

29
PAGES OF
EXPERT TUTORIALS

web designer

FREE OVER 3 HOURS
OF JS VIDEOS



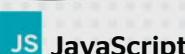
HTML5



CSS3



jQuery



JavaScript



@WebDesignerMag



/WebDesignerUK

28

PAGE
SPECIAL

COLOUR

CREATE THE
PERFECT PALETTE

TYPE

TOP TRENDS
FOR TODAY

LAYOUT

DESIGN & BUILD
GRID LAYOUTS

FONTS

10 GOLDEN RULES
OF FONT COMBOS

GET PRO DESIGN SKILLS



ISSUE 271

5 STATIC SITE
GENERATORS

10 of the best to instantly
build HTML pages

3 SHRINK
YOUR CSS

Make your style
sheets more efficient

 AUTOMATE
WORKFLOW

Utilise Gulp to get rid
of repetitive tasks

JS REACT &
TYPESCRIPT

Get the JS libraries
to work in unison

(mt)

DRAW
—
WIRE
—
DESIGN
—
CODE

Media Temple web hosting delivers premium performance and support so that creative professionals and enterprises can focus on bold ideas, not web servers.

mediatemple.net

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

Be better at design



hat makes a good web page/site/app design? It's a pretty broad question but you can guarantee that it will include fonts, colour and layout. These are all essential components of the process and getting them right will go a long way to creating a design that works.

This issue **Web Designer** has decided to bring together a swathe of tips, techniques and insight on fonts, type, colour and layout. It all kicks off on page 42 where we take a closer look at why you need to pick the right font, what a font portrays to the reader, the golden rules for picking font combinations and what is trending in the world of typography. UX/UI designer and Better Web Type

author Matej Latin reveals his six steps to achieving meaningful typography. Then it's on to colour. Discover what you need to know to get the perfect palette, how colour and accessibility should work together, the tools to help get the perfect colour scheme and what colours suggest to users. We take an in-depth look at the process of creating and designing layouts and how CSS Grid fits into the process. There is also a peek at what lies ahead for layouts. CSS Regions anyone?

Do you want to build a HTML page in double quick? Of course you do. We reveal 10 of the best static site generators to help make this happen.

Plus, we have the usual collection of must-read tutorials and interviews. Enjoy.

Layout is one of the most important aspects of the web and applications. Get it wrong and your users will get confused and leave

Highlight

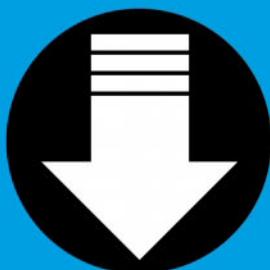


We want to maintain our commitment to the advancement of women from coding to leadership

Dataviz specialists in an information-driven world, FFunction make sense of information. **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
171 Designer resources

Video Tuition - Over three hours of expert JavaScript video guides from Pluralsight ([pluralsight.com](#))

Assets - 150 Nostalgia Lightroom presets and Dot Grid Photoshop actions from Sparklestock ([sparklestock.com](#))
- Tutorial files and assets



[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...

GET PRO DESIGN SKILLS



Carl Cahill

Carl is a creative who has been in the industry for over 10 years with a passion for UID and has previously been awarded Creative of the Year. This issue Carl uses his knowledge to reveal how to pick the right font, the rules for combining fonts, how to choose the right colours and the tools and resources to help. **Page 42**

Typography and colour are the minimum components you would expect to use to form a design and create content that has a distinct look and feel.

Steven Roberts



Steven is a multi-disciplined creative developer from Teesside. He's been designing and building layouts for over 10 years. This issue he'll be revealing how to create layouts with a practical focus on CSS Grid. **Page 62**

Mark Shufflebottom



Mark is a Professor of Interaction Design at Sheridan College near Toronto. This issue he is exploring CSS4 variables and how they can help designers experiment further with their web designs. **Page 70**

Simon Jones



This issue Simon digs into the depths of his knowledge to talk about static site generators. They offer a simple way to produce high performance static pages for the web. Here he chooses ten of the best out there. **Page 74**

Mark White



Mark White is Techniques editor on sister mag Photoshop Creative. He has been using Photoshop for over ten years. This issue he shows how to create an impressive isometric design that relies on vital features in Photoshop. **Page 56**

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 seen on the top-class sites seen in Lightbox. **Page 14**

Tam Hanna



Tam has a wealth of experience in a breadth of different development languages. This issue he takes a closer look at how to bring together TypeScript and React with the help of the module bundler webpack. **Page 86**

Matej Latin



Matej is the author of Better Web Typography for a Better Web and also the lead UX/UI designer at Auto Trader UK. This issue he unveils the six steps needed to create meaningful typography with some great examples. **Page 46**

Luke Harrison



Luke is a web developer from Sheffield, UK who is all about scalable & efficient front-end architecture. In this issue, he explores how you can automate your workflow with Node.js powered task runner Gulp. **Page 80**

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 Steve Jenkins
steve.jenkins@futurenet.com
01202 586233

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

Contributors

James Norman, Mark Billen, Leon Brown, David Howell, Carl Cahill, Matej Latin, Mark Shufflebottom, Steven Roberts, Mark White, Simon Jones, Luke Harrison, Tam Hanna, Philip Morris, Rob Mead-Green

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
Senior Advertising Manager **Mike Pyatt**
michael.pyatt@futurenet.com
01225 687538

Account Director **George Lucas**
george.lucas@futurenet.com
Account Manager **Chris Mitchell**
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

Subscriptions

Email enquiries 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
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 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 2008885) 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 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

Chief executive **Zillah Byng-Thorne**
Non-executive chairman **Peter Allen**
Chief financial officer **Penny Ladkin-Brand**
Tel +44 (0)1225 442 244

**Are you
looking for a
new.direction?
It's time to
take_action()**

Learn to code in **12 weeks** at the
Coding Bootcamp for the North.

We're not just a course. We'll secure you a career.
www.northcoders.com/web-designer



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...

steve.jenkins@futurenet.com • [@WebDesignerMag](https://www.twitter.com/WebDesignerMag) • www.creativebloq.com

08 Creative trends to watch out for

Shutterstock recently released its 2018 Creative Trends Report. Find out what's coming your way

10 WebKit: The best must-try resources out there

Discover the libraries and frameworks that will make your site a better place to visit

11 No excuse for a bad UI

James Norman from Fasthosts tells you what makes a good UI and why it's important

14 Lightbox

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

26 Angelic airplanes

Russian agency Red Collar reveals how it delivered Inculerate's distinctive new design

34 Form and FFunction

Web Designer finds out how dataviz specialist FFunction makes data beautiful and engaging

42 Get pro design skills

Learn how pick the right font, create meaningful typography, pick the perfect palette and build brilliant layouts

62 Master layouts

What makes a layout great? Discover the secrets behind designing and building what you need

74 10 best static site generators

Want to build HTML pages in double-quick time? Then try one of these carefully selected options

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



62

Master layouts

How to design and create the perfect page



74

10 best static site generators

Create HTML pages in double-quick time

FileSilo

96 Get the latest must-have resources and videos

- Over three hours of expert JavaScript video guides from Pluralsight
- 150 Nostalgia Lightroom presets
- Dot grid Photoshop actions



D
O
N

Never miss
an issue
Subscribe
Turn to page 32 now

Save 20 percent and get the
magazine delivered to your door



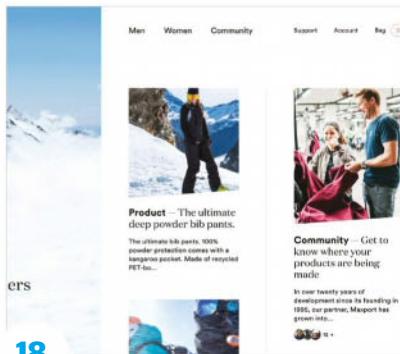
66 A good user interface design is characterised by aesthetics, ease of use and functionality

COMMENT — James Norman — p11



34

Profile: FFunction
Making data beautiful and engaging



18

Lightbox: Open Wear
Dressing up eCommerce

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

Tutorials

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



52 Code a dynamic SVG menu

Use SVG, jQuery and the GSAP animation library to create an animated radial menu

56 Design isometric typography

Learn how to build a city of letters, using the Pen tool, selections and layers in Photoshop

70 Shrink your CSS

Use CSS4 variables to change a variable and every rule that depends on it

Web Workshop

50 Overlay graphics on video

project-redspace.com

Create designs to complement video backgrounds

60 Multiple fade background colours

rocka.co

Code background colour changes on scroll

Web Developer

74 10 best Static Site Generators

Discover the best options for creating HTML pages in double-quick time

80 Automate your workflow

Learn how to remove the nuisance of repetitive and mundane tasks with the help of Gulp

86 Wire up TypeScript & React

TypeScript makes coding easier, but deploying it with a framework can be a hassle. Webpack to the rescue

Header

The tools, trends and news to inspire your web projects

What are the creative trends to watch for?

After proving pretty sage for 2017's hottest trends, the folks at Shutterstock forecast the year's most influential visual styles

Seeing the future would be a gift most designers and influencers would die for. In truth, the most talented are more perceptive than lucky and can gauge current climates or social tastes to engineer solutions tomorrow's world will want to see. Web design innovators are no exception and tend to be most adept at tapping into visual moods and fashions, not totally without insider help it seems.

Shutterstock recently released its 2018 Creative Trends Report which derives eleven styles by analysing billions of customer searches within the company's vast repository of stock imagery. Paying most attention to those browsing habits of clients in the most influential trades such as film, TV, media and advertising, Shutterstock has some form when it comes to calling out creative cool. "Our 2017 report was particularly accurate in its predictions, identifying trends like 'Glitch' early on," Shutterstock's Curator Robyn Lange

explains. So what are the trends set to turn heads in 2018 you may well ask?

Well topping the report are three major trends with "Fantasy" proving most bankable, represented by a hunger for mythical characters, imagery and even sprawling orchestral music too. Search surges for "continuous line" and "neon

Aside from growing popularity for ancient geometrics, tactile craft wares such as embroidery, plus any imagery for cryptocurrency, Shutterstock reserve special mention for holographics. It might be bringing up the rear at the report's position eleven but we recognise this one particularly among the trends observed in

“Topping the report are three major trends with “Fantasy” proving most bankable”

circle" styles place "New Minimalism" in second spot before "Space" completes the trendiest trio to watch. While the latter's recent popularity seems to reflect box-office cinema releases such as Star Wars, other tips are more abstract forms including the "Natural Luxury" of marbles and crystal, candy-coloured "Punchy Pastels" and, bizarrely, Cactus plants.

recent site picks. Shimmering rainbow, prism-like colour palettes evoking the undeniable awe that holograms inspire seem a good bet. If that wasn't enough of an insider tip, the report also managed to link many of the selected trends to specific countries. So what will be "rocking" our own creative shores by next Xmas? Terrazzo apparently... Yeah, we'll let you know!



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.

WEB DESIGNER DIGITAL EDITION

Do you want to get your hands on a digital edition of your favourite web design magazine? Head to your preferred app store – Google Play (bit.ly/2wetvGp) or iTunes (apple.co/2igtBYq) – then download, install and purchase the issue of choice from within the app.



STAT ATTACK

MOBILE BROWSERS

Who's popular, and who's not. Where's Android?

Chrome

49.72%

Popular on desktop, popular on mobile



Safari

18.34%

Popular choice on Apple devices



UC Browser

15.76%

A well-known brand from Asia



Opera

5.70%

Found on low-power smartphones



Samsung

5.49%

Popular mobile phone brand



Source: gs.statcounter.com

Sites of the month



01.



02.



03.

01. New Japan Pro-Wrestling KEIBA

umabi.jp/njpw-keiba
Japanese horse racing promo.

04.



02. Portfolio of Kadir Inan

uiix.me
Cologne-based UI/UX designer showcases talent with smoking WebGL effects.

03. Dot Agency

dot-agency.it
Italian agency crowds out the white space here with oversized SVG rollovers.

04. Daijima

daijima.jp
How to choreograph impressive 3D animation skills to music.

Graphics Thunder Rockets

bit.ly/2EXDT6p

Brazilian illustrators Thunder Rockets bring a wacky cartoonish style bursting with colour and packed full of charm.



Colour picker Verdant Peninsula

<http://bit.ly/2FStA4w>



Typesetter Decoral

bit.ly/2Djle4p

Art Deco styling with a modern twist, Decoral by Lasko Dzurovski for Totem boasts 650 glyphs across three weights.

ABCABC
1234567

WordPress Glytch

bit.ly/2rrNuAt
So called for the "glitch" effects employed throughout, Glytch is a set of music-themed homepage templates for selling sounds.



CONTACT US AT: steve.jenkins@futurenet.com | @WebDesignerMag

webkit

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

The screenshot shows a website with a tilted page scroll effect. At the top, there's a navigation bar with links like 'Home', 'About', 'Services', 'Blog', and 'Contact'. Below the navigation, there's a large image of a person sitting on rocks by a body of water with a ship in the background. Overlaid on this image is text: 'TRAVEL TO SYDNEY, AUSTRALIA' and 'Completed on 14 Feb 2014'. A blue button labeled 'Download on Github' is visible. On the left side, there's a sidebar with a red box containing 'FLAT UI STARTUP Theme for Bootstrap 3 + Free UI Pack' and a price of '\$15'. Below that is a small image of a person speaking and the word 'toptal'. At the bottom, there's another image of a landscape with a large red rock formation.

jQuery Tilted Page Scroll

bit.ly/1lijjt3

Visual designer Pete R is known for his plugin work and this one for adding a tilted page scroll effect to a page is a great example. Achieve an eye-catching 3D effect that flips <section> elements up like a stack of cards. Compatible with modern browsers and smartphones.



Animate.css

bit.ly/1qTVdjA

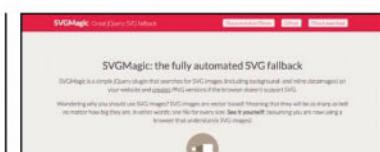
Daniel Eden's collection of "just-add-water CSS animations" can be used to add smooth motion effects and physics to page elements.



Fontjoy

fontjoy.com

Fontjoy is a web app that uses "deep learning" to grab Google fonts and generate complimentary combinations you can customise.



SVGMagic

dirkgroenen.nl/SVGMagic

This clever little plugin is a fallback that doesn't employ magic but jQuery to find SVG elements within a website and swap them for PNGs if unsupported.

TOP 5 Web conferences - March 2018

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



.concat0 2018

2018.concat.at

Inviting the web dev and UX community to Salzberg, this one day event boasts 18 industry speakers.



ngVikings 2018

ngvikings.org

This travelling Angular conference hits Helsinki Finland with a band of speakers including Google experts.



SXSW 2018

sxsw.com

The world-famous Interactive, Film and Music conference and festival open its doors in Austin Texas with a packed schedule.



UX In The City

bit.ly/2Dsd4Xs

This community-led UX conference hosts keynote speakers Sarah Richards, Molly Watt and Chris Atherton.



UX Camp Brighton

bit.ly/2BoSUfm

A smaller, more personal "barcamp" event, that appeals to UX designers.

There is no excuse for bad UI

What makes a good User Interface and why is it important?



James Norman
Content editor
fasthosts.co.uk

“ A good user interface design is characterised by aesthetics, ease of use, and functionality.

A

pple devices are famous for their intuitive user interfaces and good design. The fact these two characteristics are so prevalent in their products is no accident. The late Steve Jobs was famous for his unflinching focus on good design, because he understood how critical design was to the user experience:

“Most people make the mistake of thinking design is what it looks like. People think it's this veneer – that the designers are handed this box and told, 'Make it look good!' That's not what we think design is. It's not just what it looks like and feels like. Design is how it works.”

Having a good user interface means that a customer will know intuitively how to use your website or app without too much help. By contrast, bad user interface design can leave a user abandoning their session frustrated with an empty cart. That's why it's so important to optimise a user interface design, and think about how your customers are interacting with your website.

It is worth taking a step back to see how user interfaces have evolved, in order to understand why users have such high expectations for user interfaces today.

User interfaces have grown from a text-based command-line interface (CLI), to a two-dimensional text-based user interface (TUI), and into the graphical user interface (GUI) in use today. Where CLI and TUI were only interacted with by keyboard, a GUI is controlled by a keyboard, mouse or, ever more commonly, a touchscreen device.

Also growing in popularity are voice user interfaces (VUI) like Siri on Apple devices and Amazon's Alexa, where devices are controlled through speech. For web designers, the GUI remains the most logical interface to focus on, however interaction with the prevailing VUIs is something that should not be overlooked.

A good user interface design is characterised by aesthetics, ease of use, and functionality. The design should be quite minimalistic. An overload of content, images and buttons can confuse users who will search unnecessarily around the screen trying to find the bit that is relevant to them. Remember to include whitespace to break up the content a bit.

Make buttons and options easily recognisable by using existing icons. Think of classic user interface icons like the trash can - most computer savvy users would know what the trash can does. The same goes for the floppy-disk save icon. Even though a lot of the new internet-generation would never have seen a floppy disk in real life, they recognise that the floppy-disk icon is synonymous with saving.

These icons haven't changed over time because they're so iconic and recognisable that users immediately know what they do, and that's a fundamental of good user interface design. You want to make it intuitive. Users expect certain icons and will search for them, so there's no use having a wacky, well-designed icon in place of a traditional, simple and effective one.

You want to make the user interface as simple to use as you can, whilst still giving users all of the functionality they require. And of course you want it to look great.

The bounce rates of your website can have an effect on your SEO (search engine optimisation). If someone lands on your site and bounces straight back because they're overwhelmed and can't figure out how to navigate, then that's going to reflect on your bounce rates, which will affect your search rankings. This is particularly prevalent on mobile devices.

Make sure that your internal linking is intuitive and easy to follow for humans, because search engine web crawlers will look at the internal navigation structure of your site and analyse its effectiveness when deciding on your search ranking.

As well as that, having a page cluttered with too much content or too many features can cause the page to load slower, which can also have an effect on search engine optimisation.

Remember, design is how it works. Strike the right balance between form, functionality and content, and you'll deliver a smooth and intuitive user experience for your customers. Never forget Steve Jobs' words; “Design is not just what it looks and feels like. Design is how it works.”

webkit

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

Devices.css

Modern devices in pure CSS. The first are iPhone X and iPhone 8, and more modern devices are coming.

[Demo](#) [GitHub](#)

Latest version: 0.1.16

[PayPal Donate](#)

iPhone X



Devices.css

picturepan2.github.io/devices.css

Keen to render the very latest smartphones and devices in nothing but CSS? Well if so, Devices.css is a Github listed collection of gorgeously realised new iPhones, Google Pixels, Samsung Galaxy handsets and more. As a lightweight testament to what inventive HTML and minified stylesheets can achieve it's worth a look.



Bulma

bulma.io

Tipped for broad adoption in 2018, Bulma is an open source, mobile-first CSS framework based on Flexbox that promises to be simple to learn and use.



A-Frame

aframe.io

A-Frame is a library for creating WebVR experiences from HTML-based worlds, producing a 3D perspective from two screens at multiple angles.



mark.js

markjs.io

Usable as a jQuery or pure JS plugin, mark.js is a flexible keyword highlighter for dynamically marking search terms within paragraphs of page text.

TOP 5 WordPress themes for February 2018

Shortcut your way to blogging brilliance with a handful of the hottest new themes



Travelmatic

travelmatic.purethe.me/

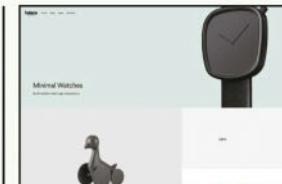
Get Google Maps integration for plotting stop-off landmarks. This theme is ideal for a journey with photos and stories.



Susastro

bit.ly/2F4CWJm

Susastro offers an elegant and clean design perfect for marketing yoga, health and gym services.



Halena

<http://bit.ly/2n6lgFK>

Want to add a touch of sophistication to your eCommerce? Halena offers minimal & modern in abundance



Esquise

<http://bit.ly/2rwXqsl>

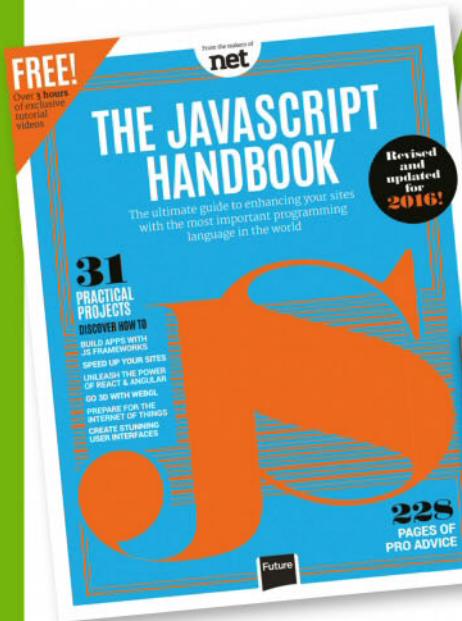
Big, bold and beautiful. This magazine theme will ensure that your content stands out and will get seen.



Ambesten MarketPlace

bit.ly/2DyTKMv

Ambesten is a flexible WooCommerce theme ideal for responsive shop fronts.



FREE The JavaScript Handbook

After completing the survey,
you'll receive a digital copy of
The JavaScript Handbook
as a thank you for
your feedback.



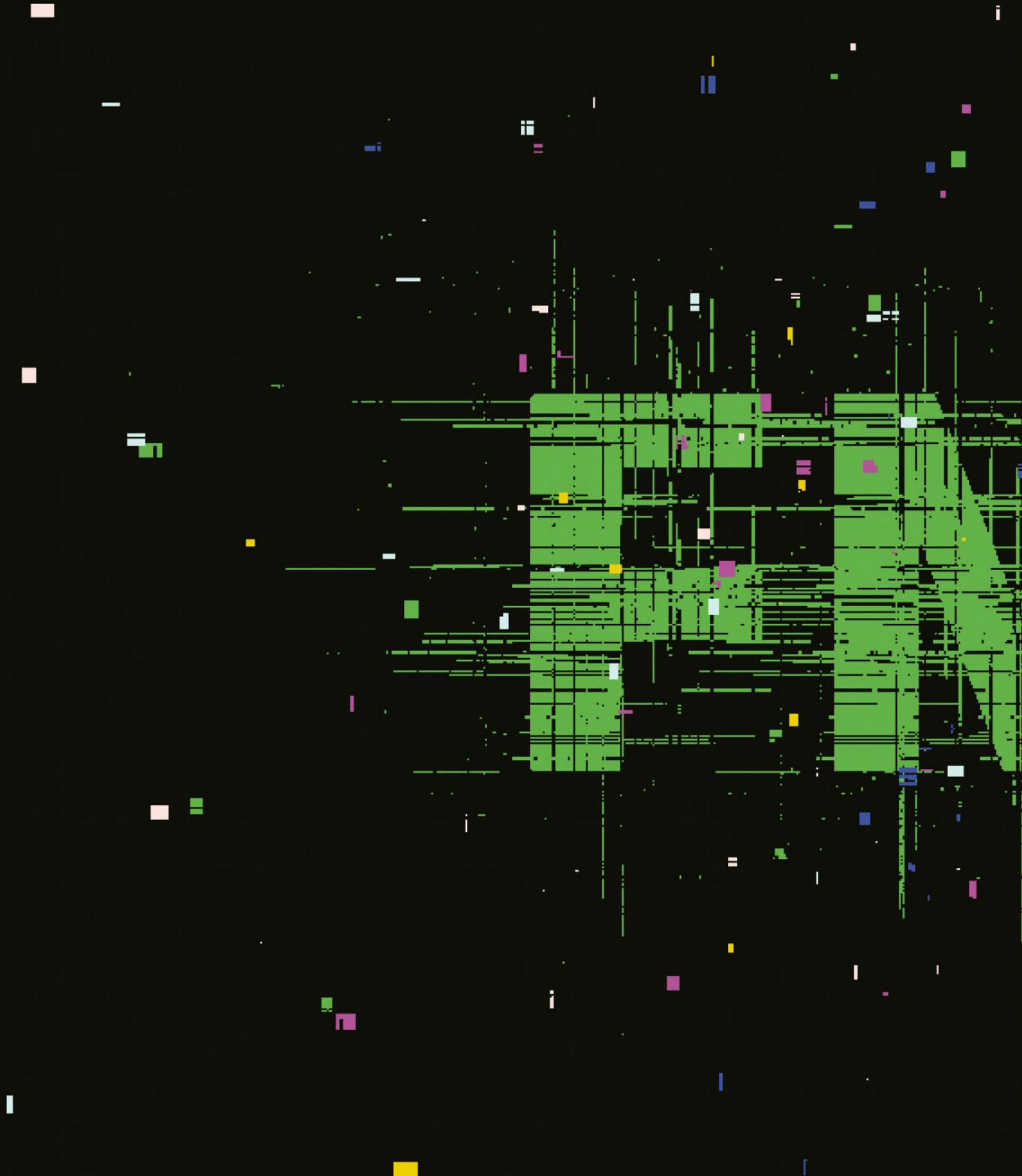
FREE DIGITAL EDITION!

Complete our reader survey at
<http://bit.ly/2BtaLmR>

Each and every issue, we do our best to create a magazine that you love to read, and we're always looking for ways to make it even better. Once again, we'd like your help in improving Web Designer magazine, and all you need to do is answer some very quick questions about yourself - we're really looking forward to hearing from you.

Steven Jenkins
Editor





Millennials Are Screwed

highline.huffingtonpost.com/articles/en/poor-millennials

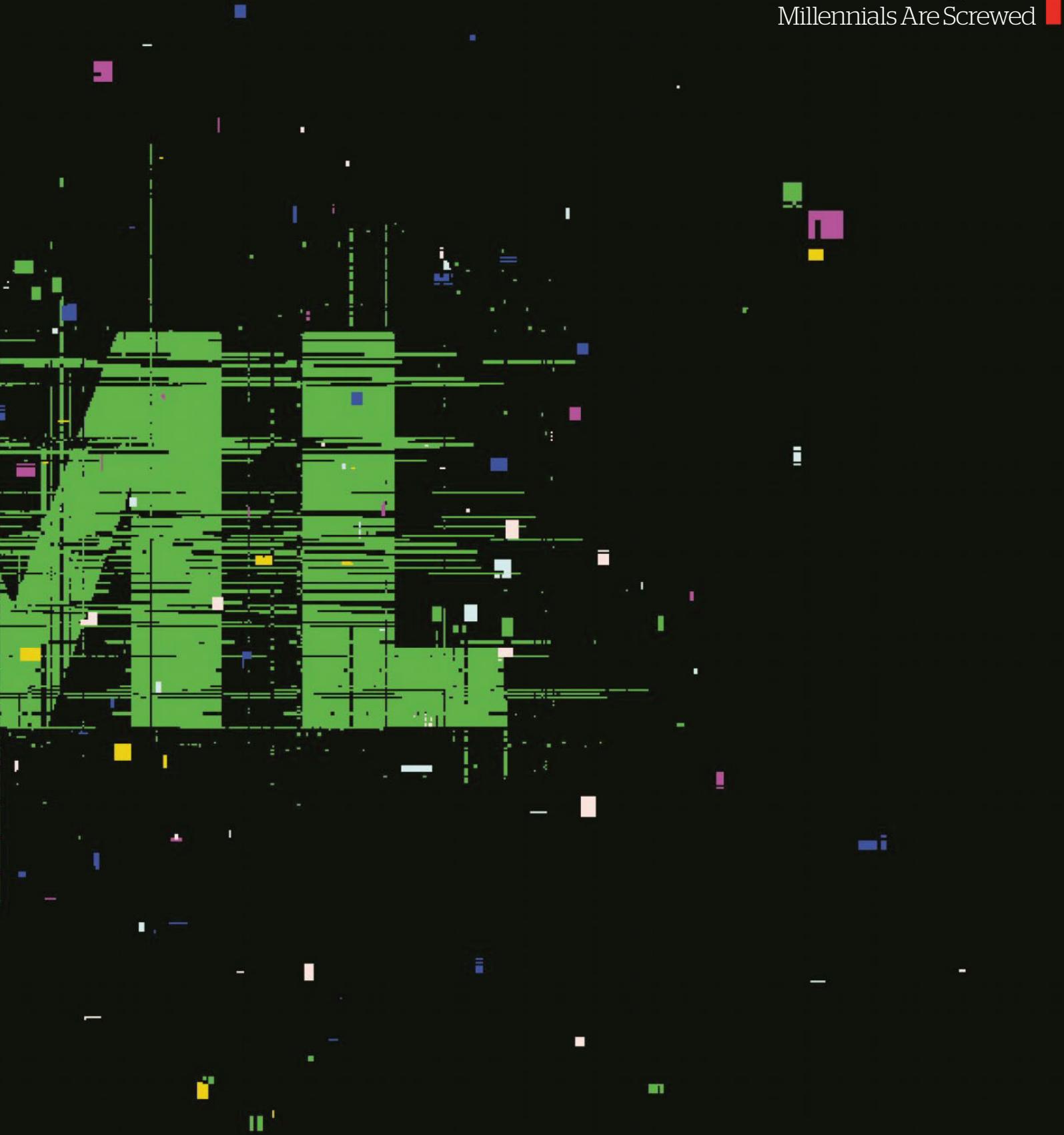
Designer: Gladeye – gladeye.com

Development technologies WordPress, jQuery, Three.js, HTML5

“Michael Hobbes and the Huffington Post present a vibrant scrolling narrative on a scary financial future”

LightBox

Millennials Are Screwed



#000000



#3BE370



#E84CF5

abcABC abcABC
1234567 1234567

Above

The GT Walsheim font, by Noël Leu for Grilli Type, is used in Regular and Bold weights, plus Oblique forms.

abcABC
1234567890

Above

The Simplon BP Mono font, by Emmanuel Rey for b+p swiss typefaces, also styles the left side <div> text boxes.

Understanding structural disadvantage is pretty complicated. You'll need a guide.

LightBox

Millennials Are Screwed

+

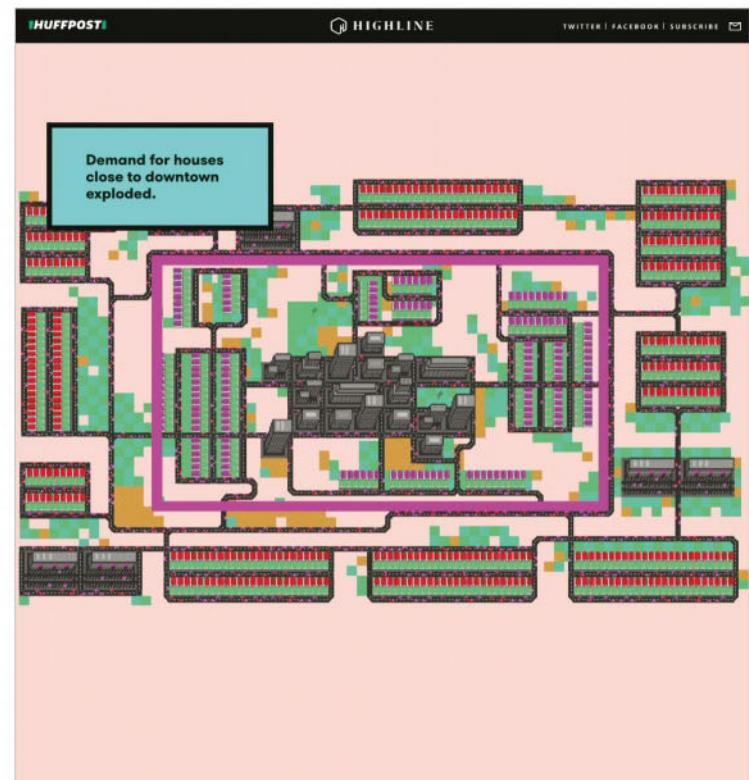
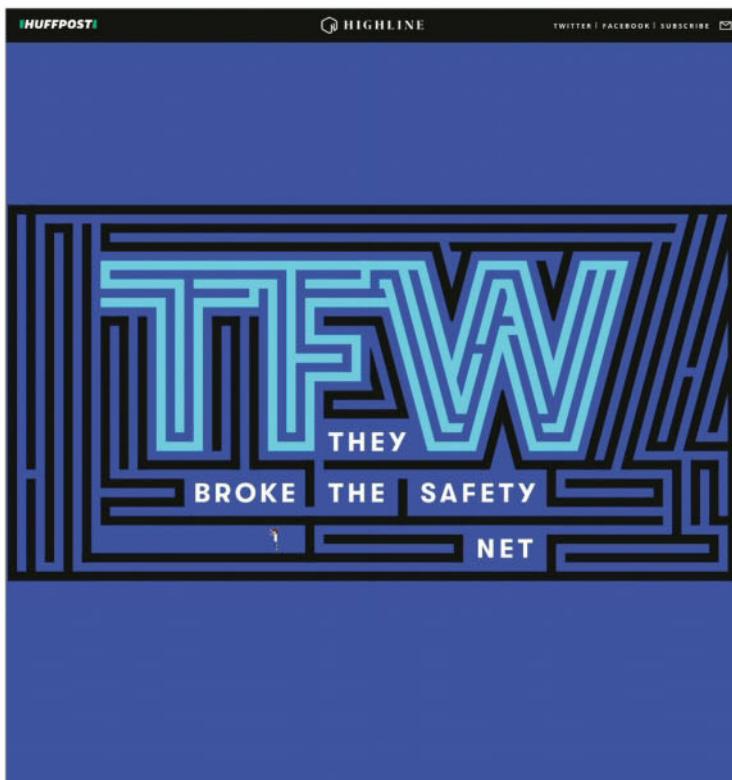
x

x



Above

As the user scrolls, the site animates the 'fall' of a pixelated character called Becky, who represents millennials



Above

Gladeye's design incorporates eye-catching typography from Jason Wong, adding real style to the headings punctuating the story

Above

A myriad of image-based illustrations are also animated during scrolling, using frames and CSS transformations to implement

Create a scroll up panel to reveal more content

Create an interactive content cover that fades to reveal page content in response to user scrolling

1. Initiate HTML

The first step is to define the HTML document. All of the document is contained within the HTML container, which consists of the head and body sections. While the head section is responsible for loading the external CSS and JavaScript resources, the body section will store the content created in step 2.

