reflections, environment maps and shadows can all be generated in real-time. You can give users access to beautiful visual experiences, on their desktops or in the palm of their hand. Sites, games and apps, can allow users to direct the action, move the camera and objects, navigate 3D worlds, and much more.

You'll be using the popular 3D library Three.js to dive into creating scenes, and animating objects. It's free and open source, light-weight and boasts countless award-winning websites that have used it. Facebook 3D objects are also now powered by this 3D library.

Continuing from the last tutorial, you will move onto lighting your scene and creating cool shadows. To do this you will learn about basic lights, materials and settings to enable your shadow casting. Other than having a JavaScript background, you can dive into this tutorial with no prior knowledge and get some great results. The goal is to demystify 3D web programming and get you inspired and well on your 3D journey.

## 1. Create HTML file

To get started, you need to set up a basic HTML file. You can set up external CSS and JavaScript files or include inline for simplicity. Three.js's renderer class will create a <canvas> element for you. Add the following code to your index.html file.

```
<!DOCTYPE html>
<html>
<head>
<style>
</style>
</head>
<body>
```

```
<script>
</script>
</body>
</html>
```

## 2. Include the Three.js library

Include a link to the Three.js library in the head of your file, either hosted externally or download it from the Three.js repository. You can find the library and minified JavaScript here: <https://github.com/mrdoob/Three.js>. Note: the code in this tutorial has been tested on the latest release of Three.js - v91.

```
<script src="libs/three.min.js"></script>
```

## 3. Add basic CSS

In your 'style.css' or between your style tags, set up a couple of basic styles. These are simple style rules to keep your canvas full screen, removing any margins or padding. We'll handle sizing of the renderer later on. In the last tutorial (WD276) you also saw how to create a window resize handler.

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

## 4. Create a 3D scene

You're going to add a basic 3D scene, which will be the container for your objects. The scene is the stage that will render with the camera. All 3D presentations will have a scene or stage of some form. What is in that stage and in view of the camera is what the user will see. Add the following code to add a scene:

```
// create a scene object
var scene = new THREE.Scene();
```

## 5. Add a perspective camera

Next, you need to add a camera. You'll use the

perspective camera, meant for 3D scenes. The first attribute is the field of view of the camera. The second is the aspect ratio ( width/height). Then you indicate the near clipping plane and the far clipping plane distances, which define what is to be visible to the camera. Also push the camera back in Z space a little to see things easier.

```
var camera = new THREE.PerspectiveCamera(
75, window.innerWidth/window.innerHeight,
0.1, 1000 );
camera.position.z = 15;
```

## 6. Add a renderer and canvas

The renderer handles the drawing of the objects in your scene that are visible to the camera. Set the 'antialias' property to true, to get smooth edges on our object. You can also define the size of the draw area to full screen. The renderer creates a 'domElement' which is actually an HTML <canvas> element that you can then append to the body. Additionally, you could specify an existing canvas element to draw to if you prefer, via the 'canvas' attribute of the renderer.

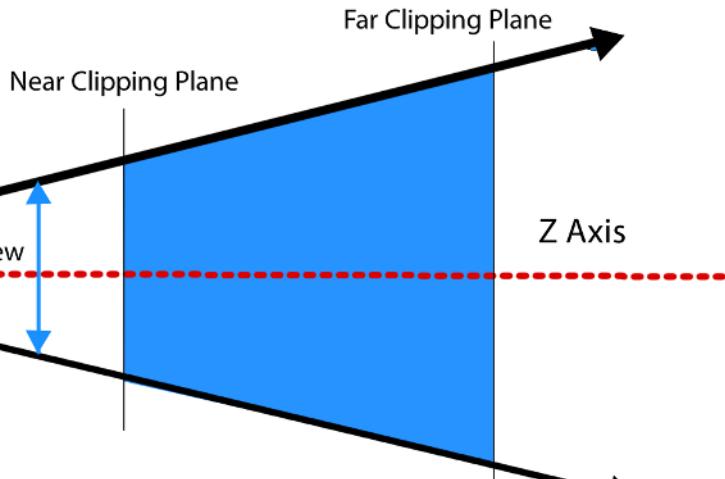
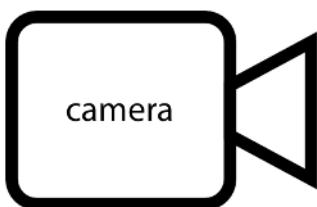
```
// create a renderer
var renderer = new THREE.WebGLRenderer({antialias:true});
renderer.setSize( window.innerWidth, window.innerHeight );
document.body.appendChild( renderer.domElement );
```

## 7. Create object geometry

Three.js includes plenty of primitive geometries to get started. Try out a Torus Knot using the

## Defining a 3D object mesh

3D objects meshes are much like physical objects. They are comprised of geometry and materials. Geometry defines the shape through vertices, faces etc. Materials are the 'skin' that textures the geometry. The Mesh is the result of combining geometry and materials together.



Setting up the camera allows you to define aspect, field of view and culling. This diagram helps illustrate these elements

# Tutorials

## Get started with Three.js - part 2



**Step 10**

Initially you use Normal Materials to see your object in 3D space without having to set up lights. With the basics set, you can move onto adding lights and shadows



**Step 13**

Using a Standard Material uses a 'physical' shader that emulates natural light and surfaces. Lights reflect accurately off its surface for a realistic finish look



**Step 17**

Adding the ground plane to receive shadows will allow you to see what is being cast by your object. This is rotated 90 degrees to lie flat like a floor

'TorusKnotBufferGeometry'. The first two attributes are torus radius and tube radius. The next two are the tube segments and radial segments, which determine the amount of divisions or faces that will be made along the surface. Try experimenting with the last fifth and sixth variables to generate some really cool shapes.

```
// create a 3D object
var geometry = new THREE.TorusKnotBufferGeometry( 4, .5, 60, 60, 13, 10 );
```

### 8. Create a material

Next, you need a material. Three.js includes a range of materials including physical shaders, lambert and phong. You can set textures using video or images as well. Use 'MeshNormalMaterial' for now. This way you can see the

## Shadows in 3D

In order to cast shadows, calculations are required by the renderer via ray-casting. You must enable shadow mapping in the renderer, set objects to cast and/or receive shadows and define shadow casting in the lights that will create them.

object without needing to light it. You'll get into lighting once you have the scene rendering.

```
var material = new THREE.MeshNormalMaterial();
```

### 9. Create a mesh and add it to the scene

To create an object 'mesh', you combine the geometry and material you just defined. Physical objects in 3D require a geometry that defines the faces, vertices and drawing of the shape. They also require a material or skin to cover that object so we can see it. Create the mesh object and add it to the scene, like this:

```
var object = new THREE.Mesh( geometry,
material );
scene.add( object );
```

### 10. Render the scene

Rendering is the function of drawing the scene data to the 'canvas' element. Now that you have a scene, camera, and a 3D object, you can render it. You'll set this up in a render loop next, but for the time being, use this code to see your 3D object so far.

```
renderer.render( scene, camera );
```

### 11. Render each 'requestAnimationFrame'

You bind your render function in a loop to the 'requestAnimationFrame' function. It will optimally run at 60fps, and ensure the browser is ready to render the next frame. To animate scenes smoothly you need to render at least 24 frames per second (ideally 60 fps). Replace that last line of code for rendering with the following snippet:

```
// render the scene
var animate = function () {
  requestAnimationFrame( animate );
  renderer.render( scene, camera );
};

animate();
```

### 12. Animate your object

To animate the object's rotation, you can add the rotation code inside the render loop to update the rotation every frame. In Three.js these are in radians not degrees. radians = degrees × (Math.PI / 180). Update your render loop to look like this:

```
var animate = function () {
  requestAnimationFrame( animate );
  object.rotation.x += 0.01;
  object.rotation.y += 0.03;
  renderer.render( scene, camera );
};

animate();
```

### 13. Use light reactive materials

Normal materials were used initially to enable seeing the object without the need for lights. Try using a standard material next, to emulate a real world look. Update your object code block to the following:

```
var geometry = new THREE.TorusKnotBufferGeometry( 4, .5, 60, 60, 13, 10 );
```

```
);
var material = new THREE.MeshStandardMaterial( { color: 0xee3333 } );
);
var object = new THREE.Mesh( geometry,
material );
scene.add( object );
```

### 14. Add an ambient light

There are a number of lights in Three.js and most 3D applications. These include directional lights, spot lights, point lights, ambient lights and many others. Ambient lights will cast a general light on the entire scene, with no direction. Add one to your scene, setting a colour and an intensity like this:

```
// create an ambient light
var light = new THREE.AmbientLight( 0xffeecc, .4 );
scene.add( light );
```

### 15. Add a primary light

Spot lights are one of the lights capable of casting shadows. These are like the classic theatrical lights, used to light a character on a stage. They are directional and have a specific area of intensity. Move outside that area and you step out of the light. Add a spot light like so:

```
//Create a SpotLight
var light = new THREE.SpotLight( 0xffffff, .6 );
scene.add( light );
light.position.set(3,4,15);
```

### 16. Add a secondary light

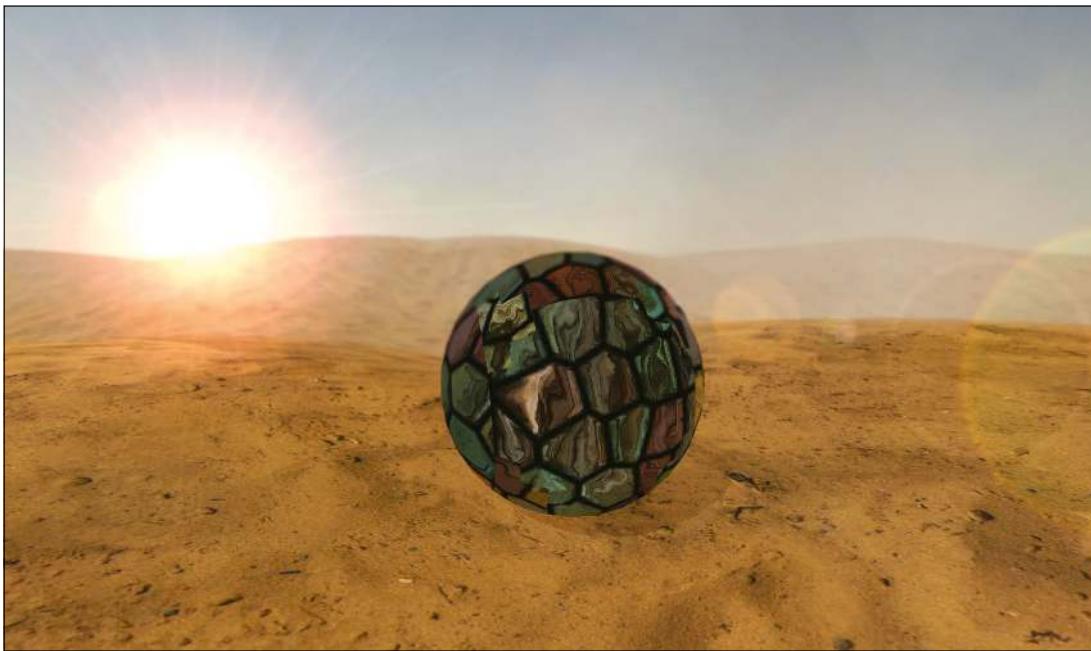
The art of lighting is complex and involves studying scenes in real life, film and various mediums. You can add as many lights as you wish and experiment with various looks. This secondary light is often called a fill light. Add a second light to your scene to create even more light and shadow effects:

```
var light = new THREE.SpotLight( 0xff0000, .4 );
scene.add( light );
light.position.set(-5,15,5);
```

### 17. Create ground plane

In order to see the shadows you'll want to cast them onto something. Many scenes will have a ground or floor. Add one by creating a new 3D object using the 'PlaneBufferGeometry'. Notice you also need to rotate the plane 90 degrees to the camera to lay flat, and we need to instruct it to 'receiveShadow'.

```
//Create a ground plane
var planeGeometry = new THREE.PlaneBufferGeometry( 500, 500, 32, 32 );
var planeMaterial = new THREE.MeshStandardMaterial( { color: 0xeeaa33 } );
var plane = new THREE.Mesh( planeGeometry,
planeMaterial );
plane.rotation.x=Math.PI/180*-90;
plane.position.y=-6;
plane.receiveShadow = true;
scene.add( plane );
```

**Lights in 3D**

Three.js includes many lights you can use to light your scene. These include spot, directional, ambient, point and several other types.

Ambient lights flood a scene with light, which you can also tint. It increases overall brightness and is great for increasing a scene's light without adding more directional lights. Spot lights shine with a beam, much like a flashlight. You can see the circle of light on the receiving object if the beam radius is tight enough.

Directional lights are like spots, except they have no circle. They cast evenly from a direction, much like how the sun lights the earth. Use directional light to emulate the sun. And point lights shine evenly in all directions from a specific position in 3D space. Think of these like a bare light bulb.

**18. Create back wall**

Next, you will add a back wall to receive even more shadows onto. This will show you the dynamics of real-time shadow rendering. Even with a more complex looking shape like the one you are using, shadows are cast accurately and smoothly. Add it like this:

```
//Create a back wall
var planeGeometry = new THREE.
PlaneBufferGeometry( 500, 500, 32, 32 );
var planeMaterial = new THREE.
MeshStandardMaterial( { color: 0xeeaa33 } );
var plane = new THREE.Mesh( planeGeometry,
planeMaterial );
plane.position.z=-50;
plane.receiveShadow = true;
scene.add( plane );
```

**19. Update renderer to use 'shadowMap'**

In order to see the shadows, you must instruct the renderer to calculate and draw them. You do this by enabling the 'shadowMap' property of the renderer. Also, to create softer, smoother shadows be sure to use the 'PCFSoftShadowMap' type:

```
//update renderer to use shadowmap
renderer.shadowMap.enabled = true;
renderer.shadowMap.type = THREE.
PCFSoftShadowMap;
```

**20. Update object to cast shadow**

You have control over which objects will cast and receive shadows in your scene. Shadows can be expensive in terms of processing costs. So always figure out the minimum lights you need to create your effect, and the minimum shadows needed. Turn on shadow casting for objects like this:

```
object.castShadow = true;
```

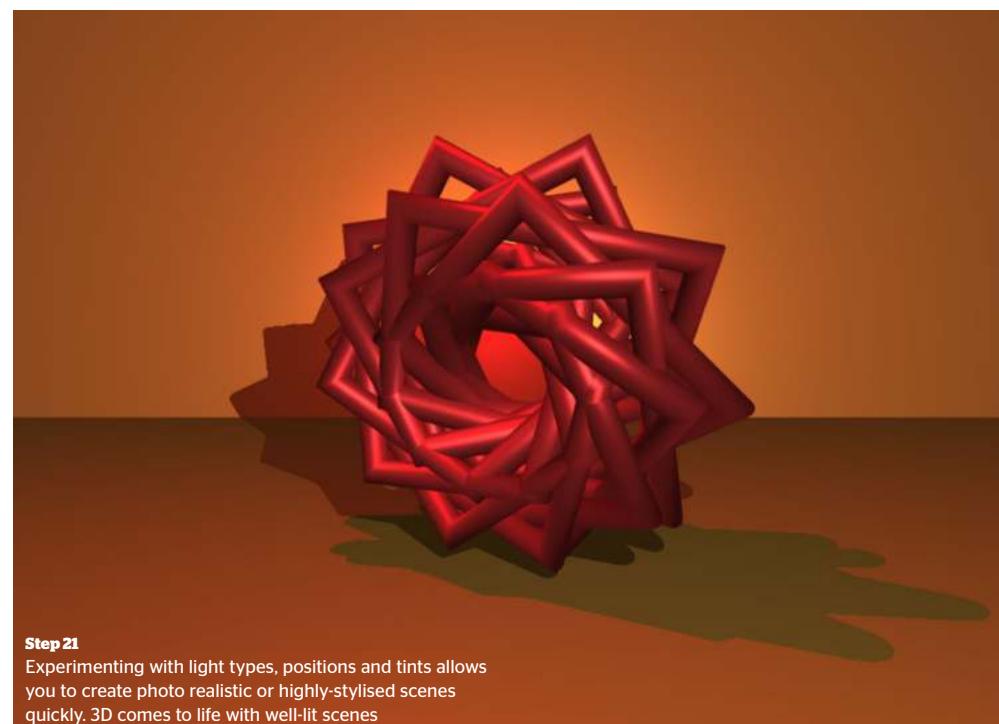
**21. Update lights to cast shadows**

Last, you need to set the 'castShadow' property for the spot lights and update some details. Near and far clipping planes should be set to avoid clipping in unintended places, and you can define the shadow map size for high resolution. Add these lines below each line you added to add a spot light. Re-run your scene and you will see really nice detailed lights and shadows!

```
light.castShadow = true;
light.shadow.mapSize.width = 2048;
light.shadow.mapSize.height = 2048;
light.shadow.camera.near = 0.1;
light.shadow.camera.far = 500;
```

**Step 18**

Adding a back wall with 'receiveShadows' set to true will reveal even more shadows, creating an interesting scene with detailed shadows

**Step 21**

Experimenting with light types, positions and tints allows you to create photo realistic or highly-stylised scenes quickly. 3D comes to life with well-lit scenes

# Convert Flash games to HTML5

Learn how to use JavaScript, WebGL and the Phaser framework to ensure your games work on all browsers and platforms





**Flash is slowly being abandoned by Adobe in favour of HTML5 and JavaScript; its official end-of-life is set for the year 2020.**

And that's where this article will come in handy. The tips described below aim to help HTML5 game developers in avoiding common mistakes when converting their Flash games to JavaScript, as well as making the whole development process go as smooth as possible. Fixing bugs and improving performance also translates to players spending more time in the game and thus on your site. All you need is basic knowledge about JavaScript, WebGL and the Phaser framework.

With the rise of HTML5 usage, many companies start redoing their most popular titles to get rid of outdated Flash and match their products to the latest industry standards. This change is especially visible in the Gambling/Casino & Entertainment industries and has been happening for several years now, so a decent selection of titles has already been converted.

Unfortunately, when browsing the internet, you can quite often stumble upon examples of a seemingly hasty job, which results in the lower quality of the final product. That's why it's a good idea for game developers to dedicate some of their time for getting familiar with the subject of Flash to HTML5 conversion, and learning which mistakes to avoid before getting down to work.

Among the reason for choosing JavaScript, apart from the obvious technical issues, is also the fact that changing your game design from SWF to JavaScript can yield a better user experience, which in turn gives it a modern look. But how to do it? Do you need a dedicated JavaScript game converter to get rid of this outdated technology? Well, Flash to HTML5 conversion can be a piece of cake - here's what our experienced JavaScript Game Developer has to say about the matter.

## 1. Improving the HTML5 game experience

Converting a game to another platform is an excellent opportunity to improve it, fix its issues and increase the audience. Below are few things that can be easily done and are worth considering:

### • Supporting mobile devices

Converting from Flash to JavaScript allows reaching a broader audience - users of mobile devices support for touchscreen controls usually needs to be implemented into the game. Luckily, both Android and iOS devices now also support WebGL, so 30 or 60 FPS rendering usually can be easily achieved. In many cases, 60 FPS won't cause any problems, which will only improve with time, as mobile devices become more and more performant.

### • Improving performance

When it comes to comparing ActionScript and JavaScript, the latter is faster than the first one. Other than that, converting a game is a good occasion to revisit algorithms used in game code. With JavaScript game development you can optimise them or completely strip unused code that's left by original developers.

### • Fixing bugs and making improvements to the gameplay

Having new developers looking into game's source code can help to fix known bugs or discover new and very rare ones. This would make playing the game less irritating for the players, which would make them spend more time on your site and encourage them to try your other games.

### • Adding web analytics

In addition to tracking the traffic, web analytics can also be used to gather knowledge on how players behave in a game and where they get stuck during gameplay.

### • Adding localisation

This would increase the audience and is important for kids from other countries playing your game. Or maybe your game is not in English and you want to support that language?

## 2. Achieve 60 FPS

When it comes to JavaScript game development, it may be tempting to leverage HTML and CSS for in-game buttons, widgets and other GUI elements. Our advice is to be careful here. It's counterintuitive, but actually leveraging DOM elements is less performant on complex games and this gains more significance on mobile. If you want to achieve constant 60 FPS on all platforms, then resigning from HTML and CSS may be required.

Non-interactive GUI elements, such as health bars, ammo bars or score counters can be easily implemented in Phaser by using regular images (the 'Phaser.Image' class), leveraging the 'crop' property for trimming and the 'Phaser.Text' class for simple text labels.

Interactive elements such as buttons and checkboxes can be implemented by using the built-in 'Phaser.Button' class. Other, more complex elements can be composed of different simple types, like groups, images, buttons and text labels.

## 3. Loading custom fonts

If you want to render text with a custom vector font (eg TTF or OTF), then you need to ensure that the font has already been loaded by the browser before rendering any text. Phaser v2.6 doesn't provide a solution for this purpose, but another library can be used - Web Font Loader (<https://github.com/typekit/webfontloader>)

Assuming that you have a font file and include the Web Font Loader in your page, then below is a simple example of how to load a font. Make a simple CSS file that will be loaded by Web Font Loader (you don't need to include it in your HTML):

```
@font-face {
    // This name you will use in JS
    font-family: 'Gunplay';
    // URL to the font file, can be relative
    // or absolute
    src: url('../fonts/gunplay.ttf')
    format('truetype');
    font-weight: 400;
}
```

Now define a global variable named WebFontConfig. Something as simple as this will usually suffice:

```
var WebFontConfig = {
```

```
'classes': false,
'timeout': 0,
'active': function() {
// The font has successfully loaded...
},
'custom': {
'families': ['Gunplay'],
// URL to the previously mentioned CSS
'urls': ['styles/fonts.css']
}
};
```

Remember to put your code in the 'active' callback shown above. And that's it!

## 4. Save the game

Now we're in the middle point of our Flash to JavaScript conversion - it's time to take care of the shaders. To persistently store local data in ActionScript you would use the 'SharedObject' class. In JavaScript, the simple replacement is the localStorage API (<https://mzl.la/2n4WEJv>), which allows storing strings for later retrieval, surviving page reloads.

Saving data is very simple:

```
var progress = 15;
localStorage.setItem('myGame.progress',
progress);
```

Note that in the above example the 'progress' variable, which is a number, will be converted to a string.

Loading is simple too, but remember that retrieved values will be strings or null if they don't exist.

```
var progress = parseInt(localStorage.
getItem('myGame.progress')) || 0;
```

Here we're ensuring that the return value is a number. If it doesn't exist, then 0 will be assigned to the 'progress' variable.

You can also store and retrieve more complex structures, for example, JSON:

```
var stats = {'goals': 13, 'wins': 7,
'losses': 3, 'draws': 1
};
localStorage.setItem('myGame.stats', JSON.
stringify(stats));
...
var stats = JSON.parse(localStorage.
getItem('myGame.stats')) || {};
```

There are some cases when the 'localStorage' object won't be available. For example, when using the file:// protocol or when a page is loaded in a private window. You can use the 'try and catch' statement to ensure your code will both continue working and use default values, which is shown in the example below:

```
try {
    var progress = localStorage.
```

## Be careful

Each time you instantiate a 'Phaser.Text' or 'PIXI.Text' object, a new texture is created to render text onto. This additional texture breaks vertex batching, so be careful not to have too many of them.

# Tutorials

## Convert Flash games to HTML5



### Top

A custom default shader can be used to replace the tinting method in Phaser and PixiJS. The tanks flash white when hit

### Middle

The picture on the left shows how a player sees the game, while the one on the right displays the effect of applying the overdraw shader to the same scene

### Bottom

The left part of the image is a scene from a game, while the right side shows the same scene with the Phaser physics debug overlay displayed on top

```
getItem('myGame.progress');
} catch (exception) {
    // localStorage not available, use default values
}
```

Another thing to remember is that the stored data is saved per domain, not per URL. So if there is a risk that many games are hosted on a single domain, then it's better to use a prefix (namespace) when saving. In the example above, 'myGame.' is a prefix and you usually want to replace it with the name of the game.

If your game is embedded in an iframe, then localStorage won't persist on iOS. In this case, you would need to store data in the parent iframe instead.

## 5. Default fragment shader

When Phaser and PixiJS render your sprites, they use a simple internal fragment shader. It doesn't have many features because it's tailored for speed. However, you can replace that shader for your purposes. For example, you can leverage it to inspect overdraw or support more features for rendering. Below is an example of how to supply your own default fragment shader to Phaser v2.

```
function preload() {
    this.load.shader('filename.frag',
        'shaders/filename.frag');
}
function create() {
    var renderer = this.renderer;
    var batch = renderer.spriteBatch;
    batch.defaultShader =
        new PIXI.AbstractFilter(this.cache.
            getShader('filename.frag'));
    batch.setContext(renderer.g1);
}
```

## 6. Change tinting method

A custom default shader can be used to replace default tinting methods in Phaser and PixiJS. Tinting in Phaser and PixiJS works by multiplying texture pixels by a given colour. Multiplication always darkens colours, which obviously is not a problem; it's simply different from the Flash tinting. For one of our games we needed to implement tinting similar to Flash and decided that a custom default shader could be used. Below is an example of such a fragment shader:

```
// Specific tint variant, similar to the
// Flash tinting that adds
// to the color and does not multiply. A
// negative of a color
// must be supplied for this shader to work
// properly, i.e. set
// sprite.tint to 0 to turn whole sprite to
// white.
precision lowp float;
varying vec2 vTextureCoord;
varying vec4 vColor;
uniform sampler2D uSampler;
void main(void) {
    vec4 f = texture2D(uSampler,
        vTextureCoord);
    float a = clamp(vColor.a, 0.00001, 1.0);
```



## Want to learn more?

In this article, we have focused mainly on Phaser v2. However, a newer version of Phaser is now available. This is also worth checking out. It introduced a plethora of features, such as multiple cameras, scenes, tilemaps and the Matter.js physics engine.

If you are brave enough and want to create truly remarkable things in browsers, then WebGL is the right thing to learn from the ground up. It's a lower level of abstraction than various game-building frameworks or

tools, but it allows you to achieve greater performance and quality even if you work on 2D games or demos. Among the websites you may find useful when learning the basics of WebGL is WebGL Fundamentals (<https://webglfundamentals.org>), which uses interactive demos. In addition to that, to find out more about WebGL feature adoption rates check WebGL Stats (<http://webglstats.com>). And remember, there's no such thing as too much knowledge.

```
gl_FragColor.rgb = f.rgb * vColor.a +
clamp(1.0 - vColor.rgb/a, 0.0, 1.0) *
vColor.a * f.a;
gl_FragColor.a = f.a * vColor.a;
}
```

This shader lightens pixels by adding a base colour to the tint one. For this to work, you need to supply negatives of the colour you want. Therefore, in order to get white, you need to set:

```
sprite.tint = 0x000000; // This colors the
sprite to white
Sprite.tint = 0xffff; // This gives red
```

## 7. Inspect overdraw

Replacing a default shader can also be leveraged to help with debugging. Below we've explained how overdraw can be detected with such a shader.

Overdrawing happens when many or all pixels on the screen are rendered multiple times. For example, many objects taking the same place and being rendered one over another. How many pixels a GPU can render per second is described as fill rate. Modern desktop GPUs have excessive fill rate for usual 2D purposes, but mobile ones are a lot slower.

There is a simple method of finding out how many times each pixel on the screen is written by replacing the default global fragment shader in PixiJS and Phaser with this one:

```
void main(void) {
    gl_FragColor.rgb += 1.0 / 7.0;
}
```

This shader lightens pixels that are being processed. The number 7.0 indicates how many writes are needed to turn pixels white; you can tune this number to your liking. In other words, lighter pixels on screen were written several times, and white pixels were written at least seven times.

This shader also helps to find both 'invisible' objects that for some reason are still rendered, and sprites that have excessive transparent areas around that need to be stripped (GPU still needs to process transparent pixels in your textures).

## 8. Why physics engines are your friends

A physics engine is a middleware that's responsible for simulating physics bodies (usually rigid body dynamics) and their collisions. Physics engines simulate 2D or 3D spaces, but not both. A typical physics engine will provide:

- Object movement by setting velocities, accelerations,

joints, and motors;

- Detecting collisions between various shape types;
- Calculating collision responses, ie how two objects should react when they collide.

There is a Phaser plugin that works well for this purpose. Box2D is also used in the Unity game engine and GameMaker Studio 2.

While a physics engine will speed up your development, there is a price you'll have to pay: reduced runtime performance. Detecting collisions and calculating responses is a CPU-intensive task. You may be limited to several dozen dynamic objects in a scene on mobile phones or face degraded performance, as well as reduced frame rate deep below 60 FPS.

## 9. Export sounds

If you have a Flash game sound effects inside of a .fla file, then exporting them from GUI is not possible (at least not in Adobe Animate CC 2017) due to the lack of menu options serving this purpose. But there is another solution - a dedicated script that does just that:

```
function normalizeFilename(name) {
    // Converts a camelCase name to snake_case name
    return name.replace(/([A-Z])/g, '_$1').replace(/\u00a0/, '').toLowerCase();
}

function displayPath(path) {
    // Makes the file path more readable
    return unescape(path).replace('file:///','').replace('|', ':');
}

f1.outputPanel.clear();
if (f1.getDocumentDOM().library.getSelectedItems().length > 0)
    // Get only selected items
    var library = f1.getDocumentDOM().library.getSelectedItems();
else
    // Get all items
    var library = f1.getDocumentDOM().library.items;
// Ask user for the export destination directory
var root = f1.browseForFolderURL('Select a folder.');
var errors = 0;
for (var i = 0; i < library.length; i++) {
    var item = library[i];
    if (item.itemType !== 'sound')
```

```
        continue;
    var path = root + '/';
    if (item.originalCompressionType === 'RAW')
        path += normalizeFilename(item.name);
    split('.')[0] + '.wav';
    else
        path += normalizeFilename(item.name);
    var success = item.exportToFile(path);
    if (!success)
        errors += 1;
    f1.trace(displayPath(path) + ': ' +
(success ? 'OK' : 'Error'));
}
f1.trace(errors + ' error(s)');
```

How to use the script to export sound files:

- Save the code above as a .jsfl file on your computer.
- Open a .fla file with Adobe Animate.
- Select Commands>Run Command from the top menu and select the script in the dialogue that opens.
- Now another dialogue file pops up for selecting the export destination directory.

And done! You should now have WAV files in the specified directory. What's left to do is convert them to, for example, MP3, OGG or AAC.

## 10. How to use MP3s

The good old MP3 format is back, as some patents have expired and now every browser can decode and play MP3s. This makes development a bit easier, since finally there's no need to prepare two separate audio formats. Previously you needed, for instance, OGG and AAC files, while now MP3 will suffice.

Nonetheless, there are two important things you need to remember about MP3s:

- MP3s need to decode after loading, which can be time-consuming, especially on mobile devices. If you see a pause after all your assets have loaded, then it probably means that MP3s are being decoded;
- gaplessly playing looped MP3s is a little problematic. The solution is to use mp3loop, read more in this article posted by Compu Phase (<https://bit.ly/2KKCEyH>).

## Complex shaders

It's important to remember that the default shader is used for all sprites, as well as when rendering to a texture. Also, keep in mind that using complex shaders for all in-game sprites will greatly reduce rendering performance.

# Code 3D interactive models with WebGL

Learn how to create and build personal interactive experiences in 3D with the Jeeliz Face Filter head tracking library





Augmented Reality (AR) is hitting social media, creating all kinds of photo opportunities – and web designers don't have to miss out on the action.

The fantastic Jeeliz FaceFilter API ([jeeliz.com](http://jeeliz.com)) is a library that sits over a slightly modified version of ThreeJS to enable full 3D head tracking in the browser. Because Jeeliz FaceFilter uses WebRTC you will need to serve it over a secure HTTPS server to your devices. The library detects faces and works out the three-dimensional transformation and uses the ThreeJS library to load and display 3D models over the top. This means that you can display any 3D model, tracked onto the face position.

In this tutorial, we're going to create a promotional web site that could be used to generate interest in a forthcoming event. In this particular case, it's going to be for a special Viking Museum exhibition. It shouldn't take much effort to enable users to save their image, and this could help generate a lot of great publicity.

You could easily adapt this technique for many different use cases – imagine using it to create a great web-based application for a band on tour, for example, where the members of that band have distinctive hairstyles or hats (Slash from Guns N' Roses springs to mind). The possibilities this technique affords are only limited by your imagination – and the content works equally well across smartphones, tablets and computers using modern web browsers.

## 1. Starting the project

From the Project Files folder, open the Start folder in your code IDE. Open up the 'index.html' page to begin the process. In the head section add in the script tags to link up all the libraries needed. Jeeliz will detect faces, there is a modified version of ThreeJS and the library to load Collada models. The 'scene.js' is where our code will go later and there is some CSS, too.

```
<script type="text/javascript" src="js/
jeelizFaceFilter.js"></script>
<script type="text/javascript" src="js/
threeModified.js"></script>
<script type="text/javascript" src="js/
```

```
ColladaLoader.js"></script>
<script type="text/javascript" src="scene.
js"></script>
<link rel="stylesheet" href="css/style.css">
```

## 2. Adding the body content

Below the head section of the page add the content for the body section. As you can see there is an onload event which will call the 'main()' function in the JavaScript. The 'div' structure places some overlays on top of the 'canvas' element where the face detection will be added.

```
<body onload="main()" style='color: white'
<canvas width="600" height="600"
id="jeeFaceFilterCanvas"></canvas>
<div id="viking">
<div id="preload">

<br>Loading Please Wait
</div>
```

## 3. Finishing the body

Now there is just some final content added to the bottom of the page that gives the context of what this page is all about, and how the face detection fits in with the theme. This is for an imaginary exhibition that might need some interesting promotion to get visitors.

```
<div id="imageWrap"></div>
</div>
```

## 4. Moving to the CSS

Now all the content code is added into the HTML, save the page and open 'styles.css' from the CSS folder. Here the basic styling of the page and the canvas will be added so that the canvas element will fill the full screen of the browser on the page.

```
body {
  overflow: auto;
  overflow-y: auto;
  margin: 0px; }
```

```
#jeeFaceFilterCanvas {
  z-index: 10;
  position: absolute;
  width: 100%;
  height: 100%;
  top: 0px;
  left: 0px; }
```

## 5. Multiple backgrounds

The Viking element is a container that will be placed over the top. In this will be the preloader and page design. Here the images are added as multiple background elements. A decorative strip is placed along the top and bottom to theme the page and then a gradient image added just to make it a little darker at the bottom.

```
#viking {
  width: 100%;
  height: 100%;
  position: absolute;
  top: 0;
  left: 0;
  z-index: 200;
  background: url(..../img/braid.png) center
  top repeat-x, url(..../img/braid.png) center
  bottom repeat-x, url(..../img/base.png) center
  bottom repeat-x; }
```

## 6. Preloading the content

While all the libraries, 3D models and images are loading, a screen will be placed over the top to tell people what is happening. You can see that this is positioned into place at a higher 'z-index' and the typography is set. It's positioned in the centre of the screen.

```
#preload {
  height: 360px;
```

## Model formats

There are a wide variety of different model formats that can be loaded through ThreeJS, check out [threejs.org/examples/](http://threejs.org/examples/) to see if there is a preferential format for your content.



### Left

To continue the 3D theme of the project the graphics have been created in Cinema4D and rendered out as a transparent PNG for use on the web

### Top

The DIV element that sits over the top of the canvas has the background pattern, shown here, repeated at the top and bottom of the page

## Tutorials

### Code 3D interactive models with WebGL

```
width: 100%;  
margin: auto;  
position: absolute;  
top: 50%;  
color: #999;  
font-family: "HelveticaNeue-Light",  
"Helvetica Neue Light", "Helvetica Neue",  
Helvetica, Arial, "Lucida Grande", sans-  
serif;  
font-weight: 300;  
transform: translateY(-50%);  
font-size: 2em;  
text-align: center;  
z-index: 200;  
}
```

## 7. Positioning at the bottom

In order to get the position of the title at the bottom of the page it's placed inside a relatively positioned DIV. This is given the full height of the page and the image is placed inside, giving a little room for it to sit above the bottom of the browser window. Save the CSS and check everything is working in the browser.

```
#imageWrap {  
    width: 100%;  
    max-width: 500px;  
    height: 100%;  
    position: relative;  
    margin: 0 auto;  
}  
  
#vikingImg {  
    width: 100%;  
    max-width: 500px;  
    position: absolute;  
    bottom: 10px;  
}
```

## Placing the model

It's easy to forget, but the 3D model of the helmet is not being placed around the head, it's being placed in front of the face video image and transformed to match the angle of the face.



#### Top

At this point all of the HTML and CSS are in place so you should see the preloader holding screen, this will be removed in the JavaScript once the page has loaded

#### Right

This is one of the six images that make up the cube map which is the environment that creates the reflection on the helmet, making it shiny

## 8. Setting up the JavaScript

Move to the 'scene.js' file now and you will see that there is already some code set up in here for the main function. All the new code will be added above what is here. The first part of this is to ensure that all the settings are correct, this is probably what will change the most when you add your own content. This sets the model file and the folder with the cube map.

```
var SETTINGS = {  
    daeModelURL: 'model/helmet.dae',  
    cubeMapURL: 'scene/',  
    rotationOffsetX: -0.1,  
    cameraFOV: 40,  
    pivotOffsetYZ: [0.3, 0.3],  
    detectionThreshold: 0.3,  
    detectionHysteresis: 0.05,  
    offsetYZ: [0.68, 0],  
    scale: 2.9  
};
```

## 9. Detecting faces

Now there are some global variables for the project to work that will be used across multiple functions. Then the face detection function runs and sets the 'THREEFACEOBJ3D' variable to 'true' or 'false' if the face is detected. This will be used later on to place the 3D object on to the face.

```
var THREEVIDEOTEXTURE, THREERENDERER,  
    THREEFACEOBJ3D, THREEFACEOBJ3DPIVOTED,  
    THREESCENE, THREECAMERA, CANVASELEMENT;  
var ISDETECTED = false;  
function detect_callback(isDetected) {  
    if (isDetected) {  
        THREEFACEOBJ3D.visible = true;  
        console.log('INFO in detect_  
callback() : DETECTED');  
    } else {  
        THREEFACEOBJ3D.visible = false;  
        console.log('INFO in detect_  
callback() : LOST');  
    }  
}
```

## 10. Setting up the scene

The scene is set up in the 'init' function, so all elements pertaining to that are added here. A renderer is set to WebGL mode and some empty objects are created. 'frustumCulled' simply refers to detecting whether these elements are in the camera's view port or not.

```
function init_threeScene(spec) {  
    CANVASELEMENT = spec.canvasElement;  
    THREERENDERER = new THREE.WebGLRenderer({  
        context: spec.GL,  
        canvas: CANVASELEMENT  
    });  
    THREEFACEOBJ3D = new THREE.Object3D();  
    THREEFACEOBJ3D.frustumCulled = false;  
    THREEFACEOBJ3DPIVOTED = new THREE.  
Object3D();  
    THREEFACEOBJ3DPIVOTED.frustumCulled =  
    false;
```

## 11. Pivoting

The pivot point of the object is set based on the settings in Step 8. This just positions the pivot point slightly differently to match where the neck and head of the person intersect rather than the model's own pivot point. The path of the cube map is set and these are all JPG images.

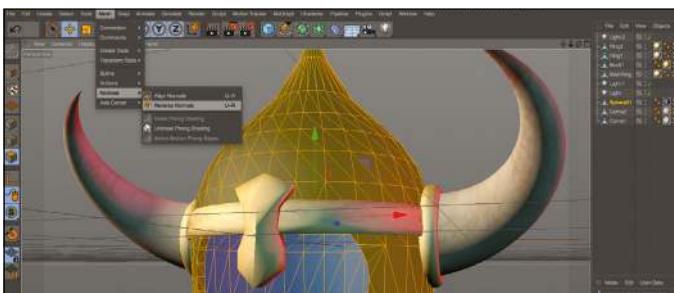
```
THREEFACEOBJ3DPIVOTED.position.set(0,  
-SETTINGS.pivotOffsetYZ[0], -SETTINGS.  
pivotOffsetYZ[1]);  
THREEFACEOBJ3D.add(THREEFACEOBJ3DPIVOTED);  
var path = SETTINGS(cubeMapURL;  
var format = '.jpg';
```

## 12. Loading the cube map

The cube map gives the helmet its shiny reflective image. Here each of the six images necessary to form a cube are loaded and stored in the variable 'envMap'. These will be added to the model when it loads as the reflection or 'environment', hence the name 'envMap'.

```
var envMap = new THREE.CubeTextureLoader().  
load([  
    path + 'posx' + format, path + 'negx' +
```





## Seeing the right geometry

When you are creating your 3D content for use with head tracking, you will want to make sure that you do not create double-sided objects. The way models work in 3D is that they have one side which is 'seen' by the renderer – this is known as the 'normal'. Even if your 3D modelling package shows both sides, you will only see one when you export it. For making a helmet model this is important. The back of the helmet

should be visible if you turn your head but not on the inside as this would obscure the face. Your modelling package should allow you to see the normal side.