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

2. Content definition

The HTML content consists of a header and main section. The header section is responsible for the 'cover' content that will be displayed to initially cover the full screen. The following main section will be used to store content that will move into view after the user starts to scroll the page.

```
<header>
  
  <h1>Once Upon a Time...</h1>
</header>
<main>
  <p>In a land far away...</p>
</main>
```

3. Scroll listener

Create a new file called 'code.js'. This JavaScript code must first wait for the page to load, upon which it will activate a page scroll event listener. When the user scrolls the page, a function will be triggered to search for the header section's image, upon which some calculations will be performed to alter its size, positioning and opacity based on the current scroll position.

```
window.addEventListener("load", function()
{
  window.addEventListener("scroll",
    function(){
      var img = document.querySelector("header
img");
      img.style.width = (100+(window.
scrollY/20))+"%";
```

```
img.style.left = (0-(window.
scrollY/50))+"%";
img.style.opacity = 1-(1/(window.
innerHeight/window.scrollY));
});
```

4. Body style

Create a new file called 'styles.css'. The first part of the CSS sets the page to cover the full screen without any border spacing. This is achieved using the width and height attributes, along with margin and padding to eliminate any default spacing set by the browser. A black background and default font are also applied.

```
html,body{
  display: block;
  width: 100%;
  height: 100%;
  margin: 0;
  padding: 0;
  font-family: "Trebuchet MS", Helvetica,
  sans-serif;
  background: #000;
}
```

5. Main container

The main content container is set to cover half of the available screen width. A minimum height of double the screen height is set to guarantee the ability to scroll, regardless of how much content is set. The auto settings in the margin attribute are used to centrally align the container so that the child content automatically appears in the middle of the screen.

```
main{
  width: 50%;
  min-height: 200%;
  font-size: 3em;
  color: #fff;
  margin: 10em auto 0 auto;
```

6. Header container

The header container is set to display with relative positioning so that any child items can be positioned in relation to its location. Its height is set to match the screen height so that it covers the full space available. Hidden overflow is also applied to avoid excess cover content conflicting with the main content.

```
header{
  display: block;
  position: relative;
```

```
height: 100vh;
text-align: center;
overflow: hidden;
}
```

7. Header's heading

The H1 element inside the header is set to have specific size, colour and shadow settings. Other unique settings can also be applied in this definition. The application of the text shadow helps to avoid any conflict in the background image where colours may be difficult for the user to distinguish the text from.

```
header h1{
  font-size: 6em;
  color: #c00;
  margin-top: 0;
  text-shadow: 2px 2px #000
}
***
```

8. Header's children

Settings for all first level children of the header are set to share properties. This approach avoids having to repeat settings for each unique item; keeping the CSS small and manageable. All children will use fixed positioning placed in the top-left corner. They will also have a width matching the header container, along with margining to centrally align them.

```
header > * {
  position: fixed;
  top: 0;
  left: 0;
  margin: 0 auto 0 auto;
  width: 100%;
```





We are Open Wear.

A collective of passionate skiers
and snowboarders.

Open Wear

www.open-wear.com

Designer: Build in Amsterdam - buildinamsterdam.com

Development technologies jQuery, Squarespace, WordPress, WooCommerce

“This catalogue and eShop for the outerwear brand marries a reduced colour palette and readable page text with bold product placement”



Product — The ultimate deep powder bib pants.

The ultimate bib pants. 100% powder protection comes with a kangaroo pocket. Made of recycled PET-bo...



Community — Get to know where your products are being made

In over twenty years of development since its founding in 1995, our partner, Maxport has grown into...



12 +



#5F9DCC



#5A293E



#OD738C



#B6AEAB

**abcABC abcABC
12345678 12345678**

Above

The geometric sans serif font Circular by Laurenz Brunner is employed in Book, Medium and Bold typefaces.

abcABC 1234567890

Above

The Druk Condensed Super font by Berton Hasebe is employed for the span headings running down the page.


[Filter](#)

Open One 3L Shell Jacket

The ultimate powder jacket 2.0. Made of recycled PET-bottles. 45K waterproofness.

CHF439

Above

A highly bold catalogue, the site blends high-impact product photography with action shots against a clean minimal template



**Open One 3L
Shell Pants**

CHF369 Traditional retail CHF589

Colour: ● ● ● ●

Size: S M L XL XXL

[Size guide](#)

[Buy](#)



Zippers – Eight water resistant YKK zippers to keep you dry.



Ethical production – Made in Vietnam under fair working conditions.




Above

Select an item to access a JavaScript-based rotator for examining a 360-degree look at the available garment on hover

Create an animated slant box border effect

Create an effect for images and other types of content components that responds to user interaction



1. HTML initiation

The first step is to create the page document architecture from HTML. This consists of the HTML document definition, which holds the head and body sections. While the head section is used to load the external CSS file, the body section is used to store the HTML created in step 2.

```
<!DOCTYPE html>
<html>
<head>
<title>Hover Frame: Slant</title>
<link rel="stylesheet" type="text/css"
      href="styles.css" />
</head>
<body>
  *** STEP 2 HERE
</body>
</html>
```

2. Visible HTML

The visible content consists of a series of containers using the 'slant' class. You can place any content inside these containers. This example uses an individual image for each example container. The HTML is designed to be kept to a minimum, without any unnecessary elements for formatting or embedded styling.

```
<span class="slant">
  
</span>
<span class="slant">
  
</span>
<span class="slant">
  
</span>
```

3. CSS initiation

With the HTML now complete, create a new file called 'styles.css'. The first definition in this file sets the HTML document and body container to display as a block with no margin or padding. This guarantees that the page will display without any visible border spacing. Optionally, you could add other properties, such as setting their min-height to 100%, meaning that the page is sized to fit the full screen.

```
html, body{
  display: block;
  margin: 0;
  padding: 0;
```

4. Container class

The slant container is identified using the 'slant' class. This element uses relative positioning, along with an inline-block display mode to allow it to flow as a normal text item while having sizing-related properties if required. Overflow is hidden to support the feature created in steps 5 and 6.

```
.slant{
  position: relative;
  display: inline-block;
  overflow: hidden;
```

5. Border slant

The slanted border feature is made from a box element overlaying the content container. This is achieved using the ':before' selector, which is used to define a virtual element inside CSS. This is set to be placed at the bottom of the slant container with a width and height that will cover enough space for the effect to work.

```
.slant::before{
  content: "";
  display: block;
  position: absolute;
  bottom: -3em;
  left: -2em;
  width: 150%;
  height: 6em;
}
```

6. More border slant

The border slant needs to match the page background colour for the effect to work. Additionally, the element needs to be rotated via the transform attribute so that it doesn't appear as a straight line. Finally, a transition with a duration of one second is applied so that any changes to the rotation transform appear animated.

```
.slant::before{
  background: #fff;
  transform: rotate(8deg);
  transition: transform 1s;
}
```

7. Hovered slant

The rotation is required to change when the slant is being hovered. This step sets the rotation angle when this occurs. The changes that occur will appear animated due to the previous step's transition attribute. Changing the -8 to another number will alter the angle that the slant rotates towards.

```
.slant:hover::before{
  transform: rotate(-8deg);
}
***
```

TOP 5 MOST LOVED BRANDS

1. HYPERX
2. MICROSOFT
3. SEPHORA
4. RAWLINGS SPORTS
5. VIZIO

PRIVACY



LIST VIEW

Brand Love Score

www.brandlovescore.com/blackfriday2017

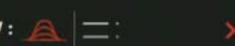
Designer: Drumroll – www.drumroll.com

Development technologies ASP.NET, Respond, jQuery, Typekit

“Leveraging social media data, Drumroll presents a dynamic web visualisation illustrating the most popular brands of Black Friday 2017”



TOP 50 BLACK FRIDAY BRANDS



DRUMROLL.COM



#FF4B33



#151D1E



#9403B4



#D5154A

abcABC
1234567890

abcABC
1234567890

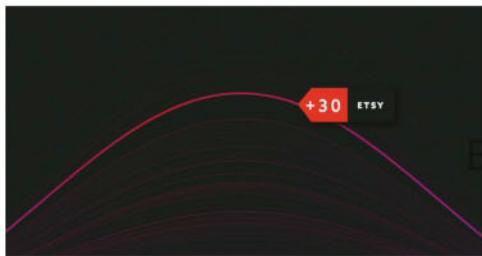
abcABC
1234567890

Above

The Azo Sans font, by Rui Abreu and available through Adobe Typekit, is used exclusively in Light, Regular and Bold

Create an interactive navigation controlled image caption

Employ the power of attributes to tell your story with the help of captions



1. Get started

The first step is to define the document architecture that will be used to present the page content. This consists of a definition of the HTML document, which stores a head and body section. While the head section is used to load the JavaScript and CSS resources, the body section is used to store the content created in step 2.

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

2. Body content

The page content consists of two child containers; an article container that stores a navigation and a figure for storing text corresponding to the navigation options. The navigation contains the interaction options, and each have a unique title attribute. The figure stores the caption text that corresponds to the equivalently positioned option located within the navigation container.

```
<article data-theme>
  <nav id="mainNav">
    <a href="#" title="Page 1">Page 1</a>
    <a href="#" title="Page 2">Page 2</a>
    <a href="#" title="Page 3">Page 3</a>
  </nav>
  <figure>
    <figcaption>Caption 1</figcaption>
    <figcaption>Caption 2</figcaption>
    <figcaption>Caption 3</figcaption>
  </figure>
</article>
```

3. JavaScript listening

Create a new file called 'code.js'. There's a need to set the value of the article's 'data-theme' attribute to match the navigation option being highlighted by the user. JavaScript will wait until the page has loaded, upon which it will find all of the navigation links in order to apply a 'mouseover' event listener. This listener will update the article's 'data-theme' attribute to match the hovered navigation option's title attribute.

```
window.addEventListener("load", function()
{
  var nodes = document.querySelectorAll("[data-theme] nav > *");
  for(var i=0; i<nodes.length; i++){
    nodes[i].addEventListener("mouseover", function(){
      var container = document.querySelector("[data-theme]");
      container.setAttribute("data-theme", this.getAttribute("title"));
    });
  }
});
```

4. Article container

Create a new file called 'styles.css'. The first definition of the CSS sets the size and positioning of the article container. The height is set using 'min-height', allowing it to expand beyond this if it has additional content. Auto is used for the margin's horizontal parameters to allow the browser to automatically position the article in the centre. The associated 'data-theme' attribute sets the article to display as a table.

```
article{
  min-height: 20em;
  width: 50%;
  margin: 0 auto 0 auto;
}
[data-theme]{ display: table; }
```

5. Article theme

Using the 'data-theme' attribute, the child figure container and its captions are referenced for styling. The figure is set to display as a table cell with relative positioning and background settings to display images centrally covering the full space. The captions are positioned out of view by default, with absolute positioning in relation to the parent figure.

```
[data-theme] figure{
  display: table-cell;
  position: relative;
```

```
background-position: center center;
background-size: cover;
}
[data-theme] figcaption{
  position: absolute;
  background: red;
  top: -9999%;
}
```

6. Unique figure captions

The unique display sections for the figure and its captions are defined in this step. When the JavaScript sets the article's 'data-theme' attribute to the navigation title, this is able to trigger unique presentation rules within the CSS. These are used to set the figure's background picture and move the associated caption to the desired position.

```
[data-theme]{ display: table; }
[data-theme="Page 1"] figure { background-image: url(image1.jpg); }
[data-theme="Page 1"] figcaption:nth-child(1){ top: 50%; left: 50%; }
[data-theme="Page 2"] figure{ background-image: url(image2.jpg); }
[data-theme="Page 2"] figcaption:nth-child(2){ top: 25%; left: 75%; }
[data-theme="Page 3"] figure{ background-image: url(image3.jpg); }
[data-theme="Page 3"] figcaption:nth-child(3){ top: 75%; left: 25%; }
```

7. Theme navigation

The navigation container is also set to display as a table cell, with its links set as block items with a border. This results in links being stacked vertically within the navigation container - which is set to cover half of the parent article. Navigation links are set to appear with a darker background colour when hovered.

```
[data-theme] nav{
  display: table-cell;
  width: 50%;
}
[data-theme] nav a{
  display: block;
  color: #000;
  border: 1px solid;
}
[data-theme] nav a:hover{
  background: rgba(0,0,0,.5);
  color: #fff;
```

generate

The conference for web designers

(mt)
mediatemple

NEW YORK | 25-27 APRIL 2018



DONNA LICHAW



WES BOS



ABBY COVERT



DAN MALL



VAL HEAD

and
more!

TICKETS ON SALE NOW

www.generateconf.com | #generateconf

ANGELIC AIRPLANES



WHEN SILICON VALLEY INVESTOR INCULERATE
DEMAND A DISTINCTIVE NEW WEBSITE, THEY
OPTED FOR THE IMAGINATION OF RUSSIAN
AGENCY RED COLLAR TO DELIVER AN IDEA FIT
FOR GIVING 'ANGEL' ITS WINGS

INCULERATE

<https://inculerate.com>

by

RED COLLAR

<http://redcollar.digital>

[@redcollars](#)

It's fair to say the modern internet experience is dominated by a clutch of incredibly successful companies. Each has cleverly found a highly profitable niche within the online world to become clear leaders in areas of search, social media, eCommerce, video and more. These enduring brands, which we need not mention, very often rose out of the dotcom bubble in the late

- a pivotal role in supporting promising new business ventures. Believing that "creation starts with an idea," Inculerate pride themselves in helping exciting startups to grow as quickly as possible, furthering innovation and hopefully in turn making people's life better. "They came to our agency with a request to create an impressive website that would reflect their principles and ideology," begins the guys at commissioned agency Red Collar. "The main basic concept agreed quickly around this time was the paper airplane, deciding between us that the take-off theme probably reflects an investment company such as Inculerate the best." If the colour of said collar seems familiar, it's because we profiled the Russian agency's equally imaginative web work for car paint manufacturer Auto Repair Materials, back in issue 259. Thankfully

"BELIEVING THAT "CREATION STARTS WITH AN IDEA," INCULERATE PRIDE THEMSELVES IN HELPING EXCITING STARTUPS TO GROW AS QUICKLY AS POSSIBLE"

1990s and swerved being burst on the rise to the top. They outlasted the tidal wave of lesser competitors by seeing the future and evolving accordingly to conquer rapidly shifting territories. However, none of them did it without cost, and an equally important driver to success is always money, inspiring those with big chequebooks to take a gamble. The website we're featuring this month relates directly to this, built to market a business that flashes the cash to fund the operations of fledgling firms. An angel investment company based in Silicon Valley, California, Inculerate play

PROJECT STATS

PROJECT DURATION
2 month

PEOPLE INVOLVED
5

PEOPLE

Creative Director
Denis Lomov

Producer
Mikhail Serkov

Designer
Daria Lomova

Junior Designer
Anna Efremova

Front-end Developer
Anton Ustinov

DESIGNING FOR DREAMS

"Every aspect of the website was about showing what the client wanted to communicate out into the world," revealed Red Collar's Creative Director Denis Lomov. "So even every animation we created was done and chosen for a reason. For instance, all the texts appear

on each screen with a movement that is reminiscent of how an idea comes to mind. We wanted to create a special atmosphere on the site so we paid considerable attention to all the animations and sounds. Such as the sound of rustling paper making the action

more immersive while making it more realistic, and the background melody helping to evoke the feeling of importance around the moment when a dream is being launched. We chose gold and a deep blue colour that was intended to be representative of a

boundless space for opportunities. We then created all the content ourselves from training and performing airplane folding gestures, buying blue cardboard, applying golden body paint and capturing more than 1,500 shots for stop-motion video."

INCULERATE

FACEBOOK

CREATION STARTS WITH AN IDEA

though we are assured no amount of that particular product was used *Goldfinger*-style to makeup the star of this latest URL! Intrigued? Well read on to discover how the story of **inculerate**.

"THE MAIN QUESTION WAS HOW TO EMPHASISE THAT THE COMPANY INVESTS IN ANY PERSON INDEPENDENTLY OF GENDER, NATIONALITY, SEXUAL IDENTITY"

com 'unfolded' to launch an angelic website worthy of investment.

TAKING FLIGHT

If you don't already know them, Red Collar are an exciting digital agency from Voronezh, Russia, with a keen sense of the unique and innovative. Recognised by a passionate attitude and

an impressive list of industry awards, the team insist on a principle of helping to make a client's wishes come true. "Creative vision, technology and digital experience are the main principals of our agency, so it's important for us to stay ahead of market strategies, technologies and creativity. Design and front-end development are our strength and so that's why in every project you can see exclusive design and interesting decisions of front-end development." These guys love what they do and the process of the work involved, and this enthusiasm was undoubtedly what drew Inculerate to a Soviet solution. "We always start think about the main idea of the website just after the contract is signed," begins Creative Director Denis Lomov when explaining perceptions of

Inculerate's requirements. "The main question was how to emphasise that the company invests in any person independently of gender, nationality, sexual identity and so on. That was really important along with an insistence they didn't want a classic investment website." What followed was a gathering of Red Collar's managers, designers and front-end developers to establish what that meant and brainstorm a fundamental concept right away. "Lots of ideas came to our mind but after a while we came up with an idea of an airplane to represent the kinds of startup businesses that Inculerate support. The great thing here was that the client absolutely trusted in us and were willing to let us create whatever we wanted."



INTO THE FOLD

So the core idea that underpins the finished site was established early and very much from all that initial creative brainstorming. This work in idea generation was however also informed by some competitor analysis that confirmed certain things that the client wanted to avoid. "We came through dozens of sites looking for good solutions," confirms Lomov. "And after that we realised that all the investment websites are absolutely similar. Given that the client had already said they didn't want to have a classic investment website, we centred around the take-off theme most enthusiastically."

The freedom Red Collar had to pursue this direction was aided by an admirably loose reign from Inculerate, who encouraged the team to go its own way throughout the project. Such an open attitude was made even more refreshing given the long-distance proximity between client and agency - an international relationship that hasn't always been so trusting. "After the brief our Junior Designer made sketches and we showed them to our client,"

- continues Lomov. "It's important to say that the company is from the USA and we're in Russia, but in spite of different state of mind we managed to establish a very good and trusting relationship. They trusted in our professionalism and let us create at our discretion. They liked our idea and so we started to implement sketches into a real website. On every stage of design and front-end development we presented what we did and the client was really excited."

- That front-end work would involve some inventive animation work, with lots of attention paid to creating an atmosphere with the visuals and sound too. Visitors to the finished site today will immediately appreciate the dedication to an idea, the choice of colours and the stop-motion gestures all performed lovingly by the team.

HOVER CRAFTING

- The front-end development of the site was all written in PixiJS with the images and animations etc rendered onto Canvas. This approach consciously ensures that the content is displayed using the fastest and most flexible 2D rendering of WebGL while ensuring

- consistency across devices. "PixiJS allows us to make cross-platform sites that open on almost any device and browser, so the site is 100% optimised for all these types of devices," Lomov confirms. The main interaction with the site of course happens via an interactive

"THE FRONT-END DEVELOPMENT OF THE SITE WAS ALL WRITTEN IN PIXIJS WITH THE IMAGES AND ANIMATIONS ETC RENDERED ONTO CANVAS"

- slider, dragging a point on the screen to symbolise the 'do-it-yourself' effort associated with a startup. This makes the whole site more like an interactive story; in a few steps a paper plane is folded and launched like a business. "We have thought out all the details to make the UI most effective," says Lomov. "When you hover the cursor to the start point it sticks to the element like a magnet because of a 60-pixel invisible hover zone around the starting point. To not confuse the user we created a 'running' light spot on the guideline in order to direct visitor's interaction. The stop-motion shots automatically play thanks to TweenMax (GSAP) technology

ABOVE: No red collars and no trousers for a few members of the team



Red Collar Creative Director Denis Lomov offers an insight into a standout feature that says something broader about why the project was successful.

"The main thing is an idea. The company has high humanistic goals and an innovative approach to startups searching. The site should clearly reflect the company's activity and arouse real emotions."

"THE SERVER AUTOMATICALLY GENERATES, INSTALLS AND RENEWS SSL CERTIFICATES FROM LETSENCRYPT.ORG AND WORKS ON THE HTTP2 PROTOCOL"

and the frames don't just follow each other. We used TweenMax to achieve the most realistic effect and to create our own easing, so at the beginning frames nicely speed up to slow down in the end for better vividness. It's this sense of smoothness that we believe delivers a more realistic perception." As far as the nuts and bolts underneath goes, the important back-end base and backbone of the site is in the platform where it's located. The server automatically generates, installs and renews SSL certificates from letsencrypt.org and works on the http2 protocol, ensuring parallel, fast and secure loading of content. This also has implications for the general adaptiveness achieved, something that has already garnered the site SOTD, WOTD and Mobile Site of the

Week accolades by Awwwards and indeed Google.

PLANE SAILING

When it comes down to talking about the site's launch after just a two-month development time, the team are keen to remember what the project was always about. Gauging any reflective notion of 'success' on that basis then comes from how well it communicates the original message at its heart. "First of all this project is a story about inspiration and searching for the right ideas. It's about people who want to change the life of the whole world for the better, and that's what both investors and startups want. The client promotes the most humanistic ideas and doesn't care about gender, skin colour or confession of a person, but only about the quality of their idea." A great and admirable philosophy always, so what about the idea at the site's heart? How effective do the people behind it think that has proved to be now it's out there live?

BELOW: The hands animation is the work of many frames



"Well the site appears to be a communicator, a place where people with valuable ideas find their Angel investor. The unusual concept of the site, the launching of a paper plane by a visitor, seems to reflect the spirit of the company and the principles of kick-starting a startup. It offers visitors to record a soundless video where they would get out a message visually. If it is clear and impressive, Inculerate reaches out and the cooperation starts. Similarly, all the components across the full interactivity of the site, sounds and colour create an encouraging picture. You want to share the site with friends and never leave it. The project already has more than 20k page views in the first week, so it's safe to say we are proud of the reception."

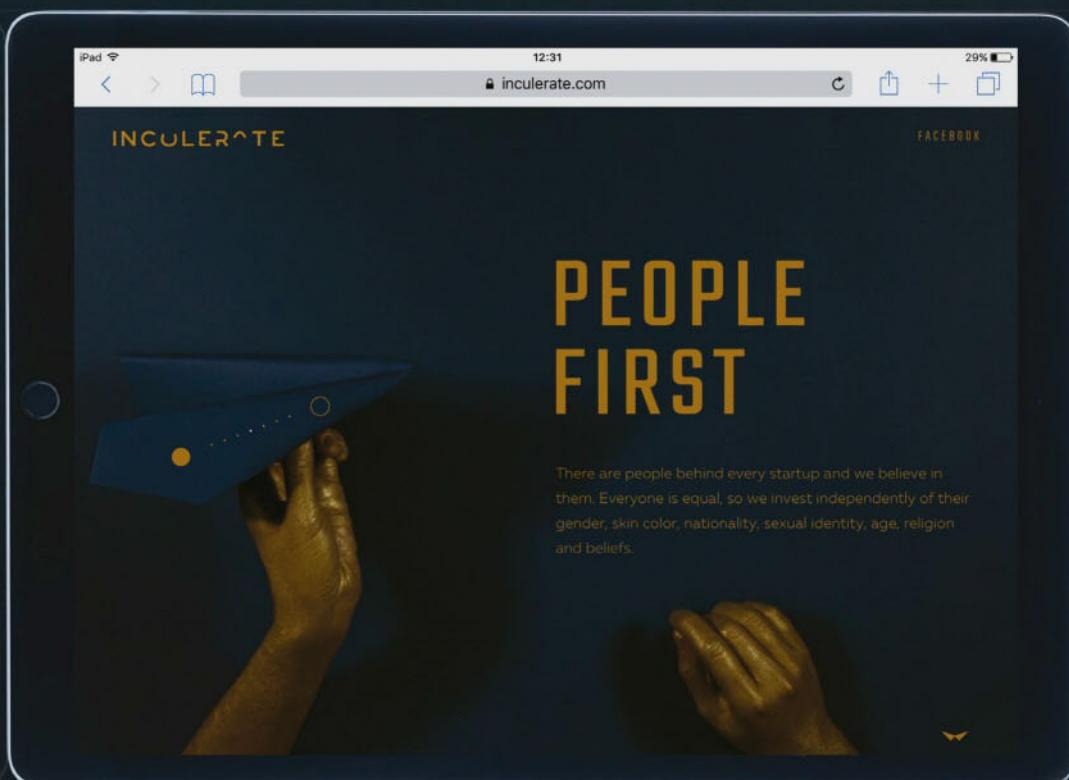
Thankfully the client echoes this understandable pride, after showing the kind of faith in Red Collar it shows it typically reserves for its investments. "We think that such a successful result is another confirmation of the importance of trust," concludes CEO of Inculerate Inc, Umit Khairollayev. "When you go to real professionals you should remember that they know what they're doing and there's no need to doubt. Only then you will have a really cool project."

PAPER TRAILS

As we all know, the period after a site's launch rarely puts a full stop on the project's story. Usually somebody, whether it's up to the client or the creator to undertake, will be concerned with marketing the final result and getting it out there to be admired. In this instance, Red Collar did some of that legwork themselves by submitting the URL to leading online awards programmes to useful effect. "In two days after the site was launched, as soon as we applied for awards, the client got a partnership request," enthuses Creative Director Denis Lomov. "A cool Brooklyn-based Art Director fell in love with the site after seeing it at Awwwards, and decided to collaborate. That's what the site is, a catchy idea and a quality implementation multiplied by the recognition of one of the web's best award platforms." So what about any further work, given the resulting

recognition and success, is any kind of subsequent aftercare on the cards? "The Inculerate website is self-sufficient, so it doesn't have personal accounts and things of that sort. It is a promotional presence that doesn't really require any on-going support. In fact, we always think about site's

sustainability, the relevance of design and technological solutions, and that's why the project's going to be relevant and trending for years. But we continue to work with the client - in fact we are already creating something innovative so rest assured that a new digital bomb is coming!"



ABOVE: Users drag one point to the other point to bring the animation to life



Subscribe and save 20%*

Every issue, delivered straight to your door



Never miss an issue

13 issues a year, and you'll be sure to get every single one



Delivered to your home

Free delivery of every issue, direct to your doorstep



Get the biggest savings

Get your favourite magazine for less by ordering direct

What our readers are saying about us...

"I've been a reader of Web Designer Magazine since the early days."
@sixrevisions via Twitter

"Love the latest magazine that I purchased today"
@navigation_web via Twitter

"My favourite magazine!!!"
@eduardomurillo via Twitter

Pick the subscription that's right for you



Subscribe and save 20%

- ✓ Automatic renewal – never miss an issue
- ✓ Pay by Direct Debit

Recurring payment of £34 every six months,
saving 20% on the retail price



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

Originator's reference
7 6 8 1 9 5

Name of bank

Address of bank

Account Name

Postcode

Sort Code

Account no

Please pay Future Publishing Ltd Direct Debits from the account detailed in this instruction subject to the safeguards assured by the Direct Debit guarantee. I understand that this instruction may remain with Future Publishing Ltd and, if so, details will be passed on electronically to my Bank/Building Society. Banks & Building Societies may not accept Direct Debit instructions for some types of account.

Signature

Date



One year subscription

- ✓ Great offers, available world-wide
- ✓ One payment, by card or cheque

A simple one-off payment ensures you never miss an issue for one full year. That's 13 issues, direct to your doorstep

UK £67.60 (saving 20% on the retail price)

Europe €81.55 USA \$103.20

Rest of the world \$103.20

Pay by card or cheque

Pay by Credit or Debit card



Visa



Mastercard



Amex

Card number

Expiry date

Pay by Cheque

I enclose a cheque for £

Made payable to
Future Publishing LTD

Signature

Date

Your information

Name

Address

Telephone number

Mobile number

Email address

Postcode

Please tick if you want to receive any communications
from Future and its group companies containing news,
special offers and product information.

Please post this form to
Future Publishing Ltd, 3 Queensbridge, The Lakes, Northampton, NN4
7BF, United Kingdom

Order securely online myfavouritemagazines.co.uk/WEDPQ17



Speak to one of our friendly
customer service team
Call **0344 848 2852**

These offers will expire on
Wednesday 28 March 2018

FORM

who FFunction

what Data visualization,
data-rich design, information
design, user interface,
infographics, UX, data portals,
and interactive

where 400 Atlantic #205,
Montreal (QC) H2V 1A5, Canada

web ffctn.com

Data is beautiful in the hands of FFunction. A dataviz specialist studio in an information-driven world, FFunction not only makes sense of the information space, but defines its parameters to communicate often complex ideas to an audience hungry for new experiences, knowledge and insights

AND

FFUNC



t was 2005, and FFunction's founder, Sébastien Pierre, was feeling disillusioned with France. He had a Masters in software engineering and a bachelor degree in design and wanted to find a way to bring the two fields together in his career.

"Where I come from, cross-disciplinary innovation is not that common," explains Sébastien. "France has a culture of excellence, but thinking out of the box and starting something new is unusual, almost to the point of being frowned upon. There, you go to the right school, get a job, pay your dues, move up the ladder, stay in your own lane. So, I came to Canada."

After working in Montreal startups for a couple of years, he decided to start his own company. But he needed a graphic designer to work with, so he went to the graduate show of UQAM (Université du Québec à Montréal) to see if there were any contenders. "I saw one display that was very strong; the designer had an innate understanding of typography and form," Sébastien continued. "Other people seemed to agree – there were no business cards left so I had to Google her name: Audrée Lapierre."

Audrée also remembers this time vividly: "Funnily enough, when I graduated I had every intention of being a print designer. I was interested in aesthetics and

decorative arts, abstract images, punchy colours and fine typography. I didn't know where I wanted to work, but I knew I didn't want to work in advertising. I wanted to design to inform people, not to sell things. One day I got an email from a Frenchman who said he'd seen my work and wanted to talk. It was Sébastien."

After several discussions, Sébastien and Audrée decided to set up shop, and rented a couple of desks in the corner of another company's office in downtown Montreal. Audrée explains: "This was before the Infographic Boom of 2010 and at that stage most people didn't even know what data visualisation was. At that time, the mix of visual communication, data journalism, information, UI and design were a very new way to get information across, and we spent a lot of time doing self-generated infographics and explaining to prospective clients why they would even want to publish their data in the first place."

A short time later, the studio moved to an industrial-zone/hipster-enclave called Mile-Ex, which has been FFunction's home ever since. The studio itself is in a century-old brick building that has been converted into large open plan loft spaces.

As a data visualisation studio, Sébastien and Audrée looked for a name for their new business that would encapsulate the kind of work they wanted to do. Sébastien explains how they named their business: "Being in a bilingual environment like Montreal adds an extra layer of complexity to choosing a name. I started with the name 'Donnéelicieuse' which translated to 'Datalicious' in English. But datalicious.com

Key Clients

UNESCO Institute
for Statistics

Google

Deloitte University
Press

Heifer International

Children's
Investment Fund
Foundation (CIFF)



The name FFunction comes from the combination of form and function, which to me is what data visualisation is all about

was taken by another company working in a parallel industry, and anyway, having different names in French and English was not ideal.

"Donnéelicieuse is still our legal name but we started trading as FFunction shortly thereafter. The name FFunction comes from the combination of form and function, which to me is what data visualisation is all about. It was also a wink to fffound.com, which at that time was a strictly invite-only magnet for graphic designers all over the world. The domain name ffunction.com wasn't available.

FORM AND FFUNCTION



but we managed to get www.ffctn.com, which gets rid of those pesky superfluous vowels. Who needs them anyway?"

FFunction was doing information design before it was really a clearly understood component of communication. As such, locating clients was difficult at first. "In the early days of the studio we created a bunch of self-generated projects for our portfolio, and presented at every conference that would have us," says Audrée. "Once we had won a couple of awards and landed our first big clients – the Gates Foundation, National Geographic, Heifer International and Edelman all came to us quite early in FFunction's journey – it was easier to maintain our momentum and attract new clients. I suppose the work began to speak for itself, and we carved out a serious niche in creating interactive data visualisations for UN agencies, major non-profits and corporations."

"Because of the nature of our clients' work and the layers of compliance that

Above Some of the FFunction team (L-R): *Wim Bruyninckx (Art Director), Chloé-Ève Levasseur (Graphic Designer & UX Specialist), Rebecca Galloway (Marketing Director), Audrée Lapierre (Creative Director) with Byron the dog, Sébastien Pierre (Director, Technical Lead), Jackie Smoak (Project Manager), Sam Dupras (Web Developer)*

they need to adhere to, we tend to respond to RFPs (Request for Proposal) as our main business development method. I know a lot of studios dislike RFPs, but they have their advantages. One major plus is this: if the client has issued an RFP it means they have data, findings and a budget, and they're ready to work. We are, however, very much against spec work, and have refused and lobbied against it on several occasions within the context of a procurement process.

"Occasionally we do cold-call organisations that we would like to work with, but in general our approach is more about communicating FFunction's work effectively than creating a sales funnel. We don't do the conference circuit so much these days, but we're active on social media, and both Sébastien and I occasionally write opinion pieces for design publications, like DesignEdge, Applied Arts and O'Reilly. It helps to stay relevant, I think."

AGENCY BREAKDOWN

- 1 Founder/CEO/Technical Lead
- 1 Creative Director
- 1 Project Manager
- 1 Marketing Director
- 1 Art Director
- 1 UX Designer
- 3 Developers
- 1 Dog (100 per cent certified Good Boy)

FREELANCE:

- 1 Data Journalist
- 1 Studio Administrator
- 2 Graphic Designers
- 1 Translator



"I think that good information design demands a certain level of customisation, which is why the notion of custom tools is very strong on the dev side of FFunction. Internally, we mostly use custom languages that transpile to JavaScript/HTML/CSS and a custom set of libraries for component-based UI and animated, interactive data visualisation"

SÉBASTIEN PIERRE
Co-founder and director



MAKING THE CONNECTION

one.org/making-the-connection/index-en.html

Right now, more than half the world is not connected to the internet, and almost 75 per cent of Africa is offline. Why does this matter? For people living in the poorest parts of the world, access to the internet isn't a luxury – it's life-changing.

ONE, a campaigning and advocacy organisation, is currently rolling out several programs to address this digital gap. It asked FFunction to create an interactive data visualisation that would highlight some key findings and contextualise them in an engaging, shareable way.

As part of ONE's larger Poverty is Sexist campaign that calls on communities, governments and private sectors around the world to help impoverished women within their countries, this site illustrates how access to the internet and digital

media can empower people in developing countries, especially women.

Displayed on a single scrolling page, the site presents two main visualisations: one shows the current rate of internet usage in African countries, and another further breaks down the first viz by gender to show the disparity between women and men who use the internet in Africa and around the world. A form at the bottom of the page directs visitors to sign the Poverty is Sexist petition.

FFunction's design team decided the best way to contextualise the data would be to contrast how the internet improves the quality of life in the wealthiest parts of the world, compared with the most poverty-stricken.

For example, every month, over 560 billion text

messages are sent worldwide. Of course, these can be useful for chatting with friends or receiving appointment reminders. But in sub-Saharan Africa, text messages can enable women who live far from a doctor to receive advice on how to have a safe and healthy pregnancy.

Poverty is sexist – in so many ways, girls and women are hit harder and have fewer opportunities, and access to the internet is no different. Women in the poorest countries are almost a third less likely to have access to the Internet than men, and the gap is increasing.

Visualising this data and putting some context around it helps engaged citizens, journalists and policymakers to understand the problem in both broad strokes and granular detail, and, hopefully, take action.



Above Digging deeper into the data, reveals personal stories that place a human face on this data-based project

The business case for working with a client can have many components. As FFunction doesn't generally do spec work, but how does it decide which clients to work with? Audrée explains the studio's approach: "We have turned down projects because the timeline is too tight, or doesn't fit with the other projects already in our pipeline. We're a boutique agency and intend to remain that way. The upside of working at this scale is that one of the directors is always personally involved in any given project. Crucially, we stay connected to the workflow. The downside is that project planning can be kind of like a game of Tetris because our human resources are limited.

"Historically we have also refused projects where our added value is minimal, like branding or website design or PowerPoint design. There are so many extremely competent branding agencies out there that should win those contracts instead. Why choose dataviz specialists for that?"

We're a boutique agency and intend to remain that way. The upside of working at this scale is that one of the directors is always personally involved in any given project

Audrée concludes: "And this is somewhat delicate, but we've also turned down projects from clients whose business practices we disagree with in some fundamental way. Last year, for example, we were approached to do a project by a cigarette company. The budget was good, but the idea of having a cigarette manufacturer on our client roster alongside the Canadian Cancer Society? It doesn't look right, and it certainly doesn't feel right."

Often an agency or studio will become known for a certain type of work. For FFunction, its approach means its clients are looking to use their data to communicate their messages. Sébastien outlines one project that spoke to FFunction's design sensibilities: "One of my favourite projects is still HP Earth Insights. It applied a Hewlett-Packard analytics platform to research provided by Conservation International, an NGO dedicated to the protection of the planet and its resources. FFunction's role was to

package data that charted fluctuating biodiversity in key national parks near the equator. HP Earth Insights contains everything I like about dataviz. There's an interesting scientific dataset, a data-driven narrative, free exploration that lets users make their own findings and gorgeous photos overlaid with simple, but rich charts."

With specialist design needs and client data that can be presented in many forms, the tools that FFunction uses are varied. "We tend to be early adopters of new tools and plugins," says Audrée. "There are new ones coming out all the time, and we like to give them a try. We draw user flows in OmniGraffle. Also, we use the Adobe suite a lot (naturally), and Sketch for user interfaces. We prototype using Invision and Craft, and the designers send their files to the devs via Zeplin. We also sketch on old-fashioned paper a lot."

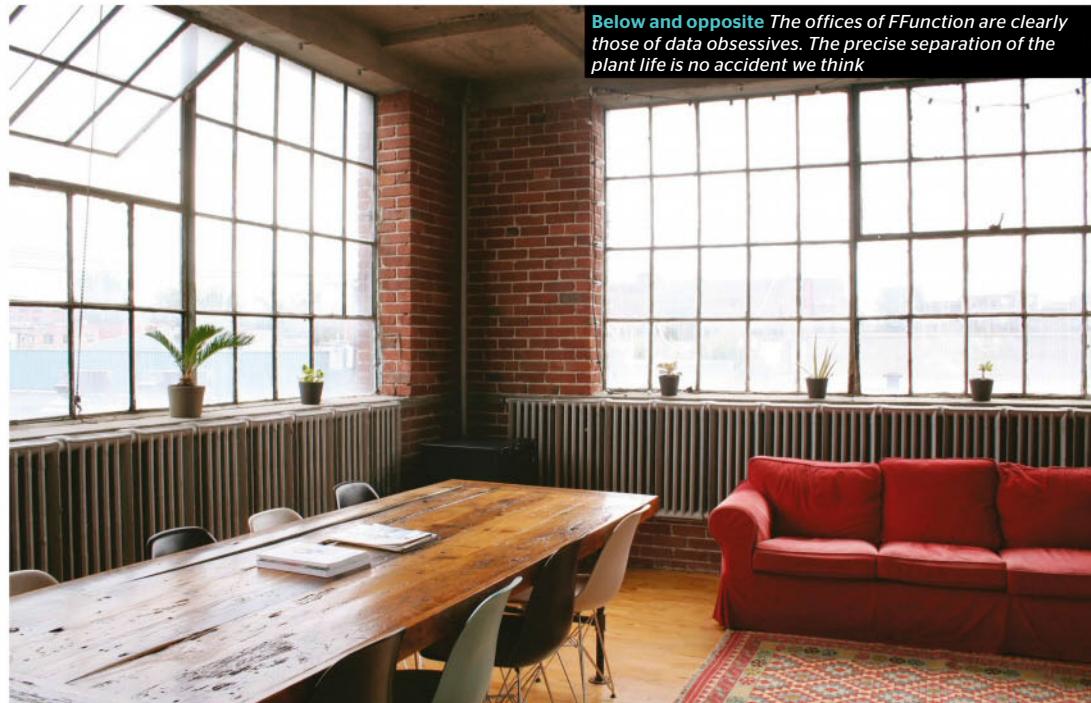
Sébastien also comments: "I think that good information design demands a certain level of customisation, which is

FORM AND FUNCTION



"We tend to be early adopters of new tools and plugins. There are new ones coming out all the time, and we like to give them a try. We draw user flows in OmniGraffle. Also, we use the Adobe suite a lot (naturally), and Sketch for user interfaces. We prototype using Invision and Craft, and the designers send their files to the devs via Zeplin. We also sketch on old-fashioned paper a lot"

AUDRÉE LAPIERE
Co-founder and creative director



Below and opposite The offices of FFunction are clearly those of data obsessives. The precise separation of the plant life is no accident we think

MAIZE SEED SYSTEM MAP

Internal tool for organisation

The Maize Seed System is an interactive map of the agricultural and trade network in sub-Saharan Africa.

This data-driven application is used for sharing knowledge and identifying connections among system actors – such as seed suppliers, farmers, importers and policymakers – in order to better understand how to improve investments, programs, and interventions within the maize seed system as a whole.

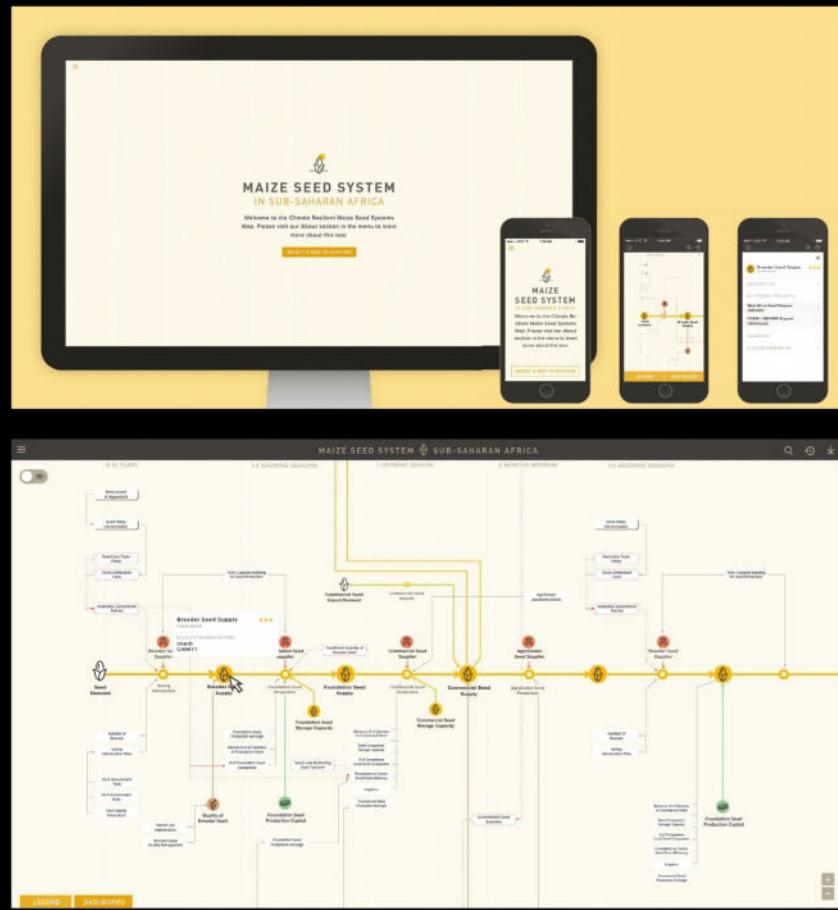
The need for a stronger and more impervious maize network is more pressing each year, as the effects of climate change become more evident worldwide.

Across much of sub-Saharan Africa where maize is the most important staple crop, production is primarily rain-fed, which makes farmers vulnerable to food insecurity.

Approximately 40 per cent of Africa's maize crop suffers a loss in yield due to drought conditions, endangering the livelihoods of millions of smallholder farmers and several hundred million consumers.

DAI, an International development company, is working with other agencies and organisations including USAID to establish a new climate-resilient maize initiative to help solve these problems. Mapping out the maize seed system as it currently stands is an important first step; the next will be using the information generated to identify the right solutions, and provide them.

Right FFunction created a detailed map of the current maize seed system as a vital first step to improving supply reliability of maize



TIMELINE

2008

Sébastien and Audrée meet over a beer. Shortly after, FFunction is born.

Employees 2

2010

The Worldwide Infographic Boom. Edelman becomes a regular FFunction client, and FFunction is busy creating infographics and dashboards for its Fortune 500 clients.

Employees 3

2012

First major interactive dataviz contract, for National Geographic.

Employees 4

2014

Recipient of the AIGA Justified, plus Top Studio at the Kantar Information is Beautiful Awards.

Employees 5

2015

Demand is ramping up for FFunction's work. The studio lands major application contracts with Google, DAI/USAID. FFunction becomes a certified B Corp.

Employees 7

2017

FFunction decides to develop its first product, a project planning software called Min.

Employees 9

“ Since we started using Zeplin, the designers spend much less time reviewing the finished projects, as they’re pretty much pixel-perfect from the get-go ”

"In our development kit, we designed primitives that scale well across this spectrum, from simple animated visualisations to complex data-driven components that interact with each other. We managed to strike this balance partly because we've used and learned from libraries such as D3, React and others, but also because our work has become more complex as the studio has grown."

As more information is now consumed on mobile devices, how does this impact FFunction's approach to each project?

"I've said this before: mobile dataviz is not just a desktop dataviz in miniature – it's a different thing entirely," said Audrée. "I've tried lots of different tactics to create dataviz for mobile. Smaller devices force you to develop succinct data visualisations that get your points across cleanly. My two main strategies are to reduce the amount of information and use small multiples.

"It's the 'smallify or simplify' approach. Small multiples (AKA trellis charts, lattice charts, grid charts, panel charts) are incredibly handy if you want to make dataviz that works equally well on mobile or desktop. It's just a series of similar graphs or charts that use the same scale and axes, which allows for easy comparison. In the desktop version, they can sit next to each other, and in the mobile version you can stack them up. When he was still with NPR, Brian Boyer championed the notion of 'small multiples on desktop, GIFs on your phone', which I think is also a super good solution."

As the environment that FFunction works within is constantly shifting, what technologies are attracting its attention at the moment? Sébastien comments: "There's been a boom in tools that facilitate the design process, from the creation (Sketch, Figma, Webflow) to the collaboration and delivery (InVision, Zeplin). This all helps having a smoother flow of work while also increasing quality and efficiency.

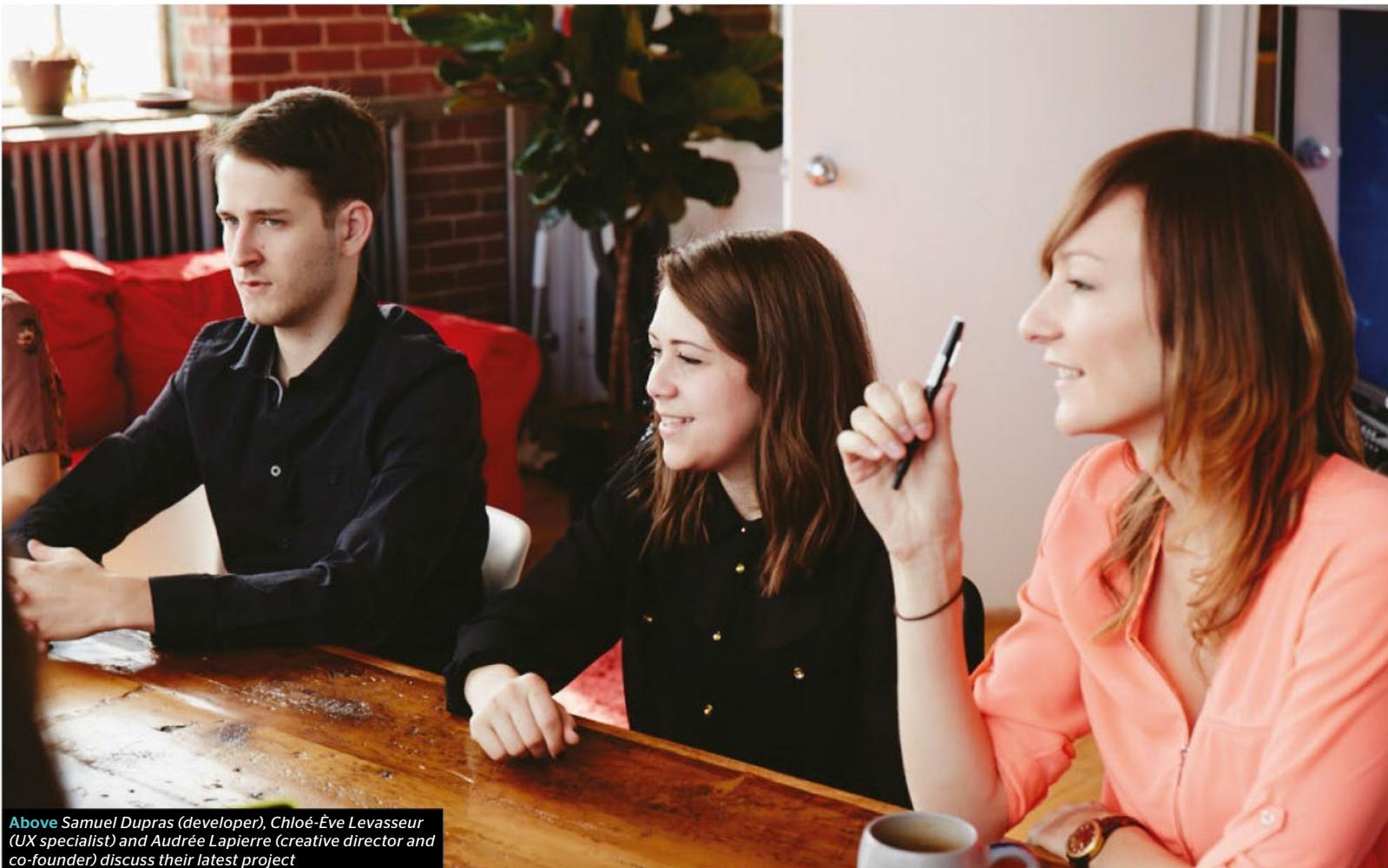
"For instance, since we started using Zeplin, the designers spend much less time reviewing the finished projects, as they're pretty much pixel-perfect from the get-go. On the other hand, there is a plethora of new data-visualisation tools, but, for me, none of them strikes the right balance of creative freedom and sufficiently powerful data manipulation features. It's really difficult, because you don't want something overly simplified like a data-driven Illustrator, but you don't want something that only Computer Science majors would be able to use. I'm looking ▶

why the notion of custom tools is very strong on the dev side of FFunction. Internally, we mostly use custom languages that transpile to JavaScript/HTML/CSS and a custom set of libraries for component-based UI and animated, interactive data visualisation. Dataviz is a demanding field that pushes web browsers to their limits. If you want to innovate, chances are there's no library out there that covers your needs – if you limit yourself to what's already available, you'll need to compromise on your ideas and, at the end of the day, limit your technical abilities.

"We're critical engineers; we believe in the value of a deep understanding of how technology works and the power of technology ownership. That is why we spend the time to create custom tools that give us the independence we need to preserve our creative freedom."

How technology is developing and how this can be harnessed for creative





Above Samuel Dupras (developer), Chloé-Ève Levasseur (UX specialist) and Audrée Lapierre (creative director and co-founder) discuss their latest project

forward to using D3 Express and see if it fulfils its promise of a more interactive and creative way of designing (and coding) data visualisation."

With a specific approach to each project, how has FFunction approached its recruitment? Audrée explains: "Most FFunction's employees come from multidisciplinary backgrounds. We have a developer with a background in education and design. A marketing director with an art history degree. An art director with a Masters in engineering. Our project manager has a chemistry degree (and, incidentally, is also a certified yoga teacher).

"We look for smart people with a rich knowledge base who can look at a problem from several angles and find the most creative, functional solution. We also want to maintain our commitment to the advancement of women from coding to leadership, so we make sure the composition of the team is at least 50 per cent women. We also hire for team culture fit; we want people who are interested in

staying current in terms of their outlook and skill set. And we want people that we're happy to eat lunch with every day tool! It's a small, close-knit team, and there's no room at FFunction for big egos or jerks.

"In terms of design positions, we never hire pure information designers. We hire graphic designers with a passion for information design. This is an important distinction because employees need to be able to work on a wide spectrum of projects and we always need the 'form' as well as the 'function'. Honestly a lot of people who specialise in information design simply don't have strong aesthetic skills, or don't understand typography."

And Audrée advises: "My advice to designers looking for work is this: design a super tight CV, and publish new work regularly. Create interesting side projects that show off your skills and interest areas."

The future will increasingly become even more data driven. For FFunction this means a landscape it can shape. "We're actually in a very exciting phase right

now," Audrée concludes. "FFunction is launching its first product; a project planning tool called Min. We're developing this software because, honestly, we've tried just about every project management tool on the market, and none of them were that great at servicing the needs of an agency.

"Min will allow the user to quickly make a project timeline and track budgets, projects and resources. But I think the clincher is this: Min auto-calculates the best delivery dates based on your task dependencies and current team availability. I always wished that we had that capability at our fingertips when planning client projects, it would have been so useful.

"The response so far to Min has been extremely enthusiastic; we've signed up more than 600 people for early access to the beta without doing any advertising. That tells us that other people are feeling just as frustrated with the current tools on the market as we are and there is demand for this combination of features.

We also want to maintain our commitment to the advancement of women from coding to leadership, so we make sure the composition of the team is at least 50 per cent women



AIGA Justified
2014

Winner 2014



GRAFIKA 2015

100 Best
Projects 2015

FORM AND FFUNCTION

FFUNCTION

ffctn.com

FOUNDER
Sébastien Pierre and Audrée Lapierre

YEAR FOUNDED
2008

CURRENT EMPLOYEES
9 (in house)

LOCATION
Montreal, Canada

SERVICES

Interactive data visualisation

Data-driven applications and portals

Infographics

Data-rich reports

Motion design

HP EARTH INSIGHTS

ffctn.com/a/earthinsights

Earth Insights is an innovative collaboration between HP and Conservation International that applies data-processing technology to monitor biodiversity loss across the world's tropical forests. Using millions of animal photos captured by camera traps, scientists are able to track how land use, human activity and climate change affect species.

FFunction's mission was to design an interactive touch-screen presentation introducing Earth Insights at HP Discover Barcelona, an industry conference which was attended by approximately 10,000 customers, partners and IT thought leaders.

Data-rich visualisations were overlaid on lush forest photography to create an intriguing nature-meets-technology ambience. The experience was structured to guide the user from a big picture overview to granular data insights through three separate

sections that each expressed a different aspect of the data.

Using slideshow, timeline, animated map and data visualisation elements, the finished presentation told stories about threatened animals around the world, the conservationists who are working to protect them and the biodiversity trends they are discovering with the help of data-analysis solutions.

It also integrated statistics and animal photos from fieldwork sites – all into a seamless, intuitive layout that captivated visitors to the exhibit and won over the jury of the AIGA Justified Design Competition in 2014.

Here's what AIGA juror Christopher Simmons had to say: "This is an awe-inspiring, seductive and immersive display of quantitative information that expertly treads the line between gratuitous technological flash and accountable and decodable data. It's a triumph of interaction and experience design."



Right
Connecting the masses of data to the real-world impact deforestation is having on wildlife is brought to life with engaging images and dataviz design



"FFunction will still remain FFunction; we're not planning on phasing out any our service offerings or pivoting into a product company. Interactive dataviz is what we're most well-known for and there is still a high demand for it. But there is another area of our work that has been growing almost exponentially over the last four years: UX for large-scale data-driven applications. Going forward, I can see that becoming a very core part of what FFunction has to offer."

The need to understand the increasing quantities of data being generated by businesses has always been important. Data has now become a highly valuable currency. Studios like FFunction help businesses and organisations make sense of the shifting sea of information to clearly communicate their messages. And these messages can be designed to not only deliver their insights, but also illustrate how beautiful that information can be.

As information denizens, the creatives at FFunction are on the cutting-edge of digital design.

GET PRO DESIGN SKILLS

Typography and colour are the minimum components you would expect to use to form a design and create content that has a distinct look and feel. The wide range of typography and colour available to us enables any creative to design with a personality in mind for a brand or

project. The way we compose our designs with textual matter hasn't evolved a great deal but the resources and trends we use and follow do.

The perfect combination of colour and typography comes from understanding the visual and tonal direction of a brand or message.

FONTS & TYPOGRAPHY

Select the right font to get on the path to great typography



PICKING THE RIGHT FONT

There are a few things to consider when choosing a font and it has a great deal to do with personality. Before starting a project, it's good to think about values and words that describe the feeling of what you're creating. They can be anything! For example; is the design going to be friendly, intelligent or confident? By limiting these values, say to three or five, we can start to get a sense of a direction that our chosen font needs to relate to. So a friendly font might be one that's nice, round and really legible. Confident might be angular, perhaps full caps?

This might be sucking eggs but running this type of exercise can help you focus your decision and so choose the right font.

WHAT DOES A FONT SAY?

So we briefly mentioned visual and tonal direction earlier. Visual is very much the font and how it looks, whilst tonal is the arrangement of words to form a message – both of which need to harmonise and be sympathetic to each other. Use the values you decided earlier to understand what a font truly says and how a message looks when you compare it to what is written. The right font can amplify words in this way so the visual characteristics of a font are important to make the relationship work. Some of the characteristics to look out for are weights, roundness, length and how they flow from letter to letter. You might be torn between Serif, San-Serif, Script or even hand-drawn styles. Each personality of a font will help evoke a feeling or message.



FONTS TO TYPOGRAPHY

A font or typeset becomes typography when the written matter becomes a visual component. You have written a message in your chosen font and it looks and reads just the way you want it to. Typography will also be a combination or composition of more words or phrases. This could include more than one font whereby you have key headline and body text fonts. It's more common to use characterful fonts for headlines and keep cleaner fonts for body text.

The mix between styles for typography is fine but there's always a limit, so never overcook all of your favourite fonts. Some work well together and a healthy contrast can work well too. For example you may have a nice large Serif font headline with a clean Sans-Serif font for body text. Experiment with scale to see what works on your canvas.

IS YOUR FONT WEB SAFE?

It's always worth making sure that your chosen font is web safe and can be rendered perfectly on a browser. Most fonts are these days, using a good font library or a web-safe font file (OTF or WOFF). This is all good but

another important factor with web-safe fonts lies in their performance.

Using a font online can be done in a number of ways. You can upload your font to a server and link to that font in your CSS. This is perfectly fine but always be cautious of font licensing and whether a heavy font file loads quickly. Using a font library such as Google Fonts or Adobe Typekit ensures that everything is taken care of and you can expect better performance. A font hosted with these guys are likely to be faster in most cases. It's good to always make sure that you're only loading the character set you need too. There's no point using all the available weights if you don't need them. The good news is that there's no need to use Arial or Verdana any more!



TOP 4 TOOLS

A must-have collection of resources to enhance your designs

GRIDLOVER

gridlover.net

Grid Lover is awesome at gauging font sizes and spacing on a web page when you're considering how big your H1 and H2 titles need to be. This free tool enables you to dynamically set scaling rules that apply to your headers and tinker with the CSS. It's also a great tool for when you need to hand over designs to a developer since the output CSS and rules are really useful for getting consistency between design and code.

CANVA FONT COMBINATIONS

bit.ly/2rfTqfU

Canva is a useful tool if you want to simply see font pairings and get some inspiration. Select your starter font and see where it takes you, providing you with a list of useful ideas of great font pairings. A great advantage of this tool is seeing the pairing in situ with live examples. Although the font listing is not huge, you can count on every font being web safe and is a great starting point for inspiration.

ADOBE TYPEKIT

typekit.com/fonts

Adobe Typekit is a healthy alternative to Google Fonts, providing you an abundance of choice on web-safe fonts. Create kits for individual projects and navigate through a variety of available fonts. What's useful about this tool is the ability to rewrite the sample text and filter through typography styles. With a font chosen it's really easy to install on a web page using a couple of lines of code in the head of your HTML and CSS.

FONTS.COM

fonts.com

A great resource if you want your project to be more individual and unique. Font libraries like Google Fonts and Adobe Typekit are always great, but you can count on your chosen fonts being used by others too. The wonderful thing about Fonts.com is that you can browse through a larger portfolio of fonts and see how they are being used by the font authors. It's again a great inspiration resource, especially for the personal and unique.

10 GOLDEN RULES FOR COMBINING FONTS

Bringing together compatible fonts is essential for great designs

1 COMPLIMENT YOUR FONT PAIRING

Don't go mad with fonts, pick a couple that work together and set the tone and mood you want to portray. Contrasting fonts tend to work but be careful of readability and scale.

2 UNDERSTAND CONTEXT

Where is your typography being displayed and who is going to read it? The right visual direction and tone is important when it comes to making your messages readable and relatable.

3 DON'T BE AFRAID OF SPACE

Clarity and breathing space is important when making a message noticeable. Don't make textual elements fight each other in a tight space. Nobody wins. Space helps keep focus.

4 SIZE MATTERS

It's important there's not a huge variety of text sizes on a page. Consistency, hierarchy and readability are key. Ensure your fonts follow set size rules and don't vary. Stick to a pattern.

5 STYLE CONSISTENCY

If you decide to use uppercase or letter spacing for your headlines, don't deviate and ensure the other style changes you make are followed.

6 APPLY COLOUR WITH ACCESSIBILITY IN MIND

Legibility is essential when choosing a colour for typography. Contrasting colours are best, but be mindful of accessibility. A resource to check is WebAIM Color Contrast Checker.

7 MIX YOUR STYLES

Use contrasting styles/fonts like Sans-Serif and Serif, to achieve more varied looks. Sizing and hierarchy is very important but the difference in styles can make a big difference.

8 WATCH YOUR WEIGHT

Most fonts come in variety of options and weights. Don't deviate from 'normal' on body text, keep text readable. Don't use too many weights, reserve for your headlines.

9 MAKE USE OF FONT FAMILIES

When pairing with other fonts, you can use variations from families but be mindful not to overuse. Use one or two weights to add variety and hierarchy.

10 PLAY AROUND

It's good to find font pairings you like that you stick to, but with the amount of fonts out there to experiment with, it's always worth having a play with new and interesting fonts.

Type

The quick brown fox jumps over the lazy dog

PROXIMA NOVA
adobe.ly/1k4Zr4S

FREIGHT TEXT
adobe.ly/2DhOexG

Type

The quick brown fox jumps over the lazy dog

MRS EAVES XL
adobe.ly/2rdMDmz

OPEN SANS
adobe.ly/2D86IMU

TYPE

The quick brown fox jumps over the lazy dog

ALTERNATE GOTHIQUE
adobe.ly/2rhkNGr

ROBOTO
typekit.com/fonts/roboto

Type

The quick brown fox jumps over the lazy dog

PLAYFAIR DISPLAY
bit.ly/2j4b4eL

ALICE
bit.ly/2DB1vR4



WHAT'S TRENDING TODAY

There are a number of different treatments and trends that you can apply to typography in order to better tell your stories. Generally, making your typography more integrated into your design is a good start and less about a headline plonked onto the canvas.

Here are some examples of typographical treatments to try and look out for:

TYPOGRAPHY CHAOS

This trend is still going and enables you to be a bit more chaotic with your composition of typography. Splitting words and even letters around the canvas can offer a really interesting look. In some cases, the unconventional arrangement can test the readers with curiosity.

CROP AND CUT TYPE

On the same lines of creating interest and making typography unique, simply crop or cut letters up within a word. Make sure it's enough that the letters are still readable. There's a really interesting neuro behaviour that readers can still identify letters within words, even if characters are missing. Take Hangman for example.



Make sure that your typography is contemporary

NEGATIVE SPACE

Mostly using artwork within your typography creates negative space. This is a wonderful way of really integrating artwork within your typography and making your text stand out. A great technique when using negative space is it enable graphical elements to pop out of the type, creating a dimensional space.

GRAPHICAL TYPE

Typography can be a little flat so enhancing your typography graphically makes and textual elements stronger on the page. It might be that your typography is made furry or even paint like. The objective here is to make your typography more like an object as opposed to a character.

RESOURCES

GOOGLE WEB DESIGNER

bit.ly/2DCT2uk

Best used for creating Google Ads but also a neat little tool to create dynamic displays whereby you can animate textual elements.

SLIDER REVOLUTION

bit.ly/2avbF9a

If you're developing in WordPress, this slider plugin is incredible for creating custom sliders with very little limitations. Using the timeline and various animation styles, your typography can really make an impact on the page. You can work with numerous layers and integrate your graphics too. A truly smart way of bringing more life to your web projects.

ADobe ANIMATE CC

adobe.ly/2rb71Vs

If you haven't dipped into this yet, it's worth doing and not being put off the association with Flash. You can export HTML5 canvases and achieve a higher level of animation. When it comes to typography and more complex graphic directions, Adobe Animate will help you out.



DYNAMIC: WHY & HOW?

In many cases, typography is really static. It doesn't move or engage other than serving one purpose and that's to be read and understood. We're all well aware of other media formats that are used to be more engaging – video, for example – but text can be just as exciting.

Have you ever seen Honda's 'Keep Up' campaign?

For a 2015 campaign, it's still a great example on how words can be the focus point and heart of a campaign. The campaign projects words quickly in the video forcing you to read in order to keep up. The central alignment and placement of the text makes it easier to read.

Speed reading is the core method used here, causing

viewers to focus entirely on the words that are displayed.

There are many methods other than speed reading to make textual elements more engaging, but thinking more about movement and integration within a design can be really effective and creative.

How? Well there are tools out there to help, but it all

depends on what you're executing and where it's going. The easiest way is to use video, but if you can achieve animations through web languages such as JavaScript and CSS3, then go for it!

Honda - 'Keep up'
bit.ly/2ritDUO



6 STEPS TO MEANINGFUL TYPOGRAPHY

1 READ THE CONTENT

Meaningful typography starts with one thing – the content. Typographers know this since they'll always read the book before they start typesetting for it. Unfortunately, it seems that many web designers underestimate the importance of content in a web design process. They will often find excuses in the fact that the website doesn't exist yet so there's no content to work with. When that's the case, use content that is similar. If you're designing a website about finance and economics, for example, find an article about that and read it.