In the image (left), they are shown as yellow polygons so that you know the outside is visible, if it isn't then there will be an option to 'reverse normals' and if only some are facing the wrong way then you can 'align normals' first.

```
format,
  path + 'posy' + format, path + 'negy' +
format,
  path + 'posz' + format, path + 'negz' +
format
});
```

## 13. Loading the model

The model is now being loaded into the code. You will see that once the model has been loaded, the environment map is added to all the materials in the scene. The model is added to an empty box in order to set the positioning to match the position of the head.

```
var loader = new THREE.ColladaLoader();
loader.load(SETTINGS.daeModelURL, function (collada) {
  collada.scene.traverse(function (child) {
    if (child.isMesh) {
      child.material.envMap = envMap;
    }
  });
  collada.scene.frustumCulled = false;
  var bbox = new THREE.Box3();
  expandByObject(collada.scene);
  var centerBBox = bbox.getCenter(new THREE.Vector3());
```

## 14. Repositioning the object

The model is repositioned to the settings offset from Step 8. The scene is also scaled to fit. If you want to customise this with your own model, then the scale variable in the settings will be the place, which you will need to change the size of the object so that it fits on to the head.

```
collada.scene.position.add(centerBBox.multiplyScalar(-1));
collada.scene.position.add(new THREE.Vector3(0, SETTINGS.offsetYZ[0], SETTINGS.offsetYZ[1]));
var sizeX = bbox.getSize(new THREE.Vector3()).x;
collada.scene.scale = multiplyScalar(SETTINGS.scale / sizeX);
THREEFACEOBJ3DPIVOTED.add(collada.scene);
```

## 15. Setting up the video

The 3D scene is created and the model is added into this scene. The video needs to run in the background of the scene. To do this the video is taken from the webcam and is turned into a texture that can be placed on to a quad mesh to fill the screen. It requires a lot of code.

```
THREESCENE = new THREE.Scene();
THREESCENE.add(THREEFACEOBJ3D);
THREEVIDEOTEXTURE = new THREE.DataTexture(new Uint8Array([255, 0, 0]), 1, 1, THREE.RGBFormat);
THREEVIDEOTEXTURE.needsUpdate = true;
var videoMaterial = new THREE.RawShaderMaterial({
  depthWrite: false,
  depthTest: false,
```

## 16. Making a shader

A shader is a program that runs on the graphics card, very fast. Because all the pixels are calculated at once, or in parallel, it's written in a language similar to C. A vertex shader can change the shape of objects, but it isn't doing so here; it's just mirroring the webcam image.

```
vertexShader: "attribute vec2 position;\nvarying vec2 vUV;\nvoid main(void){\n  gl_Position=vec4(position, 0., 1.);\n  vUV=0.5+vec2(-0.5,0.5)*position;\n  //inverse X axis for mirror\n}
```

## 17. Shading fragments

Now the fragment shader is written and can be thought of as writing all the pixels inside the shape of an object or between the geometry. As such this is again written in the C-like language. This adds the video image into the 'videoMaterial' variable for use later.

```
fragmentShader: "precision lowp float;\nuniform sampler2D samplerVideo;\nvarying vec2 vUV;\nvoid main(void){\n  gl_FragColor=texture2D(samplerVideo,\n  vUV);\n}
```

## 18. Pressing the uniform

A 'uniform' is a consistent input into the shader program. In this case it is the video texture from the webcam since this is what needs to be processed and spat out onto the background of the scene. This will be placed on to geometry in the following step. All ThreeJS objects consist of a geometry and a material.

```
uniforms: {
  samplerVideo: {
    value: THREEVIDEOTEXTURE
  }
},
```

## 19. Creating the video geometry

The video geometry is created and it is set to be

positioned into the screen corners so that it fills the screen. The video object is created out of the geometry and the material so that it is a visible object in the scene. A new function is created at the end of this code.

```
var videoGeometry = new THREE.BufferGeometry();
var videoScreenCorners = new Float32Array([-1, -1, 1, -1, 1, 1, -1, 1]);
videoGeometry.addAttribute('position', new THREE.BufferAttribute(videoScreenCorners, 2));
videoGeometry.setIndex(new THREE.BufferAttribute(new Uint16Array([0, 1, 2, 0, 2, 3]), 1));
var videoMesh = new THREE.Mesh(videoGeometry, videoMaterial);
videoMesh.onAfterRender = function () {
```

## 20. Updating the texture

The texture is told that it is updated every single frame – most textures don't do this as they are just mapped onto the object, but being a video it needs to change. Some filtering is done to smooth the edges of the pixels out and make it appear better, since the webcam image is being scaled up considerably.

```
THREERENDERER.properties.update(THREEVIDEOTEXTURE, '__webglTexture', spec.videoTexture);
THREEVIDEOTEXTURE.magFilter = THREE.LinearFilter;
THREEVIDEOTEXTURE.minFilter = THREE.LinearFilter;
delete(videoMesh.onAfterRender);};
```

## 21. Finishing up

The webcam video object is placed at the front of the render queue so that it is rendered first, then the model will be placed over the top when a head is detected in the screen. A perspective camera is added to the scene to match the camera from the webcam. Finally, the preloader is turned off so that this screen is visible.

```
videoMesh.renderOrder = -1000;
videoMesh.frustumCulled = false;
THREESCENE.add(videoMesh);
THREECAMERA = new THREE.PerspectiveCamera(SETTINGS.cameraFOV, 1, 0.1, 100);
setFullScreen();
document.getElementById("preload").style.visibility = 'hidden';}
```

# web workshop

## Create a large fade-in content navigation

Inspired by [www.hinderlingvolkart.com](http://www.hinderlingvolkart.com)

### Animated background

Full-screen navigation is complemented by a video background that helps to increase the visual appeal of the webpage presentation.

### GDPR notification

This is a smart approach to presenting the data privacy notification for GDPR compliance without being too intrusive.

### Sub navigation items

A secondary navigation is presented with a smaller font size, allowing lesser visibility for options that aren't the main focus.



### Main navigation

The main navigation options are presented with a font size that clearly stands out from the background and any other page content.

### Link containers

Each navigation link uses spacing to clearly separate themselves from other items. It's also used as part of the intro animation.

## Create a large fade-in content navigation

◀ DOWNLOAD TUTORIAL FILES [www.filesilo.co.uk/webdesigner](http://www.filesilo.co.uk/webdesigner)

### EXPERT ADVICE

#### Focused navigation

Full-screen navigation is a useful approach for content design where the only focus is to get people to the content they want. The absence of other content avoids any scope for distraction, but may make the page appear to be a bit boring. This problem is solved with the application of animation.



#### <comment>

What our experts think of the site

#### A tool for simplification

Increasing complexity of web content and its associated functionality requires presentation to allow people to easily make sense of what's being presented. Navigation is a tool that allows people to avoid irrelevant content and functionality so that they can more easily understand what's relevant.

**Leon Brown, developer and author of e-learning content at nextpoint.co.uk**

## Technique

### 1. Initiate HTML document

The first step is to define the structure of the HTML document that becomes the webpage. This consists of a HTML container for the head and body sections. The head section references to the external CSS and JavaScript resources, while the body section stores the page content created in step 2.

```
<!DOCTYPE html>
<html>
<head>
<title>Fade In Navigation</title>
<link rel="stylesheet" type="text/css" media="screen" href="styles.css"/>
<script src="code.js" type="text/javascript"></script>
</head>
<body>
*** STEP 2 HERE
</body>
</html>
```

### 2. Page content

The webpage content consists of a navigation container that stores a collection of links. For the fade in effect to be possible, the text inside the links need to be presented as a child element - hence the use of span tags. This allows CSS to control the text independently of their link container.

```
<nav>
<a href="#"><span>One</span></a>
<a href="#"><span>Two</span></a>
<a href="#"><span>Three</span></a>
<a href="#"><span>Four</span></a>
</nav>
```

### 3. CSS navigation

With the HTML now complete, create a new file called 'styles.css'. Each link inside the navigation needs to have a default colour and padding to show spacing. Hidden overflow is also applied so that the span text can appear out of view.

```
nav a{
display: block;
padding: 1em;
color: #000;
font-size: 3em;
text-decoration: none;
overflow: hidden;
background: transparent;
text-align: center; }
```

### 4. Navigation extras

The remaining settings for the navigation require a border to appear on link items when they are hovered. Additionally, span items inside the links need to have a negative top margin that moves them out of view from the link container. The 'menuOpen' animation is applied to occur over one second.

```
nav a:hover{
border: .01cm solid #000
}
nav > a > span{
display: block;
margin-top: -2em;
animation: menuOpen 1s forwards;
}
```

### 5. Animation definition

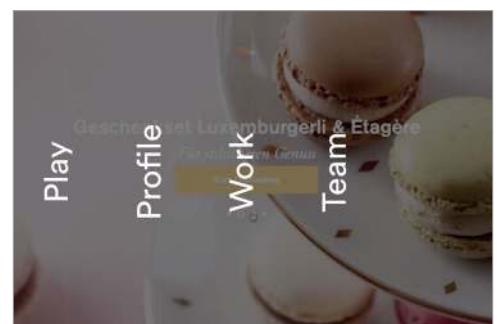
The 'menuOpen' animation referenced in the previous step uses just one frame. By using 'to', keyframes will instruct the browser to animate items from step 4 towards having a top margin value of zero. With links using a hidden overflow, this results in the menu link text appearing into view.

```
@keyframes menuOpen {
to{ margin-top: 0; }
}
```

### 6. Animation delays

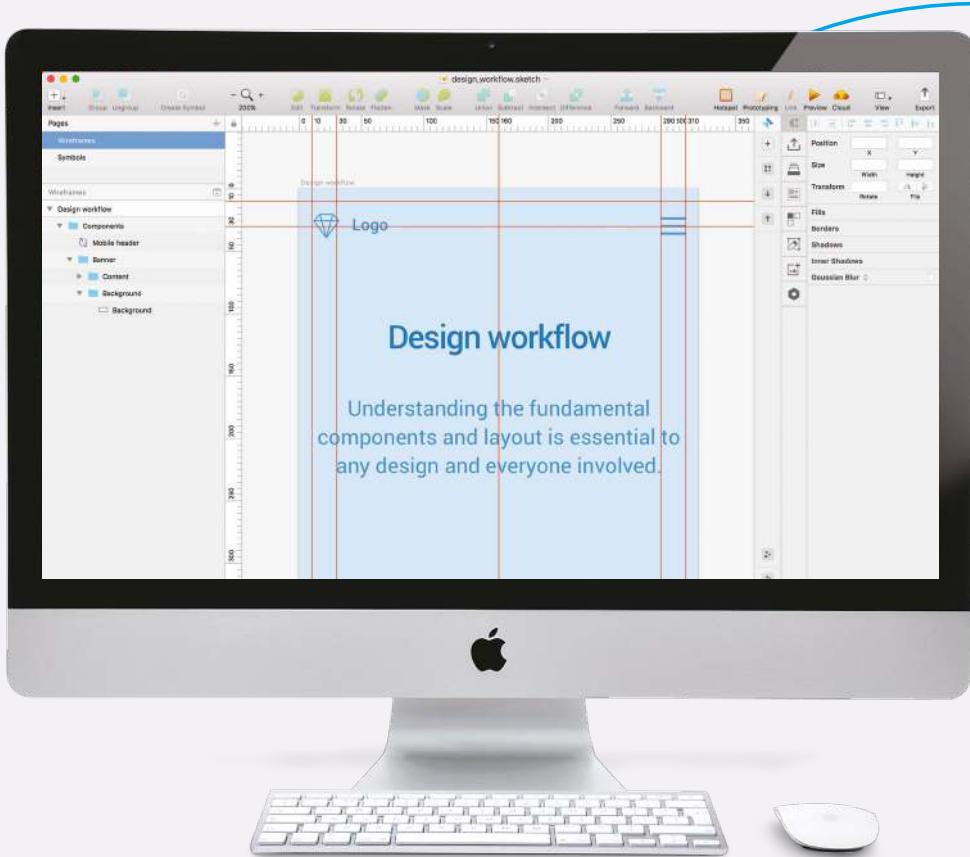
The CSS applies the animation for all link text to start at the same time. A unique delay is required for each text item to appear animated afterwards. JavaScript is used to find each navigation span and calculate their 'animationDelay' based on their index position.

```
window.addEventListener("load", function(){
var nodes = document.querySelectorAll("nav > a > span");
for(var i=0; i<nodes.length; i++){
nodes[i].style.animationDelay = (i/2)+"s";
}
});
```



# STREAMLINE YOUR DESIGN WORKFLOW

A design process is a method of steps to ensure you and your team create great products. Take a look at the tools and techniques used by the pros on a daily basis



## CREATE A WIREFRAME IN SKETCH

Strip back your design and present only the fundamental components

To some, a wireframe may seem like a laborious and unnecessary task that can drag out a project with no real long-term benefit. Jumping straight into a design may seem like a compelling and rewarding alternative, but by doing so you could be forgetting the most important thing of all when creating a design: the user!

The ideation and wireframing stage of any design enables you to consider the layout and user experience from the very start of your project. By only using the core components for your web design – header, footer, navigation, buttons – to begin with, you can always ensure your focus is on the user, without getting distracted by which colour palette you had in mind.

There are plenty of tools out there to get started with wireframing, but today many web teams use Sketch. Its usability and unrivalled speed make it the obvious choice.

So a project has been set up, and a font has been chosen for the wireframe, but what should be added? This is where the core components are decided and added to the wireframe – the

components that the design can't live without.

Starting with the header, what do we need if we are approaching this project mobile first? Is the logo the main focus? Should we consider using a 'burger' navigation to ensure best practice? These are the questions any designer should keep asking themselves throughout the

**Symbols act as a super component**

process when building their wireframe components.

The first steps should be to create an artboard for mobile or desktop, and begin to build the components on the page with simple shapes, in order to define the outline of the components, like the header. To highlight areas of importance and interactivity in the wireframe, use varying shades of a

monochrome colour palette whilst wireframing. For example, within the header, inserting a logo and navigation in a bold blue colour, on a pastel blue background, will present them as the important elements in your component.

It's also important to consider a grid system, padding and margins as soon as possible when defining the wireframes. Showing rulers in a Sketch file (Ctrl+R) and clicking on these rulers (down the left and above your artboard) will create guides, which in turn saves a lot of stress when sharing designs, as consistency is key when defining a design system.

When the essentials are in place for one component, move onto the next. Only include what you need. Keep in mind the bare minimum that is needed for the user to complete their task! Planning to use these components in more than one artboard? Why not create a symbol. Symbols act as super-components that update all the artboards they currently sit in. To create a component, right-click an element, and choose 'Create symbol'. This will be saved into one packaged item manageable from the symbols page.

## 3 important wireframe techniques in Sketch

### 1. Define your guides

Every design, unfortunately, needs rules from time to time, make sure you add guides by going to View > Canvas > Show Rulers or hit Ctrl+R. Don't have any guides? Add some by clicking on the rulers, and then make sure that they are consistent.

### 2. Create symbols

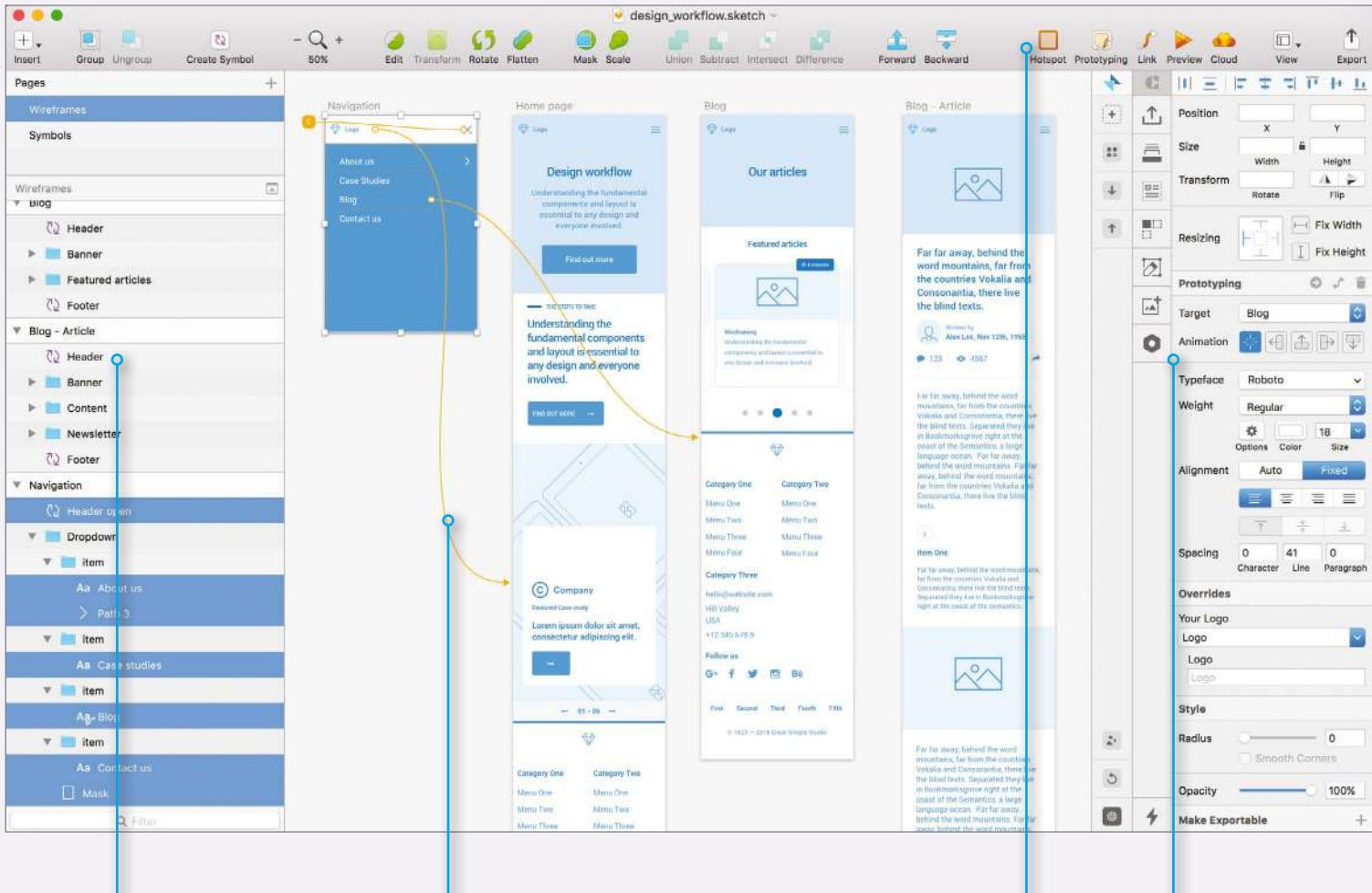
Symbols are by far the best feature in Sketch. Right-click on any of your assets and then click 'Symbol' to create a universal asset that updates across all your artboards when changed.

### 3. Use a monochrome colour system

Too many colours can be distracting, so minimise your colour palette to clearly define important elements in your wireframes. Highlighting the important stuff (call-to-action buttons, header text) with bolder colours is a subtle yet effective way of creating a visual hierarchy when sharing the designs with colleagues and clients.

# PROTOTYPING IN SKETCH

Bring your wireframes to life by following these simple steps



### Prototyping links

If you want to know where your prototype links are going, highlight them to view the prototype arrows built into Sketch.

### Symbol links

You can add prototyping links from inside symbols, which will always go to the same artboards, but you also have the option to overwrite.

### Prototyping GUI

Use these icons to quickly build your prototype, and click Preview to get a live linking version.

### Animation effects

Supercharge your prototypes by adding app-like animations, available in the the prototyping panel.

## 3 great ways to improve your Sketch prototypes

### 1. Animations

Underneath the target area on the right-hand panel there is the option to add animations between your artboards.

Using animations is a great way to add more life to your prototype. However, the current options are more tailored towards app prototypes, and don't offer much flexibility if demonstrating a website design. That being said,

it's expected the list will grow in future updates of Sketch with options like fading transitions, for example.

### 2. Flag your starter page

When previewing a prototype, the flag icon that sits in the header of the popup next to the select dropdown is a great way to indicate which of the artboards is the home page. Once selected,

every time the prototype is opened, this artboard will be the first page. This will also set the homepage when sharing the prototype in the cloud.

### 3. Quick tip — link colours

It's worth noting that it's possible to update the link colours in your Sketch preferences, just in case there is a colour clash in a design with the default orange.

# COLLABORATING IN SKETCH

Share, collaborate and comment on the designs using the Sketch Cloud feature

## Sketch Cloud explained

Sketch's latest cloud feature is an online showcase of all pages and artboards in one place. Anyone can sign up to use Sketch Cloud to view, download, and comment on Sketch files, which have been shared publicly or privately straight from Sketch.

Pages act as sections on your Sketch Cloud link and are displayed in order of their structure in Sketch (from top to bottom). Artboards within these pages do the same, so make sure the structure is correct (for artboards, order from left to right) to show designs in a specific order.

It's worth noting that this is also a great platform to perform some user testing once you've shared your cloud link.

## How to use it

Sketch have outdone themselves with the free sharing and cloud feature. When the prototype is complete, click on the Cloud icon, sign up with an account, and upload the designs to the cloud.

From this point onwards, the link will remain the same, so any future amends added to the project will be uploaded when clicking on the cloud icon and selecting Update in the share popup.

The dashboard on the cloud is self-explanatory with prototypes, artboards and symbols available to view in order.

### Prototypes

If you viewed your prototype in your Sketch file earlier, and flagged one of your pages as the starting page, Sketch cloud will create a new section with this

prototype ready to go!

Can't see a prototype? Simply go back to your Sketch file, highlight the artboard you want to be the starting page, click Preview so you see the popup, and finally click the flag icon. On your next push to the

cloud, this prototype will be waiting for you!

## Comments and Sharing

All users with access can leave comments on each artboard, enabling all feedback to be kept in one place, so there's no need for that long-winded email anymore.

If you are looking to get feedback on your prototype or designs, or want to share the prototype on a public link, click on the cog icon in the top-right corner of your cloud dashboard. Here, you can enable

comments (which will appear once viewing a design by clicking the bottom-right icon), and create a public link for your designs for sharing. You can keep the designs private, and give access via email to a lucky few.

# Essential shortcuts



## Pushing to cloud

Push your latest changes to cloud by selecting this icon – your browser will open the link once completed.

## Hide your symbols

Declutter your Sketch cloud link by selecting only the page you want to show in the top left dropdown, and share the URL from here instead.



## Keep your audience in the loop

In the settings panel popup, tick 'Use as library' to automatically let your team know there's been a change.

## Show your prototypes first

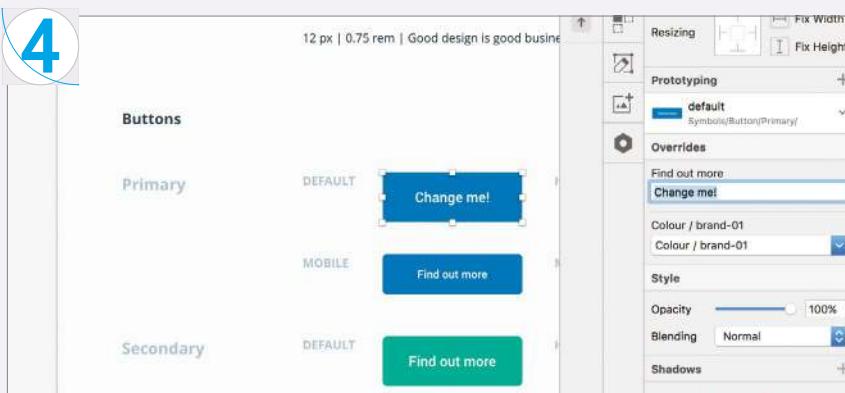
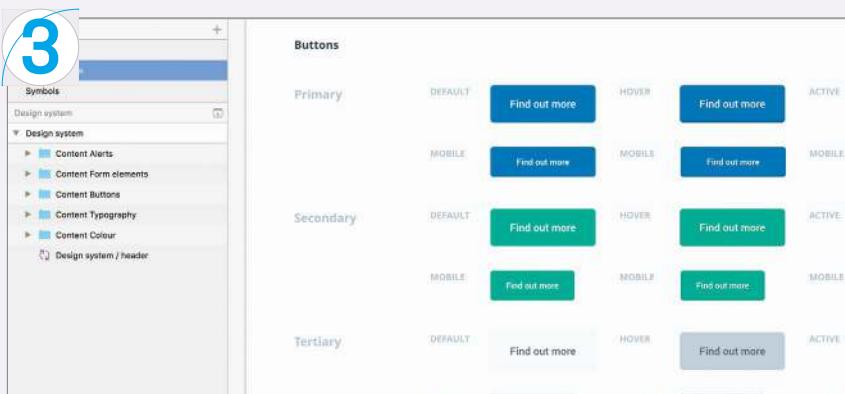
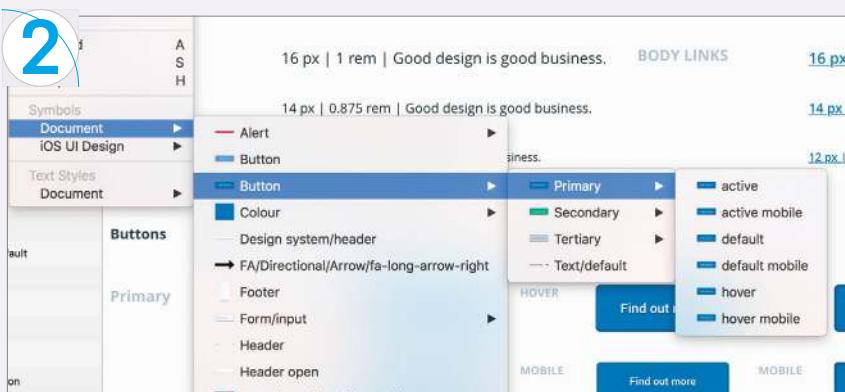
Flag your opening artboard in preview mode before pushing to cloud to make it appear first on your Sketch cloud link.

The screenshot shows the Sketch application window with the following sections:

- Wireframes:** Displays four wireframe prototypes labeled "Navigation", "Home page", "Blog", and "Blog - Article".
- Symbols:** Displays a list of symbols including "Logo", "About us", "Case Studies", "Blog", and "Contact us".
- Cloud View:** Shows a list of documents under "Documents" with the file "design\_workflow" selected. It includes a "Preview" button and a "Share" icon.
- Top Bar:** Includes icons for Prototyping, Link, Preview, Cloud, View, and Export.

# BUILD A DESIGN SYSTEM IN SKETCH

Generate a coherent user experience by building a design system



## 1. Define your colour palette

Keep your brand colours organised in your document palette by choosing the said colour and clicking the '+' icon.

## 2. Nested symbols

Keep your favourite symbols organised in their multiple states by using slashes when naming your symbols, for example naming your symbols 'Button / Active' and 'Button / Disabled' will group your symbols together under the same category. It's that easy!

## 3. Create a design system page

Separating all your elements that define your brand into its own page in Sketch is a simple way to show everyone where they can grab the assets.

## 4. Symbol overrides

You can overwrite the content in your symbols, such as text in your button (see example), so you don't need to worry about making new elements if it's only a small change.

# Popular design systems

## Google's Material Design system

[material.io/design/](https://material.io/design/)  
Google created a very well put together design system called Material, which is full to the brim with great information.

## IBM Carbon design system

[carbondesignsystem.com/](https://carbondesignsystem.com/)  
Carbon is the design system for IBM Cloud products. It is a series of individual styles, components, and guidelines used for creating unified UI.

## MailChimp grid system

[ux.mailchimp.com/patterns/](https://ux.mailchimp.com/patterns/)  
If you're looking for a more web-focused design system, MailChimp offers a perfect insight in to all the elements you need to get started on your own.

## Apple Human Interface Guidelines

[apple.co/2HY5vtf](https://apple.co/2HY5vtf)  
A very complex but thorough example of a design system across multiple devices.

## Lonely Planet

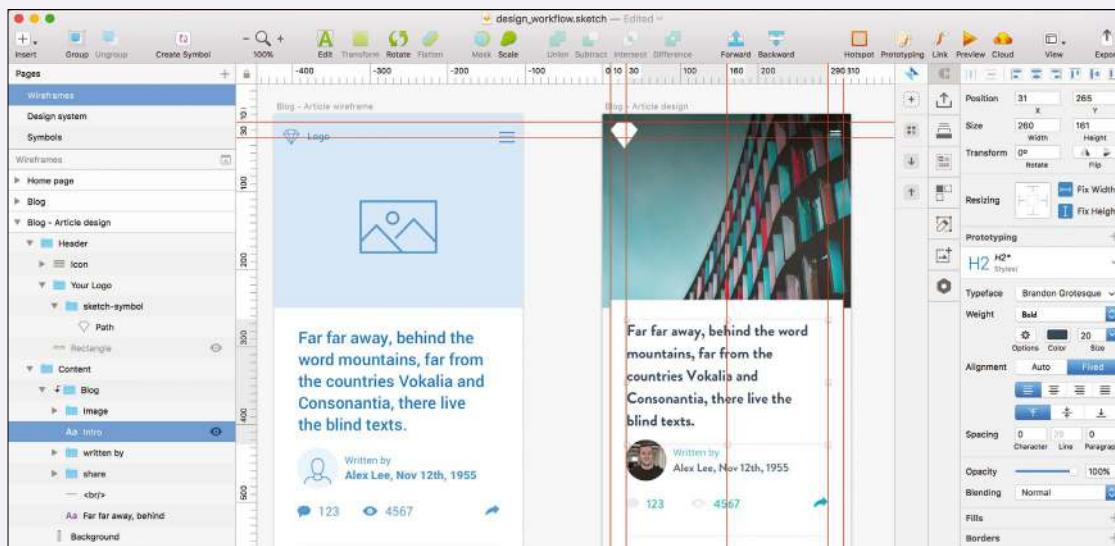
[bit.ly/2KAqmbq](https://bit.ly/2KAqmbq)  
If you're looking for a detailed colour and icon design system, check out Lonely Planet's design elements.

## GitLab Design System

[design.gitlab.com/](https://design.gitlab.com/)  
An open-source design guideline for web apps, which covers everything from their iconography, through to the use of popup windows.

# CREATE A PAGE DESIGN

By using the wireframes and design system as a guide, you can piece together a well-planned page design



Consistency transforms a good design into a great design, and following the wireframe and design system ensures that the look, feel and user experience of the product isn't compromised.

## Visual hierarchy

A strong visual hierarchy ensures clarity in any user-centric design. To get started, design the most universal components – such as colour and typography – first, and then work down to smaller ones such as buttons and input components. To insert just about anything into your design, hover over to Insert tab at the top of the Sketch UI, click, and then start importing elements onto your artboards. Keep in mind your wireframe and design system to ensure consistency in your work.

## Colour

Colour is arguably the most important element in any design workflow. With colour, we can set the overall look, feel and tone of a design, so always ensure you use it in correlation to the importance of the elements that they are assigned to. Try grouping your colours like so:

- **Primary colours:** the main brand colours, used to create the basic color scheme of a project and crucial elements like buttons.

- **Secondary colours:** these accompany the primary colours, and compromise different shades, gradients and tints from the primary colours.

- **Tertiary colours:** an important group, which display system messages, such as alerts, warnings, and notifications.

To make your colours as efficient as possible, create each colour as a symbol in your Sketch file to ensure any changes update the elements across your design.

## Typography

It's important to design the style and size of all the headings (H1, H2, H3, and so on) and paragraphs to create a visual hierarchy. Usually, typography doesn't have many stylistic variations, such as colour or weight, so consider using your colours effectively to present the brand personality. Once you're happy, define your text styles in the right-hand panel, by clicking on the 'no text style' dropdown and selecting 'new text style', once saved, you can use this typographic style throughout your design, like symbols!

## Icons

When used effectively, icons add context to more complex components such as buttons, labels, or tables. Icons in your

design can be functional by encouraging users to interact – using an 'X' in a button to signify a 'Remove' action, for example.

Consider creating a set of UI icons as nested Sketch symbols to help complement other UI elements in the design framework, – arrows, for example – that can be used in sliders, previous and next buttons, and pagination.

## Buttons and inputs

When designing buttons and inputs, make sure to design their individual 'states'. Each have multiple states and provide visual feedback to users to indicate the current state (for example, hover, clicked). It's good practice to create each state as a separate symbol, since this adds flexibility.

## Complex components and Sections

At this stage, the design could be considered complete since it has everything needed to create a functioning product. However, it's worthwhile spending more time creating components for the UI framework, such as cards, tables and forms.

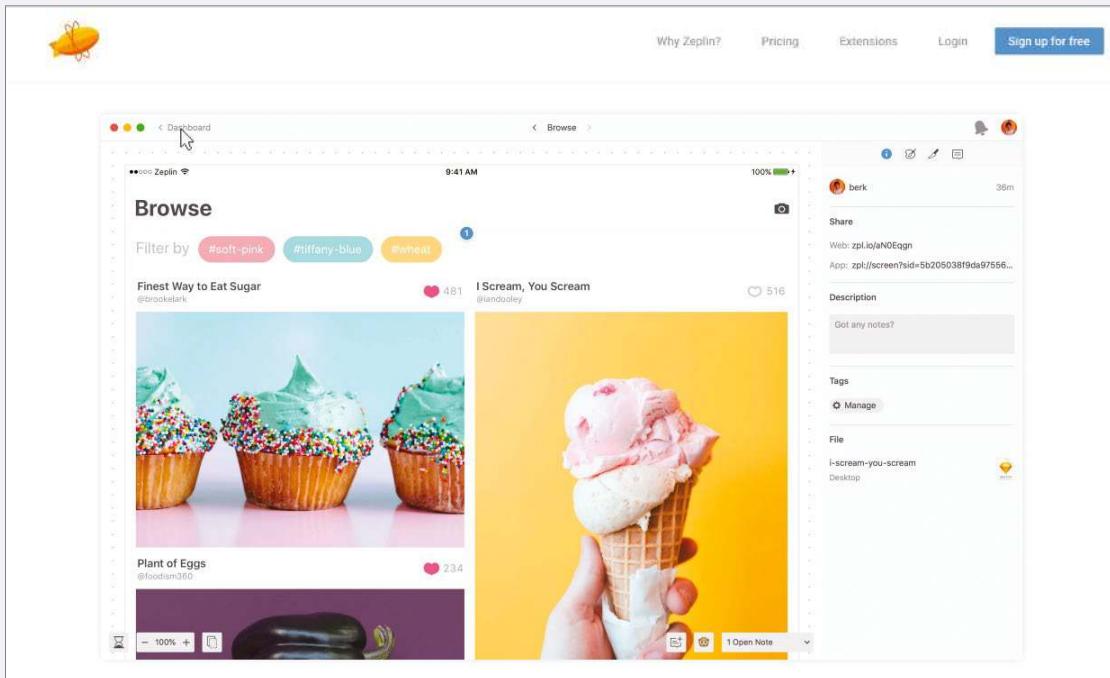
These can then be combined to develop sections, shaping the blocks to which our websites and applications rely on such as headers, navigations and banners.



**By understanding all of the user-centric steps you went through to get to your final design, colleagues & clients will be more likely to adopt your approach and thought process**

Alex Lee, UX designer  
[www.impression.co.uk](http://www.impression.co.uk)

## Design workflow



## EXPORT A DESIGN TO ZEPLIN

Export your designs into something that developers can finally work with

If you're looking to bridge the gap between design and development, look no further than Zeplin, a program that turns a Sketch file into something that all developers can work with. If you haven't already downloaded Zeplin as an app or signed up online (it's free!) and are working in a team of developers, go and do it!

Zeplin pulls out fonts, line heights, CSS, colours, symbols, images and many more, allowing them to be previewed through their web app. This is essential for presenting your concept to the

development team, as everyone can keep track of the look and feel when inevitable changes happen in the development cycle.

Once you're signed up, go ahead and install it to your plugins. To push selected artboards to Zeplin, highlight them, click the plugins tab, then Zeplin, and finally export selected artboards. Zeplin will ask you to name your project, do so accordingly, hit next, and your artboards will be available to view once you've invited your team!

It's worth pointing out that Zeplin will convert everything in

your Sketch file, so organisation is key! Stay on top of your folder structures, colours, variables and keep your assets as efficient as possible (with symbols of course!) to make sure your development team has no excuses when building your favourite elements.

If working on multiple projects, consider downloading the Zeplin app compared to the web app. Both are great, but the desktop app makes life so much easier when switching between Sketch and Zeplin in a fast-paced working environment.

## Getting a team set up on Zeplin

When you've got your designs ready to share, download and install Zeplin as a Sketch plugin, and push your artboards to the web app. Get your team to head over to [app.zeplin.io/register](http://app.zeplin.io/register) and sign up for a free account to get started. Once everyone has signed up, open your project in Zeplin, and click on the Invite button in the Members tab on the right, send out the emails and you're sorted... it's that simple! If your colleagues aren't so keen, show them the generated CSS and they will quickly get onboard!



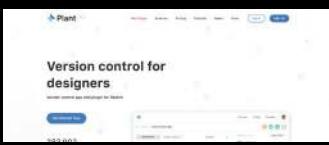
**Zeplin will convert everything in your Sketch file, so organisation is key**

## 8 killer plugins to use with Sketch

### Plant

[plantapp.io](http://plantapp.io)

Want to add version control to Sketch? Plant is the best. Push your changes directly from Sketch to a version-controlled file on the cloud for free.



### Bootstrap grid

[bit.ly/2KHDNWk](http://bit.ly/2KHDNWk)

Depending on your dev setup, getting a grid system guideline is essential for any web project. Bootstrap is a great plugin to get going.



### Icon font

[bit.ly/1FNn2wO](http://bit.ly/1FNn2wO)

A simple way to add icons into a project in one place. Download and install your favourite icons and easily insert them into your design with a few clicks.

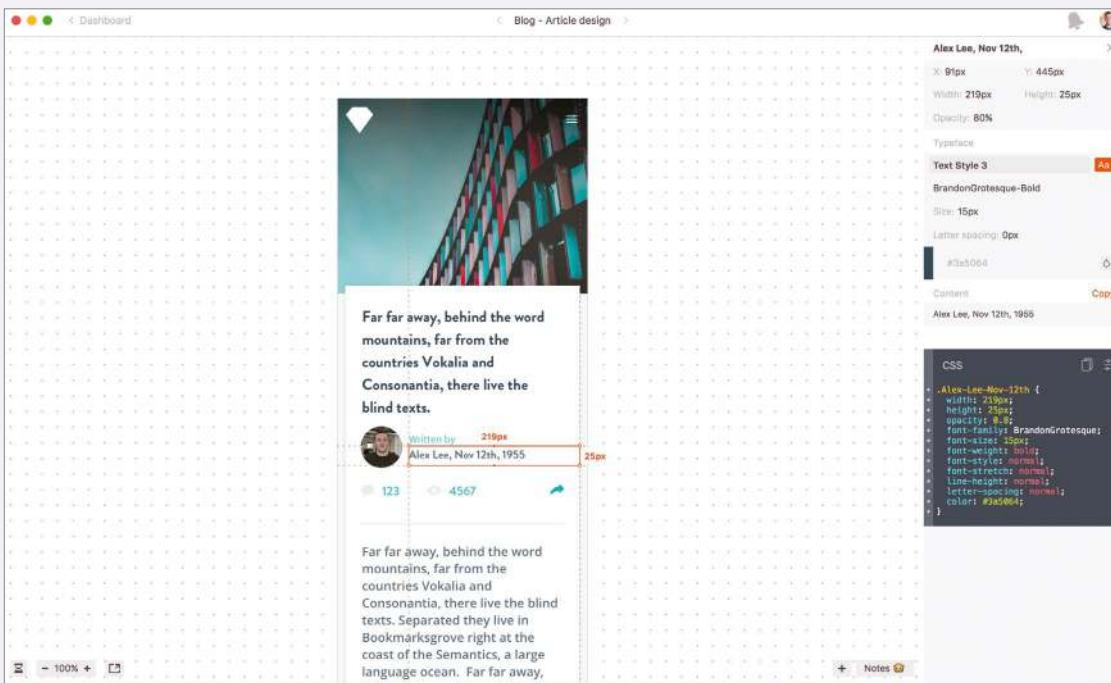


### Lightwire

[lightwireplugin.com](http://lightwireplugin.com)

If you're looking to build an app wireframe, this plugin has a collection of popular pages ready for you to download and use for free! It's a lifesaver.





“

To keep on top of the style guide, you can add fonts and colours independently

## HOW TO USE ZEPLIN

Take advantage of Zeplin's features to help organise projects and take hours off your development time

### Exportable CSS

By far the most valuable feature in Zeplin is the ability to view the designs as code. To get started, highlight a specific area such as the banner in Zeplin and look to your right, where you should be presented with some CSS.

It will definitely save hours on development time in the long run! It's fair to say the code isn't as clean as it could be, but it's a fantastic starting point for any dev.

### Creating a style guide

You may have noticed in Zeplin a

switch at the top for Dashboard and 'Styleguide'. If you haven't already, click on the 'Styleguide' option. With any luck, Zeplin has worked out all your assets and compiled a pre-made style guide for you to hand over to your developers, including any fonts and colours.

If you want to keep on top of the style guide, you can add fonts and colours independently by selecting them in Zeplin, hovering over the font or colour in the right-hand panel, and clicking on the droplet or 'Aa' icon to add them

to your colour palette or font book. This is handily compiled into the CSS block on the right, ready for you to export!

### Tags

If you find your project has a fair amount of artboards to review then tags is the way forward. It's useful tagging your artboards in the dashboard area (top left) by highlighting an artboard and hitting Cmd+T. By setting up tags, you can switch between different categories to improve the organisation of your project.

### Comments & Feedback

If you're keen to get feedback on your design, the 'notes' feature in Zeplin is very useful. Open up an artboard, and click the '+' icon next to notes in the bottom right corner and select somewhere on the design. Having the option to be specific on where you click is great for pointing out intricate details. (You can turn the notes on and off by clicking the monkey emoji next to the + icon.)

You can also keep your team updated by syncing your Zeplin project to Slack via the dashboard.

### Craft

[invisionapp.com/craft](http://invisionapp.com/craft)

Craft is a plugin which includes features, such as duplicating content, pulling live content from URLs, and of course syncing to Invision prototypes.



### Paddy

[bit.ly/2oxRbQv](http://bit.ly/2oxRbQv)

Simple yet effective, Paddy is a quick keyboard shortcut to adding padding to your elements and its children, like buttons, text blocks and more.



### Content generator

[bit.ly/1mLVrXp](http://bit.ly/1mLVrXp)

Add more life to your designs with randomly generated pictures and text, such as user profile pictures, names and biographies, all in a few clicks.