2 CHOOSE A TYPEFACE BASED ON CONTENT

Now you have read the content, you're ready to choose your main typeface. If a website is about technology, but is expected to have medium to long articles, use a typeface that looks a bit modern but is easy to read. If it's an art gallery portfolio, you can get away with something edgy. Don't use Lorem Ipsum as placeholder text – it's a strange form of the Latin language that has nothing to do with your website. Use the content from Step 1, in the language that will be used, and then design around that.

3 START MOBILE-FIRST

An important step is to design the best reading experience for the screen that's hardest to design for – mobile. Mobile-first is a fundamentally different approach to web design where progressive enhancement is favourable to graceful degradation. Don't design the best reading experience for desktop screens and then adapt for mobile or, even worse, forget about mobile altogether. Choose a combination of font size and line height that works best on smaller screens. Your starting point should be the agreed-on browser default of 16 pixels.

4 ADAPT FOR LARGE SCREENS

Don't let mobile-first turn into mobile-only. The tools for shaping the best reading experiences for different screens are in place and they should be used. Larger screens are usually further away from reader's eyes so the base font size needs to be larger. 18 pixels is widely considered a good starting point. Don't forget to limit the width of paragraphs – recommended for best reading experiences is up to 60 characters per line. The line height needs to be looked at again – 1.4 or 1.5 times the font size is usually best.

GREAT EXAMPLES

Matej Latin picks a selection of typography that works well

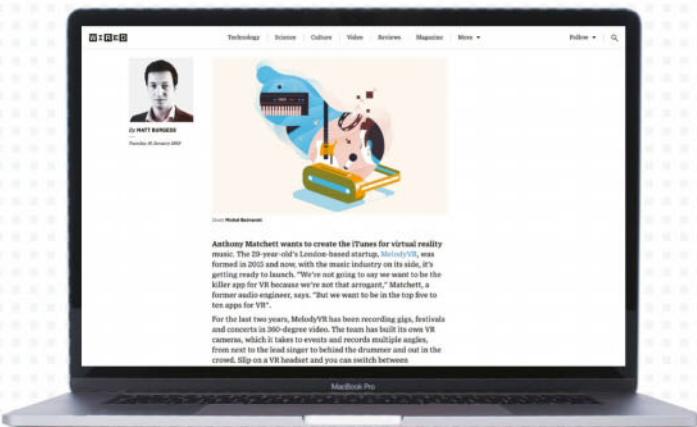


The Independent – independent.co.uk

The Independent benefits from a bespoke typeface, Indy Serif, which seems very capable at delivering any type of news content on any medium. Well-shaped paragraphs offer a great reading experience.

Wired – wired.co.uk

Wired comes with a typeface designed for web usage (Exchange Web) and neatly designed reading experiences for both, mobile (17 pixels and 1.3 line height) and desktop (19 pixels and 1.4 line height).



5 USE A SCALE

It's now time to define a range of reusable font sizes based on a scale and the most common way to do that is to use a modular scale. Go to modularscale.com, enter your base font size and choose a scale. It will give you a range of font sizes to choose from. Defining a scale and trying to stick to it adds meaning to font size choices and prevents the chaos that often arises from randomly assigning them instead.

6 SET A BASELINE GRID

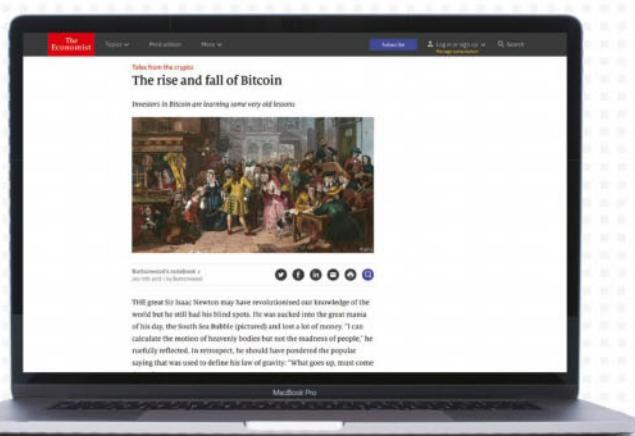
The next step is to start thinking about other text elements around the body text you should have designed by now. Titles, lists, captions, side comments, etc. To add meaning behind placing these elements on your website, it's best to use a baseline grid. This grid originates from your body text line height. If your line height is 22 pixels, you need a vertical grid based on that. When that is in place, you're ready to set the sizes and margins of other text elements so they'll fit inside this grid.

MATEJ LATIN

Lead UX/UI Designer at
Auto Trader UK

matejlatin.co.uk

@matejlatin



The Economist — theeconomist.com

The Economist comes with generously sized body text, topped with a crisp, slightly lighter typeface — Milo Serif Pro. Mobile-dedicated font size and line height help improve the reading experience.

The Great Discontent — thegreatdiscontent.com

The Great Discontent uses Leitura News for its body text. Its unique style complements the type of content (interviews) very well. A large base size makes the content very easy to read.



Describe your path to becoming a designer and illustrator.

It's really the only thing I ever thought I'd do. As a kid, I drew all the time and my grandma embraced that early on because she was an oil painter. We were the only artists in our very sports-oriented family. My dad was a college baseball player and he's been a coach all of his life; my brother played college football, and my sister was all-state in a bunch of sports. I think my grandma was the first artist I ever saw and bought me a drawing desk and all the art supplies I wanted. I also played sports, but that was the only thing I ever loved.

Fast forward to college — it was pretty clear that I was going to do something art related. There was one slight hiccup where I thought I might be a preacher like my dad and grandpa, but that only lasted one semester because I just missed too much.

As far as illustration, that was an interesting addition. When I got to college and discovered Photoshop and Illustrator, I did not want to do anything else. I started the projects where the first week was sketches on individual. I never filled one sketchbook the whole time I was in college. I fell in love with drawing on the computer — that did it for me.

I graduated from college and didn't sketch at my first two jobs. It wasn't until I started *Facebook* with my buddy, Paul, that I began to draw again. The first year I worked directly with clients and had to deal with a lot of stuff getting ironed out. Because it was only a two-person team, I wasn't as

"As a kid, I drew all the time and my grandma embraced that early on because she was an oil painter. We were the only artists in our very sports-oriented family."



COLOUR & ACCESSIBILITY

Colour and accessibility are key things to be mindful of, but are too often forgotten about by designers. You may be designing something on your nice shiny iMac with an incredibly glossy Retina display but not everyone will have one of those. Screens render colours differently and you can count on older screens having a harder time telling the difference between closely matched colours. Not only that, you have accessibility considerations whereby the viewer may have eye



sight difficulties or colour blindness. Contrast will always save you here and there are many good guides to avoid these issues.

Check out Web AIM (Web Accessibility in Mind) to see how your chosen colours rate within the AA and AAA web standards.

SCREENS RENDER COLOURS DIFFERENTLY AND YOU CAN COUNT ON OLDER SCREENS HAVING A HARDER TIME TELLING THE DIFFERENCE

CHOOSING COLOURS

1 BEST FIT

Colours can be very subjective but are nevertheless important when it comes to look and feel. The personality of your project still has a complete hold of the colours you choose. The first step would again be to think about your values and what colours best fit.

It's best to acknowledge colour psychology before you start picking any colour. This is because you can really influence human behaviour and perceptions of what is being seen by the colour used – take warm and cold colours, for example. Although obvious, we have the ability to control a reader's feelings and behaviour just by the colours we use.

2 LESS IS MORE

As with text, hierarchy is important. Having some ground rules for colour usage prevents your artwork looking like a pack of fruit pastilles. Think about primary, secondary and even tertiary colours that can be used. There's no real limit to how many colours you can use but less is often more. Knowing how many colours you're planning to use, will help maintain consistency.

ESSENTIAL RESOURCES

ADOBE COLOR CC

adobe.ly/2wLlqg

Adobe provides a range of rules that will help discover closely matched colours from selecting a primary colour. If you work with Adobe products, save your palette to a library to get rid of copying and pasting.



It's common for more corporate-related designs to have less colours than something perhaps more friendly.

Another thing to be very mindful of is reserving a single colour for your call-to-actions and buttons. It's a good idea to single out a colour so that links and buttons are more individual and user-friendly.

CHOOSING COLOURS

Let's start by choosing colours since we now know the personality and hierarchy. You can use a tool like Adobe Color CC to play around with your colour choices. This tool is simply a colour wheel that gives you the freedom to select and find colour palettes.

Focus on your primary colour first and play with the rules that Adobe gives you if you want help finding other colours. These calculated rules are a useful headstart but not restrictive. Once you have selected your primary colour, work through selecting your other colours. Remember accessibility and contrast in your options.

SHADES OF COLOUR

Once you have selected your colours, there's a few

more things to think about. Lightness and darkness will be one. Although you have your chosen colours, there are no rules to restrict you from using lighter or darker versions. For example, we may have a bright blue as our primary colour and we are using it for some of our foreground content. We could darken the same blue (with a shade of black) for a treatment in the background. Be mindful of consistency and not create too many variations of your colours, but this method will steer you clear of the fruit pastilles and help with creating contrasting colours based on your own palette.

TEST AND TRY

Apply, play and change if necessary. It's only when using everything in context that you can truly gauge if your design is working. Look back over your values since this will help validate whether your creation works.

Create a number of versions and try new arrangements of colour. There's never a single right answer to any creative project and the process of experimenting is fun. If you have options and a history of experiments, it's easier to see how your design has progressed and evolved.

Make sure to get your colour right

COLBLINDOR

bit.ly/1qSSJSu

A great tool to check your design against colour blindness. You can safely upload your artwork and get a feeling of how it is to have a colour vision handicap.

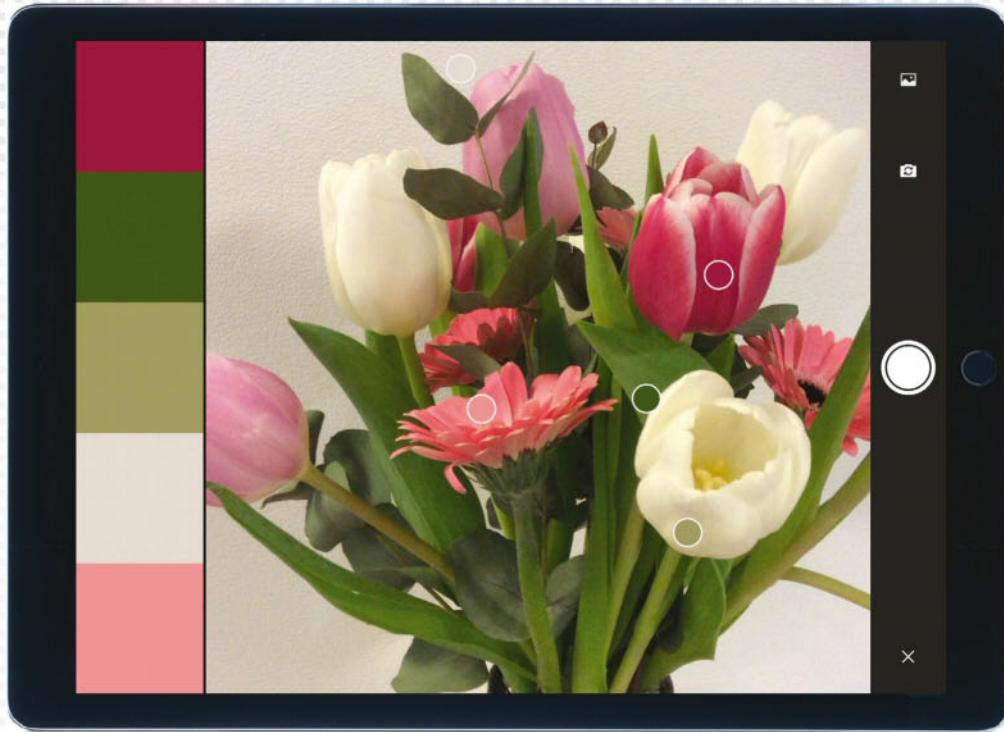


WEB AIM COLOUR CONTRAST CHECKER

bit.ly/2ALPtTp

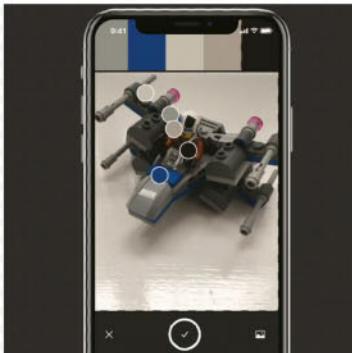
Using this tool will help ensure that your chosen colours meet accessibility criteria. The AA and AAA web standards will make sure that your designs are compliant and user friendly.





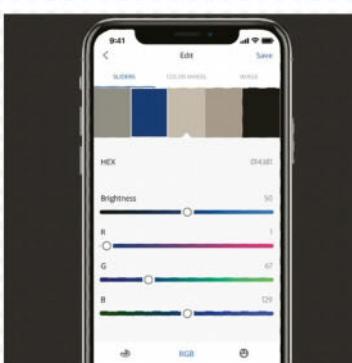
COMBINE CAPTURE & COLOUR

Adobe Capture CC is a fantastic app that enables you to find fonts and colours simply by taking a photo. You may be wondering what fonts are used on a favourite magazine, ahem. Or you may wish to take colour references from an object or scenery.



1 DOWNLOAD THE APP

You can find the app on the iOS App Store or Google Play. Simply search for Adobe Capture, download and register/log in.



2 FIND COLOURS FROM A PHOTO

From the Colors tab within the app, simply tap the '+' button and point the camera towards the object or scenery you wish to capture. You will see that the app is already working at finding popular colours. Once you're happy with your arrangement, take the photo and refine it further.

You can proceed to save your captured colours to a library, picking them back up and using them in other Adobe tools.

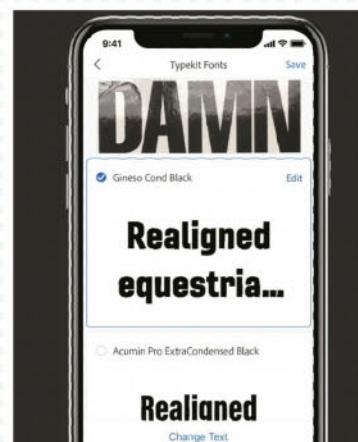


3 WHAT THE FONT?

You can also find out what fonts have been used or find closely matched fonts in printed material. Head over to the Type tab within the app and direct the camera over the typography you wish to capture. There will be a blue line that helps you with alignment. Once you have the typography in view, take the

Quickly create a palette with Adobe

photo and then make sure the app has cropped the text correctly. Hit the big tick and let Adobe analyse the image. Once it has finished analysing, it will provide you with a list of best matched fonts within Adobe Typekit that you can then go on to use.



4 CREATE A SHAPE

Although it's not strictly typography or colour-related, Shape is a very useful feature for a number of reasons. It's a great way of vectorising an object so that you can play around with it further in Illustrator.

For example, you may wish to vectorise a signature or perhaps some freehand typography. It's great for capturing elements that are just sometimes easier to draw with pen and paper.

COLOUR PERSONALITY

BLUE

- Intellectual
- Coolness
- Trust
- Calm
- Efficiency

RED

- Courage
- Warning
- Strength
- Excitement
- Warmth

GREEN

- Refreshment
- Environmental
- Restoration
- Peace
- Reassurance

YELLOW

- Optimism
- Friendliness
- Confidence
- Creativity
- Extroversion

web workshop

Overlay graphics on video backgrounds

Inspired by <http://project-redspace.com>

Main navigation
The main navigation is available, hidden behind the main burger menu. Clicking this animates the overlay menu down from the top of the screen.

Video background
The site is set up to tell a story, very similar to a documentary, and the main background is a full screen video.

Overlaid SVG images
When the video starts to play, the crisp SVG images spring forward and expand into place in the centre of the display.

Call to Action
After the images have animated onto the screen, the text and call to action button appear on the screen enabling the user to go deeper into the experience.

Keeping you oriented
The bottom of the screen has three stages of the story and locates the user where they are as a means of navigation to efficiently get around.

Overlay graphics on video backgrounds

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

EXPERT ADVICE

Engage the user

The Project Redspace site works so well because it engages the user with a strong narrative. Having a story at its core makes the user intrigued, wanting to know more as they are drawn deeper into the unfolding events. With the user actively engaged in the content, they will take time to explore all that the site offers.



<comment>

What our experts think of the site

An easy to follow process

Making use of video segments throughout the site shows the design process of the car in a very visual way that creates a compelling story to hook the user into the ideology of the project's goals. Three main pages work together to tell different aspects of the story and make for easy navigation.

Mark Shufflebottom, Professor of Interaction Design

Technique

1. Creating graphics over video

The first step in creating graphics over video background is to add the video to the document. The video placeholder here is an open source movie that will serve as a background. This is placed inside a div tag and appropriate classes are added for CSS control.

```
<div class="fullscreen-bg">
<video loop muted autoplay class="fullscreen-bg_video">
<source src="http://dev2.slicejack.com/
fullscreen-video-demo/video/big_buck_bunny.
mp4" type="video/mp4">
<source src="http://dev2.slicejack.com/
fullscreen-video-demo/video/big_buck_bunny.
ogg" type="video/ogg">
<source src="http://dev2.slicejack.com/
fullscreen-video-demo/video/big_buck_bunny.
webm" type="video/webm">
</video>
</div>
```

2. Adding the graphics

Below the video in the body section, the content is added. In this case that content is going to be two SVG graphic elements and a heading. The content is placed over the top of the video and then the graphics will be stacked up through CSS.

```
<div class="content">
<div class="graphics">

</div>
<div class="graphics">

</div>
<div class="graphics">
<h1>Journey</h1>
</div>
```

3. Creating your CSS

In the heading section of the page, or in a separate style sheet, add the following code. The first rule sets all elements to have border-box sizing so that the border and padding are included in sizes. The body of the page is given the full height of the browser.

```
* {
  box-sizing: border-box;
}
html, body {
  margin: 0;
  padding: 0;
  height: 100%; }
```

4. Position the background video

The background video is positioned absolutely over the full dimensions of the screen and given a minimum width and height so that they always completely fill the screen. This video element is centred vertically and horizontally in the browser.

```
.fullscreen-bg_video {
  position: absolute;
  top: 50%;
  left: 50%;
  width: auto;
  height: auto;
  min-width: 100%;
  min-height: 100%;
  transform: translate(-50%, -50%); }
```

5. Content positions

The container for the video is given a fixed background position and is positioned to fill the entire screen, while being placed behind all other elements on the display. The content runs, by default, over the top of the other media and is told to be relatively positioned.

```
.fullscreen-bg {
  position: fixed;
  top: 0; right: 0;
  bottom: 0; left: 0;
  overflow: hidden;
  z-index: -100; }
.content {
  margin: 0 auto;
  width: 100%; height: 100%;
  max-width: 960px;
  padding: 0 15px;
  position: relative; }
```

6. Positioning the graphics

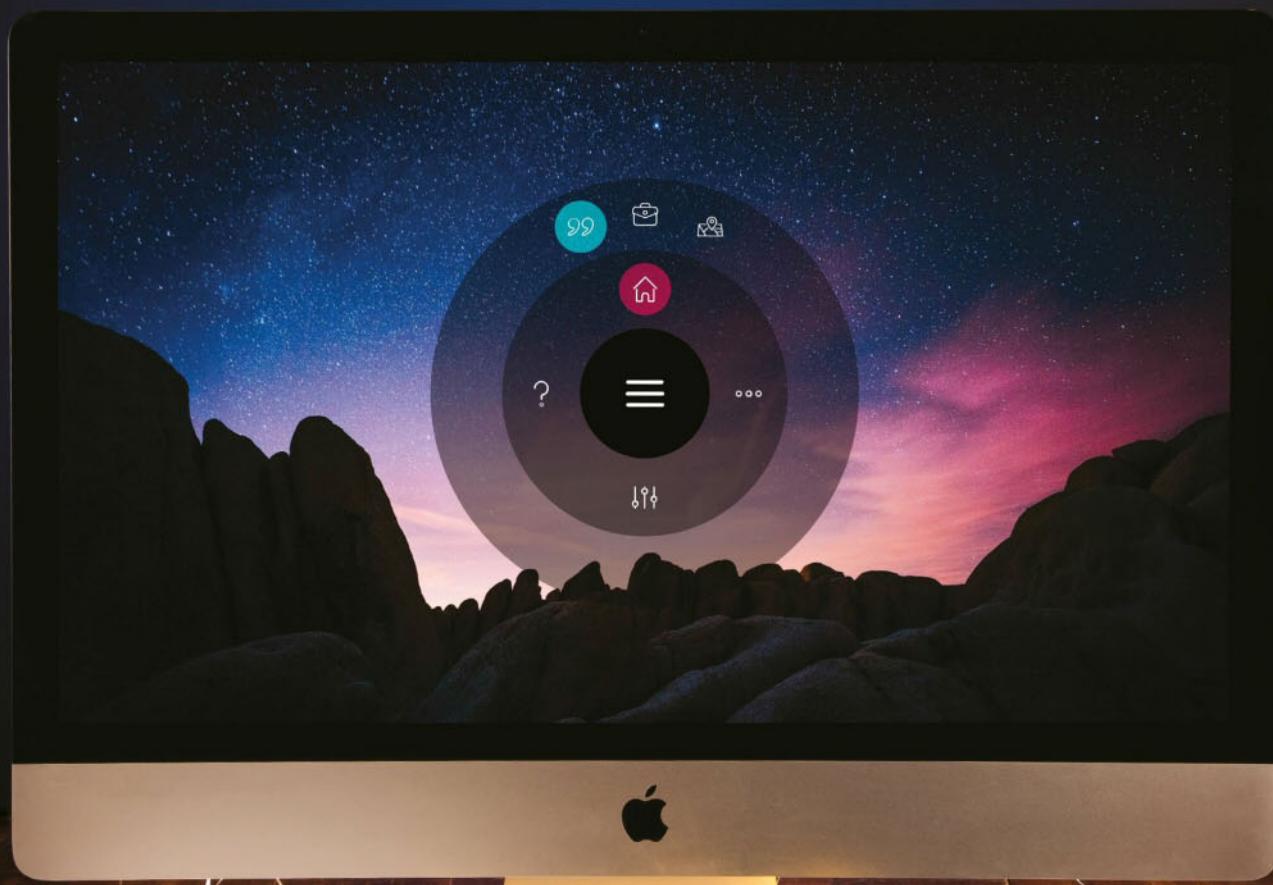
The graphics are placed over the top by placing them absolutely in the relatively positioned content box. This class is applied to graphics and heading so that it's over the top. Finally the heading is styled up with the appropriate font and colours.

```
.graphics {
  width: 100%;
  position: absolute;
  top: 50%;
  transform: translateY(-50%); }
h1 {
  font-size: 8em;
  text-align: center;
  font-family: "Helvetica Neue", Helvetica,
  Arial, sans-serif;
  color: #fff;
}
```



Code a dynamic SVG radial menu

Use SVG, jQuery and the GSAP animation library to create an animated radial menu



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

There's so much that can be achieved natively in the browser using CSS3 or the Web Radial menus. It's something we've seen used on the web before, mainly back in the days of Adobe Flash – although recently implementations in JS and/or Canvas have been spotted in the wild – but they're often seen in software used on touch screens. We'll be creating our very own radial menu but we'll be using SVG for the drawing and jQuery and GSAP to handle the interaction and animation.

SVG is perfect for these kinds of elements due to its navigable DOM and responsive nature. We can capitalise on these properties to create a complete user interface (UI) or specific parts of a UI – such as a menu.

The idea behind our radial menu is that each circular layer would be revealed when the layer before is interacted with. The menu contains three layers; the first layer is the menu button in the middle; the second consists of four icons; and the final layer contains three buttons for each of the four buttons found on the previous layer.

We'll be creating the design in a graphics editor (this example was created in Sketch) but you can use any vector drawing program that exports to SVG.

1. Design the menu

The first step is to create a design for the menu using a graphics editor. The grouping of elements here is important as we need to create all of the layers we'll need later for the reveal process. The circular backgrounds on the icons are the hover states and the hit area for each button. Once complete we need to export the SVG file. Check out the code snippets here: <http://srt.lt/cOH1>

2. Create our first timeline

GSAP offers two timeline types: TimelineLite and TimelineMax. The 'max' version offers us access to additional features such as the ability to repeat animations, as well as playing them in reverse among other features. Our first timeline will be the water which



we will repeat infinitely and yo-yo back and forth.

```
var animation_water = new TimelineMax({
  repeat: -1,
  yoyo: true
});
```

<http://srt.lt/D9rN0m>

3. Container styles

Next we'll create a new pen on CodePen and paste our optimised SVG code into the HTML editor. We'll wrap a container div around the SVG and provide some simple styles to position and size the container, as well as applying the background image.

```
.container {
  position: relative;
  display: flex;
  align-items: center;
  justify-content: center;
  min-width: 100vw;
  min-height: 100vh;
  ...
}
```

<http://srt.lt/k7EeS>

4. SVG styles

Next we'll add a class to our SVG and provide some simple styles to control the size of the menu.

```
.menu {
  width: 100%;
  height: auto;
  max-width: 792px;
  max-height: 100%;
}
```

<http://srt.lt/zYcA4c>

5. Hover and active states

We can use CSS to create our hover states and use a class name of 'is-active' to control the basic interaction. We'll change the cursor to a pointer and change the opacity of background circles. We will also add some similar styles to the main button in centre.

```
[class$="-button"] {
  cursor: pointer;
  circle {
    opacity: 0;
    transition: opacity 250ms;
    &:hover {
      opacity: 0.6;
    }
  }
}
```

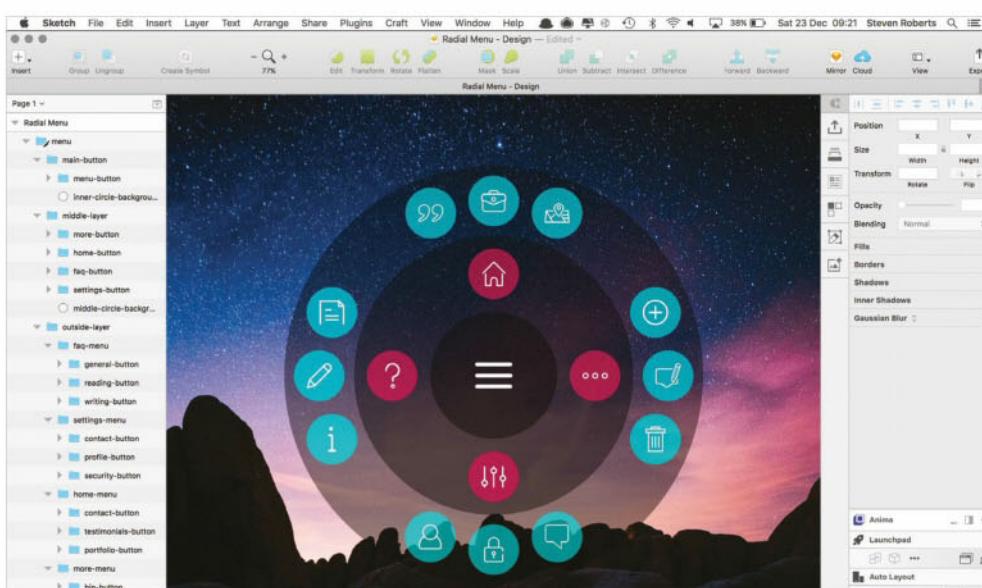
<http://srt.lt/RtXn>

6. JavaScript

For the remainder of the interaction we'll need to turn to JavaScript. Make sure you've included the GSAP and jQuery libraries, so we can create our first timeline and animation. We're going to animate the middle layer of buttons from a scale of '0' to its original size.

Accessibility

The code we are writing for this demo is meant as an example of creating interactive SVGs. The code is not accessible and shouldn't be used for production without modification.



Left

Create the design using a vector graphics editor such as Sketch for Mac

Top

Using CSS we've provided a simple interaction when the buttons are hovered over

Tutorials

Code a dynamic SVG radial menu

```
var animate_middle = new TimelineLite();
animate_middle
    .from(".middle-layer", 0.5, {
        transformOrigin:"50% 50%",
        scale: "0"
    }, 0, 0)
;
http://srt.lt/kLc8M
```

7. Spin and fade in the icons

The GSAP library allows us to layer animations onto a single timeline. We'll utilise this to create two more animations to the 'animate_middle' timeline. We'll add rotation and opacity animations to the buttons of the middle layer. The animation will now grow the circle while spinning and fading in the icons.

```
animate_middle
...
.from(".middle-layer .button-group",
0.75, {
transformOrigin: "50% 50%",
rotation: "-=135" }, 0, 0)
.from("middle-layer .button-group",
0.5, {
opacity: "0" }, 0, 0)
;
http://srt.lt/YjKm5C
```

8. Animate the outside layer

Next we'll create a new timeline for the outside circle. The animation will be exactly the same as the first animation we created to animate the inside layer. This time we won't be adding anything else to the timeline as we won't

always need to animate the buttons and circle together.

```
var animate_outside = new TimelineLite();
animate_outside
    .from(".outside-layer", 0.5, {
        transformOrigin:"50% 50%",
        scale: "0"
    }, 0, 0)
;
http://srt.lt/S1jNa
```

9. Timeline play states

Now we have both of the layers animating we can update their initial play states. This will allow us to pause the animations until we interact with them, hiding the two layers of buttons we've created animations for. We also want to create a new variable to set the menu state.

```
var animate_middle = new
TimelineLite({paused: true});
var animate_outside = new
TimelineLite({paused: true});
var menuOpen = false;
http://srt.lt/ZgCi7
```

10. Create our first click function

Now we have our animations created and paused when the page loads we need to create a function to trigger the animation when the centre circle is clicked. Within our click function we will check to see if the menu is open using our 'menuOpen' variable.

```
$('.main-menu').click(function() {
    if(menuOpen == false) {
        ...
        animate_middle.play();
        menuOpen = true;
    }
});
http://srt.lt/LaXOf
```

11. Close the first layer

In the last step we checked to see if the menu was closed, played the animation and updated the menu state variable. We now need to provide what happens when

the button is clicked and the menu is already open. GSAP allows for timelines to be played in reverse so we can have the centre layer animate back into the centre.

```
else {
    ...
    animate_middle.reverse();
    menuOpen = false;
}
http://srt.lt/k3PsOA
```

12. Additional variables

In the same vein as the main menu button we need to create a number of variables to keep track of the menu's current state. The outside layer contains four separate menus and the circle. Here we're creating a new variable for each of those menus and the outside circle.

```
var outsideCircleOpen = false;
var homeOpen = false;
var faqOpen = false;
var settingsOpen = false;
var moreOpen = false;
http://srt.lt/kG7pW
```

13. Outside layer logic

We now need to create a number of click functions, so we'll start with the home button at the top. When we click this button we need to check for two things; if the home menu is already open, and then inside of that logic, if the outside circle is already open. If the outside circle is already open we don't need to animate it, we can simply animate the buttons and leave the circle open.

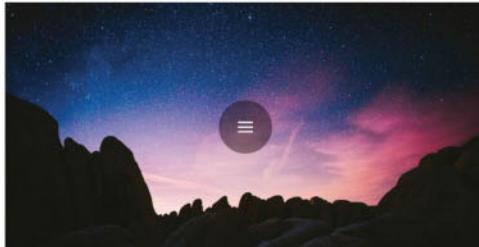
```
$('.home-button').click(function() {
    if(homeOpen == false) {
        if(outsideCircleOpen == false) {
            animate_outside.play();
        }
        outsideCircleOpen = true;
        homeOpen = true;
    }
});
http://srt.lt/E8tG
```

GSAP documentation and forums

The Greensock site is an invaluable resource for learning to utilise GSAP. The site includes learning resources, examples and a beautiful showcase of sites built using the library. <https://greensock.com>

Topleft

With our animations paused to begin with they are hidden when the page is loaded



Topright

We've now created the click function to play the first animation, revealing the first layer of buttons



Bottom left

The home buttons are now hidden due to the initial paused state of the timeline

Bottomright

The finished product including the active states for our buttons on each layer

GSAP animations

The GSAP animation library is simple yet extremely powerful. The library gives us a really simple syntax and handles all of the complex calculations under the hood for us. In its simplest form the library allows us to create timelines and then add animations to these timelines. This concept will be familiar if you've used Flash or any

other animation GUI.

```
var t1 = new TimelineLite();
```

Once we have created a timeline we can create an animation to add to it. The example below will animate the 'circle' element from a scale of '0' to its original scale, over the duration of one second. We're also providing the delay and start times of '0' although

this isn't required as they are the default values.

```
t1.from(
  'circle', 1, {
    scale:'0'
  }, 0, 0
);
```

The library also has a number of more complicated functions we can use

like the staggerTo and staggerFrom functions, which allow us to animate a number of elements one after another, as well as a number of paid plugins for really specific uses.

The library is performant and handles much of the browser compatibility issues you would run into making it ideal for production.

timeline.to/from(

```
'element to animate', animation duration, {
  property : value,
}, delay time, start time or label
);
```

14. Outside layer else logic

In the last step we created the logic for when the home menu was not already open, so here we create the opposite logic. We need to make sure to update the variables to ensure we can keep track of the menu's state. We also play the 'animate_outside' timeline in reverse when clicking the main menu button.

```
else {
  homeOpen = false;
  outsideCircleOpen = false;
  animate_outside.reverse();
}
```

<http://srt.lt/zDcYw>

15. Animate the home menu buttons

We now need to create a new timeline for our home menu buttons and change the initial play state to paused. We can utilise GSAP's 'staggerFrom' function to apply the same animation to the buttons, one at a time, with a given delay between each iteration.

```
var animate_home = new TimelineLite({paused:
  true});
animate_home
  .staggerFrom(".home-menu g", 0.5, {
    transformOrigin:"50% 50%",
    opacity: "0",
    scale: "0"
  }, 0.125, 0.25)
;
```

<http://srt.lt/q1SyG>

16. Add animations to click functions

We've created our home button's timeline so we now need to update our click function to include playing and reversing the animation when the home button is clicked.

```
$('.home-button').click(function() {
```

```
if(homeOpen == false) {
  ...
  animate_home.play();
} else {
  ...
  animate_home.reverse();
}
});
```

17. More menu

We now need to do exactly the same for the More menu by creating a click function and staggered animation timeline for the menu and its buttons. We also need to update the menu state variables within each of our click functions.

<http://srt.lt/i3rT>

18. Settings and FAQ menu

As you may have guessed we need to repeat the same thing for both the Settings and FAQ menus. Remember to add the menu state variables to each of the layers and play the animation timelines in reverse.

<http://srt.lt/x5wTx>

19. Almost finished

We now have all of the animations being triggered and played, or reversed given the appropriate click on the middle layer. We need to make sure all of the animations are reversed and the state variables updated when the main menu is clicked and is already open.

```
$('.main-menu').click(function() {
  animate_middle.reverse();
  animate_outside.reverse();
  ...
  menuOpen = false;
  outsideCircleOpen = false;
  homeOpen = false;
```

```
...
});
```

<http://srt.lt/Bk2Hz4>

20. Active states – middle layer

That's all of our animations and click functions created - the only thing left to do is to add the 'is-active' class to the button that has been clicked. jQuery allows us to use advanced CSS selectors to first listen for any click to buttons in the middle layer and then to remove the class from all the other buttons.

```
$('.middle-layer [class*="-button"]').
click(function(){
  $('.middle-layer [class*="-button"]')
    .removeClass('is-active');
  $('.outside-layer [class*="-button"]')
    .removeClass('is-active');
  $(this).addClass('is-active');
});
```

<http://srt.lt/T8Z3f5>

21. Active states – outside layer

We now do the same thing for the buttons on the outside layer, targeting, removing and adding our active class to the clicked button. That's it, our radial menu, created with SVG, animated with GSAP and coded with jQuery is finished. The animations are simple yet effective and the active states and hover interactions allow the user to keep track of their place in the menu.

```
$('.outside-layer [class*="-button"]').
click(function(){
  $('.outside-layer [class*="-button"]')
    .removeClass('is-active');
  $(this).addClass('is-active');
});
```

<http://srt.lt/k9Gj1x>

Collection of tutorial pens - <http://srt.lt/AmTp>

Design isometric typography

Learn how to build a city of letters, using the Pen tool, selections and layers in Photoshop





Perspective is everything in design. If something you've drawn has a perfect perspective, it will help your work look more realistic; alternately, Pablo Picasso made history by drawing with a very skewed sense of the reality around him.

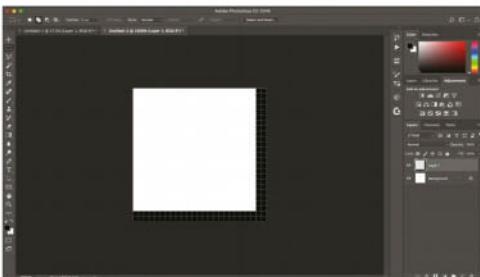
Isometric art is as much of a science as anything, though. It has a very strict set of rules when it comes to what goes where. That's pretty difficult to sit down and draw yourself, but when it comes to Photoshop, you can create a guide for anything.

In this tutorial, we're going to set out a perfect isometric grid before we build a world on hexagonally facing lines. In this London-inspired image, we're going to trace the Thames isometrically, and even create isometric versions of landmarks. You'll find that anything can be created on this grid, since it simply holds cubic shapes together. Curves can simply be drawn from corner to corner.

Everything in the image is going to follow this structure too; you can use images to copy, you can follow our steps religiously, or you can design your own icons following isometric patterns. It's entirely up to you – within the limits of isometry, of course.

1. Start the grid

Begin by creating a new document of 20 x 20 pixels. Create a black strip down the right and bottom of the document with the Rectangular Marquee tool. Use the Grid if you need to, by hitting Cmd/Ctrl^h:



2. Define Pattern

Go to Edit>Define Pattern. Name it and then click OK. Create a new 1920 x 1080 document by going to File>New, and then go to the Paint Bucket (G). Use Pattern fill and on a new layer, fill with the pattern you've created.



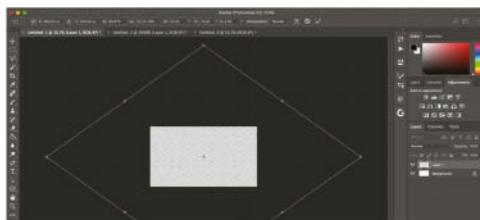
5. Lay out your text

Use the Type tool and create the text that you want to use across the document; we're going with 'London, England'. This is important, as it will dictate the size, spacing and whole look of your isometric poster.



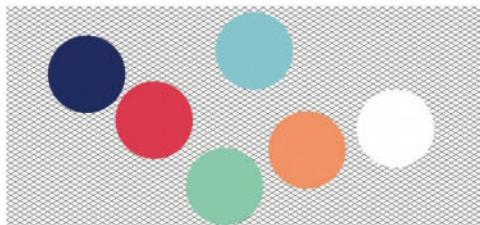
3. Transform the pattern

Now we're going to make the pattern isometric! Make the width and height much bigger, the angle 35 degrees and place in the centre of the document. The grid should be made of diamond shapes.



4. Choose your palette

We're going to use five different colours for this, but different shades of each. The colours are Royal Blue (#021574), Muted Red (#f22f50), Spearmint (#6ec5cf), Peppermint (#6ecfb0), Tan (#ff935e) and White (#ffffff).



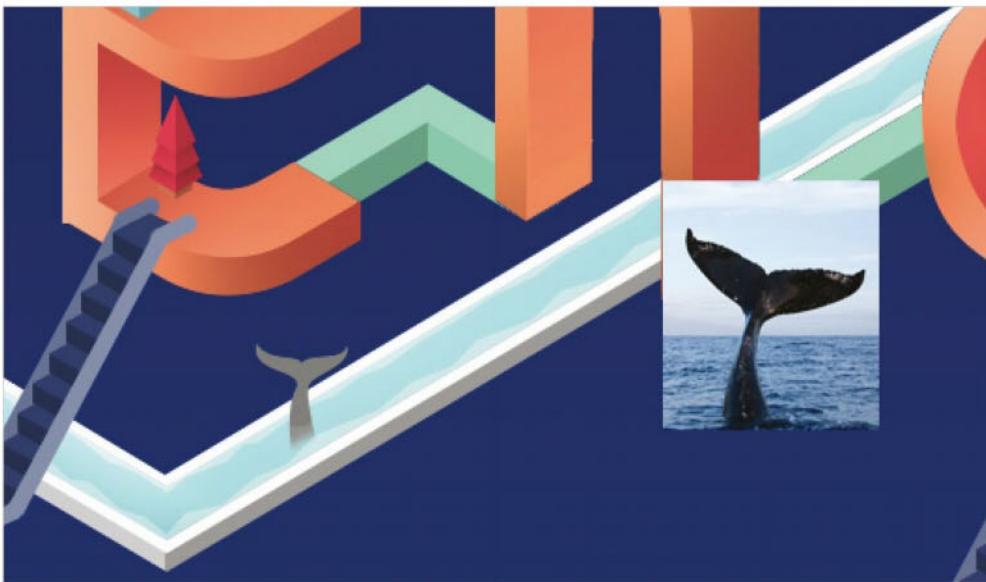
6. Create some letters

Create isometric letters with the Pen tool, following the guide that you created. This is a trial-and-error part of the process, and will take the longest of everything. Experiment with shapes and Fill with #808080 for now.



Selecting pixels on a layer

Hold Cmd/Ctrl and click on the preview layer of a window to select all the pixels on that layer



Above

Create a flat guitar with the Pen tool, and transform it up so it's in line with the grid. Duplicate and darken the lower layer by -20 lightness before moving this layer up slightly too, to make it 3D

Left

Use a reference photo of a whale tail to draw your own whale in the Thames with the Pen tool

Tutorials

Design isometric typography

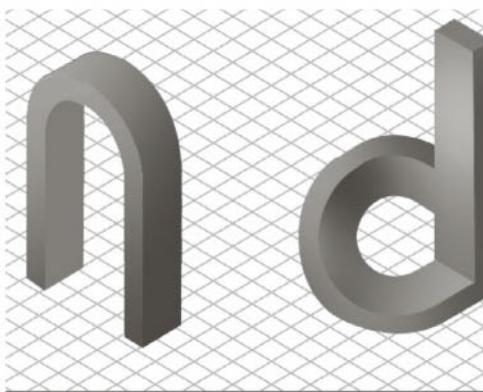
7. Work on the letters

The isometric letters are going to have to have one side that predominantly faces forwards. Work out where the curves are on each of the letters by reducing the opacities. This might take some time to figure out, and you may need to redraw some letters.



8. Shade the letters

Group each set of layers for each letter. Create clipping masks for each, and add either light or shade to the letters so that they appear 3D. Work out where the light is in the image and judge accordingly.

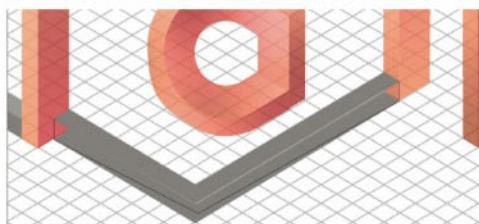


Grouping and merging

Group a set of layers that you may still want to edit individually; merge a set of layers that you want to start editing as one

9. Colour and build bridges

Create new Overlay layers above each of the letters and fill with one of your colours. Add Colour layers, clip, and brush in another colour to flesh it out. Next, create bridges between some of the letters. Colour this with another colour.



10. Create a speaker

Now, we're going to add objects to the scene. Create a box using the isometric guide, lightening and darkening either side of the box to create perspective. Create two circles, and Transform these to fit into the box along the guide; add a Stroke to the circles using Layer Styles.



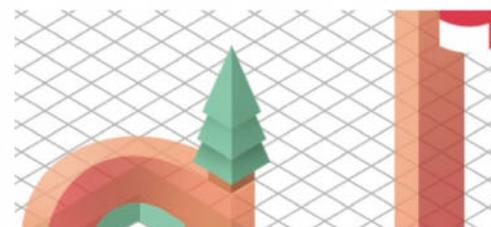
11. Add a flag

Next, we're going to create a flag. Again, use the Pen tool to draw along the guide, and then fill with white. Clip layers to this, and touch soft black brushstrokes over one side of the flag to add shade.



12. Plant some trees

Trees are easy to create with the Pen; create a triangle with the two bottom sides lining up along the isometric guide and then darken one side. Repeat this twice, then add a stump. Colour with an Overlay layer.



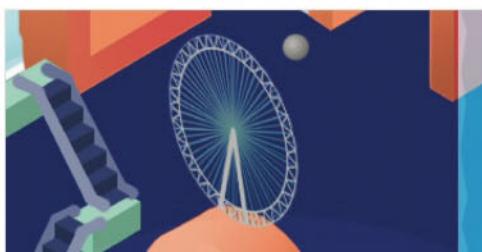
13. Escalators and map planning

We've created escalators using the same methods we used to create the other objects, using the isometric guide to draw over. After that, we've used a real map of London to roughly trace the Thames and draw a white outline for some water to fill.



14. Curve the edges

Use the Pen tool to curve the edges of the white structure that you've just created, and fill with #808080. Use a soft, black brush to create the illusion of a curved space for the water to sit in.



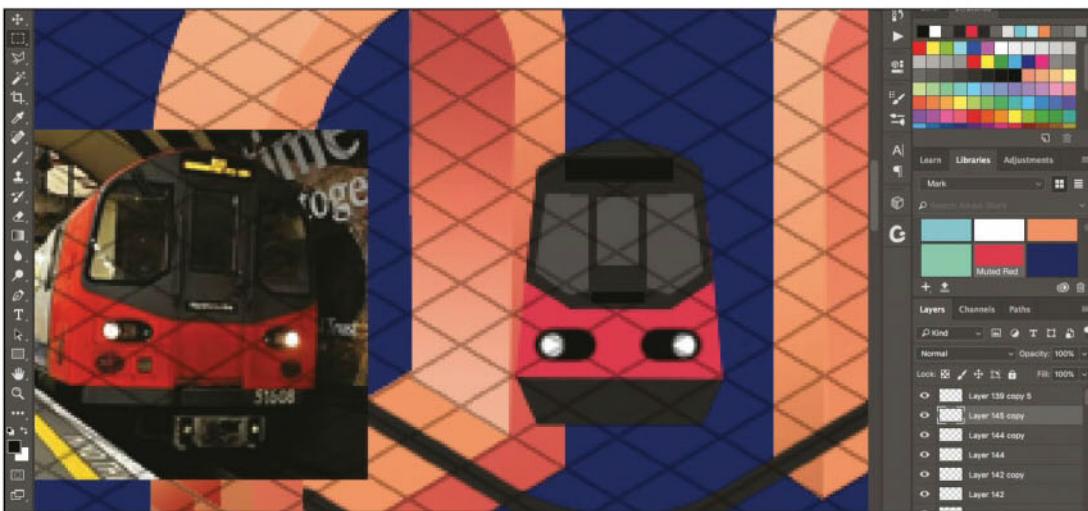
Above

Create a circle with the Elliptical Marquee and Ctrl/right-click to choose Stroke to draw your main circle. Draw lines within it, and draw a structure to hold it up. Transform this to the grid

Right

With your London Eye in place, draw the round brush strokes around the outside of the wheel. Make them smaller the further away they are in perspective.

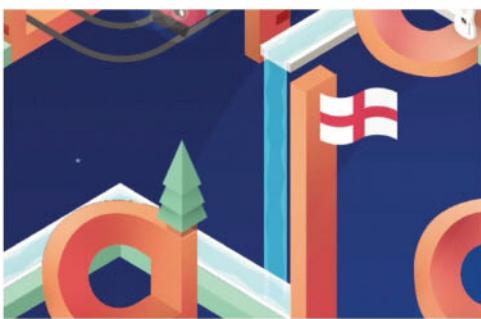


**Drawing the train**

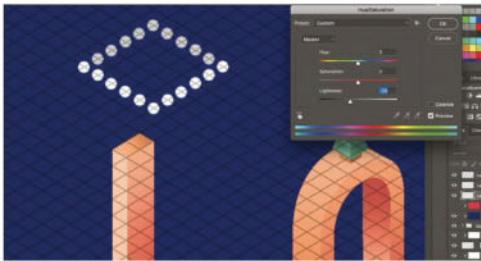
Start by drawing the train track along the grid, and use a reference photo to design the front of a tube train with the Pen tool. Transform it to the grid and place it over the tracks. Create a new layer below the front of your tube train, and create straight lines leading up under the 'N' with the Pen tool, before you create the rest of the train by filling in selections with blue, red and white. Create a window in grey, and then create three new layers. Reduce these new layers to 20% opacity, and on each create a white circle getting progressively bigger with each layer.

15. Fill with water

Use the Pen tool to select the grey space in each of the white structures and falling from the top structures, and fill with blue on a new layer. Set this layer to Hard Light to make it look like it's actual liquid. Clip a white layer and delete a jagged pattern in the middle for added effect.

**16. Start the chandelier**

Create a new layer and add dots in the intersections of the lines like you can see in the screenshot. Select the back rows of these dots and reduce the Lightness to -10 by using Hue/Saturation (Cmd/Ctrl+U). Duplicate this downwards to create more dots.

**17. Finish the chandelier**

Create the same effect three times with multiple rows of dots, then resize the two latter ones. Stack these to create a chandelier effect. On new 20% layers, create white dots to suggest light emanating outwards.

**18. Create Big Ben**

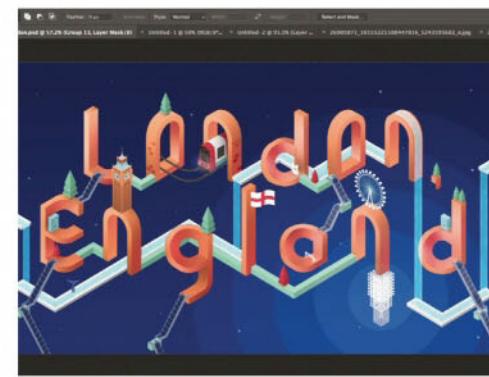
Use a stock photo of Big Ben to draw the clock with the Pen tool; you don't have to be as detailed, though, of course. Draw it in shades of grey; draw the spire to one side of the building rather than straight upwards, too.

**19. Place the landmark**

Colour Big Ben with some Overlay layers, duplicate the layer and then flip this new one horizontally. Resize both layers and Skew with Free Transform to fit along the guide that you have created.

**20. Finishing touches**

Make any final flourishes that you'd like to add more to the image; this might include stars in the background with little dots, a whale in the Thames or altering the colour with a Curves adjustment.

**21. Add noise**

Finally, select black and white in your Swatches (D). Create a new layer, fill with black (Alt/Opt+Delete) and go to Filter>Noise. Choose 400%, check Monochromatic and click OK. Set this layer to Soft Light, 10% opaque.



web workshop

Create multiple fade background colours

Inspired by <https://rocka.co>

Page navigation
Page navigation uses fixed positioning, always appearing in the same place regardless of how far down the page the user scrolls to.

Go home logo link
A logo is presented as a distinct part of the navigation, allowing users to return to the home page from any location and position.

Changing background
JavaScript event listening enables the colour of the background to change at specific scrolling locations generated by the user.

Page content
This is the page content that the user can scroll through. The effect only works if there's enough content to allow for scrolling.

Navigation prompt
A notification to help make sure that users of the website are aware of the additional content made available by scrolling the page.

Create multiple fade background colours

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

EXPERT ADVICE

Using body labelling

Using JavaScript to apply a status label to the body section is an easy way to keep code simple and allowing visual reactions to be controlled from CSS. Take this further by applying additional labels to describe specific events that have occurred relating to user interactions or other events.



<comment>
What our experts think
of the site

Avoid complexity to avoid headaches

It's worth spending extra time to identify the simplest way to create effects like this. Placing too much responsibility within JavaScript increases scope for future changes breaking functionality, costing time to fix. This example avoids the problem by keeping JavaScript's involvement to a minimum.

Leon Brown, freelance web developer and trainer

Technique

1. Initiate HTML document

The first step is to initiate the HTML document that will be the framework for storing the content and loading external resources. This document contains a head section that loads CSS and JavaScript resources, along with a body section for defining the visible content in step 2.

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

2. Body content

The main content can consist of any HTML, providing that it covers enough space to allow for scrolling. This example will use an article container that will be controlled via CSS to allow for scrolling.

```
<article>
<h1>Content</h1>
</article>
```

3. JavaScript scrolling

Create a new file called 'code.js'. This JavaScript listens for a scrolling event, upon which it calculates a screen number based on the vertical scroll position and sets the DOM body's 'data-screen' attribute as this value. Any visual changes to 'data-screen' can now be defined via some CSS.

```
window.addEventListener("load",function(){
window.addEventListener("scroll",function(){
var screen = Math.round(window.scrollY/
window.innerHeight);
document.body.setAttribute("data-screen",
screen);
})
});
```

4. CSS screen definition

Create a new file called 'styles.css'. The 'data-screen' attribute applied to the DOM body via the JavaScript is used to define the default screen properties. Its background colour is set to white, while a transition is set to animate changes to the background colour over a duration of one second.

```
[data-screen]{
background: #fff;
transition: background-color 1s;
}
```

5. Screen colour changes

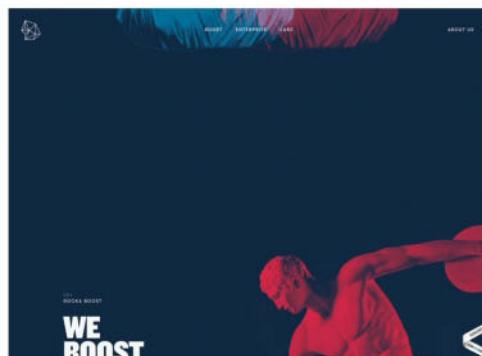
The JavaScript processing of scrolling events from step 3 allows CSS to define presentation rules for specific positions. Each count of 'data-screen' is the full height of the browser screen. This step sets different background colours as the user scrolls down the equivalent of a full screen height.

```
[data-screen="1"]{ background-color: red; }
[data-screen="2"]{ background-color: green; }
[data-screen="3"]{ background-color: blue; }
```

6. Example content size

Due to this example not having five screen heights of content to produce the required scrolling, this step sets the article container to fit this size. This allows you to test the effect before committing to inserting the content.

```
article{
display: block;
min-height: 400vh; }
```



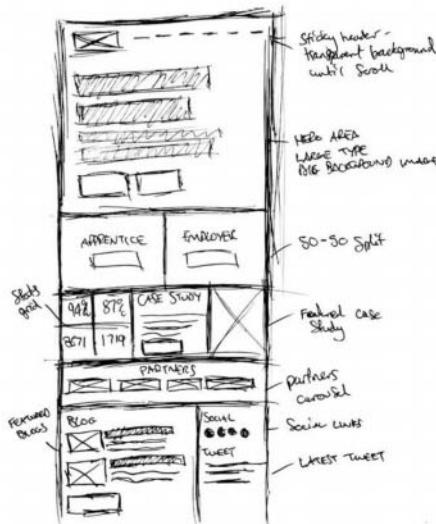
MASTER LAYOUT

Layout is one of the most important aspects of the web and applications. We need to consider a number of different things from the target audience to the brand and the content. Get it wrong and your users will get confused and leave

to find the information elsewhere. Here we take you through every aspect of the process, from the initial inception of the project through to building and structuring the code, and touch on which display properties you should use.

DESIGNING LAYOUTS

The Design phase of a project typically includes a number of processes



This image shows the design process for the Baltic Training website, from initial sketches through wireframes and design to the final live website

THE PROCESS

The design phase of a project usually begins with a scoping document of some description. This document will contain a specification of what you're designing.

The starting point for designing layouts really depends on the complexity of the project:

Brochure websites – for example – will usually use a fairly simple design phase and usually include the following processes: content gathering > site map creation > wireframing > high-fidelity concept design.

Whereas a more complex web app design phase might include a few extra processes: gathering content > site map creation > wireframing > low-fidelity prototype > high-fidelity concept design > high-fidelity prototype.

There's no right or wrong way to do this and different designers use different processes. It's up to you – as the designer – to find the right process for the project at hand. This may also be dictated by the project's budget, when this is the case you need to decide which of these processes are the most important for the project you're working on.

HIGH- VS LOW-FIDELITY

In the process diagrams explained above they mention both high- and

low-fidelity designs and prototypes – this simply is how far you go with the design. A low-fidelity prototype, for example, is usually made up from the elements in the wireframes; whereas a low-fidelity design may be slightly more *designed* than the wireframe you might use for the project's brand colours and fonts.

APPLYING THE BRAND

This includes obvious things such as the logo and colours and typography, but also everything else that comes with it: the language; the intent (is it playful or corporate? Is it confident and loud?); any printed materials; and anything the company has that

ELEMENTS AND RULES ARE USUALLY REPRESENTED BY THE BRAND GUIDELINES

represents them. This will help you understand how to display the content on the screen.

These elements and rules are usually represented by the brand guidelines document – if you have one, be sure to study it!

Once you understand the brand you can look for ways to apply it to your design – are there any brand elements you can use, such as shapes or patterns? Are there any ways to use

what you know about the brand to influence the layout and design of your project?

These details are what will give the design meaning and help cement the brand into the company's DNA. For a lot of users the website will be the first time they see the brand and, since you don't get to make a first impression twice, you need to get this right! Besides, being on-brand should help keep the client happy too, and that's always good!

STRUCTURING THE DATA

You can use the wireframing stage of the design in order to work out the content that needs to be structured.

It's always best to get the content before you start the high-fidelity design stage, in the real world this very rarely happens, but it's much easier to design something when you know what information you need to include. Once you know the content of the page you can start to structure the data by breaking it down into sections or chunks. Depending on the quality of the content this can be a really simple or a really difficult thing to do. It's important to keep linked information together so you don't confuse the user, but you can use these sections or chunks of content to influence the type of layout you're going to create.

SITEMAPS

Visual Sitemaps

A necessary part of the design. They tell everyone involved in a project what pages are going to be in the project and how they are going to be linked together.

You can simply draw these on paper or create them using a graphics editor, you could use Microsoft Word or even Excel for that matter. The idea here is to structure the data for the project.

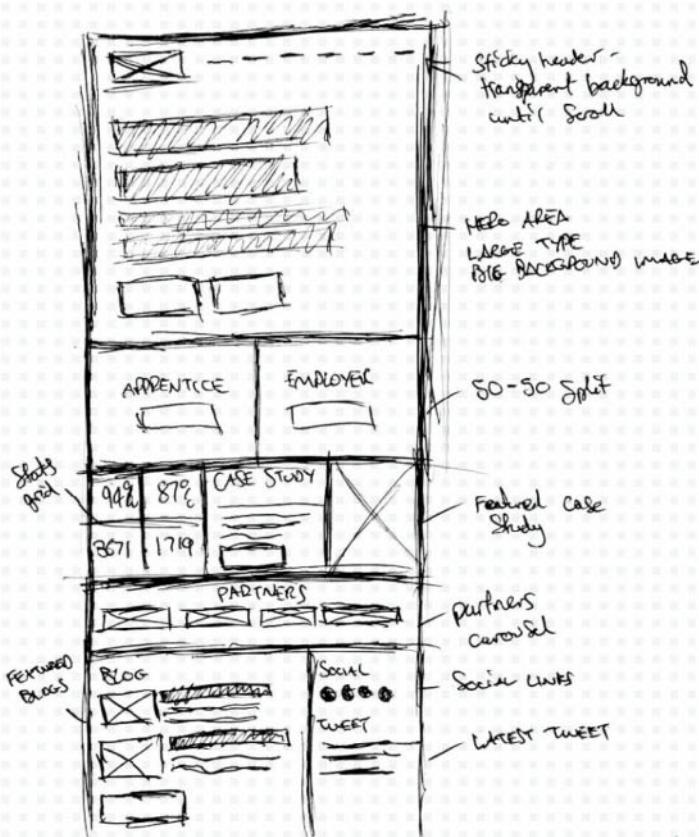
There are a number of tools built specifically for creating sitemaps, such as Slickplan, MindNode and WriteMaps.

XML Sitemaps

These are created once the project has finished, and are used by search engines to index your project's contents and structure. While they can be created by hand, there are a number of tools available for generating sitemaps – the tool you need will depend on your project. If you're using WordPress as the backend you can use a plugin to generate the XML sitemap. Once created you need to submit the sitemap to the search engines using their webmaster tools.

WIREFRAMES

Wireframes are structural designs of the layout



Initial sketch for the Baltic Training website homepage

These structural designs are meant to be kept simple and they need to be quick and easy to make. They serve as a visual representation of the data that's to be provided to the user on a particular page.

Multiple iterations on the same page can be made at this stage.

Wireframes should be used to flesh out initial ideas; the good, the bad and the ugly. Wireframes aren't meant to be beautiful and the initial wireframes don't even need to be shown to the client or stakeholders. Write down a list of elements and content you need to include and create every variation you can think of – it's very rare that your first idea is your best! During the wireframing stage designers tend to be much less attached to an idea too, probably because of the amount they've created.

The amount of time you spend wireframing a project will usually depend on its complexity and on budgetary constraints but wireframing a project and getting everything right at this stage can save a lot of money and heartache later on. By sharing these wireframes with the client or stakeholders early in the design phase,

problems in the layout and usability can be spotted and fixed much more quickly and less expensively than in a concept design, or worse during the build process.

CREATING WIREFRAMES

The best place to get all of your ideas down is on paper. A good method is to use some kind of felt-tip pen since this won't let you get into too much detail, we're just creating structure at this point. You can draw them again later with a finer pen to flesh out more of the details.

Draw blocks on the page and think about how they can be changed and moved and how this will affect the

WIREFRAMES SHOULD BE USED TO FLESH OUT INITIAL IDEAS; THE GOOD, THE BAD & THE UGLY

DESIGNS

Now you have the wireframes in place your next step is to choose which of the layouts you're actually going to design.

By sharing wireframes with the client or stakeholders you will have a clear indication of which of the wireframed designs is going to work best for them.

The design is where you will apply the brand to the project (the fonts, colours and logo); the brand language and content (if you have it); plus imagery and design assets. This is where you make the design truly unique to the project. These designs can be created in a number of different software packages. However, the two most popular are the Adobe Creative Suite and Sketch.

WHAT TO USE:

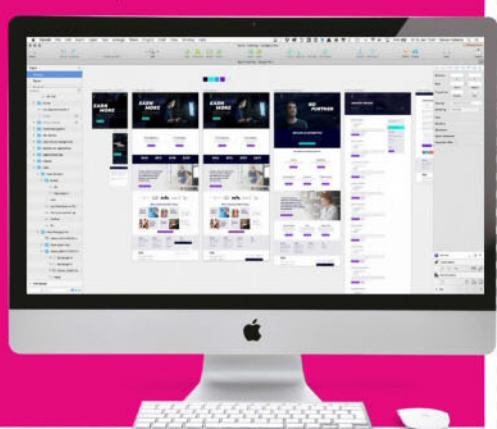
- **Adobe Creative Suite**

This is an industry-standard powerhouse and includes many different applications that you can use for your design, with Photoshop and Illustrator being the most popular. They both offer a number of useful features for web design. The grid system and smart guides are really useful if you're following a rigid grid (like Bootstrap) and the custom document presets are useful for creating canvases sized to particular devices. Photoshop has a useful feature enabling you to copy elements from the layers panel to either SVG or CSS, while Illustrator excels when it comes to vector graphics.

- **Sketch**

Sketch is a vector-based design tool for the Mac with a focus on screen design. When compared with Adobe's Creative Suite apps it is closest to Illustrator. Sketch makes designing for the screen a breeze with an intuitive interface and simple-to-use tools. The third-party plugins allow for the functionality to be extended, offering almost endless possibilities. Sketch also integrates with a number of prototyping apps such as InVision. The rendering in Sketch

(unlike Photoshop) is close to how the web renders content, ensuring that the design will be truer to the final product. Sketch is also fast and creates much smaller design files than some of its competitors.



flow of the content and the user's journey through the website or app that you're developing.

From all of the designs you've drawn you can then select a few that you think will work best and then refine them further. At this point you could use any number of tools at your disposal – you could draw them again

using a finer-point pen while taking the time to flesh out some of the details, or you could use apps such as Sketch, Photoshop or Illustrator to create these higher-detail wireframes. Alternatively you could use software specifically built for wireframes and prototypes such as Balsamiq Mockups, Axure or Adobe XD.

PROTOTYPES

A prototype is a design (high- or low-fidelity) that has some interactive elements to it. A prototype is used to show what happens when buttons are clicked, how the interactions will work and will simulate some aspects of how the project is going to function. Prototypes can be used to show how a full application will work or just how one small interaction will work. They're much cheaper and quicker to create than building the product and can give you valuable feedback before building the project. There are a number of applications we can use to create these prototypes.