### Magic Mirror

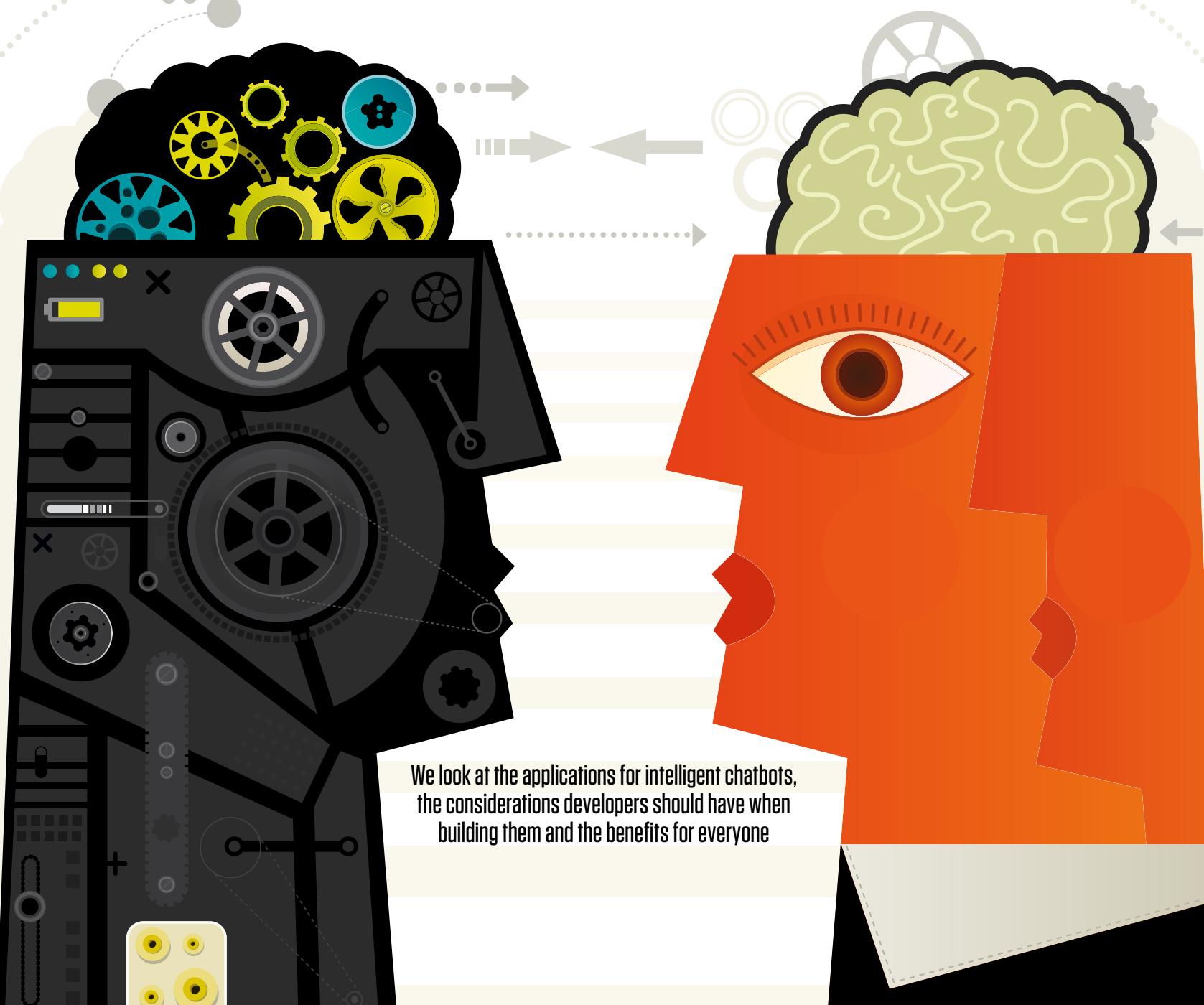
[magicsketch.io/mirror](http://magicsketch.io/mirror)

If you want to present your mockups in 3D, Magic Mirror can transform your flat design onto a collection of devices to give it that realistic edge.



# A NEW CONVERSATION

## THE COGNITIVE CHATBOT REVOLUTION



We look at the applications for intelligent chatbots,  
the considerations developers should have when  
building them and the benefits for everyone

# WHY DEVELOPERS NEED TO EMBRACE CHATBOT TOOLS

FROM END-USERS TO DEVELOPERS, INTELLIGENT CHATBOTS CAN MAKE LIFE A LOT EASIER

No one really likes the traditional line-of-business applications - neither the end-users nor the developers. The reason behind that is called 'fixed screens'. Think about how you do simple things like booking a meeting within a traditional fixed-screen application today - you have to input the names of all attendees, choose a room, select start and end time, etc. Ask yourself how natural it feels to go through all these steps? Probably the interaction with the computer isn't very natural because the UI defines its flow and it feels like a compromised user experience that we are all now used to. At the same time, developers have to spend a lot of time building every single one of those 'fixed screens' and then update them to reflect the

another human. If you look at science fiction, the desire for computers that resembles a human has always been popular, and now we are seeing the first bits in action. Everything from booking a flight to receiving banking advice is achievable through the use of intelligent chatbots, with the cognitive element offering a far more sophisticated level of personalised advice to the consumer. For businesses deploying chatbots, the increased interaction with consumers will give them more data and therefore insights into their behaviour. We are already seeing a lot of businesses building bots for channels like Viber and Facebook Messenger. At the same time a lot more chatbots are expected to be built into their own web, desktop and mobile

**"When you look at how cognitive chatbots function, the concept of a screen disappears, allowing you to re-define the UI to follow the natural conversation rather the other way around"**

constantly changing business requirements. Intelligent chatbots are changing that.

The increased availability of data and cognitive machine learning capabilities are enabling the intelligent chatbots to bring the interaction between humans and computers to a whole new level that hasn't been possible before. Now we can interact with computers in a close-to-human way that enables the UI to follow the conversation.

So what does this mean in reality for developers and end-users?

For consumers, they are going to see a far more enjoyable, seamless and natural interaction with chatbots. Rather than having to ask exact questions or follow a fixed path, they will be able to speak naturally and jump around with their requests in much the same way they would if talking to

applications in order to improve the employees' performance and the customer experience.

For developers, the benefits of cognitive chatbots are great. The availability of out-of-the-box tools, such as Progress' NativeChat ([www.progress.com/nativechat](http://www.progress.com/nativechat)) and Telerik and Kendo UI Conversational UI components ([www.telerik.com](http://www.telerik.com)) means that it's never been easier for developers to start building cognitive chatbot applications capable of being deployed at scale. If you look at how applications are built today, especially from the front-end, you have to build a series of windows with different properties. But, when you look at how cognitive chatbots function, the concept of a screen disappears, allowing you to re-define the UI to follow the natural conversation rather the other way around. Another benefit for developers is using them to

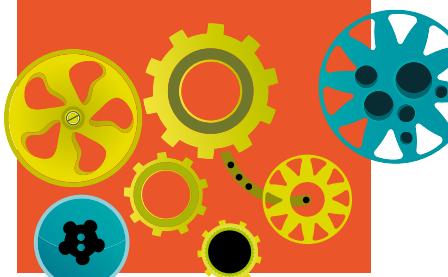


automate some of their own processes. Cognitive chatbots offer a means to automate a number of processes of the application development lifecycle and integrate them with the team's Slack channel for example.

The rise of cognitive chatbots is inevitable. What's most exciting is that as more companies introduce them, allowing them to gain and access more and more data, the smarter and more helpful they will become. Developers will not be required to constantly update as they will in many ways update themselves, learning at their own rate. At the same time, developers would need to work on the UI far less because the UI will be almost auto-generated based on the flow of the conversation, allowing for more time to address the next business problems.

Get on board before the chatbots start to build themselves!

**VESKO KOLEV IS AN EXPERIENCED BUSINESS LEADER AND SOFTWARE ENGINEER. AS VP OF PRODUCT, DEVELOPER TOOLING AT PROGRESS, VESKO IS RESPONSIBLE FOR A TEAM OF 150+ SOFTWARE ENGINEERS, PRODUCT MANAGERS AND UX DESIGNERS. HE BELIEVES AI AND VR/AR WILL REDEFINE THE CONCEPT OF APPLICATIONS.**





### WHAT DO YOU THINK ARE THE KEY BENEFITS THAT CHATBOTS BRING TO BUSINESS APPLICATIONS?

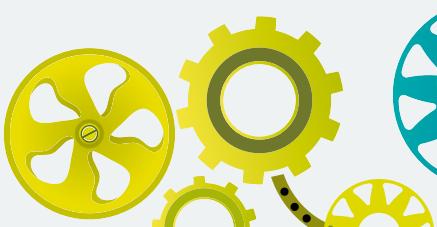
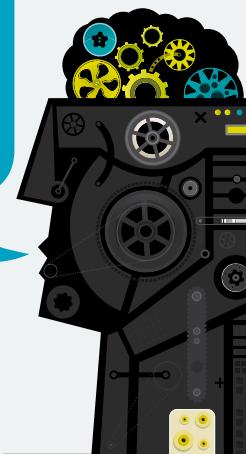
The key benefits are time, money and improved user experience. Developers can iterate multiple orders of magnitude faster on defining and improving a chatbot compared to the traditional way of building applications, thus being truly agile and able to get feedback from actual users almost in real-time. Chatbots also get smarter over time and the UI is in many cases being generated automatically, requiring less human intervention. Finally, if you have powerful front-end tooling, then you are able to have cognitive chatbots in all your channels.

### WHAT ARE THE ACCESSIBILITY ISSUES THAT CONVERSATIONAL UI CREATES FOR DESIGNERS AND DEVELOPERS?

I can't think of an accessibility issue that the conversational UI creates. Actually, it is quite the opposite. One of the most exciting features of cognitive chatbots is that they will allow for a far more enjoyable experience for those with visual defects. Up to now, unless the application has been specifically developed for visually impaired people, it is practically not accessible for them. With the voice recognition technology, intelligent chatbots allow visually impaired people to get an experience that almost equals the experience of everyone else.

### HOW MUCH MORE ADVANCED DO YOU THINK CHATBOTS WILL BE IN 5 YEARS?

The tipping point will be when enough organisations start to use them in their daily operations. If they begin to experiment with cognitive chatbots then they will see their power. As more data is generated and fed in, the more they will offer the business from a competitive edge perspective. Heightened user personalisation, increased productivity, sophisticated advice, will become available to apps that have this technology. The beauty of where we will be is that so much is dependent on how the cognitive chatbots teach themselves!



# CREATE THE IMPOSSIBLE

[www.photoshopcreative.co.uk](http://www.photoshopcreative.co.uk)

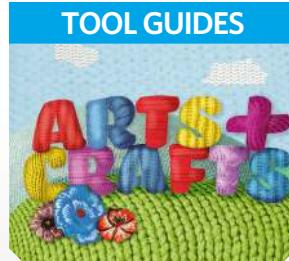


# Photoshop creative

Available  
from all good  
newsagents and  
supermarkets

ON SALE NOW

• Striking imagery • Step-by-step guides • Essential tutorials



## BUY YOUR ISSUE TODAY

Print edition available at [www.myfavouritemagazines.co.uk](http://www.myfavouritemagazines.co.uk)  
Digital edition available for iOS and Android



[facebook.com/PhotoshopCreative](https://facebook.com/PhotoshopCreative)

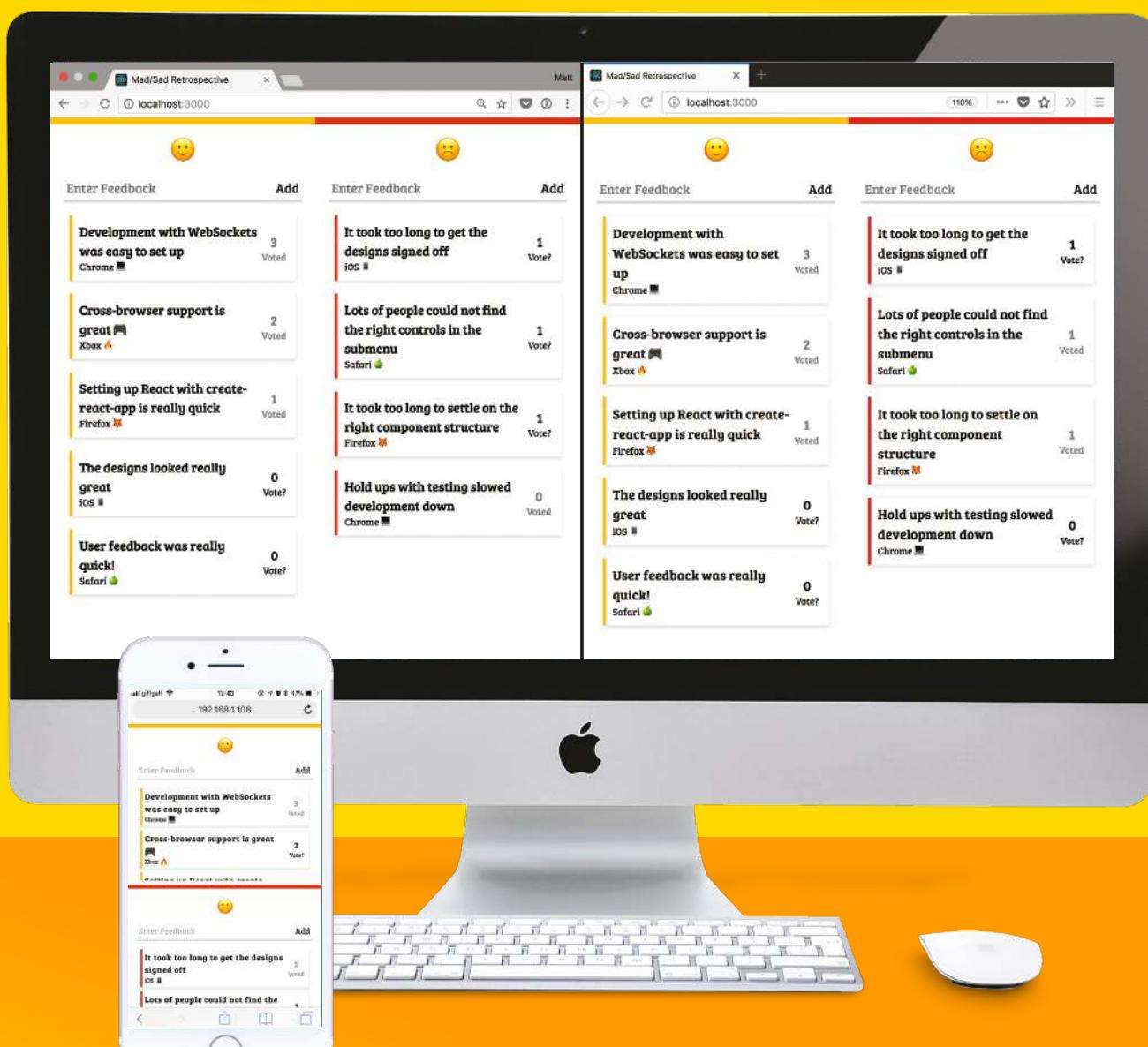


[twitter.com/PshopCreative](https://twitter.com/PshopCreative)

## Developer tutorials

# Build a feedback app with React

Combine React's powerful state management with real-time messaging to make a flexible feedback tool



 **DOWNLOAD TUTORIAL FILES**  
[www.filesilo.co.uk/webdesigner](http://www.filesilo.co.uk/webdesigner)



Applications on the web are commonplace nowadays. Developing these constant communication systems can be tricky using traditional methods.

REST APIs are a solution, but frequent requests to the server can slow things down and can end up being a poor experience for the user.

WebSockets are a step up from regular HTTP requests. They allow real-time communication between a server and a browser. Any update from either side will instantly notify the other without the overhead of a regular HTTP request.

React is, in its nature, a dynamic framework. By updating an application's state, we can change how it behaves. That information can come from any source - including WebSockets. By passing data from a socket into React's state, communication between clients can become instantaneous.

In this tutorial, we will be making a retrospective feedback tool similar to that used at the end of projects in large organisations. Users on multiple devices can provide good or bad responses and have others vote on which ones they agree with. Traditionally this would be done in-person using sticky notes, but our tool will allow remote workers to participate from around the world.

We will be making the WebSocket server with Node.js and Yarn. Make sure these are both installed and up to date before we begin.

## 1. Install dependencies

This application has two sides - the server and the client. The Node server uses the 'ws' package to start and manage WebSocket connections, while 'create-react-app' powers the client side.

Locate the project files on the command line and install the dependencies. Once that is finished, start up the React development server.

```
> yarn install
> yarn start
```

## 2. Start a WebSocket server

The 'ws' package provides everything needed to get a server running within Node. Before we go any further, we need to get the server up and running.

Head to `server/index.js` and initialise the server. Then open up a new command line window and run this file in it. We will need to re-run this command every time a change is made to the server in order to reload it.

```
/* server/index.js */
const server = new
  WebSocket.Server({ port: 8080 });
/* Separate command line */
> yarn run server
```

## 3. Link React to connection

On the client side, all the communication happens within a special 'WebSocketConnection' object we will make. This lives outside of React, but we pass a callback into it that will let React know when any messages are received.

Open up `src/App.js` and create a new connection. Do this inside 'componentDidMount()', which allows React to render the app initially without having to wait for the

connection to set up.

```
componentDidMount() {
  this.connection = new
    WebSocketConnection(this.onMessage);
}
```

## 4. Connect object to server

When we initialise 'WebSocketConnection', it will connect to and manage the WebSocket on React's behalf. When any information arrives that React would need, it's passed to the callback from the previous step.

In `src/WebSocketConnection.js`, start the WebSocket in the constructor and hook into the 'message' event. This will be useful later on.

```
constructor(messageCallback) {
  let pathToServer =
    CONSTANTS.WEBSOCKET_URI;
  this.connection =
    new WebSocket(pathToServer);
  this.connection.onmessage =
    this.onMessage.bind(this);
  this.messageCallback =
    messageCallback;
}
```

## 5. Identifying a client

While we can identify each client on the server, if the user refreshes the page or loses connection they will appear as a new client. To keep track of a user, the server can supply a generated ID to pass through on connection if they have one. Just after creating the 'pathToServer' variable, add the ID as a query parameter if one is available.

```
const existingId = getId();
if (existingId) {
  pathToServer += `?id=${existingId}`;
```

## 6. Detecting a message

All communication is performed through messages. When the server sends messages to the client, it will call the 'onMessage' method inside 'WebSocketConnection'. Messages are limited to a handful of formats, but one of those is a string. We can convert objects to JSON strings and pass those across instead.

Create the 'onMessage' method. For now, all it needs to do is convert the string back to an object and pass it through to React's callback from earlier.

```
onMessage(event) {
  const data = JSON.parse(event.data);
  if (this.messageCallback) {
    this.messageCallback(data);
  }
}
```

## 7. Detect a server connection

Once a client has tried to connect, the server should acknowledge it. The one we set up in step 2 receives a 'connection' event when that happens. Back in `server/index.js`, underneath the initialisation of the server, add a listener for the connection event. We will take the ID passed as a query parameter and pass it to another

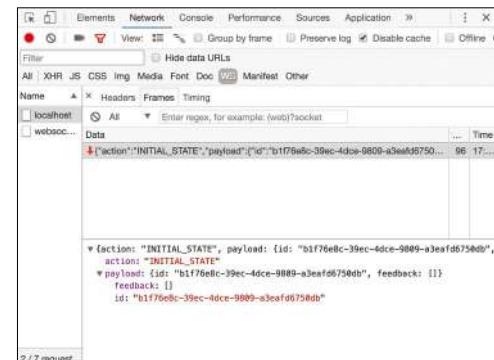
function that sets up the client.

```
server.on("connection",
  (client, req) => {
  const {
    query: { id }
  } = url.parse(req.url, true);
  addClient(client, id);
});
```

## 8. Set up a client

Every client is tracked by the 'ws' module and is represented by an object. We can add the ID to that object to link a user to it. If the user did not supply an ID, we can generate one and send it over. Once that is set up, add a listener for when that client sends a message. As you can see from the code below, add the following to the 'addClient' function.

```
function addClient(client, connectionId) {
  const id = connectionId ? connectionId : uuid();
  client.id = id;
  client.on("message", onClientMessage);
```



**Above:** Chrome's developer tools are great for working with WebSockets. We can see the connection being made and messages being sent around in real time

## 9. Pass initial state

Once the client has been set up, we can start sending information back again. The first of that would be initial state, which contains any feedback already entered along with the client ID and a username that we will set up later. Add this command to the end of `addClient`, which will send over the initial state. The object we pass has an 'action' to describe the intent with a 'payload' providing the data.

```
client.send(
  JSON.stringify({
    action: CONSTANTS.INITIAL_STATE,
    payload: {
      id, username: users[id],
      feedback: feedback.map(f => ({
```

## Use local IP address

When testing the site on another device, be sure to use the local IP address rather than 'localhost' in any client-side code. In this project, this can be updated in `src/constants.js`.

## Tutorials

### Build a feedback app with React

The screenshot shows the Network tab in the Chrome DevTools developer tools. A single message is listed under the 'Logs' section. The message is an incoming WebSocket message from 'ws://localhost:8080'. The payload contains an action object with an id, feedback array, and other properties. The file 'WebSocketConnection.js' is shown as the source for this message.

```
INCOMING MESSAGE
▶ message { target: WebSocket, isTrusted: true, data: "{\"action\": \"INITIAL_STATE\", \"payload\": {\"id\": \"dc93db41-7060-45d3-9fa6-9e1c3bcf6c75\", \"feedback\": []}}", origin: "ws://localhost:8080", lastEventId: "", ports: Restricted, eventPhase: 0, bubbles: false, cancelable: false, defaultPrevented: false, ... }
```

**Above:** In other browsers without WebSocket inspection, log the event inside `onMessage` within `WebSocketConnection.js` to check messages as they arrive.

```
... f, username: users[f.clientId]
})
)
);
});
```

### 10. Save the ID from initial state

Now the server is responding, we should capture the data it sends. The first thing to get is the client ID, which is passed in through the initial state. We can do this step outside of React to keep things simple.

In `src/WebSocketConnection.js`, just after we parse the JSON in '`onMessage`', check to see if the message is the initial state. If it is, store the ID for use later on.

```
if (data.action ===
  CONSTANTS.INITIAL_STATE) {
  setId(data.payload.id);
}
```

### 11. Update React state

The '`onMessage`' function will run '`messageCallback`' from React once a message is received. We can use this to

update the state, or in this case initialise it. Open up `src/App.js` and fill out the '`onMessage`' callback function here. Add the following to the switch statement in place, which we will add to in later steps.

```
case CONSTANTS.INITIAL_STATE:
  this.setState({
    ready: true,
    id: data.payload.id,
    username: data.payload.username,
    feedback: data.payload.feedback
  });
  break;
```

### 12. Submit a username

Now we see a screen asking for a username. While our generated ID identifies a client, it should be linked to a human-readable name to identify the user.

The `<UsernamePrompt/>` component is hooked up to submit using the '`setUserName`' method in `src/App.js`. Capture that username and send it through the WebSocket.

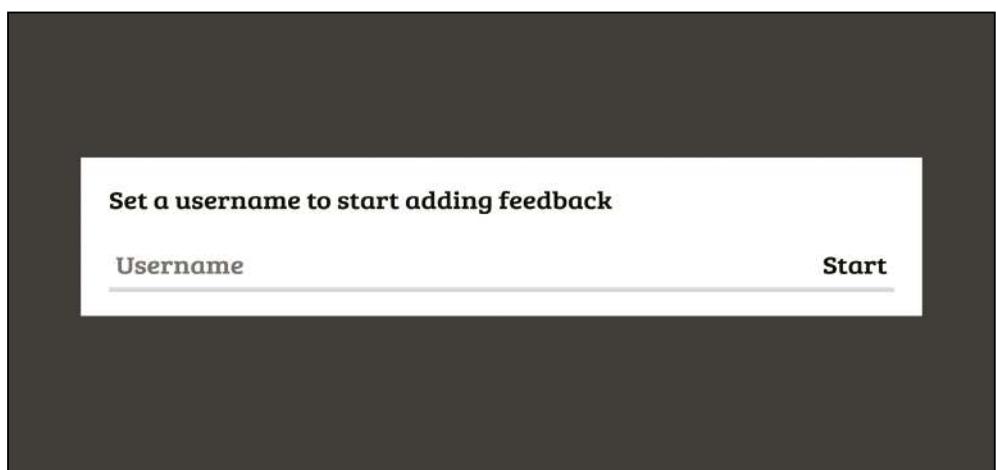
```
this.setState({
  username
});
this.connection.send(
  CONSTANTS.SET_USERNAME,
  username);
```

### 13. Send data via WebSocket

While adding a username removes the prompt, it does not yet send it to the server. We need to hook up the '`send`' method in '`WebSocketConnection`' first.

Open up `src/WebSocketConnection.js` again, find the '`send`' method and hook it up. All it will do is convert it to a JSON string in the expected format.

```
send(action, payload) {
  this.connection.send(
    JSON.stringify({
```



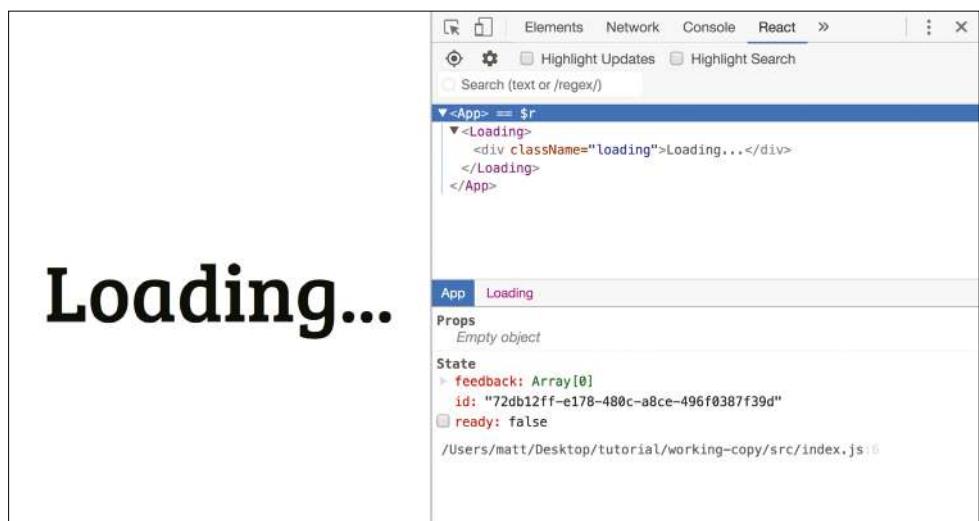
## Hot reloading pages

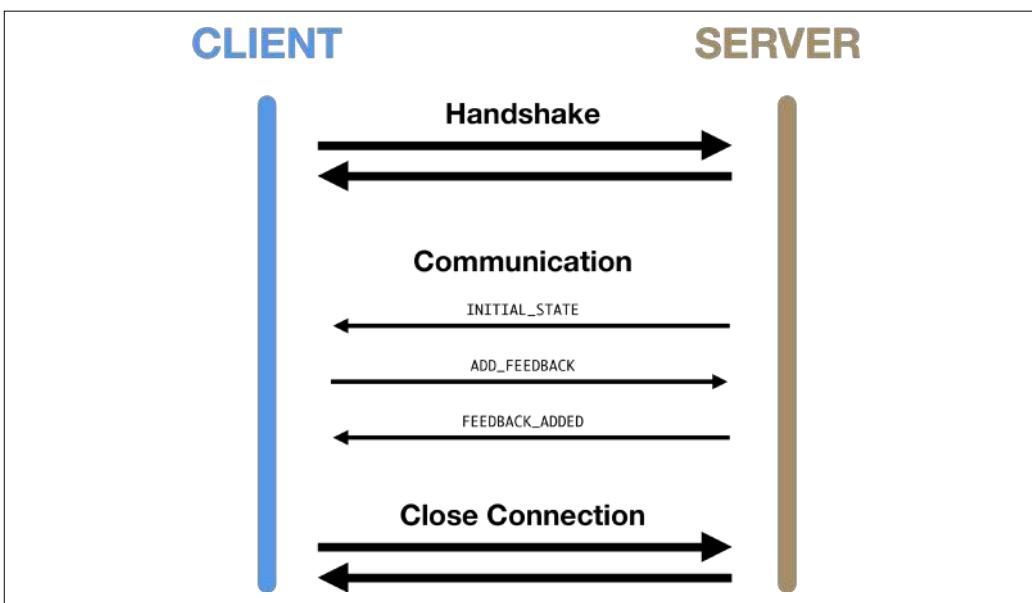
The '`create-react-app`' package uses WebSockets to reload the page once the code changes. This will appear as a request from '`websocket`' and should not be confused with our server.



The '`loading`' state value is `false` when the initial state is received. This removes the `<Loading/>` component and shows the page

**Top**  
The feedback input fields will only be visible once a username is present in the state - either by adding it from the prompt or set by initial state





### How do WebSockets work?

WebSockets are a relative newcomer to browsers, but they work off HTTP - the same principle that browsers have used for years.

Each WebSocket connection starts as a regular HTTP request to the server. A special 'Upgrade' header tells the server it would rather use WebSockets to communicate. If the server accepts, it sends back its own 'Upgrade' header telling both parties to use a WebSocket instead. This is known as the handshake phase.

From then on, both the browser and the server can start sending messages in simple formats such strings. This all comes without the overhead of headers and cookies that accompany regular HTTP requests. The connection stays open for as long as either party is connected. The result is real-time communication with low latency.

```

        action,
        payload
    })
}
}

```

## 14. Link username to ID

The final step in setting a username is to save it against an ID on the server. This needs to be kept separate from the client so we can link them back if the connection closes. In `server/index.js` add a case to the switch statement in '`onClientMessage`'. Be sure to trim the username so it avoids breaking the layout of other users.

```

case CONSTANTS.SET_USERNAME:
    users[this.id] = trim(data.payload,
        CONSTANTS.MAX_USERNAME_LENGTH);
    break;
}

```

## 15. Submit feedback

With a username set, we can start sending feedback. The process is similar to setting a username, but we won't set any state as we submit.

Open up `src/App.js` and add the functionality for the '`submitFeedback`' function. This will use the same 'send' method as before, but the payload will be an object telling the server if it is good or bad feedback.

```

submitFeedback(type, content) {
    this.connection.send(
        CONSTANTS.ADD_FEEDBACK, {
            type,
            content
        });
}

```

## 16. Save feedback server-side

With the feedback submitted, we need to pick up that message server-side. These are stored as objects in a separate feedback array on the server. Head back to `server/index.js` and add another case in

'`addClientMessage`'. If the feedback object is created with no issues, push it to the array.

```

case CONSTANTS.ADD_FEEDBACK:
    const feedbackObj =
        createFeedbackObject(
            this.id, data.payload.type,
            data.payload.content
        );
    if (feedbackObj) {
        feedback.push(feedbackObj);
    }
    break;
}

```

## 17. Broadcast to all clients

If we refresh the page, we will see our feedback showing up, but it should show up instantaneously. We need to tell everyone that feedback has been added.

Use the '`broadcast`' function in `server/index.js` to send the feedback object, plus the username linked to the ID, to all connected clients. All this function does is loop through the connected clients and sends the same

message to each of them.

```

broadcast(CONSTANTS.FEEDBACK_ADDED, {
    ...feedbackObj,
    username: users[feedbackObj.clientId]
});

```

## 18. Receive feedback client-side

Once the feedback has been broadcast, we should add it to the local state of each connected client - including the one that sent it.

In `src/App.js`, update the '`onMessage`' function to accept the new feedback. We will use the spread operator to copy the existing array, rather than mutate it directly.

```

case CONSTANTS.FEEDBACK_ADDED:
    this.setState(prevState => ({
        feedback: [
            ...prevState.feedback, data.payload
        ]
    }));
    break;
}

```

## Enter Feedback



Add

This is some example feedback

Matt

0

Voted

**Above:** The vote button next to each piece of feedback will be disabled if the user has already voted for it, or it was submitted by them

## Tutorials

### Build a feedback app with React

The screenshot shows the official socket.io website. At the top, there's a navigation bar with links to Home, Demos, Get started, Docs, Blog, Donate, and Slack. A 'Star' button indicates 41,878 stars, and a 'Download v2.1.0' button is available. The main headline says 'SOCKET.IO 2.0 IS HERE' with the subtitle 'FEATURING THE FASTEST AND MOST RELIABLE REAL-TIME ENGINE'. Below this, there are two code snippets: one from 'tweets/index.js' and another from 'your-node-app.com'. The 'tweets/index.js' snippet shows how to use socket.io to track tweets. The 'your-node-app.com' snippet shows a feed of tweets from various users. Below these examples is a 'slack' logo with the tagline 'All the tools your team needs in one place. Slack: Where work happens.' and a note 'ADS VIA CARBON'.

**Alternative real-time tools**

While WebSockets are well supported in modern browsers, they will not work in IE9 and Opera Mini and can be hard to scale up to lots of users. While providing support for these cases may be tricky, there are some tools that can make things easier.

Socket.IO provides real-time communication between devices - primarily through WebSockets but with fallback options like long polling. They provide both server and client APIs to enable easy communication. This is done through channels, which can send messages to specific clients.

SockJS is a similar platform that caters for different server environments, ie Python and Erlang. It creates an abstraction over WebSockets to make communication easier. Both are set up to work with thousands of clients with ease. To help work with WebSockets, it makes sense to use an external tool to help the project scale.

## 19. Add a vote

When one user submits feedback, another user can vote on it to give it prominence. This can be done by clicking the vote number next to each feedback item. Like with the username and feedback submission, send a message to the server when a vote has been cast.

```
submitVote(id) {
  this.connection.send(
    CONSTANTS.ADD_VOTE, id);
}
```

## 20. Record the vote

We still need to record the vote on the server side. To do that, we need to find the object in the feedback array, check the user hasn't voted on it already and add their vote if they haven't.

Once that is done, we can broadcast that change to all clients. In `server/index.js`, add another case to '`onClientMessage`' to add the vote.

```
case CONSTANTS.ADD_VOTE:
  const votedOn = feedback.find(
    f => f.id === data.payload);
  if (votedOn) {
    if (!votedOn.votes.includes(
      this.id)) {
      votedOn.votes.push(this.id);
    }
    broadcast(CONSTANTS.VOTE_ADDED, {
      id: votedOn.id,
      votes: votedOn.votes
    });
  }
  break;
```

## 21. Sync votes on the client

Finally, the votes should be kept up to date on all clients. Once the message is broadcast, we can update each client's state by finding the feedback and updating the votes array within it. Add the following to '`onMessage`' within `src/App.js`.

```
case CONSTANTS.VOTE_ADDED:
  const index = this.state.feedback
    .findIndex(
      f => f.id === data.payload.id);
  if (index !== -1) {
```

```
this.setState(prevState => {
  const newFeedback =
    [...prevState.feedback];
  newFeedback[index] = {
    ...newFeedback[index],
    votes: data.payload.votes
  };
  return { feedback: newFeedback };
});
}
break;
```

The screenshot shows a feedback application interface. On the left, there's a text input field labeled 'Enter Feedback' and a 'Add' button. Below this, there's a list of feedback items:

- This is really popular feedback!** 3 Vote?
- This is quite popular feedback** 2 Vote?
- This is slightly popular feedback** 1 Vote?
- Nobody has voted for this feedback** 0 Vote?

A note on the left side explains the sorting logic: 'Above: As votes come in, React will sort the feedback based on the number of votes it has. This makes sure the most popular feedback gets noticed.'

Special offer for readers in *North America*



# 6 issues FREE

When you subscribe\*

**FREE**  
resource  
downloads  
every issue

29 PAGES EXPERT ADVICE

**FREE 61 MINUTES OF JS VIDEOS**

**web designer**™

**CHATBOTS**  
What do developers need to consider?

**STREAMLINE DESIGN**  
Speed up workflow with Sketch

**HOT NEW CSS**

**Top properties for today and tomorrow**

CSS Grid • @supports • CSS Variables • CSS Filters

**REAL-TIME REACT**  
Create a feedback app

+ Build interactive 3D • Get started with Three.js  
Deploy to Google Cloud • Convert Flash to HTML

**“The only magazine you need to design and develop stunning websites”**



Order hotline **+44 (0) 344 848 2852**  
Online at **myfavouritemagazines.co.uk/WEDPQ17**

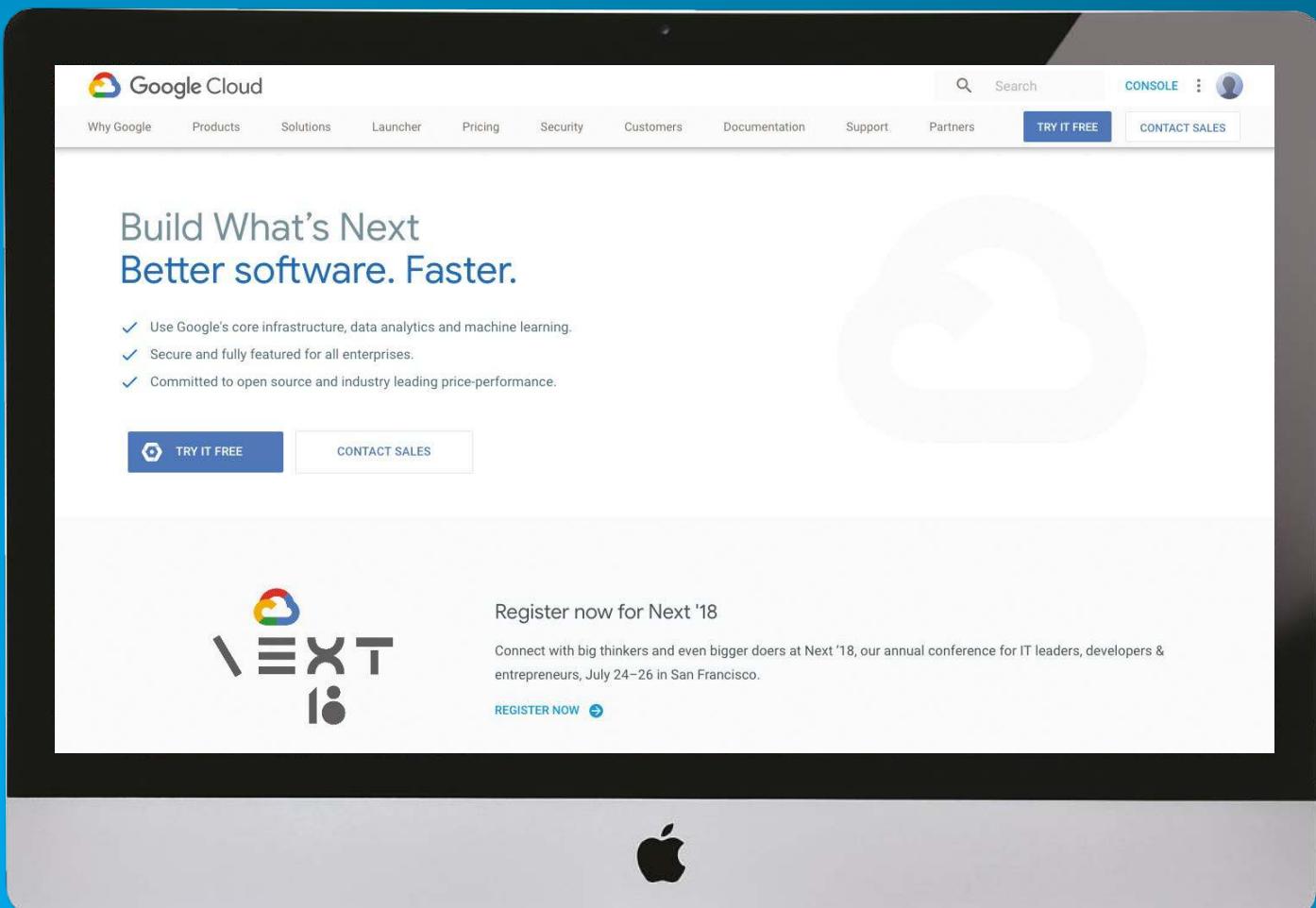
**\*Terms and conditions** This is a US subscription offer. 6 free issues refers to the USA newsstand price of \$14.49 for 13 issues at \$188.37, compared with \$103 for a subscription. You will receive 13 issues in a year. You can write to us or call us to cancel your subscription within 14 days of purchase. Payment is non-refundable after the 14 day cancellation period unless exceptional circumstances apply. Your statutory rights are not affected. Prices correct at point of print and subject to change. Full details of the Direct Debit guarantee are available upon request. UK calls will cost the same as other standard fixed line numbers (starting 01 or 02) are included as part of any inclusive or free minutes allowances (if offered by your phone tariff). For full terms and conditions please visit: [bit.ly/magtandc](http://bit.ly/magtandc) Offer ends September 30 2018.

**Expires  
30 Sept  
2018**

## Developer tutorials

# Deploy your SPA to Google Cloud

Learn how to set up a simple Node.js server and deploy  
to Google App Engine or Firebase Hosting



 In this article we're going to show you how to deploy your single-page application (SPA) to a cloud-hosting platform, such as Google Cloud Platform (GCP). We'll show two different options; deploying to Google App Engine and deploying to Firebase Hosting, both of which are based on Google Cloud infrastructure, but provide different offerings – we will come to this later.

Your SPA could be built using React, Vue.js, Angular or even something bespoke – it doesn't really matter, the process is the same. For this tutorial we'll use Angular's CLI tool to create a very simple app so we actually have something with routing to deploy, then we'll set up a simple Node.js server and finally deploy the whole thing to GCP. To do this you will need a Google account to access Google Cloud Platform (GCP). With this you get \$300 free credit for a year.

All code is available via GitHub: [github.com/danielcrisp/deploy-your-spa-to-google-cloud](https://github.com/danielcrisp/deploy-your-spa-to-google-cloud) or FileSilo.

## 1. Install Angular CLI

First we need to install Angular's CLI tool. This is super simple and requires at least Node v8.9.0 and npm v5.5.1. We can install it globally like this:

```
■ npm install -g @angular/cli
```

Once the install is complete we can go ahead and create our app. We're going to call the app 'gcp-demo' so we need to run this command to scaffold the app with a routing module:

```
■ ng new gcp-demo --routing
```

When the process completes you can check everything is OK by starting the app:

```
cd gcp-demo  
ng serve
```

Open up <http://localhost:4200> and you should see your basic app.

## 2. Add Bootstrap

Just to keep life simple we're going to add Bootstrap to provide our basic styling requirements. Head over to [getbootstrap.com/](https://getbootstrap.com/) and copy the CDN link for the CSS from the Get Started section.

Open up 'src/index.html' and paste it at the bottom of the <head> element. Done.

We don't need the JS, but if you do want to add interactivity to Bootstrap components in an Angular app we'd recommend ng-bootstrap ([ng-bootstrap.github.io/](https://ng-bootstrap.github.io/)) rather than using Bootstrap's own JS.

## 3. Set up some routes

In order to demonstrate one of the hosting requirements of a SPA (more on this later) we need to create some routes in our app. Again, we can use the CLI to do the legwork for us.

We used the '--routing' flag when we created the app so we already have a routing module set up for us. Let's create a couple of components that we can use as routes. Tip: We'd recommend stopping the dev server before creating new components.

As a nod to the World Cup, let's call the first one 'route-one'. And as a nod to, erm, the first route let's call the second one 'route-two'. Create them like this:

```
■ ng g component route-one  
■ ng g component route-two
```

## 4. Wire them up into the router

Open up 'src/app/app-routing.module.ts' in your editor of choice. The CLI has kindly already imported our new components into the main 'app.module.ts', but we'll still need to import them into this file manually.

```
■ import { RouteOneComponent } from './route-one/route-one.component';  
■ import { RouteTwoComponent } from './route-two/route-two.component';
```

You will see the following array definition:

```
■ const routes: Routes = [];  
We need to create two simple route configurations inside this array as follows:  
■ const routes: Routes = [{  
  path: 'one',  
  component: RouteOneComponent  
}, {  
  path: 'two',  
  component: RouteTwoComponent  
}];
```

This just tells Angular that if the URL path matches '/one' then render the RouteOneComponent and if it matches '/two' then render the RouteTwoComponent.

## 5. Add a navigation

The routes are set up, but we can't actually get to them yet. Run 'ng serve' again and you'll see that nothing has visibly changed.

Open up 'app.component.html' in your editor and then delete everything apart from the <router-outlet> element. This is the component that will render your routed components. Anything outside of here will be visible on all routes.