WHAT TO USE:

• Adobe XD

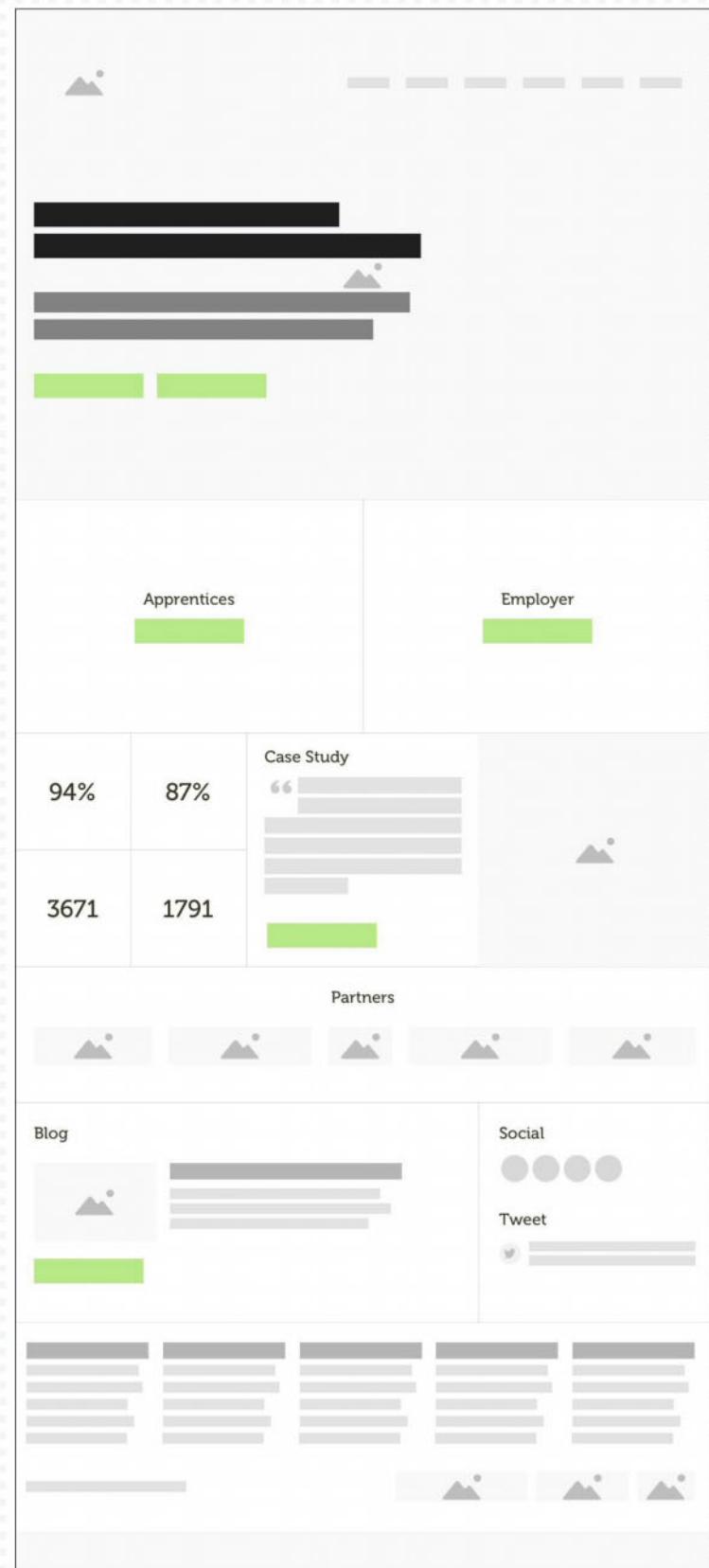
Short for Experience Design, this is a fairly new application from Adobe. You can do pretty much every part of the design process here, but the ability to create interactive prototypes is what makes it different from the other Creative Cloud applications. The interface is simple and intuitive, enabling you to link artboards together by simply dragging lines. Adobe has taken user feedback (including from the author of this article) and implemented requested features. The ability to connect a mobile device for testing designs in real-time is another feature that stands out. Adobe XD is an all-in-one UI/UX design tool that you'll definitely want to try if you haven't yet.

• InVision

This is a web application used to link together static designs. The interface is intuitive and the tools are simple but powerful. The collaboration tools enable multiple designers to update the designs and make changes. With powerful integration with Sketch updating prototypes is quick and easy. The presentation and communication tools offered by InVision enable clients and stakeholders to add comments and feedback directly onto the designs, while the development workflow – called Inspect – enables developers to get the information they need (fonts, colours, images, and so on) without ever having to open a design application.

• InVision Studio

The soon-to-be-released InVision Studio looks like it might actually be able to live up to its tagline – "The world's most powerful screen design tool". Much like Adobe XD, it's a tool that can facilitate the entire design process. The interface looks clean and in the demos and videos that InVision has been showing on its website, Studio looks fairly similar to Adobe XD too – albeit with the addition of animation and collaboration tools. If you're interested, you can request early access from invisionapp.com/studio. The first wave of invites have already been sent out.



Wireframe for the Baltic Training website homepage, created using Sketch

BREAKING DOWN DESIGNS FOR BUILD

Methodologies, tooling
and project structure



When it comes to building the project there are a number of methodologies, tools and project structures which can be very useful for helping you manage the codebase.

These methodologies are useful when working in teams. No matter which member(s) of the team wrote the code it will be familiar to everyone else should they need to make changes or take over the project.

TOOLING

Both SASS and LESS enable you to write BEM code nested using the ampersand (&). This is a really nice way to quickly see all the elements belonging to any given block.

```
.block
...
&__element {
...
&--modifier {
...
}
}
```

The code above will then be compiled to:

```
.block { ... }
```

There is no right or wrong way to do this, you can use whichever methodology you choose or even a mix of a few, you just need to make sure everyone in the team is working to the same principles and guidelines.

METHODOLOGIES

There are a number of different methodologies available to you, or you could simply create your own. Three

of the most popular are BEM, OOCSS and SMACSS.

BEM

BEM or Block Element Modifier is, at its heart, a naming convention. It enables you to split the code into reusable components by scoping elements to a block. The naming convention works like this; blocks are the start of the class name:

.block

Elements are then scoped to a block using a double underscore:

.block__element

Then modifiers are used for smaller changes or additions, a different colour for example:

.block__elements--modifier

```
.block__element { ... }
.block__element--modifier {
... }
```

You can also use this to apply hover and pseudo classes:

```
.block {
...
&:hover,
&:focus,
&:active {
...
}
&::before {
...
}
}
```

This helps keep problems with specificity to a minimum as well as creating styles that can be reused throughout the project.

It's important to note that the BEM structure doesn't necessarily follow the exact structure of the DOM. If you have a block with a container element, for example, subsequent elements do not need to be scoped to the element but to the block:

```
<div class="block">
<div class="block__content">
<h1 class="block__title"></h1>
</div>
</div>
```

OOCSS

OOCSS stands for Object Orientated CSS, the two main principles of this methodology are:

- **Separate structure and skin**

This means that the entire structure of the website should be separate to the styles that apply visual and decorative properties.

- **Separate container and content**

This means "rarely use location-dependent styles". In other words, any given element should look the same no matter where it appears on a site.

SMACSS

SMACSS stands for Scalable and Modular Architecture for CSS. There is nothing to download and include in your project – it's more of a style guide, a set of rules to help keep the specificity demons at bay. The architecture is complex, which is why some teams overlook it but there are a lot of useful rules to apply.

PROJECT STRUCTURE

Folder structure within your project is important, keeping everything organised can be hard work and take a lot of your time. Luckily, there are a couple of ways you can manage unruly projects. One architecture that's currently gaining in popularity is ITCSS (Inverted Triangle CSS). This is a

way to structure your files within your CSS. The styles at the top are the most far-reaching, whereas the files at the bottom of the triangle are the most specific. By implementing the same structure for all projects you can make every project, even those you haven't worked on, seem familiar.

CREATING LAYOUTS

Creating layouts is about using the right HTML and CSS layout modules

TABLES

Tables in HTML should be used for tabular data – elements that are presented in a table, a package comparison for example. In the early days of web design, tables were used in order to create complicated layouts using background images sliced to fit each table cell. These layouts are definitely a thing of the past now, even before we had the responsive web.

BLOCK

Understanding the box model is imperative when learning frontend development. Block and inline-block level elements are used to enable content to display next to each other or one after another. These properties can be used to create full grid systems like the one used in inuitcss and older versions of Bootstrap, but newer technologies like Flexbox and CSS Grid are now taking over for layout due to their responsive nature and powerful underlying layout algorithms.



inuitcss

FLOATS

The ‘float’ display property allows for content to flow around an object. This should be used for wrapping text around an image, in an article for example. Floats used to be used for layout when combined with the Clearfix workaround. The workaround would clear the next object in the DOM, enabling columned layouts to be created using less-than-elegant CSS.

USING CSS GRIDS

CSS Grid is perfect for creating 2-axis layouts (rows and columns). The syntax is simple and makes page layout a breeze. Layouts that would have required multiple nested containers can now be simply described in CSS.

Grid works differently to the other layout properties – since applying the ‘grid’ value to the ‘display’ property will affect any direct children. These elements are now grid items and will be positioned according to the rules you set on the parent (unless being specially placed).

GRID FEATURES

Grid brings a number of new features:

- **Grid gap**

Instead of using margins, grid has its own property for defining gaps between grid items, enabling the layout of the grid to be defined and unaffected by any gaps. Grid systems (like those found in inuit and Twitter Bootstrap) usually rely on setting negative margins in order to align everything correctly.

- **The Fr unit**

Short for fraction, this new unit of measurement is used to split the layout into fractions, with or without a gap.

- **Grid placement**

CSS Grid allows for any element to be prioritised and positioned anywhere on the grid before all other items, which are then positioned automatically.

- **Grid areas and templates**

Named areas on the grid are defined, which can then be referenced to define the layout in an almost ASCII fashion.

- **minmax()**

The ‘minmax()’ property value enables you to apply a minimum and maximum size for grid items, columns and rows. This feature is sadly missing from Flexbox and every other layout module in CSS.

BUILDING GRIDS

CSS Grid’s powerful areas and templates enable complicated layouts to be easily achieved. You start by applying a name to the elements within your grid using the ‘grid-area’ property; for example ‘header { grid-area: header; }’. Once you have named all of the areas in your grid you can then use the ‘grid-template-areas’ property to describe the layout.

grid-template-areas:

```
"header header header"
"left-sidebar article"
```

```
right-sidebar"
"footer footer footer"
";
```

The code example above describes a 3 x 3 grid; you simply repeat the name of the grid item if you want it to span multiple columns or rows. You can also use media-queries in order to change the grid layout for different-sized screens.

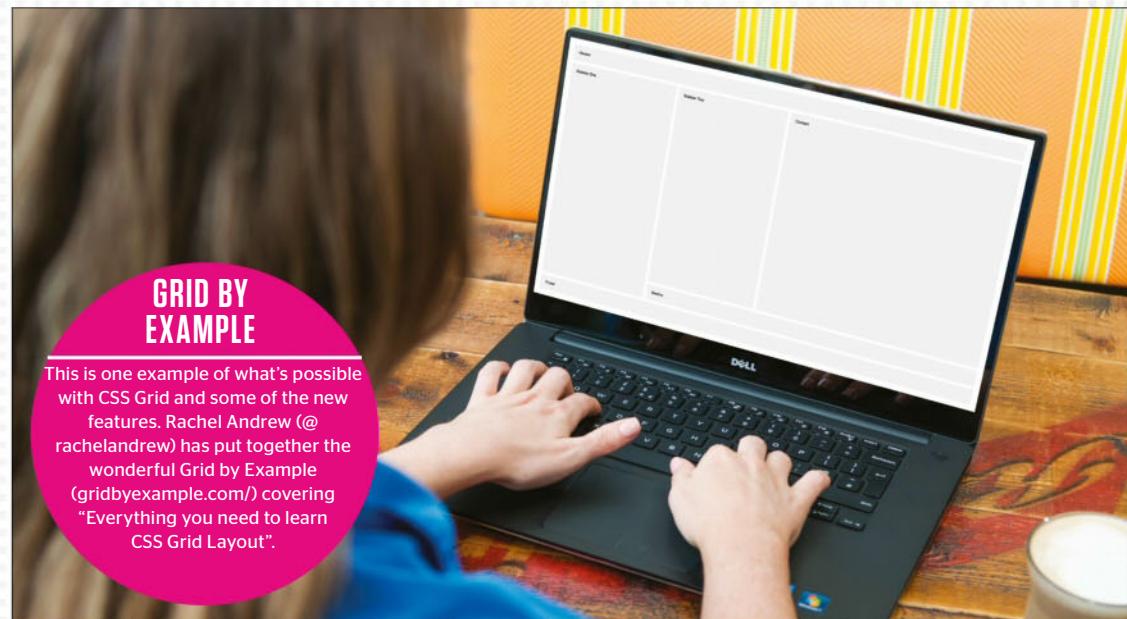
You can then use the new ‘fr’ unit to define how the width is split among the different columns:

```
grid-template-columns: 1fr 3fr
1fr;
```

The example shown above will split the available width into five equal width values, in other words it will apply one-fifth to the first column, three-fifths to the second column and the final fifth to the final column.

In order to size the rows you can use the new ‘minmax’ property value to size the top and bottom columns to their content and then allow the middle row to span the remaining space in the grid.

```
grid-template-rows:
  minmax(min-content, max-
content)
auto
  minmax(min-content, max-
content) ;
```



FLEXBOX

Perfect for single-axis layouts

Flexbox is great for creating single-axis layouts, either vertically or horizontally. Supported by the browser's powerful layout algorithm, spacing and sizing elements along an axis has never been more satisfying.

The syntax can be confusing sometimes and there is a lot to learn but once you are familiar with this new way of aligning items you'll wish it had always been this way.

Flexbox shares some similarities with CSS Grid in that applying the value of 'flex' or 'inline-flex' to the 'display' property will have an effect on direct children in the DOM.

When Flexbox is applied the browser will align all of the direct children, or flex-items as they are called. By default these items will be placed in a row next to each other and each item will take up as much space as it needs, something that's either

dictated by its content, or by setting a width on each item.

FLEX-GROW

The 'flex-grow' property enables you to apply the leftover width in the container once the items have been aligned according to their content. By setting the flex-items grow property to the value of '1' the remaining space in the container will be shared among the items equally. *See fig.1*

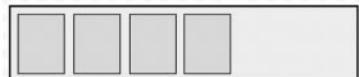
If we set the 'flex-grow' property on the last flex-item to the value of '2' then the leftover space will not be split equally, the last item will be given twice as much of the left over space in the container. *See fig.2*

FLEX-SHRINK

The 'flex-shrink' property does the opposite when the container is made smaller than its items' widths. If you set

the 'flex-shrink' property on all the items to the value of '1' they will be reduced in size equally. *See fig.3*

If you set the last item to '2' the width will be reduced in size twice as much as the other flex-items. *See fig.4*



Default – flex-items – (flex-grow: 0; flex-shrink: 0;)



Fig.1 – flex-items – flex-grow: 1;



Fig.2 – flex-items – flex-grow: 2; for the last item, 1 for the others



Fig.3 – flex-items – flex-shrink: 1;



Fig.4 – flex-items – flex-shrink: 2; for the last item, 1 for the others

WHEN FLEXBOX IS APPLIED THE BROWSER WILL ALIGN ALL OF THE DIRECT CHILDREN OR FLEX-ITEMS AS THEY ARE CALLED

FRAMEWORKS AND GRID SYSTEMS

Powerful battle-tested systems

You can save yourself hours of time in development since frameworks have been battle-tested by a number of developers and users.

However, they can also add an unnecessary amount of code bloat to the project. When using a framework it is worth spending time choosing only the elements that you want. This will help keep down the size of the files to be used.

However, thanks to the widespread adoption of CSS Grid and Flexbox these systems are starting to fall out of favour. These frameworks usually offer much more than just a grid system, but since we're discussing layout we'll concentrate on the most popular frameworks grid system.

BOOTSTRAP

The Twitter Bootstrap grid works much like old design grids – they apply 12 columns onto the container and you (as the developer of the project you're working on) choose how many of these columns any given element should span.

The grid itself is fully responsive and built using Flexbox. Grids are defined

by using a series of containers, rows and columns.

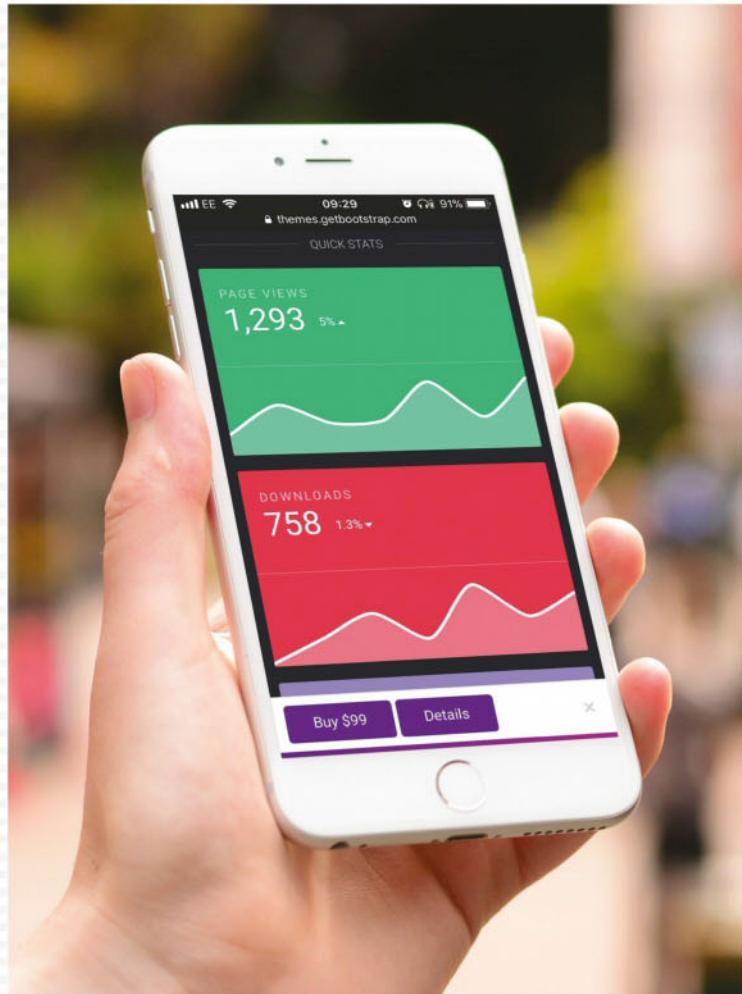
The number of columns an element spans at any given screen size is dictated by part of the class name. While this is a powerful grid system it is very rigid, the breakpoints for the grid are always the same and the syntax isn't very intuitive for someone who is new to the system.

HOW DOES IT WORK?

```
<div class="col-12 col-md-9 col-xl-8">
```

The element shown in the code above will span 12 columns by default and then on medium-sized screens span 9 columns and then on extra-large screens span 8 columns. The screen sizes are defined in Bootstrap and if you are just including Bootstrap as an external resource it is very difficult to update these values.

While these grid systems and frameworks can be good for beginners it's worth noting that unless you know how all of this works under the hood you're cheating yourself out of learning the fundamentals of CSS and relying on someone else's code.



WHAT NEXT FOR LAYOUTS?

Standard layouts are still the staple diet of many sites, but new specifications can mean new direction for layouts. Here we check out what's happening in the world of layouts with expert insight and the resources to help get you to the next level



STEVEN ROBETS

Web designer and developer

[codepen.io/
matchboxhero](http://codepen.io/matchboxhero)

"CSS GRID'S NEW TEMPLATING ENGINE ALLOWS FOR QUICK AND EASY EXPERIMENTATION WITH LAYOUT AND DESIGN. BY SIMPLY CHANGING THE TEMPLATE WE CAN RE-ORDER THE CONTENT."

"CSS REGIONS ALLOW FOR ELEMENTS TO BE PLACED INSIDE OF FLOWING CONTENT, THIS ALLOWS ELEMENTS BELOW THE FOLD ON MOBILE TO BE SHOWN AS PART OF THE CONTENT, ABOVE THE FOLD."

"CSS SHAPES ALLOW US TO WRAP TEXT AROUND A SHAPE BRINGING YET MORE DESIGN ELEMENTS FROM THE PRINT WORLD OVER TO THE WEB"

4 MUST-READ RESOURCES

1 THE EXPERIMENTAL LAYOUT LAB OF JEN SIMMONS

<http://labs.jensimmons.com/index.html>

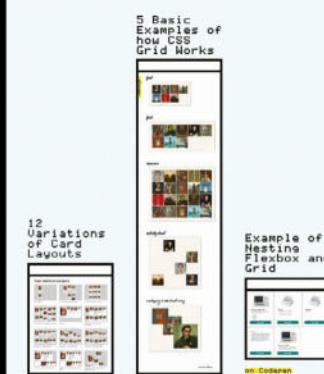
Jen Simmon site provides a host of experiments that reveal what can be done CSS Grid. It's not just CSS Grid that is included take a look at Whitespace, Flexbox & more.

3 CSS GRID LAYOUT AND ACCESSIBILITY

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

Accessibility is an often forgotten aspect of layout, but it shouldn't be. Check out this resource from Mozilla. It asks and answers how users should approach accessibility with grid layouts.

Intro to CSS Grid



2 A COMPLETE GUIDE TO GRID

<http://bit.ly/1UMOgSd>

As the title suggests this is an extensive look at CSS Grid. It introduces the basics, browser support and terminology before going in-depth on the specifics. Make sure you bookmark this as it is a great reference point.

4 EXPERIMENTAL 3D ROOM EXHIBITION LAYOUT

<http://bit.ly/2n6qp05>

This one is a bit more out there. It is an experimental 3D layout that uses CSS 3D transforms and CSS Flexbox to create its layout. View the demo at <https://tympanus.net/Development/Exhibition>

CLOSE UP WITH NEW CSS SPECS

We know that CSS Grid is taking layouts to new levels, but what else does the W3 have in the pipeline? Find out more

CSS REGIONS

<http://bit.ly/2n6D6cE>

The CSS Regions module, a working draft, allows content from multiple elements to flow through multiple boxes.

CSS SHAPES

<http://bit.ly/2DB6r8L>

This specification allows the use of geometric shapes in designs. Currently supported in Chrome and Safari but not Firefox.

CSS BOX ALIGNMENT

<http://bit.ly/2BqrtS1>

A working draft this module relates to the alignment of boxes within their containers in various CSS layout models.

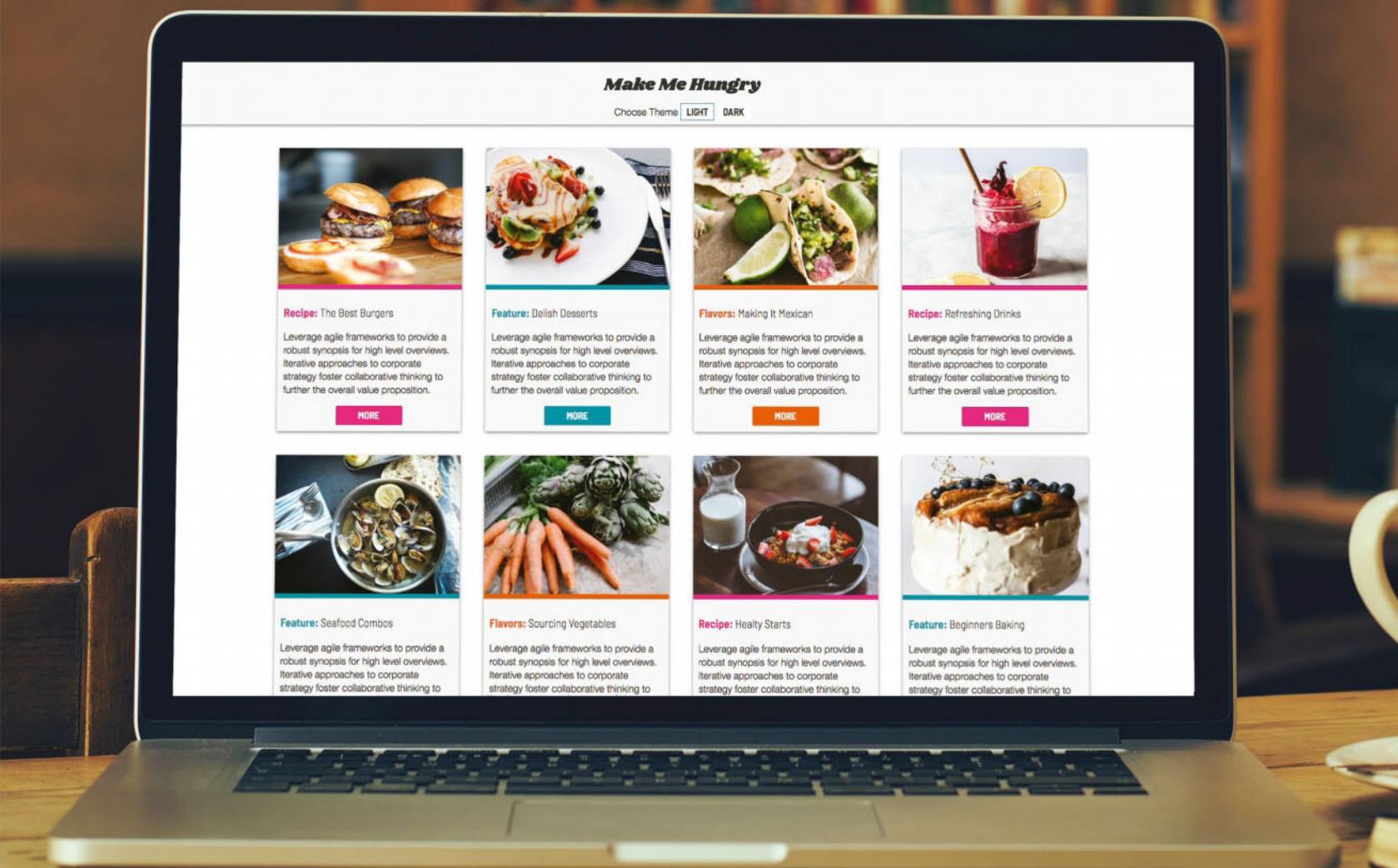
CSS EXCLUSIONS

<http://bit.ly/2F9UR18>

Currently a working draft this specification defines arbitrary areas around which inline content can flow.

Shrink your code with CSS4 variables

Change a variable and every rule that depends on that is updated, giving smaller style sheets and a more efficient way to experiment with your design





Web designers have had access to variables for some time when using CSS preprocessors to create style sheets. Now with major

browser support, CSS4 variables are poised to take the power of CSS preprocessors and place this natively in the browser for designers to harness.

CSS preprocessors work by compiling their variables down to regular CSS. The benefit of having a variable in the browser is not part of what they offer, and that's where CSS4 variables have the advantage over their preprocessor counterparts. It's completely possible to change the value of variable by the user at any time.

In this tutorial, a few different use cases are going to be explored. The first just being for the designer to change the colour in one variable and see every instance of it updated to keep consistent colour. This is a great feature because it's easy to experiment with ideas more. The second will be to control the widths of elements and just update the width in media queries for different sized screens, meaning the only change is a variable. Finally, the third way to use it will be to create two themes and let the user switch between them.

Here, three variables can be changed that completely updates the design on display in the web browser.

1. Declaring CSS variables

Open the 'start' folder in your code IDE and launch the 'style.css' file for editing. The first task to perform is to declare some CSS variables that will control some of the colours within our design, which will make it very easy for a designer to experiment with colours. The following lines declare three variables to the whole page.

```
:root {
  --primary: #ff368b;
  --secondary: #0098ac;
  --tertiary: #F06D06;
}
```

2. Applying the colours

Applying these colours to elements is very easy. In your code find the 'info.primary' class, which should be on line 81. Replace the existing border-top colour with the variable as shown here. Repeat the process for the class 'info.secondary'. This time the variable for the secondary colour is used instead.

```
.info.primary {
  border-top: 8px solid var(--primary);
}

.info.secondary {
  border-top: 8px solid var(--secondary);
}
```

3. Different colours

Now the tertiary colour is added to the 'info.tertiary' class. That has now used all three colours controlled by a variable. At present there's no benefit to using this but it's possible to use those colours in other contexts such as the background colour in links - as shown here.

```
.info.tertiary {
  border-top: 8px solid var(--tertiary);
}

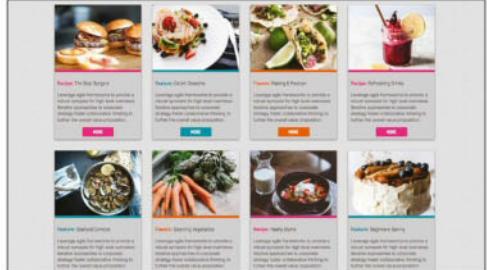
.a.primary {
  background: var(--primary);
}
```

4. Linked in

Now the remaining secondary and tertiary links are also styled with the relevant background colours by replacing their existing colour formatting with the colours from the variables. Now is a good time to save this file and just make sure your browser is supporting CSS variables. Do this by opening the 'index.html' page and checking the colours are there.

```
a.secondary {
  background: var(--secondary);
}

a.tertiary {
  background: var(--tertiary); }
```



5. Text colours

Let's target the text colour now by finding the class 'span.primary'. Again this class is given the appropriate primary variable. As you can see, when colour is being used consistently to theme up elements, controlling from one variable is incredibly powerful.

```
span.primary {
  font-weight: 600;
  color: var(--primary); }
```

6. Outlining the benefit

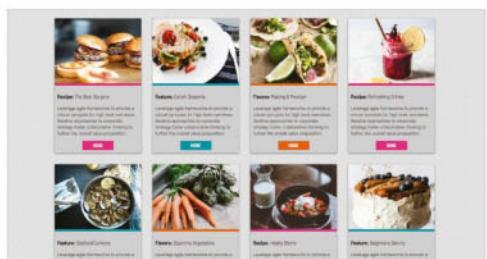
Add the variable to the code shown here and then save your CSS. If you return to the code added in step one and change the colours, you will see it consistently change with the elements that have been styled up that way.

```
span.secondary {
  font-weight: 600;
  color: var(--secondary); }

span.tertiary {
  font-weight: 600;
  color: var(--tertiary); }
```

Declaring in the root

By declaring variables in the root, they are globally accessible by any object. If you declared a variable in a style applied to an element, only itself or children would be able to access it.



Left

The start of the project has a finished website design, but the tutorial will control many of the colours and widths using CSS variables

Top

Controlling multiple elements by the same colour variables mean that it is easy to update colours in a website when it has otherwise needed a CSS preprocessor

Tutorials

Shrink your code with CSS4 variables

7. Going responsive

At present the design is not responsive so let's set up a way to do that and use a CSS variable to control the width of the card layout. This means all that needs to be updated in media queries is the width variable. Add the width variable to the code you had in step 1.

```
--width: 25%;
```

8. Boxing clever

Now find the 'box' CSS class, where the width is being calculated from 25% minus 40 pixels. Remove the '25%' and replace it with the width variable. This can now be controlled dynamically by changing the width CSS variable.

```
.box {  
    float: left;  
    width: calc(var(--width) - 40px);  
    margin: 20px;  
    box-shadow: 0 2px 8px rgba(0, 0, 0, 0.5);  
    background: #ccc;  
    color: #666;  
}
```

9. Break point

Placing a media query into your CSS at the end of the CSS gives us the kind of mid-sized screen that is often found on tablets. Here the root section of CSS is updated to have the width set to 50% and therefore give us a two-column card layout that would better suit that sized screen. Resize your browser to see it switch.

```
@media (max-width: 960px) {  
    :root {  
        --width: 50%;  
    } }
```

Changing via JavaScript

It's possible to update the CSS variables via JS as a quick way to update many items at once. It shows off the runtime power of CSS variables.

Right

By controlling the width property using a CSS variable, the variable is the only thing that needs updating when making a break point media query for tablet screens

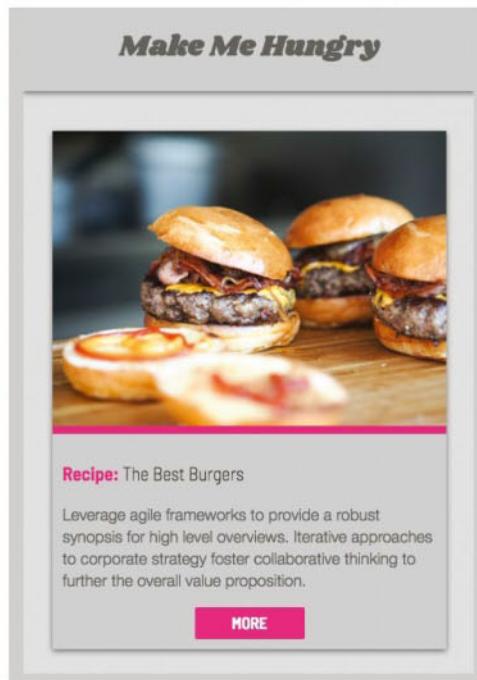
Farright

Dynamically changing the variables can be easily achieved, and here the user can switch between a light theme and dark theme. Only three CSS variables are updated to change the entire theme

10. Mobile friendly

The final breakpoint will be for devices that are less than 500 pixels, such as mobile screen sizes. Here the width variable is updated to take full advantage of the full width of the display in order to be able to see the card content. Again save this and resize your browser to see the display update when a smaller screen size is used.

```
@media (max-width: 500px) {  
    :root {  
        --width: 100%;  
    } }
```



11. Creating colour themes

The next part of the tutorial will focus on using JavaScript to dynamically alter the CSS variables to create different colour themes for the page. To get started some new variables need to be declared in the 'root' of the CSS...

```
--main-bg: #fff;
```

```
--main-type: #333;  
--alt-bg: #f5f5f5;
```

12. Changing to variables

Now various elements will start to get these variables added in order to control the theme of the page. Find the 'main-header' class and change the colour and background colour to be controlled via variables added in the previous step.

```
.main-header {  
    color: var(--main-type);  
    background-color: var(--alt-bg);  
    text-align: center;  
    padding: 15px 0 10px 0;  
    box-shadow: 0 2px 8px rgba(0, 0, 0, 0.5);  
}
```

13. Styling buttons

Find the 'button' class. This time the variable will be the main background colour, whereas the header used an alternative background colour. By having two different background colours it allows the content to have some contrast as they are viewed. Controlling the text colour means that a high contrast text can be added.