Now let's add a navigation just above <router-outlet> and wrap the whole thing in a container, like so:

```
■ <div class="container-fluid">  
  <nav class="navbar navbar-expand navbar-light bg-light mb-3">  
    <a routerLink="/" class="navbar-brand">GCP Demo</a>  
    <div class="navbar-nav">  
      <a routerLink="/" routerLinkActive="active" class="nav-item nav-link">Home</a>  
      <a routerLink="/one" routerLinkActive="active" class="nav-item nav-link">One</a>  
      <a routerLink="/two" routerLinkActive="active" class="nav-item nav-link">Two</a>  
    </div>  
  </nav>  
  <router-outlet></router-outlet>  
</div>
```

Save the changes and your app should update with a nice shiny nav. Click on the links and the URL will change and text will appear in the page for routes one and two. Home doesn't have any text yet because nothing has been routed here.

## 6. Deployment decisions

Now we've got our ready-to-deploy app we need to make a choice: shall we host on Firebase or Google App Engine (GAE)? The key difference is that – at least for this example – Firebase does a lot of the work for you, including providing a production server with HTTP2 support, pushing out to a global CDN and managing your (free) SSL certificate. GAE, on the other hand, expects you to do all of this legwork yourself.

Sounds like Firebase wins hands-down. But there is a price to pay for all of this magic, and that is, well, the price.

Firebase does offer a generous free tier but if you need to go big, or if you want to get your hands dirty with the nitty-gritty and be in total control, then GAE is a great alternative option.

We're going to take a look at GAE first, and then switch over to Firebase.

## 7. Google App Engine

As we mentioned Firebase provides you with a production server, but Google App Engine doesn't. So this is probably a good place to start. GAE supports Java, Ruby, C#, Go, Python, PHP and Node.js, but as we're all familiar with JavaScript we're going to use Node.js.

Before we write any code we first need to figure out where to put this server. It needs to be in the project repo and be able to serve the compiled app files. If you run a production build with 'ng build --prod' you'll see that the compiled files are output to 'dist/gcp-demo'. This is pretty handy as it means we can create our server inside 'dist'.

First, though, we need to modify '.gitignore' so that the git only ignores the build. Open up '.gitignore' and on line 4 change '/dist' to '/dist/gcp-demo'. Also remove the leading slash from '/node\_modules'. Save it and commit it. Next, create an empty index.js file inside '/dist'.

## 8. Install some packages

Now, thankfully, there are some great 'npm' packages out there that will do most of the hard work for us.

First, open up your terminal, cd into the 'dist' directory, run the following command and follow the prompts:

```
■ npm init
```

This just creates a 'package.json' file in your directory ready for you to install some packages. Let's do just that:

```
■ npm install express serve-static compression helmet --save
```

This will install four separate 'npm' packages, which we'll now configure.

## 9. Express

Express will be our server. It's hugely popular and very simple to use. Open up your 'index.js' file and add this:

```
■ const express = require('express');  
■ const app = express();  
// Check if the host env has specified a port to run on  
■ const port = process.env.PORT || 3000;  
■ app.listen(port, () => console.log(`App listening on port ${port}`));
```

You should now be able to start this server from your terminal using 'node index.js'. Nothing much will happen but you should see the message logged to your console.

## Developer tutorials

### Deploy your SPA to Google Cloud

#### 10. Serve the static files

The next step is to serve the compiled files as static files. Express supports lots of clever things like routing, but we only need a very simple, lightweight server here, so import the 'serve-static' package at the top of your file.

```
const serveStatic = require('serve-static');
```

Then just before the port declaration, enter the following piece of code:

```
app.use(serveStatic(__dirname + '/gcp-demo'));
```

This tells your Express app to serve the contents of 'gcp-demo' statically.

If you run a production build with 'ng build --prod' and start your server again you should be able to access your app at <http://localhost:3000>.

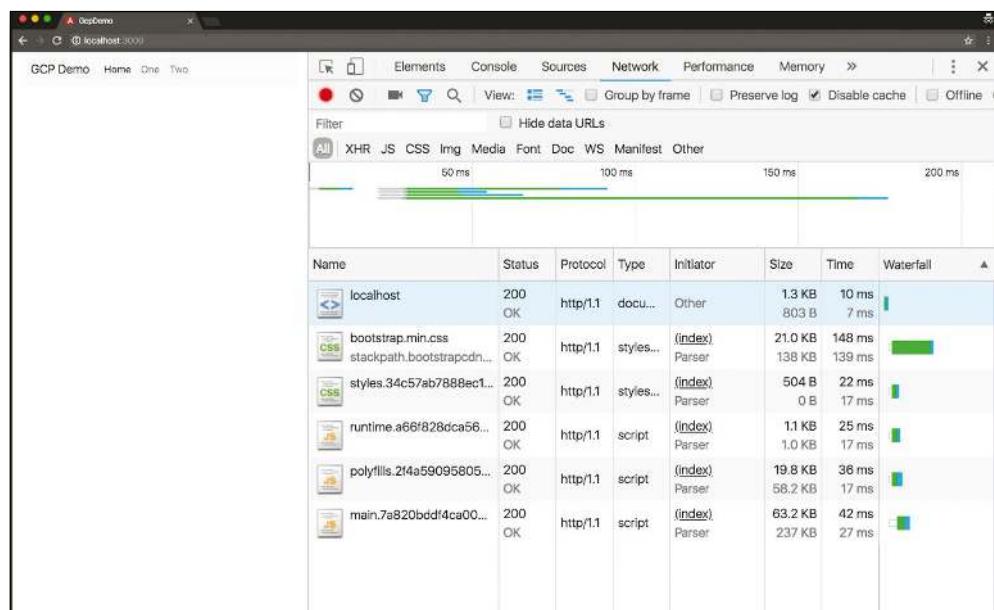
#### 11. Serving the index

Click around in your app and you should see that your routing is working nicely. You can access Route One and Route Two, and the URL in the browser's address bar should update accordingly.

There is one gotcha, though, that it is important to be aware of with single-page applications of any flavour, e.g. React, Angular. The routing is managed client-side using the HTML5 History API, which makes it look like you are navigating to different pages. However, if you request a client-side route from the server the server doesn't know anything about it and will return a 404.

We can demonstrate that now by going to Route Two at '/two' and then refreshing the page. You should see the default Express error page showing 'Cannot GET /two'.

That's because Express doesn't know anything about a route or directory called '/two'. The solution to this problem is remarkably simple. We simply tell the Express app to serve the 'index.html' file for all paths, other than



existing files. We do that like this:

```
app.use('*', (req, res) => {
  res.sendFile(__dirname + '/gcp-demo/index.html');
});
```

Add this just after the 'serveStatic' line – the ordering is very important. By putting this line after the static files we ensure that if the request is for a file that actually exists – 'styles.css', for example – then the static handler will intercept the request first and serve the requested file. The request will never reach the '\*' (wildcard) route handler. Anything that doesn't match an existing file will be handled by the catch-all wildcard route.

'main.js' is a weighty 237 KB! Gone are the days when we'd aim for less than 100 KB.

If you open up the Network panel in DevTools and switch to the 'large request rows' view you will see that the two sizes listed for each of our app's files are either similar or identical. Bootstrap, however, has two very different numbers. The number at the top, in black, is the size of the file sent over-the-wire – the response size – and the number below in grey is the size of the file on disk – the content size.

Let's enable compression and see what effect it has. Currently the total size is showing 319 KB.

Open up your server's 'index.js' again and import the compression package.

```
const compression = require('compression');
Then before the serveStatic line add this:
app.use(compression());
```

That's it! Restart your server and refresh. Now the 'main.js' file is just 63 KB – a saving of over 70 per cent! The entire app is coming in at just 109 KB, almost a third

## Get the app

Amazon Web Services are probably still the market leader for cloud computing but Google have made a concerted effort to catch up. Particularly impressive is the Google Cloud Console app.

**Right**  
After creating your new project on Google App Engine this is what you should see

Welcome to App Engine

A powerful platform to build web and mobile apps that scale automatically [Learn more](#)

Your first app

Learn how to build and deploy on App Engine with a simple "Hello World" app. If you're new to App Engine, then start here.

Select a language

App Engine Docs

Learn more about App Engine's capabilities and features, and download the App Engine SDK to set up your local environment.

Browse docs

Download Google App Engine SDK

The screenshot shows the AWS Free Tier landing page. At the top, there's a navigation bar with links like Contact Sales, Products, Solutions, Pricing, Getting Started, Documentation, More, English, and My Account. Below the header, the title "AWS Free Tier" is prominently displayed. A sub-header below it reads: "The AWS Free Tier enables you to gain free, hands-on experience with the AWS platform, products, and services." A yellow button labeled "Create a Free Account" is visible. The main content area is titled "AWS Free Tier Details" and includes filters for "FEATURED", "12 MONTHS FREE", "ALWAYS FREE", "TRIALS", "PRODUCT CATEGORIES", and "ALL". It lists three services with their respective free tiers: Amazon CloudFront (Storage & Content Delivery), Amazon DynamoDB (Database), and Amazon EC2 (Compute). Each service has a summary, a value (e.g., 50 GB for CloudFront), a unit (e.g., data transfer out), and a brief description.

Service	Free Tier	Unit	Description
Amazon CloudFront	50 GB	data transfer out	Web service to distribute content to end users with low latency and high data transfer
Amazon DynamoDB	25 GB	of storage	Fast and flexible NoSQL database with seamless scalability
Amazon EC2	750 Hours	per month	Resizable compute capacity in the Cloud

of what it was before – perhaps we could get it to under 100 KB after all.

Now, GAE does actually do this for you automatically, but it's such an impressive improvement for such a simple thing that we thought it was worth demonstrating.

### 13. One last package

The last thing we like to do is add some basic protections to help secure your app. There is a great package called Helmet, which we've already installed, that adds 12 different middleware functions to Express that set various headers. Out of the box only seven are enabled, but you can configure it further to suit your needs. Again we need to import the package:

```
■ const helmet = require('helmet');
```

And then add it to Express, before the compression:

```
■ app.use(helmet());
```

Restart the server and refresh. Open up the Network panel and check the Response Headers for an app file. You'll see the defaults, such as 'X-XSS-Protection', and you may also notice that the 'X-Powered-By: Express' header has disappeared.

As the website says, it's not a silver bullet, but it helps. We'd definitely recommend spending some time to learn about and set up the Content Security Policy.

### 14. A little npm trick

Currently we have two 'package.json' files: one for the Angular app, and one for the production server. This means we have to run 'npm install' twice. But there is a neat little trick we can do with npm to save us that additional step.

In your terminal ensure you're at the app root where the Angular 'package.json' is. Then we can install the production server as a dependency of the Angular app, like this:

```
■ npm install ./dist --save
```

Now when we run 'npm install' at the root it'll also install the server's dependencies. Neat.

### 15. Cloud Console

Now it's time to login to the Google Cloud Console at [console.cloud.google.com](https://console.cloud.google.com). If you've not already created an account now is the time to do so. Once you're in click on New Project and give your project a name.

Once it is ready, click on App Engine in the menu and you should see the Welcome page.

Then click on 'Select a language' in the blue box and choose Node.js, which will start a step-by-step wizard to walk you through the process. At the Region step choose 'europe-west2' – this is the London datacentre.

Once the billing has been enabled, Google Cloud will offer you a tutorial – we're going to skip this option for now, though.

### 16. Prepare for deployment

So we've got our production server and we have a project in the cloud waiting for us to deploy something.

We just need to complete a few more steps so that we're deployment ready.

First, we need to add a little configuration file for GAE. Create a file call 'app.yaml' in the 'dist' directory and add this to tell GAE to use Node.js v8:

```
■ runtime: nodejs8
```

### Deploying to the cloud

Back in the old days, a 'traditional' hosting company charged around £40 per month for your own VPS. It could be used to host various client projects ranging from WordPress installs to simple, one-page static sites. But rarely did it ever get even close to being used to its full potential. Ultimately, it was more hassle than it was worth; a user had to maintain the VM, update the OS, fix vulnerabilities and generally do all the things that were hard to understand and didn't want to do – and pay more for the privilege!

An alternative is to start hosting a portfolio or project on AWS. It's a lightweight, static site so it should be quick anyway, but on AWS it is blazing fast. Plus AWS's CloudFront distributes to edge locations so it is fast globally. There's no need to maintain anything or monitor uptime – there are leagues of experts to do that for you 24/7. And the best bit is this only costs about £1.50 per month.

Deploying to the cloud is definitely the way forward. You only pay for what you use. Scaling is easy. Someone else is responsible for keeping it online. And you get access to global, cutting-edge infrastructure.

Next, we need to add a couple of things to the 'package.json' in our 'dist' directory. In the 'scripts' object we are going to add two things:

```
■ "start": "node index.js",
  "deploy": "gcloud app deploy"
```

These are like aliases that we can run from npm, so now 'npm start' will run 'node index.js' and 'npm run deploy' will tell the Google Cloud SDK to deploy our app.

And finally we need to make sure we've got a production build ready by running 'ng build --prod'.

Before we go any further, search for Google Cloud SDK and follow the instructions to download and install the command-line tool. Initialize the SDK and then login from the prompt.

### 17. Let's deploy

OK, go ahead... from your 'dist' directory run this and follow the prompts:

```
■ npm run deploy
```

This will zip up your files, upload them to the server, unzip them and then start the relevant services. You can run it multiple times and it will only upload modified files.

Once the upload has completed you should be able to run 'gcloud app browse' and your freshly deployed app will be opened in your browser.

Take a look at the Network panel in DevTools, right-click on the column headings and enable the Protocol column. You'll see our files are served using 'h2' (HTTP/2) and 'https' without requiring any complicated configuration from us.

So that wasn't too bad, was it? But how does it compare to Firebase?

## Developer tutorials

### Deploy your SPA to Google Cloud

The screenshot shows the Netlify homepage. At the top, there's a navigation bar with the Netlify logo, 'Features', 'Pricing', 'Docs', 'Blog', and a search icon. To the right are 'Log in' and 'Sign up →'. Below the navigation is a large blue banner with the text 'Build, deploy, and manage modern web projects'. Underneath, it says 'Get an all-in-one workflow that combines global deployment, continuous integration, and HTTPS. And that's just the beginning.' A green button labeled 'Get started for free' is visible. To the right of the banner is a stylized illustration of a person working at a computer. Below the banner, the text 'Deploy your site in seconds\*' is prominently displayed. At the bottom, there are three steps: '1 Connect your repository', '2 Add your build settings', and '3 Deploy your website'. A dropdown menu for 'Branch' is set to 'master'.

#### Cloud providers

This tutorial only covers Google Cloud Platform and Firebase, but there are plenty of other great options out there.

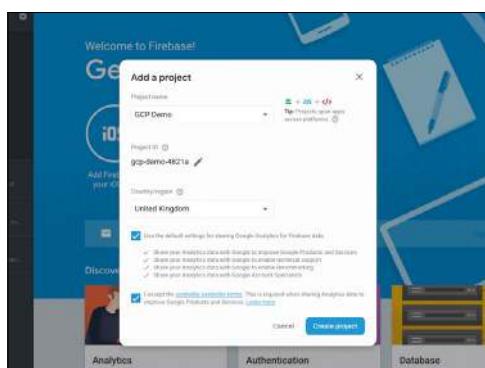
AWS, Google Cloud Platform and Azure are the big players and offer all sorts of complicated and wonderful things, however the learning curve can be pretty steep.

Because of this there are a number of other providers that utilise the big player's infrastructure but provide their own abstraction for particular purposes that greatly simplifies the developer's job. Firebase does this by providing a super-easy deployment process to GCP. Heroku does a similar thing for AWS. Netlify – the new kid on the block – sits on top of (at least) GCP, AWS and Rackspace Cloud to minimise downtime. The compromise of this simplicity is usually the cost.

## 18. Firebase

Firebase has its own console over at [console.firebaseio.google.com](https://console.firebaseio.google.com). Log in (or sign up if you haven't already) and click on the Add project button.

Give the project a name, choose your region and click on 'Create project'. It'll take a few moments to provision.



For your public directory you need to enter 'dist/gcp-demo' as this is where your compiled project resides.

At the next prompt it'll ask you if you want to configure as a single-page app. Enter 'Yes' here. You'll then probably be asked if the existing 'index.html' should be overwritten, in which case enter 'No'.

All done. This has simply created two files in your project that configure Firebase.

## 20. Time to deploy

Now we're ready to hit the deploy button. To do that, simply run the following:

**firebase deploy**

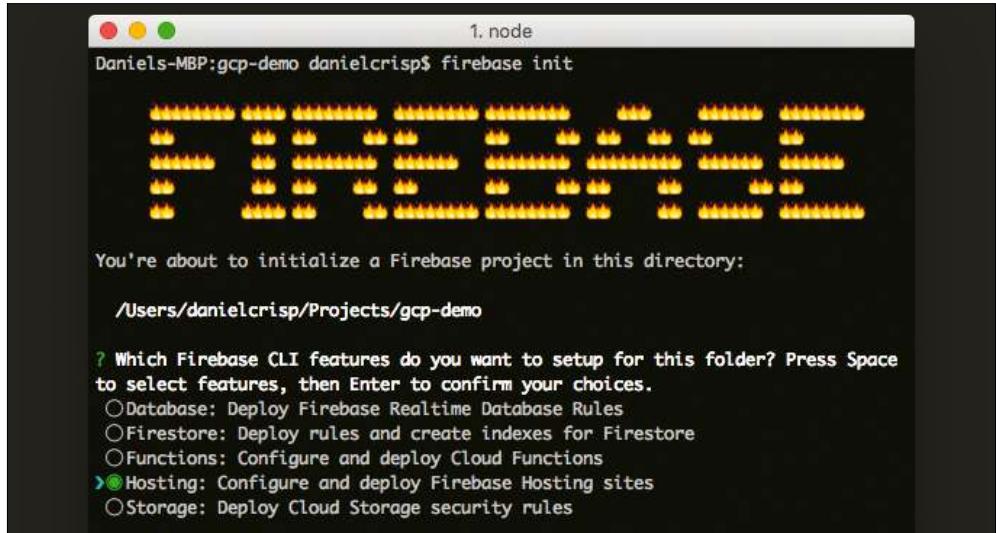
As with Google App Engine, this will upload all of your files and manage the process for you. Once it has completed

you'll be shown your app's URL – open it in your browser to check that everything is OK.

## 21. That's all there is to it

Clearly Firebase simplifies the process but the result is pretty much the same. Once you've done the legwork of creating your own production server the deployment process of GAE is just as simple as Firebase's. Some (unscientific) performance analysis – AKA refreshing repeatedly – shows both hosts load in the same kind of time, but perhaps Firebase's global CDN would help if we were not in the app's region. Although maybe GAE has this, too – we're not really sure.

Either way the choice is up to you, and what is right for your project.



## 19. Install the tools

Once it is ready continue to the 'Get started' page and scroll down to the Hosting box. Click the 'Get started' button. You should see a prompt instructing you to install the 'firebase-tools' package with npm. Do that.

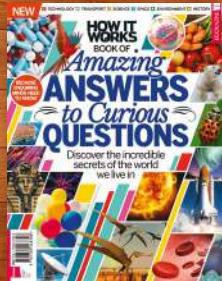
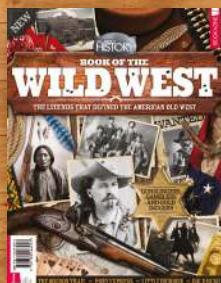
Once it is installed ensure you are in the root of your project and run the following command to login:

**firebase login**

Now you need to run the 'init' command to add the configuration files for Firebase:

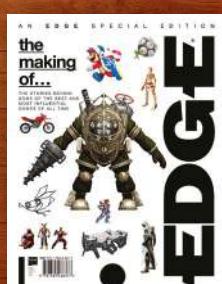
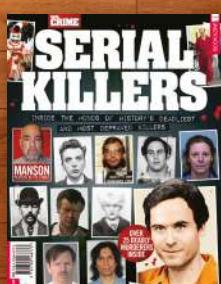
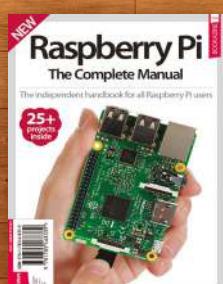
**firebase init**

At the first prompt select Hosting. If you are prompted to select your Firebase project, choose the one you've just created. Next you'll be asked about your Hosting Setup.



## Discover another of our great bookazines

From science and history to technology and crafts, there are dozens of Future bookazines to suit all tastes



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



**www.myfavouritemagazines.co.uk**  
Magazines, back issues & bookazines.



## Get your listing in our directory

To advertise here contact *Chris*

[chris.mitchell@futurenet.com](mailto:chris.mitchell@futurenet.com)

+44 (0)1225 687832

# HOSTING LISTINGS



## Featured host: Netcetera

[netcetera.co.uk](http://netcetera.co.uk)  
03330 439780

### About us

Formed in 1996, Netcetera is one of Europe's leading web hosting service providers, with customers in over 75 countries worldwide.

As the premier provider of data centre colocation, cloud hosting, dedicated servers and managed web hosting services in the UK, Netcetera offers an array of services designed to more effectively manage IT

infrastructures. A state-of-the-art data centre environment enables Netcetera to offer your business enterprise-level colocation and hosted solutions.

Providing an unmatched value for your budget is the driving force behind our customer and managed infrastructure services. From single server to fully customised data centre suites, we focus on the IT solutions you need.

### What we offer

- Managed hosting** - A full range of solutions for a cost-effective, reliable, secure host.
- Cloud hosting** - Linux, Windows, Hybrid and Private Cloud Solutions with support and scalability features.