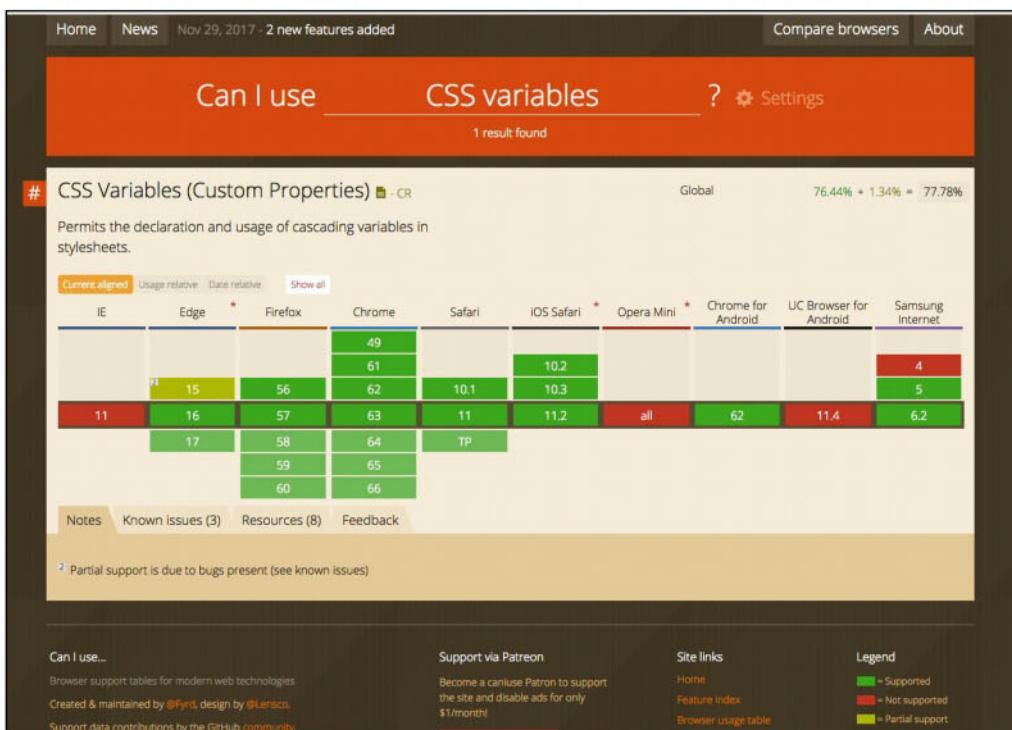
```
.button {  
    text-decoration: none;  
    padding: 5px 10px;  
    color: var(--main-type);  
    background-color: var(--main-bg);  
    font-family: 'Barlow Semi Condensed', sans-serif;  
    font-weight: 600;  
}
```

14. Footing the bill

Now move to the class that controls the footer - 'main-footer'. As with the header this will also have an alternate background colour so that it will stand out from the remainder of the page. The colour is also changed so that it will be visible.

```
.main-footer {  
    clear: both;
```





Is it safe to use CSS variables?

As with any relatively new web technology the question every web designer wants to know is: "what's the support like?". The great news if you visit www.canuse.com is that currently most of the major browsers support this, the exceptions being IE, Opera Mini and UC Browser for Android.

So all in all this has very good support from the main desktop and mobile browsers at just under 79% browser support globally. As always whether you can use this depends on who your audience is.

If you need to know more about CSS variables then the Mozilla Developer Network has good documentation on this, and it's easy to understand (https://developer.mozilla.org/en-US/docs/Web/CSS/Using_CSS_variables). A more in-depth look at CSS variables, featuring a good description of why you would want to use them can be found on Medium (<https://medium.com/dev-channel/css-variables-no-really-76f8c91bd34e>).

```
padding: 20px;
text-align: center;
background-color: var(--alt-bg);
color: var(--main-type);
```

15. Body building

The next part of the page to be targeted is the body element, so find the appropriate CSS that controls the look of this. The background will now be controlled by the main background CSS variable to give global control to the variables.

```
body {
margin: 0;
padding: 0;
font-family: "Helvetica Neue-Light",
"Helvetica Neue Light", "Helvetica Neue",
Helvetica, Arial, "Lucida Grande", sans-serif;
font-weight: 300;
line-height: 1.4em;
background: var(--main-bg);}
```

16. Box fresh

The 'box' CSS is going to be updated now, so find the code for this and change both the background colour and text colour to have control from CSS variables. Save this CSS page and just check the page has updated in the browser. Now move into the 'index.html' page.

```
.box {
float: left;
width: calc(var(--width) - 40px);
margin: 20px;
box-shadow: 0 2px 8px rgba(0, 0, 0, 0.5);}
```

```
background: var(--alt-bg);
color: var(--main-type);}
```

17. Adding buttons

In the 'index.html' page find the main header section and just after the heading two tag, some content will be added. Just to help you locate this in the code we've shown where the code will go with the HTML comment in this step. Once you've found it add the code in the next step.

```
<div class="main-header">
<h2 class="title">Make Me Hungry</h2>
<!--Code Goes Here-->
</div>
```

18. Span and chips

The span is added to the page. This gives the instruction to change the theme. Two buttons are placed after this, and they will talk to the JavaScript to update the CSS variables so that they switch between light and dark.

```
<span>Choose Theme <a href="#" class="button" onclick="light();">LIGHT</a>
<a href="#" class="button" onclick="dark();">DARK</a></span>
```

19. Stick to the script

The code here should be added right before the final body tag of the page. The code in the remaining two steps will be placed inside these script tags. This function will be used to set a specific CSS variable with a value dynamically. Change it at run time via user control.

```
<script>
function setDocumentVariable(propertyName,
value) {
document.documentElement.style.
```

```
setProperty(propertyName, value);
};</script>
```

20. Going dark

When the user clicks the 'dark' button the function here will be called. As you can see it sets three variables with new values and these update the page immediately, changing the theme. Save this and try the button in your browser to see the theme change.

```
function dark() {
setDocumentVariable('--main-bg',
'#918d82');
setDocumentVariable('--alt-bg',
'#57534c');
setDocumentVariable('--main-type',
'#ddd');}
```

21. Seeing the light

The final step is to add the functionality for the 'light' button. This switches the colours back to the regular light colours of the page. Save your work and refresh your browser to see the update take effect. Now you can switch between themes just by updating three CSS variables that, in turn, control many other elements on the page.

```
function light() {
setDocumentVariable('--main-bg', '#fff');
setDocumentVariable('--alt-bg',
'#f5f5f5');
setDocumentVariable('--main-type',
'#333');}
```

Most sites you visit on the web are probably generated dynamically – that is, rather than having all of their content encoded into the HTML stored on the server, they retrieve content from a database and construct pages to serve to you on the fly. Many also provide user interactivity through logins, forms and so on. Think of Facebook for an example of both of these things. To build this type of functionality yourself, you might look to a CMS such as WordPress.

For some use cases, however, this is overkill. Simple pages, such as a personal profile, information about a business, or even a blog really don't need this sort of overhead or complexity. You could of course just build static HTML yourself in a text editor, and indeed, many years ago this was how all sites were built, but

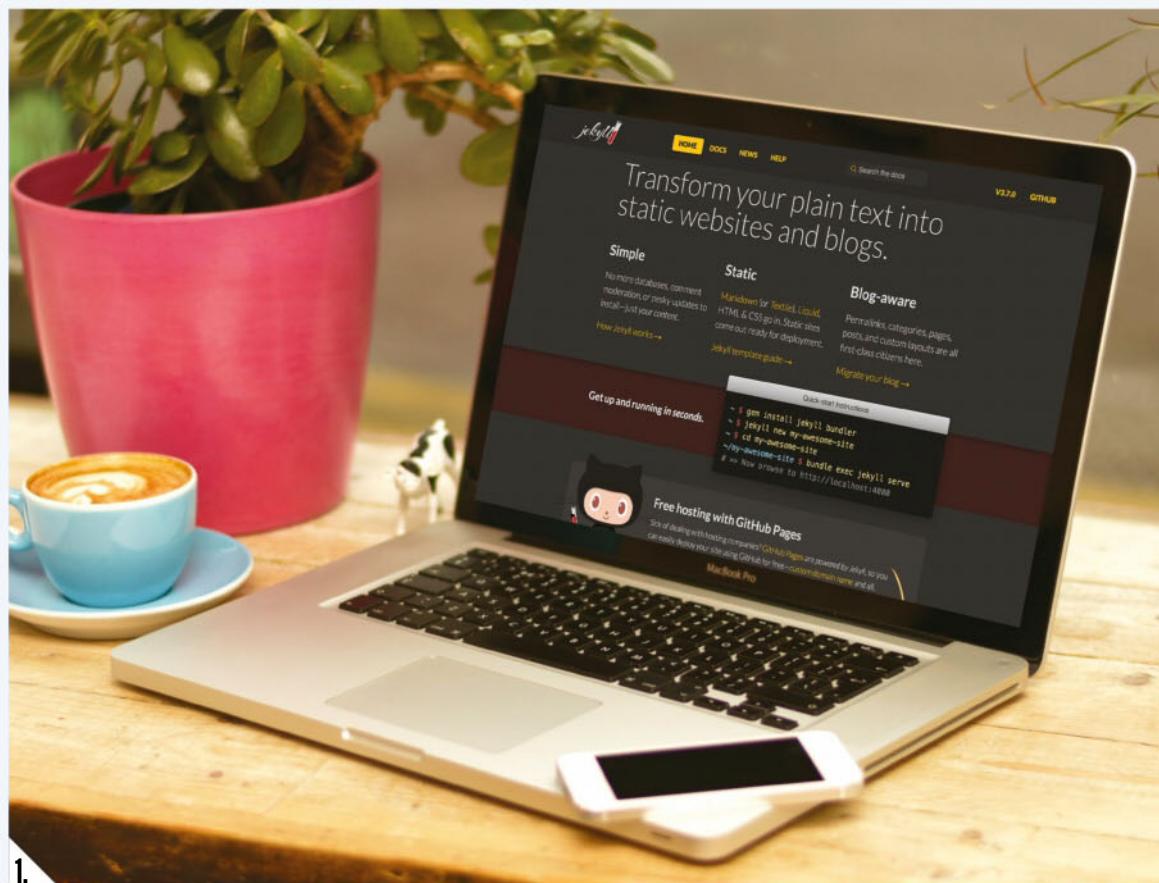
"PERFORMANCE WILL BE GREATLY IMPROVED"

this rapidly becomes cumbersome to maintain when you want to scale up or make changes. Static site generators offer a solution to this, by enabling you to build static HTML pages using templates.

Essentially, static site generators are command-line tools which shift the creation of the final HTML page forward from the point the user requests it to the point you write the content. When you make an update, you build the new page which can then be served as-is to every user who requests them. This offers several advantages. Performance will be greatly improved compared to a dynamic site, since serving static HTML and CSS has a very low footprint. Your server-side setup will be much simpler, which also means fewer security worries. Conversely, however, you'll lose the opportunity to deliver real-time content or receive user input.

Static site generators have exploded in popularity in recent years, so navigating the wide range of choice can be difficult. Here, we've taken a look at some of the best options you should be considering.

10TH JEWEL



1.

STATIC SITE GENERATORS

DO YOU NEED THE COMPLEXITY OF A DYNAMIC SITE?

1. JEKYLL

<https://jekyllrb.com>

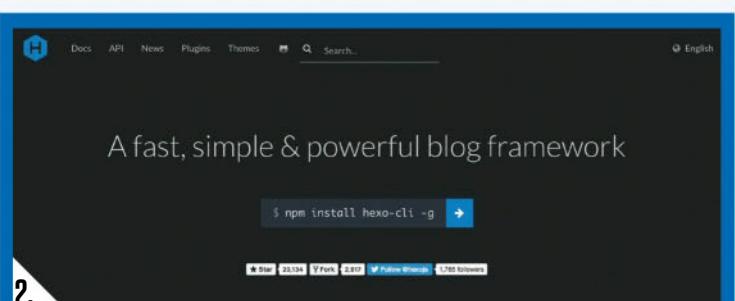
PRO: Jekyll is one of the oldest static site generators out there, and still the most popular

First released in 2008 by Tom Preston-Werner, the co-founder of GitHub, Jekyll arguably popularised the concept of static sites and remains probably the most widely used static site generator. With Jekyll, you'll typically work with content in Markdown, a lightweight markup language designed for text formatting. The Liquid templating engine is used to place this Markdown content into a HTML template, and to combine templates representing various parts of a page (say, header, footer and content) in a modular and re-usable manner. Support for Sass is built in for those with a preference for CSS preprocessing, and it'll play fine with libraries like Bootstrap. Also included with Jekyll is an HTTP server which can be used to easily deploy and test your static pages locally.

One of Jekyll's key selling points is its wide range of 'importers', which enable an existing site to be migrated to Jekyll with relative ease. If you have a WordPress site, for example, you can switch to using Jekyll using one of the importers. It's also trivial to convert existing static HTML sites to Jekyll, which can be great if you've been coding static HTML yourself or see a template you like the look of.

Notably, Jekyll is used to power GitHub pages, a static site hosting service which is provided with GitHub. If you have a GitHub repository, you're able to create a GitHub pages site for free using Jekyll. This can be a convenient way to give a polished landing page to your GitHub project.

The big downside of Jekyll - and this applies to most generators - is that it can seem complex at first and is a new technology to master. You might not be up and running as quickly as with a CMS. However, it's very well documented and the learning curve is quickly overcome.



2. HEXO

<https://hexo.io>

PRO: Focused on blogging, Hexo is fast to get started and simple to use, especially for JavaScript developers

Hexo is powered by Node.js and aimed at blogging. While the JavaScript implementation shouldn't in theory make too much difference to how you work with it, since you'll be using with markup and templating languages, it can make installation and configuration more familiar for JavaScript developers. If you're already using

npm and git then it's extremely simple to get up and running.

Like many other general purpose generators, you'll probably want to write your content in Markdown. The default templating engine is Swig, which is once again well suited for JavaScript developers. However, Hexo is extensible to allow other templating engines to be used if you want.

The Hexo website includes a range of pre-built themes for you to try out, and one especially popular feature of the tool is its support for single-command deployment.

3. GATSBY

www.gatsbyjs.org

PRO: Gatsby uses some of the latest web technologies such as React to create static PWAs

Like Hexo, Gatsby is powered by Node.js and so will be more familiar territory for experienced JavaScript developers. However, several things set it apart from other similar tools.

Gatsby uses React, which means that everything is built with components, and allows it to benefit from React's approach to DOM rendering. This means it can be

picked up quickly by developers who have worked with React, but for those unfamiliar with it, learning React will be necessary.

It also utilises GraphQL at build time to retrieve data and content from data sources, which provides a modern, consistent interface to ensure that each page receives the exact data needed when it is built.

Finally, Gatsby will build pages as progressive single page apps, meaning that the entire site is downloaded and subsequent navigation is immediate.

The Gatsby homepage features a dark blue header with the Gatsby logo and navigation links. Below the header, there's a large purple diagonal graphic. The main content area has several sections with headings and descriptions. At the bottom, there's a large '3.' followed by a list of six features: 'Blazing-fast static site generator for React', 'Modern web tech without the headache', 'Bring your own data', 'Future-proof your website', 'Static Progressive Web Apps', and 'Speed past the competition'. Logos for Fabric and Segment are visible in the bottom right corner.

HUGO

The world's fastest framework for building websites

Hugo is one of the most popular open-source static site generators. With its amazing speed and flexibility, Hugo makes building websites fun again.

[Quick Start](#)

4.

4. HUGO

<https://gohugo.io>

PRO: Aims to be the fastest static site generator. Hugo is the other big hitter alongside Jekyll

Widely regarded as the 'other' leading static site generator, it's only natural to compare Hugo with Jekyll. Hugo is the newer of the two, and one of its key focus areas is speed, which for some has been a complaint with Jekyll. Nobody likes waiting for a site to build, and Hugo can put together a simple site from your markup and templates in milliseconds, or even blaze through thousands of pages in seconds.

With Hugo it's also typical to write content with Markdown, and the templating engine is based on Go templates since Hugo itself is implemented in the Go programming language. Like Jekyll, it ships with a lightweight HTTP server to quickly serve your pages locally. The two tools' build workflows are overall fairly similar.

For many, the key benefit of Hugo is its quicker, simpler path to getting started, with very little need for configuration and no dependencies other than the core binary. Its documentation and tutorials are very

good, and it has an ethos of maintaining simplicity which makes for a very approachable learning curve.

One disadvantage Hugo has relative to Jekyll is that it lacks the extensive plugin ecosystem available for the latter. Given its wide range of built-in functionality, however, this is unlikely to be a problem for most users. It also may be marginally lighter on support on sites such as StackOverflow since it hasn't been around as long; however, it continues to grow in popularity and many believe it may become the leading static site generator in the near future.

"HUGO CAN PUT TOGETHER A SIMPLE SITE FROM YOUR MARKUP AND TEMPLATES IN MILLISECONDS"

5. NUXT

<https://nuxtjs.org>

PRO: Use NUXT to take advantage of Vue.js for your static sites

Vue.js has gained immense popularity as a front-end framework in recent years, due to its combination of a gentle learning curve, high performance and powerful feature set. Nuxt.js is actually a framework for creating server-rendered Vue applications – that is, dynamic pages which are rendered by the server

before being passed in their completed form to the client to display. However, it can also be used to build static sites, with a command line parameter that will build static HTML pages for all routes through a Vue project.

Since Nuxt is a Vue framework, familiarity with Vue will be necessary to use it, but developers who have worked with Vue before will feel right at home. And, like Vue, it also brings a component-based approach to development of your sites.

5.

6. MKDOCS

www.mkdocs.org

PRO: MkDocs might well be the easiest way to create websites for project documentation

MkDocs is more specialised than the tools we've looked at up until now. It sets out with one very simple objective in mind: to provide a fast and easy-to-use way to generate HTML sites for project documentation. And it absolutely succeeds at doing this. It's built in Python, so you'll need Python and pip (Python package manager) installed to get it.

After that, getting started is easy. Documentation is written in Markdown, and the tool is configured using a single YAML file. The command-line build process is as simple as it could be.

The MkDocs GitHub page includes a number of themes which are specially designed for documentation, or you can build HTML for your own theme.

MkDocs is one of the easiest static site generators to get started with, and if documentation is your use case, there really isn't much cause to look at other tools.

6.

7. PELICAN

<http://docs.getpelican.com>

PRO: Pelican is a highly popular static blog generator built in Python

Pelican supports content written in multiple languages, including the ubiquitous Markdown. It uses the Jinja templating engine, which is both easy to use and extremely powerful, meaning that while Pelican is primarily optimised for building blogs, it is also well suited to building a wide range of different types of other sites. It's also fast, and can comfortably handle sites with thousands of pages without making you wait for them to build.

Like many of the leading static site generators, Pelican allows you to import your site from a variety of

blogging platforms. This makes it trivial to convert an existing site built with WordPress or many other popular content management systems. If you're familiar with Python, and especially if you've used

Jinja templates before, Pelican is a very safe choice.

Conversely, however, it may be a little harder to get to grips with for developers who are more familiar with JavaScript or Ruby.

" WHILE PELICAN IS PRIMARILY OPTIMISED FOR BUILDING BLOGS, IT IS ALSO WELL-SUITED TO BUILDING A WIDE RANGE OF DIFFERENT TYPES OF OTHER SITES "

METALSMITH

An extremely simple, pluggable static site generator.

8.

8. METALSMITH

www.metalsmith.io

PRO: Metalsmith is an extremely simple tool where all functionality is provided by plugins

Metalsmith takes a different approach to many other static site generation tools, in that it doesn't try to do very much at all. Essentially, static site generators take a set of source files, manipulate them, and then generate a set of output files which is the static site itself. Metalsmith provides a

framework for doing this, but leaves all of the actual manipulations to plugins. These manipulations are things which typically come out of the box with other static site generators, such as utilising templates, substituting variables, or interpreting languages like Markdown.

When run through Metalsmith, all source files are converted to JavaScript objects, which means that manipulations by plugins are essentially modifications to properties

"METALSMITH IS IMMENSELY CUSTOMISABLE, BUT REQUIRES A LITTLE MORE CONSIDERATION DURING SETUP"

of these JavaScript objects. There is, for example, a `markdown()` plugin which transpires Markdown to HTML.

The result of this approach is that Metalsmith is immensely customisable, but requires a little more consideration during setup than some of the more monolithic tools. Don't like Jekyll's use of the Liquid templating engine? Here you can pick your own. On the Metalsmith page, you'll find a lengthy list of plugins to provide a wide range of functionalities, ranging from compiling Sass to CSS through to computing a word count.

If you have a preference for unopinionated frameworks, Metalsmith is about as unopinionated as you can get.

HONOURABLE MENTIONS

THERE ARE MANY MORE STATIC SITE GENERATORS OUT THERE. HERE ARE A FEW MORE WHICH ARE WORTH A LOOK

OCTOPRESS

A blogging framework for hackers.



Octopress 3.0 Is Coming

The way Octopress is being distributed and maintained is changing its end. There are many things I've always disliked about how Octopress works. So before I talk about the exciting part, I'd like to tell you what's wrong with Octopress.

What's wrong?

If I'm being harsh, Octopress is basically someone's Jekyll blog you can fork and modify. The first, and most obvious flaw, is that Octopress is distributed through GitHub. I want to push through a wall when I think about users wanting with merge conflicts from updating their sites. It's absurd.

Octopress is released as a single product, but it's a collection of plugins and configurations which are hard to disentangle. If you want to change or remove anything you're leaving the "golden path" and code will be broken if you make changes, absent of drama and errors. There's no room to move when Octopress is released in

OCTOPRESS <http://octopress.org>

Octopress is a framework for Jekyll which tailors specifically for blogging, and bundles a number of commonly-used plugins. If you like Jekyll's style, it may take some of the setup work off your hands.



PUBLII <https://getpublii.com>

If you don't want the trouble of setting up a command-line tool, Publii is a desktop application with GUI designed to develop static sites. Think of it like a CMS that sits on your local machine rather than a server.

HAKYLL

OVERVIEW

Static sites are fast, secure, easy to deploy, and manageable using version control.

Hakyll is a Haskell library for generating static sites, mostly aimed at small-to-medium sites.

It's built on top of Haskew, a tiny, fast, and very configurable way to serve files across the web.

It's also designed to be easy to extend and integrate with existing Haskell applications.

It's a great way to get started with Haskell and static sites.



HAKYLL <https://jaspervdj.be/hakyll/index.html>

Someone made a Haskell library for static site generation? It might not be that widely used, but fans of Haskell (and there are many around) will love this.



Build static websites with an easy-to-use framework

Middleman is a static site generator using all the shortcuts and tools in modern web development.

9.

9. MIDDLEMAN

<https://middlemanapp.com>

PRO: Middleman has been around for a long time, and is designed with more complex sites in mind

Middleman was released around the same time as Jekyll, and will be most familiar to developers who have worked with Ruby on Rails. Its default template engine is ERB (Embedded RuBy) and it also includes built-in

support for Haml, Sass, SCSS and CoffeeScript, and can be extended further to support more.

Some leading static site generators are heavily geared towards blogs, but Middleman sets its ambitions wider and aims to provide the flexibility to develop any type of site. It's highly unopinionated and extensible. This means, if all you are doing is a blog, the setup is a little more complex since you'll have to configure it.



PRO: Spike is the spiritual successor to Roots, a popular static site generator that is no longer maintained

Spike is built by the same team as Roots, which saw great popularity while it was actively maintained. It provides a familiar ecosystem for JavaScript developers, utilising webpack, Postcss, Reshape and Babel. Much like Metalsmith, Spike is designed to provide a very simple framework and allow plugins to handle your transformations.

CREATE THE IMPOSSIBLE

www.photoshopcreative.co.uk

FREE! OVER 260 RESOURCES
✓ BRUSHES ✓ TEXTURES ✓ BACKGROUNDS ✓ STOCK IMAGES ✓ ACTIONS AND MORE

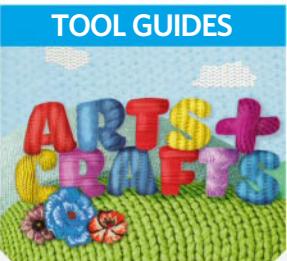
Photoshop CC File Edit Image Layer Type Select Filter 3D View Window Help



Photoshop® creative

ON SALE NOW

• Striking imagery • Step-by-step guides • Essential tutorials



Available
from all good
newsagents and
supermarkets

BUY YOUR ISSUE TODAY

Print edition available at www.myfavouritemagazines.co.uk

Digital edition available for iOS and Android



facebook.com/PhotoshopCreative



twitter.com/PshopCreative

Developer tutorials

Automate your workflow with Gulp

Learn how to remove the nuisance of repetitive and mundane tasks with the help of Gulp, a Node.js task runner



In today's web development world, building a modern, complex website which supports a variety of devices, both old and new, often constitutes a lot of work. Much of this work involves small, repeated tasks such as minifying JavaScript, optimising images, or compiling SASS.

While these types of tasks are useful, they also have to be performed again and again every time the code is changed. Understandably, this quickly grows repetitive, and eventually, it can be tempting to just ditch them altogether and spare yourself the headaches.

Thankfully, this is an issue we can solve. The solution comes in the form of Gulp, a popular task runner which enables us to automate these types of mundane, time-consuming tasks through the use of Node.js streams, leaving us free to focus on the important stuff.

In this tutorial, we're going to create a basic build system using Gulp, which compiles and optimises a small project and its assets. To get started, download the project files from FileSilo and open 'website-template' in your favourite text editor. Then in Terminal, CD into 'website-template' and run 'npm install' to install the initial project dependencies. Let's begin.

1. Install Gulp

We first need to install Gulp as a project dependency, and then its command line interface (gulp-cli), by running the commands below in Terminal. The latter will enable Terminal to recognise any commands beginning with 'gulp', making it possible for us to actually run the tasks we'll be building in this tutorial.

```
npm install gulp-cli -g
npm install gulp -D
```

2. Create gulpfile.js

Gulp searches for tasks in a JavaScript file called 'gulpfile.js'. Let's create one in the 'website-template' directory, and then in it, copy the contents of 'support-files/step-2.txt' from the FileSilo package. This gives us a basic structure so we can organise our code.

The first thing we need to do in the gulpfile is import Gulp itself as a dependency. This can be done by adding the code below under the "Set Dependencies" header. We'll use an ES6 const declaration for this and other imports since we won't be needing to reassign them.

```
const gulp = require('gulp');
```

3. Project directory structure

When building a gulpfile, it's worth remembering that we may want to reuse it in future projects. These projects may all have different directory structures, which means we would then have to go through each Gulp task and edit the paths accordingly. Let's spare our future-selves this inconvenience and declare our directory structure in reusable variables which can be easily changed. Add the declarations below under the Options header.

```
const dev = 'dev';
const dist = 'dist';
const htmlDev = `${dev}/html`;
const htmlDist = dist;
const imgDev = `${dev}/img`;
const imgDist = `${dist}/img`;
const sassDev = `${dev}/sass`;
const sassDist = `${dist}/css`;
const jsDev = `${dev}/js`;
const jsDist = `${dist}/js`;
const dataDev = `${dev}/data`;
```

4. Minify variable

Next, again under the Options header, create a new variable called 'minify' and set its value to 'true'. This will be used by some Gulp tasks to tell them if they should minify the compiled code.

What minification does is remove all unnecessary characters such as whitespace and line breaks from the code without changing any of its functionality. This in turn, reduces its filesize, making it load a little faster in our end user's browser.

```
const minify = true;
```

5. IF statements and file renaming

By having a minify boolean in our build system which can be changed, we thus have to account for two differing scenarios in our Gulp tasks: One where we want Gulp to minify our code; and one where we don't.

To help us define this logic, we will install two new dependencies. The first is called gulp-if, and enables us to integrate simple ternary operators into our Gulp streams. The second is called gulp-rename, and simply enables us to rename files. Run the command below in Terminal to install these:

```
npm install gulp-if gulp-rename -D
```

Then import them into our gulpfile by adding this code under the Set Dependencies header:

```
const rename = require('gulp-rename');
const gulpif = require('gulp-if');
```

6. Creating our first Gulp task

As our Gulp tasks run, files will be taken from the specified 'dev' directory, processed, and then deposited into the specified 'dist' directory. To make sure no leftover files from previous builds can break things, we should delete any existing 'dist' directory first. To do this, we'll build our very first Gulp task. We'll require another dependency called 'del', which can be installed using the command below.

```
npm install del -D
```

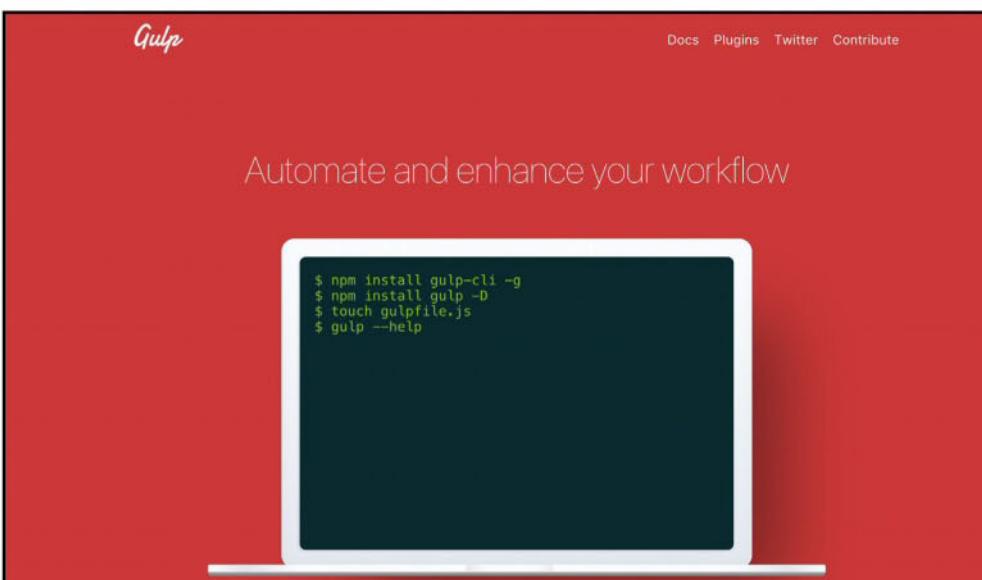
Then add it to our list of dependency imports.

```
const del = require('del');
```

Now, beneath the task's header, let's use the 'gulp' object we defined back in Step 2 and call its 'task()' method. This

It's just JavaScript

Remember, Gulp is running on JavaScript, so we can use all the usual suspects like conditional statements and loops to enhance our tasks. A very simple example is using 'console.log' to print feedback, but we'll also be using more complex 'map' functions soon enough.



```
$ gulp --help
  gulp [options] tasks
Options:
  --help, -h          Show this help.
  --version, -v        Print the global and local gulp versions.
  --require            Will require a module before running the gulpfile
                      useful for transpilers but also has other applica
  --gulpfile           Manually set path of gulpfile. Useful if you have
                      multiple gulpfiles. This will set the CWD to the gulpfile
                      well.
  --cwd               Manually set the CWD. The search for the gulpfile
                      the relativity of all requires will be from here.
  --verify             Will verify plugins referenced in project's packa
                      against the plugins blacklist.
  --tasks, -T          Print the task dependency tree for the loaded ou
```

Left

If you'd like to learn more about Gulp, you can visit its homepage at www.gulpjs.com.

Top

With 'gulp-cli' installed, you can also type 'gulp --help' in Terminal to see what other commands are available to you.