- Data centre colocation** - Single server through to full racks with FREE setup and a generous bandwidth.
- Dedicated servers** - From QuadCore up to Smart Servers with quick setup and fully customisable.

### 5 tips from the pros

#### 1. Reliability, trust & support

Reliability is a major factor when it comes to choosing a hosting partner. Netcetera guarantees 100 per cent uptime, multiple internet routes with the ability to handle DDOS attacks, ensuring your site doesn't go down when you need it.

knowledgeable staff available 24/7 to provide you with assistance when you need it most. Our people make sure you are happy and your problems are resolved as quickly as possible.

#### 2. Secure and dependable

Netcetera prides itself on offering its clients a secure environment. It is accredited with ISO 27001 for security along with the options of configurable secure rackspace available in various configurations.

#### 4. Value for money

We do not claim to be the cheapest service available, but we do claim to offer excellent value for money. We also provide a price match on a like-for-like basis, as well as a price guarantee for your length of service.

#### 5. Eco-friendly

Netcetera's environmental commitment is backed by use of eco-cooling and hydroelectric power. This makes Netcetera one of the greenest data centres in Europe.



### Testimonials

#### Roy T

"I have always had great service from Netcetera. Their technical support is second to none. My issues have always been resolved very quickly."

#### Suzy B

"We have several servers from Netcetera and their network connectivity is top-notch, with great uptime and speed is never an issue. Tech support is knowledgeable and quick in replying. We would highly recommend Netcetera."

#### Steve B

"We put several racks into Netcetera, basically a complete corporate backend. They could not have been more professional, helpful, responsive or friendly. All the team were an absolute pleasure to deal with, and nothing was too much trouble, so they matched our requirements 100 per cent."

## Supreme hosting



[cwcs.co.uk](http://cwcs.co.uk)  
08001777000

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 per cent network uptime

## UK-based hosting



[cyberhostpro.com](http://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.co.uk](http://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. Services can be self-managed through the Fasthosts Control Panel.

- Dedicated servers
- Cloud servers
- Hosted email



## Budget hosting

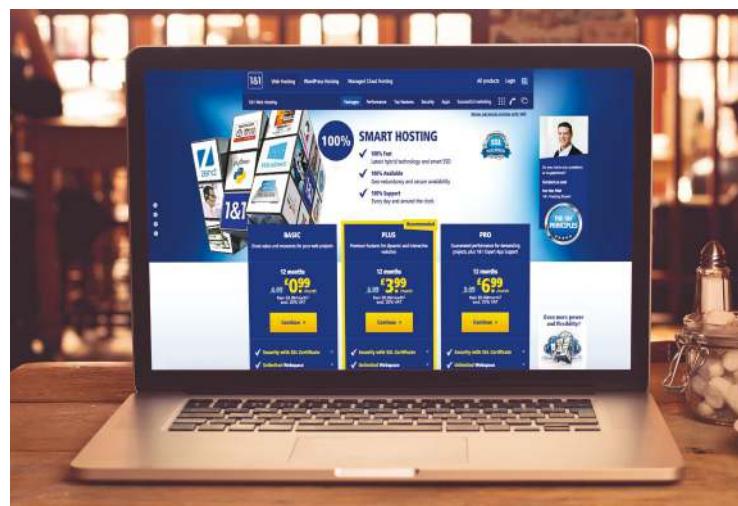


[hetzner.com](http://hetzner.com)  
+49 (0)9831 505-0

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 nationally and internationally.

- Dedicated/shared hosting
- Colocation racks
- SSL certificates



## All-inclusive hosting



[1and1.co.uk](http://1and1.co.uk)  
0333 336 5509

1&1 Internet is a leading hosting provider that enables businesses, developers and IT pros to succeed online. Established in 1988, 1&1 now

operates across ten countries. With a comprehensive range of high-performance and affordable products, 1&1 offers everything from simple domain registration to award-winning website building tools, eCommerce packages and powerful cloud servers.

- Easy domain registration
- Professional eShops
- High-performance servers

## SSD web hosting



[bargainhost.co.uk](http://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

## Value Linux hosting



[patchman-hosting.co.uk](http://patchman-hosting.co.uk)  
01642 424 237

Linux hosting is a great solution for home users, business users and web designers looking for cost-effective and powerful hosting. Whether you are building a single-page portfolio, or you are running a database-driven eCommerce website, there is a Linux hosting solution for you.

- Student hosting deals
- Site designer
- Domain names

## Flexible cloud servers



[elastichosts.co.uk](http://elastichosts.co.uk)  
020 7183 8250

ElasticHosts offer simple, flexible and cost-effective cloud services with high performance, availability and scalability for businesses worldwide. Their team of engineers provide excellent support 24/7 over the phone, by email and with a ticketing system.

- Cloud servers with any OS
- Linux OS containers
- 24/7 expert support



## Get your listing in our directory

To advertise here contact *Chris*

[chris.mitchell@futurenet.com](mailto:chris.mitchell@futurenet.com)

+44(0)1225 687832

# COURSE LISTINGS



## Featured: **Northcoders**

[northcoders.com](http://northcoders.com)  
Twitter: @northcoders  
Facebook: Northcoders

### About us

Northcoders is the coding bootcamp for the north, based in the heart of Manchester and built upon northern values of grit, determination and community spirit. No matter what your background, you can fast-track your career and become a web or software developer in 12 weeks at their

full-time bootcamp, or fit their course around your life with their 24-week part-time bootcamp. Their internal career support team will help find you work as a developer, setting up interviews with your choices of Northcoders Hiring Partners across the north of England.

### What we offer

- **Full-time:**  
Fast-track your career in just 12 weeks.
- **Part-time:**  
Fit our curriculum around your life in 24 weeks.

### 5 tips from the pros

#### 1. Get started with coding

The best way to know if coding is for you is to just try it! We recommend the free, online JavaScript track of Codecademy to get you started with the basics.

for you, set aside a few evenings each week to really start making progress! If coding is for you, this should be fun.

#### 4. Be prepared

We'll be with you every step of the way when you apply. Make sure you go through all the materials we recommend and ask for help if you're stuck.

#### 5. Get social

With Northcoders, you're not just on a course, you're part of a community that will stay with you long after you graduate. Make the most of it!

#### 2. Do your research

Make sure you read plenty of student reviews to make sure you're applying somewhere reputable. Read their blog and have a look at their social channels.

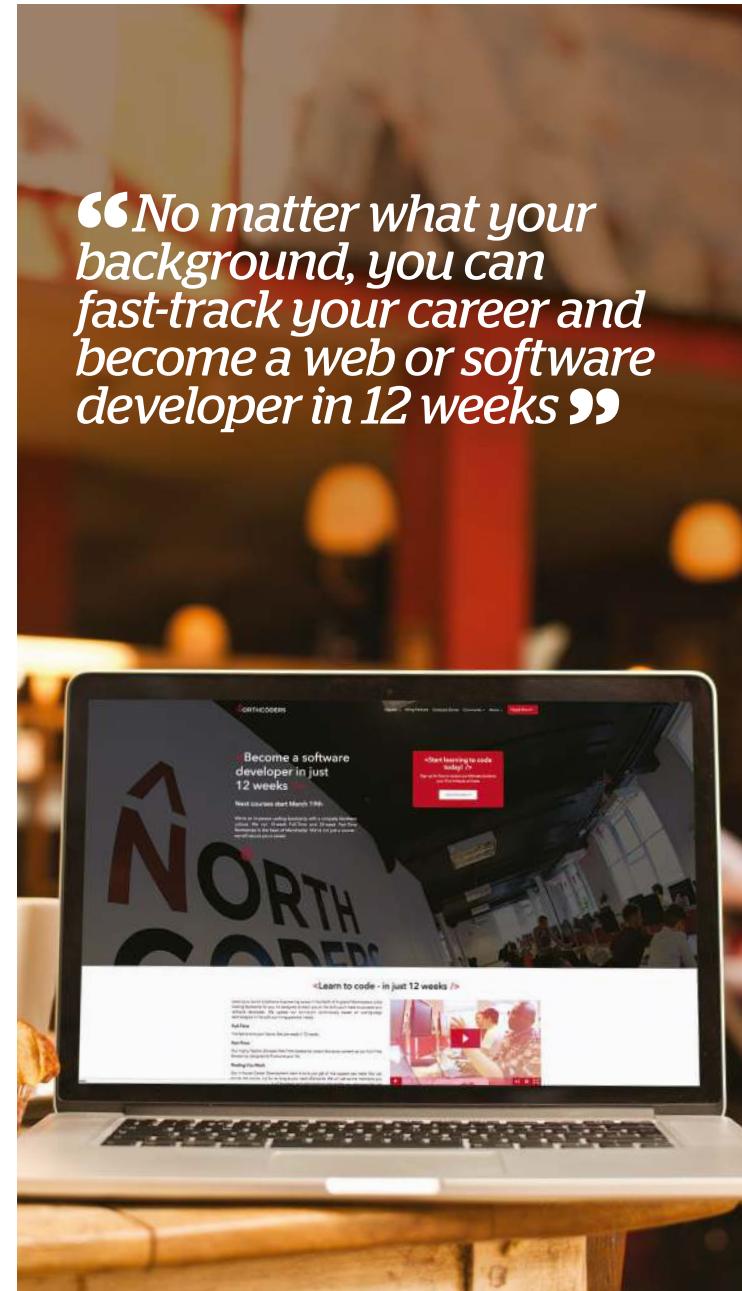
#### 3. Throw yourself in

Once you've decided it's right

**“** Becoming part of this vibrant, caring community was something I hadn't expected before the course, but now I couldn't be without it. To be a Northcoder is to be enlightened, inspired and supported.

**Joanne Imlay**

*Primary school teacher to software developer at Careicon*



**“** Northcoders delivered their part of the bargain in spades. They provided tremendous assistance in turning me into the full product - a well-rounded, capable, future tech employee - and they have the contacts to deliver the opportunities for such people.

**Joe Mulvey**

*Maths teacher to software developer at Auto Trader*

# udemy

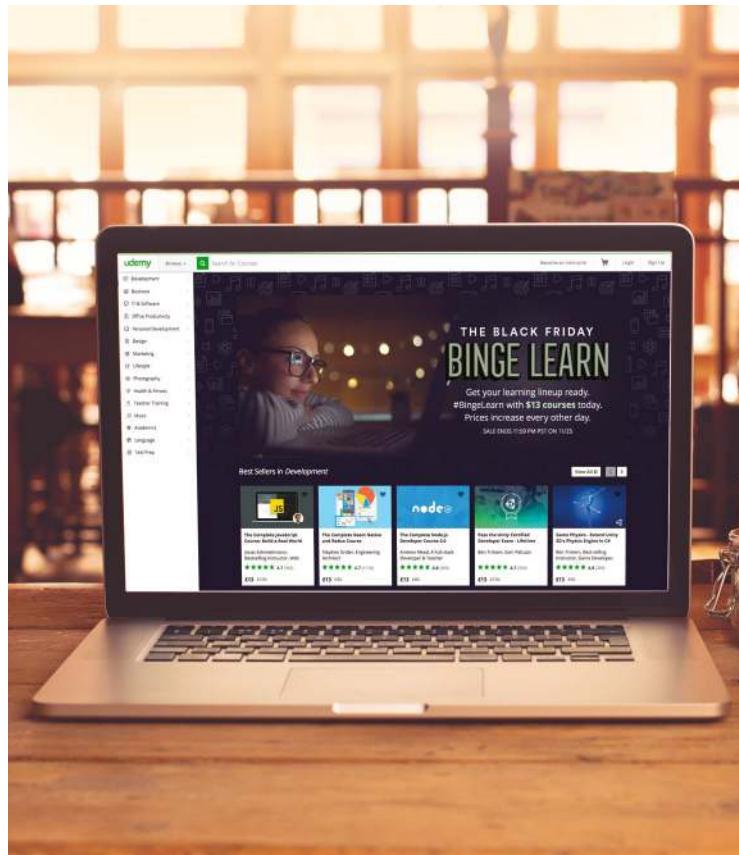
## UDEMY

[udemy.com](http://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.



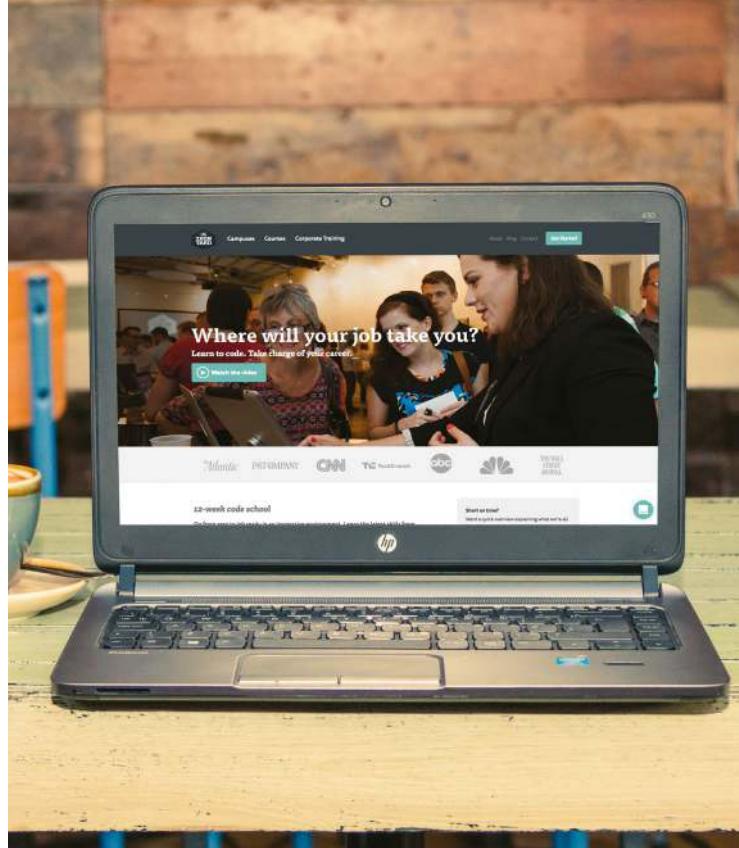
## THE IRON YARD

[theironyard.com](http://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 backend engineering, frontend 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



## WE GOT CODERS

[wegotcoders.com](http://wegotcoders.com)

[hello@wegotcoders.com](mailto: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 courses 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



[futurelearn.com](http://futurelearn.com)

[feedback@futurelearn.com](mailto: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



[thegymnasium.com](http://thegymnasium.com)  
[help@thegymnasium.com](mailto: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.

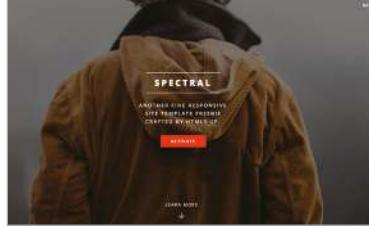
- Gain real-world skills
- Get expert instruction
- Career opportunities

# Free with your magazine

Instant access to these creative resources...

## Essential assets and resources

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
- Two more chapters from the Beginners JavaScript video series from Killersites ([shop.killervideostore.com](http://shop.killervideostore.com))
- 121 Duotone PS filters and 12 Simple business card mockups from Sparklestock ([www.sparklestock.com](http://www.sparklestock.com))

→ Log in to [www.filesilo.co.uk/webdesigner](http://www.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



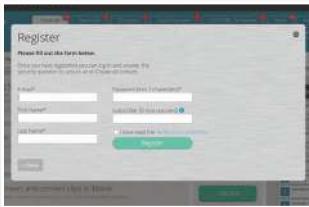


The home of great downloads – exclusive to your favourite magazines from Future!

- 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@futurenet.com](mailto:filesilohelp@futurenet.com)

# An incredible gift for subscribers



## Subscribe today & unlock the free gifts from more than 50 issues

Access our entire library of resources with a money-saving subscription to the magazine — that's more than 900 free resources

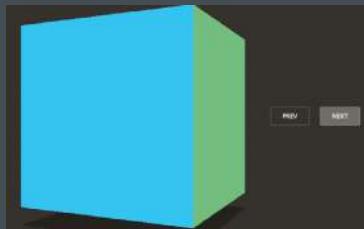
### Over 60 hours of video guides

Let the experts teach you to create and code



### More than 400 tutorials

Get the code you need to get creative



### Over 250 creative assets

Templates, fonts, textures and backgrounds



Head to page 32 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 ten-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

# NEXT MONTH

---

# JAVASCRIPT

---

## What's now? What's next?

---

Get a closer look at the best new ES8 techniques, plus  
the latest and greatest JS libraries and frameworks

---

### CODE VISUAL MOTION DESIGN

Learn how to create contemporary dynamic design, realised through the power of WebGL

### GET STARTED WITH THREE.JS – PT3

In the third part of the series learn how to add textures and materials to objects

### SAY HELLO TO FLUTTER

Learn what Google's mobile app SDK has to offer and how to start creating for iOS and Android

Visit the **WEB DESIGNER** online shop at



**myfavouritemagazines**

[myfavouritemagazines.co.uk](http://myfavouritemagazines.co.uk)

for the latest issue, back issues and specials

ALL IN YOUR NEXT  
**WEB DESIGNER**  
Issue 278 on sale  
**Tuesday 21st August 2018**

# PUT A PAUSE IN YOUR DAY

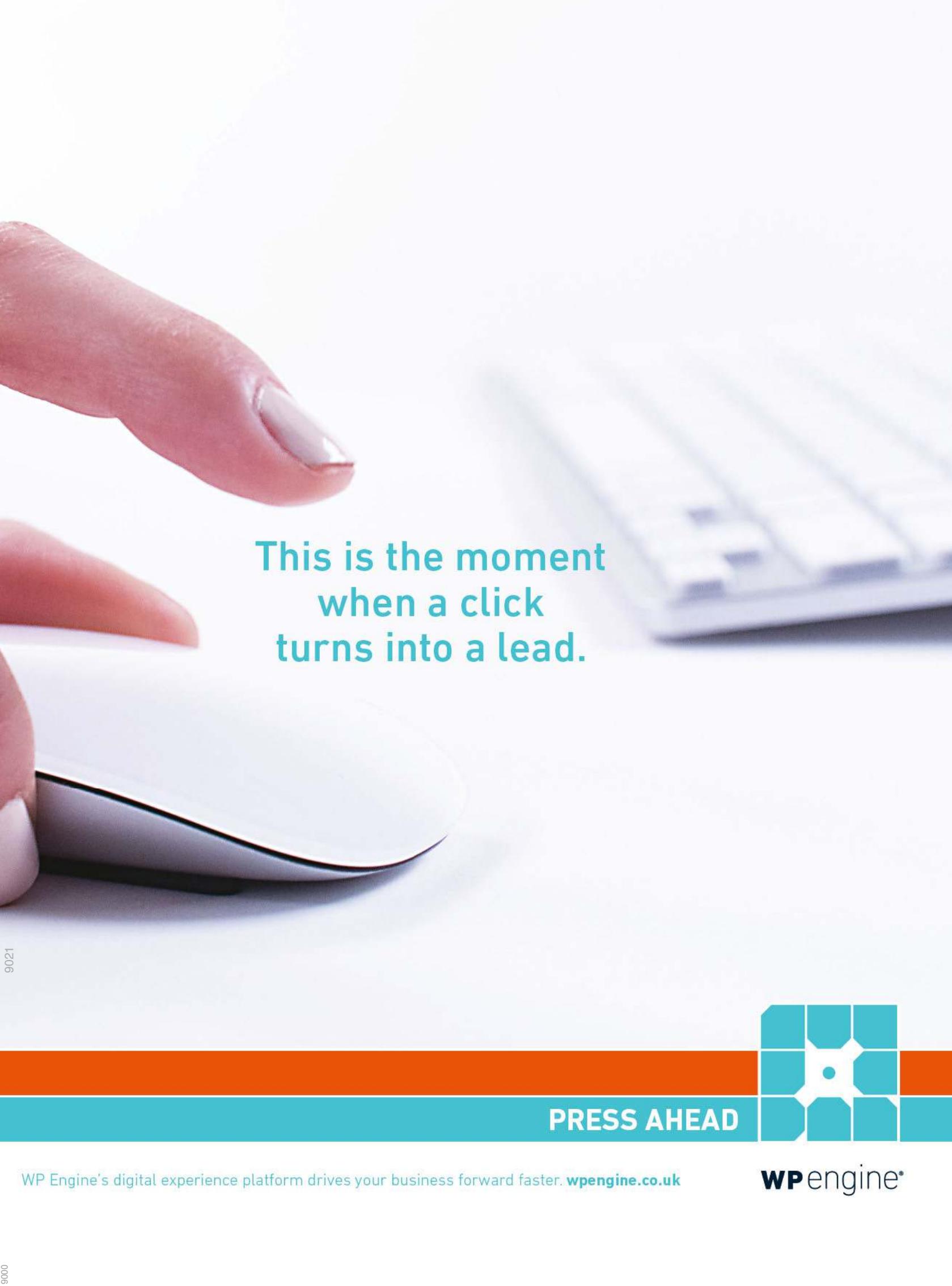
With so many demands from work, home and family, there never seem to be enough hours in the day for you. Why not press pause once in a while, curl up with your favourite magazine and put a little oasis of 'you' in your day.



**PRESS PAUSE**  
ENJOY A MAGAZINE MOMENT

To find out more about Press Pause, visit:

**pauseyourday.co.uk**



This is the moment  
when a click  
turns into a lead.

PRESS AHEAD

WP Engine's digital experience platform drives your business forward faster. [wpengine.co.uk](http://wpengine.co.uk)

**WP** engine\*