Developer tutorials

Automate your workflow with Gulp

method accepts two parameters, a task name and a function specifying what it should do. In the function, we then simply tell it to delete the directory we've assigned to the 'dist' variable using the 'del' function we've just imported. In the snippet, there's also a helpful emoticon-powered 'console.log' which will send feedback to the Terminal. The first of many.

```
gulp.task('deleteDist', function(){
  console.log('Deleting old \'${dist}\' directory ');
  return del(dist);
});
```

7. Compile SASS with Gulp - part 1

As mentioned, Gulp can be used with the preprocessor SASS to automate the compiling of SCSS files into CSS. To do this, we'll be installing more dependencies. 'gulp-sass' is a gulp wrapper for node-sass. 'gulp-sass-glob' enables us to use globbing patterns in our SASS to import many files at once using wildcard characters (See '/dev/sass/main.scss'). Finally, 'gulp-autoprefixer' adds vendor prefixes to the compiled CSS based on browser usage data. Add them all in the usual manner, as below.

```
npm install gulp-sass gulp-sass-glob
gulp-autoprefixer -D
```

Then add it to our list of dependency imports.

```
const sass = require('gulp-sass');
const sassGlob = require('gulp-sass-glob');
const autoprefixer = require
('gulp-autoprefixer');
```

8. Compile SASS with Gulp - part 2

Now we've installed the SASS dependencies, create a new task called 'sass'. Within it, use Gulp's 'src()' method to grab all the SCSS files from our root 'sassDev' directory, and

Running tasks

At any point in this tutorial, if you need to test that a Gulp task is working correctly, run it by typing 'gulp my-task-name' in the Terminal.

then – using Node's 'pipe()' method – pass them to sassGlob so their own SASS imports can be parsed in preparation for compile.

```
gulp.task('sass', function() {
  console.log('Compiling SASS ')
  return gulp.src('${sassDev}/*.scss')
    .pipe(sassGlob())
});
```

9. Compile SASS with Gulp - part 3

Next, pipe our stylesheet to gulp-if, which looks at the truthiness of the minify variable from Step 4, and then compiles the SASS either in a compressed (AKA minified) state or an expanded state. Let's then pipe the compiled CSS to the rename dependency, which will add a '.min' filename prefix if required.

```
.pipe(gulpif(minify, sass({outputStyle:
  'compressed', precision: 8}), sass
  ({outputStyle: 'expanded', precision: 8})))
  .pipe(gulpif(minify, rename({ suffix: '.min'
})))
```

If 'minify' is true, we'll have a stylesheet called 'main.min.css' in the Node.js stream. If there were compile errors, it's important that we print these to the Terminal with the 'on()' method. Pipe the stylesheet to autoprefixer for vendor prefixing, and then finally send it to gulp's 'dest()' method, which places it in the correct 'dist' directory, as defined back in Step 3.

```
.on('error', sass.logError)
.pipe(autoprefixer({
  browsers: ['last 2 version'],
  cascade: false
}))
.pipe(gulp.dest(sassDist))
```

10. Implementing UnCSS

With Gulp, we can also leverage a powerful plugin called UnCSS, which analyses HTML files and then removes any unused styles from our CSS. Among other advantages, this is great way to reduce stylesheet file sizes. As usual, install the dependency like so:

```
npm install gulp-uncss -D
```

Then add UnCSS to our list of dependency imports.

```
const uncss = require('gulp-uncss');
```

Now create a new Gulp task called 'uncss' and grab any CSS files from the 'sassDist' directory using 'src()'. Next, after piping these to UnCSS, we have to tell the plugin where our HTML is located so that it can be analysed, before sending the edited CSS files back to 'sassDist' using the 'dest()' method.

```
gulp.task('uncss', function() {
  console.log('Removing unused CSS ')
  return gulp.src('${sassDist}/*.css')
    .pipe(uncss({
      html: ['${htmlDist}/*.html']
    }))
    .pipe(gulp.dest(sassDist))
});
```

11. Image optimisation

Another useful thing we can automate in Gulp is lossless image optimisation. It requires 'gulp-imagemin', which we should install in the usual way and then include in the gulpfile as a dependency called 'imagemin'. Create a new task called 'images', use 'src()' to grab any bitmap images, and then pass these to imagemin.

```
gulp.task('images', function(){
  console.log('Optimising bitmaps ');
  return gulp.src
(` ${imgDev}/*.{png|jpg|jpeg|gif}`)
  .pipe(imagemin([
    imagemin.jpegtran({progressive: true}),
    imagemin.optipng({optimizationLevel: 5})
  ], { verbose: true }))
  .pipe(gulp.dest(imgDist));
});
```

12. Bundling JavaScript - part 1

If you open '/dev/js/main.js' you'll see 'require' statements, which import project functionality. We can automate the bundling of these into one concatenated, minified file using Browserify. Using Babelify, we can also auto transpile any ES6 JavaScript into JS, which older browsers can understand. Let's install some dependencies.

```
luketheron@lukes-MBP:~/git/webdesignergulpworkflow/package/filesilo/heres-one-in
$ gulp dev
[34146:08] Using gulpfile /git/webdesignergulpworkflow/package/filesilo/heres-one-in-
[34146:08] Starting 'dev'...
[34146:08] Starting 'deleteDist'...
Deleting old 'dist' directory ...
[34146:08] Starting 'serve'...
[34146:08] Server started at http://localhost:8000
[34146:08] First run 'build' after 10 ms
[34146:08] Finished 'build' after 34 ms
[34146:08] Starting 'dev'...
[34146:08] Dev server after 14 ms
[34146:08] Finished 'autoDist' after 88 ms
[34146:08] Starting 'makes-list'...
Generating homepage ...
[34146:08] Starting 'models-list'...
Generating models ...
[34146:08] Finished 'models-list' after 2.27 ms
[34146:08] Starting 'sass'...
Compiling SASS ...
```

Above

When you run 'gulp dev' in the Terminal after finishing the tutorial, you should see the results of the build process in the Terminal, including the feedback we've been adding via 'console.log' statements.

Right

Since the 'serve' task has created a local server, you'll also see the finished result running live at the localhost port mentioned in the Terminal. This is typically 'localhost:8000'.



The screenshot shows a search bar at the top with the placeholder "Search 3399 plugins...". Below it, there are four plugin cards:

- gulp-css-cache-bust**: Adds query to CSS for cache buster. Tags: gulpplugin, cachebuster.
- gulp-css-condense**: A gulp plugin for the CSS compressor that shows no mercy. Tags: gulpplugin, minify, css.
- gulp-css-flipper**: Gulp plugin for Twitter's css-flip - a CSS BiDi flipper. Tags: gulpplugin, gulp, css, flip, bidi.
- gulp-css-hairline**: (partially visible)

The Gulp ecosystem

At the time of writing, there are 3,399 plugins listed at gulpjs.com/plugins. This means there's an incredible amount of potential automation we can apply to our workflows. Here's a brief overview of some of our favourite plugins not featured in this tutorial:

'gulp NOTIFY' enables you to send OS notifications from Gulp tasks. Possible uses include notifying when a build is complete, or if there's an error.

'gulp-plumber' changes Gulp error handling so that when things break (such as SASS not compiling) the task won't quit prematurely.

'gulp-stylelint' and 'gulp-eslint' bring these popular frameworks to Gulp, enabling you to automatically lint your SASS and JavaScript on build.

'gulp-git' enables you to run Git commands from a Gulp task, so for example, when you complete a build, you could choose to commit it also.

15. Bundling JavaScript - part 4

Next, let's send the file to Browserify through the transform function so it can be bundled and its ES6 transpiled. The vinyl-source-stream dependency is then used to send the file back using Gulp so it can be renamed if minify is true. Finally, if minify is true, we run the bundled file through uglify to minify it and then send it to the 'jsDist' directory.

```
return browserify({entries: file})
  .transform('babelify', {presets:
    ['es2015']})
  .bundle()
  .pipe(source(file))
  .pipe(gulpif(minify, rename({
    dirname: '',
    basename: filename,
    suffix: '.min',
    extname: '.js'
  }), rename({
    dirname: '',
    basename: filename,
    extname: '.js'
  }))
  .pipe(gulpif(minify, streamify(uglify())))
  .pipe(gulp.dest(jsDist));
```

13. Bundling JavaScript - part 2

Next, create a new Gulp task called 'js' and within it, use the glob plugin to grab all the JavaScript files in our root 'jsDev' directory and return them in an array. We'll then loop through this using a map function, enabling us to process each JavaScript file individually.

```
gulp.task('js', function() {
  console.log('Bundling JavaScript ');
  var files = glob.sync(`${jsDev}/*.js`);
  files.map(function(file) {
});
```

14. Bundling JavaScript - part 3

Within the map function, the 'file' variable references a string containing the full path of the current file (For example '/dev/js/main.js'). If we want to rename a minified 'main.js' in a later step to 'main.min.js' like within the SASS task, we will need its base filename 'main'. Let's filter out the path information which isn't required, by adding the snippet below to the map function.

```
const filename = file.replace('${jsDev}/',
  '')
  .replace('.js', '');
```

```
gulp.task('makes-list', function() {
  console.log('Generating homepage ');
});
```

17. Import JSON data

In the task, the first thing we'll be doing is importing the car data, which will feed into the static site generation. However, before we do even that, we first need to delete any leftover cache from previous builds to make sure the car data we're using is always up to date. If we don't, when we make changes to the car data in future, the out-of-date cached version may be used by the task instead.

```
delete require.cache[require.resolve(`./${dataDev}/cars.json`)];
let config = require(`./${dataDev}/cars.json`);
```

18. Inject data into a single page

Using Nunjucks, we'll make the car data available to the 'index.html' template, enabling it to display a list of car makes once compiled. With the 'src' method, grab the 'index.html' template and then pipe it to Nunjucks. Next, we'll use Nunjucks' 'compile' method to inject car data from the JSON we've imported to the template and then render it. Finally we'll send the compiled HTML to the 'htmlDist' directory.

```
return gulp.src(`.${htmlDev}/index.html`)
  .pipe(nunjucks.compile({
    date: config['date'],
    makes: config['makes']
  }))
  .pipe(gulp.dest(htmlDist));
```

19. Render new pages - part 1

It's also possible in Gulp to generate lots of HTML files

Developer tutorials

Automate your workflow with Gulp

from a single template. In this case, one page for each car make, which displays a list of available models. Create a new task called 'models-list' and grab the same car data file, deleting cache from any previous builds like before.

```
gulp.task('models-list', function() {
  console.log('Generating car pages ');
  delete require.cache[require.resolve(`./${dataDev}/cars.json`)];
  let config = require(`./${dataDev}/cars.json`);
});
```

20. Render new pages - part 2

We'll now loop through the makes from the car data using a map function. By default, there are three different makes in the car data, so the loop should run three times. For each one, we'll first pass the template to Nunjucks and inject the car data, then rename it to include the make in its filename before finally sending the compiled HTML to the 'htmlDist' directory.

```
config['makes'].map(function(make) {
  return gulp.src(`${htmlDev}/template.html`)
    .pipe(nunjucks.compile({
      date: config['date'],
      make: make.name,
      cars: make.cars
    }))
    .pipe(rename({basename:
      `${make.name.toLowerCase()}-lease-
      prices`}))
    .pipe(gulp.dest(htmlDist));
});
```

21. Automating XML sitemaps

Next, we'll use Gulp to generate an XML sitemap based on our statically generated HTML pages. Typically, we would need one for SEO purposes, and having to update it every time a page is added would be a pain. Install 'gulp-sitemap' and include it as a dependency called 'sitemap' in gulpfile.

22. Create sitemap task - part 1

Create a new task called 'sitemap' and within it, use 'src' to grab any compiled HTML files from the 'htmlDist' directory. To save a bit of processing time, we'll set the method's 'read' option to false, meaning it won't read each files contents, instead only registering its existence.

```
gulp.task('sitemap', function() {
  console.log('Generating sitemap ');
  return gulp.src(`${htmlDist}/*.html`, {
    read: false
  });
});
```

23. Create sitemap task - part 2

We need to pass the compiled HTML files to the sitemap plugin so they can be registered. The options we're configuring here largely correspond to tags within an XML sitemap page entry. One thing to note is that in places, we're using regex through the 'replace' method. This removes the '.html' file extension where visible, as

typically, on a live web server, these wouldn't be used in web addresses.

```
.pipe(sitemap({
  siteUrl: 'http://www.example.com',
  mappings: [
    pages: ['**/*.html'],
    changefreq: 'monthly',
    lastmod: Date.now(),
    hreflang: [
      {
        lang: 'en-GB',
        getHref(siteUrl, file, lang, loc) {
          return siteUrl + file;
        }
      }
    ],
    getLoc(siteUrl, loc, entry) {
      return loc.replace(/\.\w+$/, '');
    }
  ]
})
.pipe(gulp.dest(htmlDist));
```

24. Copy static files

In any build system, sometimes the only thing we need to do with certain files is copy them from directory A to directory B. In this project, there's a collection of fonts where we need to do just that. Create a new task called 'copy', call the 'src' method and grab anything in 'fonts' within the 'dev' directory, before using the 'dest' method to copy them to 'dist'.

```
gulp.task('copy-fonts', function() {
  console.log('Copying fonts ');
  return gulp.src(`${dev}/fonts/**`)
    .pipe(gulp.dest(`${dist}/fonts`));
});
```

25. Create a webserver

Being able to easily create a simple web server to serve our project from would be a big convenience. This is something we can do in Gulp with the 'gulp-webserver' plugin, which we should install and then list as a dependency called 'webserver' in gulpfile.

Under the Production Tasks header, add the task below. We'll set the plugin's 'livereload' option to true, so that when changes are made, the page automatically refreshes in the browser.

```
gulp.task('serve', function() {
  console.log("Serving the project ");
  gulp.src(`${dist}`).pipe(webserver({
    livereload: true
  }));
});
```

26. Install run-sequence

Now we have all of our tasks in place, it would be useful if we could run them all with one easy command, rather than being forced to run them individually.

The most readable way for us to do this requires a plugin called 'run-sequence'. Install it in the usual manner and include it as a project dependency called 'runSequence'. Next, create a new task under the Production Tasks header called 'build'.

```
gulp.task('build', function() {
});
```

27. Creating the build task

With run-sequence, we can define whether tasks should run synchronously or asynchronously, which is helpful when one task (Such as UnCSS) depends on another being completed first (such as SASS).

Gulp comes with this kind of functionality out of the box in the form of task dependencies, but run-sequence enables us to define this behaviour in a much more obvious, readable way, in large tasks. Add the snippet below to the build task. Tasks passed on their own run synchronously, while tasks passed in an array all run asynchronously.

```
runSequence(
  'deleteDist',
  ['makes-list', 'models-list',
  'sass', 'js', 'copy-fonts', 'images'],
  ['uncss', 'sitemap']
);
```

28. Dev task - part 1

So far, we have created a build task which does exactly that, build the task. One issue with this is that during development, we'll keep having to run this or individual tasks as we make changes to our files, as well as serve our project. Let's create another task called 'dev' that automates all of this.

You may notice that before the task's function, we're adding an array mentioning the 'build' and 'serve' tasks, which tells Gulp to run these first. These are the same tasks dependencies mentioned in Step 27, and are more suited to this simple use.

```
gulp.task('dev', ['build', 'serve'],
  function() {
});
```

29. Dev task - part 2

Next, we'll use a method belonging to Gulp called 'watch'. This keeps an eye on any files we pass to it via the first parameter, and if they change, runs the task passed in the second. This removes the need for us to manually run tasks as we develop the project. Add these watch tasks to 'dev' and that should cover everything.

```
gulp.watch(['${htmlDev}/**/*.html',
  '${dev}/data/cars.json'],
  ['makes-list', 'models-list']);
gulp.watch('${imgDev}/*.(png|jpg|jpeg|gif)',
  ['images']);
gulp.watch('${sassDev}/**/*.scss', ['sass']);
gulp.watch('${jsDev}/**/*.js', ['js']);
```

30. Run dev task

Finally, run the dev task in terminal with 'gulp dev' and watch as the project is built and a server is created at localhost:8000. From then on, if we change any of the files which are being watched, the appropriate task will then run and recompile your edited file.

Congratulations! You now have a build system powered by Gulp.js.

Special offer for readers in *North America*



6 issues FREE

When you subscribe*

FREE
resource
downloads
every issue

28 PAGE SPECIAL

COLOR CREATE THE PERFECT PALETTE

TYPE TOP TRENDS FOR TODAY

LAYOUT DESIGN & BUILD GRID LAYOUTS

FONTS 10 GOLDEN RULES OF FONT COMBOS

STATIC SITE GENERATORS 10 of the best to instantly build HTML pages

SHRINK YOUR CSS Make your style sheets more efficient

AUTOMATE WORKFLOW Utilise Gulp to get rid of repetitive tasks

JS REACT & TYPESCRIPT Get the JS libraries to work in unison

FREE OVER 3 HOURS OF JS VIDEOS

web designer™

@WebDesignerMag /WebDesignerUK

HTML5 CSS3 jQuery JavaScript

GET PRO DESIGN SKILLS

“ *The only magazine you need to design and develop stunning websites* **”**



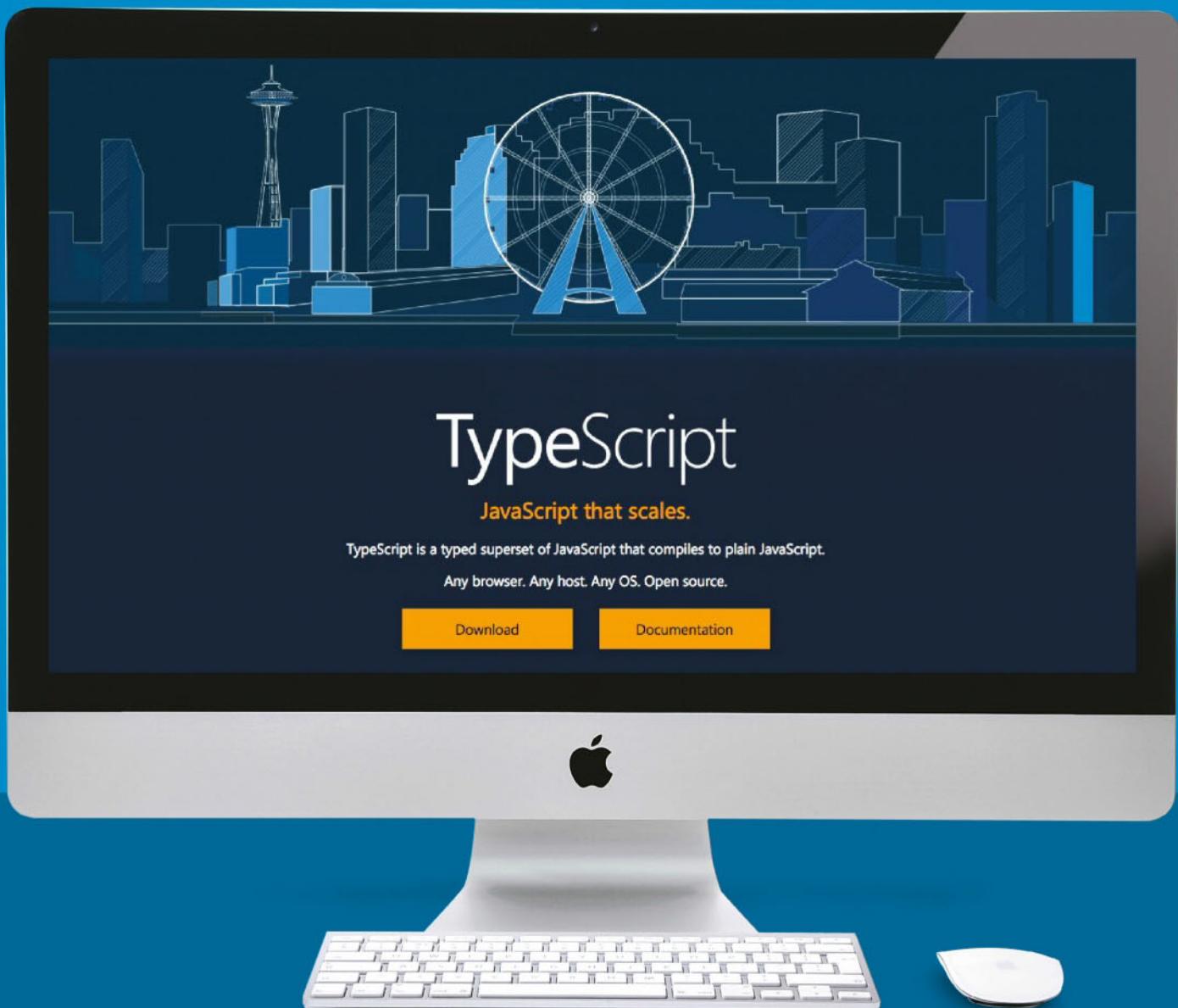
Order hotline **+44 (0) 344 848 2852**
Online at **myfavouritemagazines.co.uk/WEDPQ17**

***Terms and conditions** This is a US subscription offer. 6 free issues refers to the USA newsstand price of \$14.99 for 13 issues being \$194.87, compared with \$103.20 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 Offer ends March 31 2018.

**Expires
31 March
2018**

Deploy TypeScript with React

While TypeScript makes coding easier, deploying it with a framework can be a hassle. Here's webpack to the rescue...



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



While TypeScript brings a variety of advantages to the arsenal of the JavaScript programmer, transpiling (and syntax-checking)

large amounts of TS code in the browser is ineffective. Sadly, an old adage teaches us that manual processes tend to break down over time – if each and every deployment takes a lot of console work, deployment frequency will take a hit. webpack's automatic packing feature absorbs some of the heat so that when the pipeline is set up, transpilation is performed with minimal developer input.

This, furthermore, frees up capacity for other tasks. In our example, the powerful React framework is integrated into the webpack compilation process via a group of helper utilities. React's JSX syntax is augmented via a proprietary language called tsx, thereby enabling tighter integration between React and TypeScript's advanced run-time checking.

In short, working through this tutorial demonstrates the creation of a workflow combining the best of both worlds. Save time by using React's logic, and avoid errors via the enhanced syntax. If you are not a bug, there is little not to like here.

1. Deploy webpack

Seasoned Node.js developers will not be surprised about the fact that the use of webpack requires its global installation via the npm 'install -g' command. When done, webpack can be invoked from your workstation's command line.



[nodejs.org](#)

[HOME](#) | [ABOUT](#) | [DOWNLOADS](#) | [DOCS](#) | [GET INVOLVED](#) | [SECURITY](#) | [NEWS](#) | [FOUNDATION](#)

About

[Governance](#)
[Community](#)
[Working Groups](#)
[Releases](#)
[Resources](#)
[Trademark](#)
[Privacy Policy](#)

About Node.js®

As an asynchronous event driven JavaScript runtime, Node.js is designed to build scalable network applications. In the following "hello world" example, many connections can be handled concurrently. Upon each connection the callback is fired, but if there is no work to be done, Node.js will sleep.

```
const http = require('http');
const hostname = '127.0.0.1';
const port = 3000;
const server = http.createServer((req, res) => {
  res.statusCode = 200;
  res.end('Hello World\n');
```

2. Create the scaffolding structure

Microsoft recommends the creation of a specific folder structure to contain the various files constituting our project. When working on an Ubuntu workstation, simply execute the command sequence shown in a convenient location.

```
tamhan@TAMHAN14:~$ mkdir tsreactspace
tamhan@TAMHAN14:~$ cd tsreactspace/
tamhan@TAMHAN14:~/tsreactspace$ mkdir src
tamhan@TAMHAN14:~/tsreactspace$ cd src
tamhan@TAMHAN14:~/tsreactspace/src$ mkdir components
tamhan@TAMHAN14:~/tsreactspace/src$ cd ..
```

3. Create a Node.js project

As with all Node.js-related programs, the next step involves the creation of a repository and the loading of the various dependencies that are needed. Simply enter the commands shown below to complete the creation of the structure.

```
tamhan@TAMHAN14:~/tsreactspace/src$ npm init
tamhan@TAMHAN14:~/tsreactspace/src$ npm
install --save react react-dom @types/react
@types/react-dom
tamhan@TAMHAN14:~/tsreactspace/src$ npm
install --save-dev typescript awesome-
typescript-loader source-map-loader
```

4. Automate your TypeScript process

Most developers use TypeScript via the in-browser transpiler or an IDE. The capabilities of the tsconfig.json file are less known – it allows the pre-definition of a compiler toolchain. Create an empty file in root folder.

5. Populate the file

In the next step, the TypeScript compilation process must be set up. Open the file created in step 4, and replace its contents with the ones shown accompanying this step.

```
{
  "compilerOptions": {
    "outDir": "./dist/",
    "sourceMap": true,
    "noImplicitAny": true,
    "module": "commonjs",
```

Be awesome and keep Microsoft happy

Authors of TypeScript examples like to substitute 'awesome-typescript-loader' with 'ts-loader'. This is bad for two reasons. First of all, 'awesome-*' has better multithreading capabilities. Secondly, Microsoft's 'official' guidelines prefer 'awesome-*'.

```
tamhan@TAMHAN14:~$ npm install -g webpack
/home/tamhan/.npm-packages/bin/webpack -> /home/tamhan/.npm-packages/lib/node_modu
> uglifyjs-webpack-plugin@0.4.6 postinstall /home/tamhan/.npm-packages/lib/node_mod
> node lib/post_install.js

/home/tamhan/.npm-packages/lib
└── webpack@3.10.0
    ├── acorn@5.3.0
    └── acorn-dynamic-import@2.0.2
        └── acorn@4.0.13
            ├── ajv@5.5.2
            │   ├── co@4.6.0
            │   ├── fast-deep-equal@1.0.0
            │   ├── fast-json-stable-stringify@2.0.0
            │   ├── json-schema-traverse@0.3.1
            │   ├── ajv-keywords@2.1.1
            │   ├── async@2.6.0
            │   │   └── lodash@4.17.4
            │   ├── enhanced-resolve@3.4.1
            │   │   ├── graceful-fs@4.1.11
            │   │   └── object-assign@4.1.1
            │   └── escape@3.6.0
            └── es6-map@0.1.5
```

```
tamhan@TAMHAN14:~/tsreactspace$ tree -L 2
.
└── src
    ├── components
    └── node_modules
        └── package.json

3 directories, 1 file
```

Left

NPM's graphically verbose output showcases the colour capabilities of Ubuntu's terminal emulator

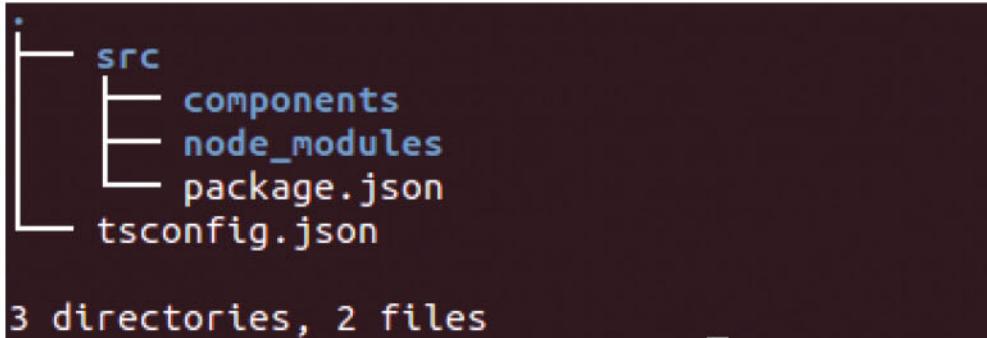
Above

Once the package.json file is created, the rest takes place almost completely automatically

Developer tutorials

Deploy TypeScript with React

Below Thanks to the use of the './' syntax, the 'config' file can sit above the source folders



```
"target": "es5",
"jsx": "react"
},
"include": [
"./src/**/*"
]
```

6. Understand the structure

One interesting aspect of the 'tsconfig' file involves the use of './'. A single dot represents the local folder, while .. represents the folder 'above' the current one. In our case, './dist/' points at the dist folder below the location of the configuration file.

7. Configure webpack

As workstations age, your Node.js config can get messed up. This is particularly problematic for webpack: if the product can't be run from your shell, install it again locally. Next, use the npm bin to find the folder containing it, and invoke it from there to get the error message shown accompanying this step.

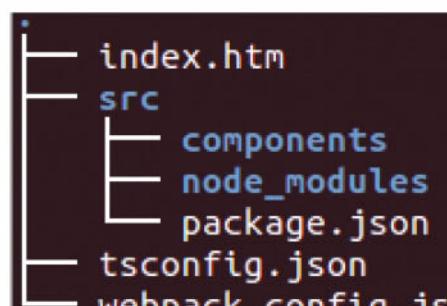
Even more options!

The 'Awesome TypeScript Loader' can be considered the Swiss Army knife of TypeScript - look at its options described at <https://github.com/s-panferov/awesome-typescript-loader> to adjust its behaviour even further.

8. Get configuring

Create a file called 'webpack.config.js' in the root folder of the current project. Next, replace its contents with the ones shown accompanying this step - we will discuss relevant parts of the configuration file next.

```
module.exports = {
entry: "./src/index.tsx",
output: {
filename: "bundle.js",
path: __dirname + "/dist"
},
devtool: "source-map",
resolve: {
extensions: [".ts", ".tsx", ".js", ".json"]
}
}
```



Above Index.htm must stand outside of the '/src' folder and its contents

Top

If webpack cannot be found, re-install it and run it from the npm bin folder

Bottom

Microsoft's preferred arrangement does not work with recent versions of webpack

```
tamhan@TAMHAN14:~/tsreactspace$ /home/tamhan/tsreactspace/src/node_modules/.bin/webpack
Hash: 8711e01e588d8432dbbb
Version: webpack 3.10.0
Time: 48ms

ERROR in Entry module not found: Error: Can't resolve 'awesome-typescript-loader' in '/home/tamhan/...'

tamhan@TAMHAN14:~/tsreactspace/src$ webpack
webpack: command not found
tamhan@TAMHAN14:~/tsreactspace/src$ npm bin
/home/tamhan/tsreactspace/src/node_modules/.bin
tamhan@TAMHAN14:~/tsreactspace/src$ /home/tamhan/tsreactspace/src/node_modules/.bin/webpack
No configuration file found and no output filename configured via CLI option.
A configuration file could be named 'webpack.config.js' in the current directory.
Use --help to display the CLI options.
tamhan@TAMHAN14:~/tsreactspace/src$
```

Documentation

tsconfig.json

Overview

The presence of a `tsconfig.json` file in a directory indicates that the directory is the root of a TypeScript project. The `tsconfig.json` file specifies the root files and the compiler options required to compile the project. A project is compiled in one of the following ways:

- By invoking tsc with no input files, in which case the compiler searches for the `tsconfig.json` file starting in the current directory and continuing up the parent directory chain.
- By invoking tsc with no input files and a `--project` (or just `-p`) command line option that specifies the path of a directory containing a `tsconfig.json` file, or a path to a valid `.json` file containing the configurations.

When input files are specified on the command line, `tsconfig.json` files are ignored.

Examples

Example `tsconfig.json` files:

- Using the "files" property

```
{
  "compilerOptions": {
    "module": "commonjs",
    "noImplicitAny": true,
    "removeComments": true,
    "preserveConstEnums": true,
    "sourceMap": true
  },
  "files": [
    "index.ts"
  ]
}
```

Analyse your options

Our discussion of `tsconfig.json` is relatively superficial in scope. The limited space in the magazine does not allow us to go into depth. TypeScript comes with literally dozens of options, most of which are documented in the portal accessible at <https://www.typescriptlang.org/docs/handbook/tsconfig.json.html>. Keep in mind that the 'root' webpage does not list all options. Instead, click the Project Configuration node at the left-hand side of the screen to reveal further parameters and fields which might be of interest. Finally, be aware that some of the options require the presence of Visual Studio - Microsoft, after all, wants to push its own IDE.

11. Create an entry point

The next step involves the creation of an application entry point using the structure shown accompanying this step. We simply load a bunch of predefined resources and place the file outside of the `/src/` folder in accordance with the tree output below.

```
<!DOCTYPE html>
<html>
  <head> </head>
  <body>
    <div id="example"></div>
    <script src="./node_modules/react/umd/react.development.js"></script>
    <script src="./node_modules/react-dom/umd/react-dom.development.js"></script>
    <script src="./dist/bundle.js"></script>
  </body>
</html>
```

12. Create a TypeScript file

Displaying information in TypeScript is relatively simple. Due to the inclusion of React, 'index.tsx' - to be placed in `/src/` - requires some additional code to include various modules.

```
import * as React from "react";
import * as ReactDOM from "react-dom";
ReactDOM.render(
  <h1>Hello, world!</h1>,
  document.getElementById("example")
);
```

Fortunately, the problem can be fixed by adjusting the folder structure (see figure).

14. Pack it again!

With that out of the way, adjust the path for calling webpack and take another stab at the compilation process. It will pass, yielding output similar to the one shown in the listing accompanying this step.

```
tamhan@TAMHAN14:~/tsreactspace$ /home/tamhan/tsreactspace/node_modules/.bin/webpack
...
Asset      Size  Chunks
Chunk Names
bundle.js  2.95 kB      0 [emitted]
main
bundle.js.map 3.19 kB      0 [emitted]
main
[0] ./src/index.tsx 252 bytes {0} [built]
+ 2 hidden modules
```

13. Take a stab

At this point, it is time to try and compile the project for the first time. If you followed Microsoft's instructions carefully, the error shown in the figure will pop up.

15. Say hello

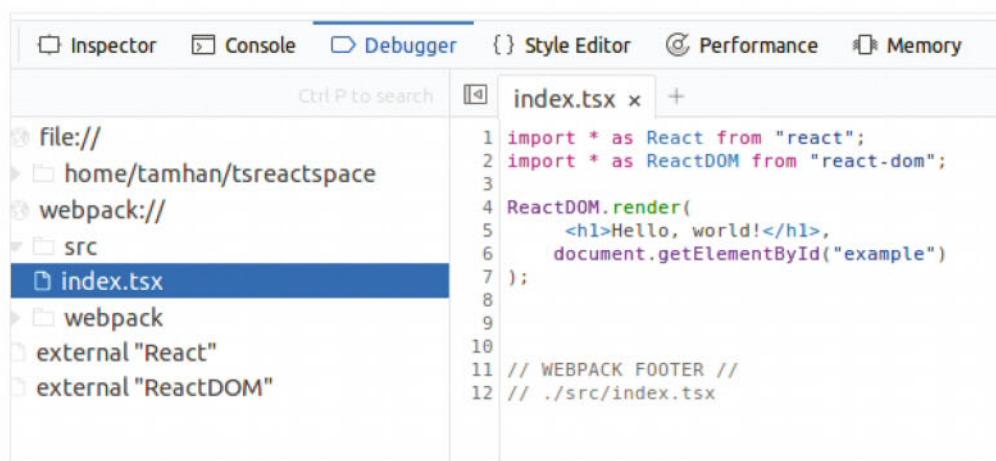
With that out of the way, it is time to open 'index.htm'. The browser will load the page, and will present our simple, React-based greeting to the world. When distributing the finished application, make sure to include the 'dist' folder.

16. Use source mapping

One frequent criticism of TypeScript is that debugging is tiresome. Browsers, after all, cannot run TypeScript directly. This problem is thankfully mitigated via Map files such as 'bundle.js.map'. They contain a data structure which permits correlation between the output and the source files.

17. Celebrate the map file

While many browsers demand the manual loading of the map file, using our webpack output with Firefox leads to the output shown in the figure. This is made possible by a comment at the bottom of 'bundle.js', which references



Developer tutorials

Deploy TypeScript with React

```
tamhan@TAMHAN14:~/tsreactspace$ tree -L 1
```

```
.
├── dist
├── index.htm
├── node_modules
├── package.json
└── src
    ├── tsconfig.json
    └── webpack.config.js
```

3 directories, 4 files

Above Moving the body of the Node project to the same level as the webpack 'config' file fixes weird problems

the source map directly.

```
    ...
    /**/ })
/***** ];
//# sourceMappingURL=bundle.js.map
```

18. Create a component

One of TypeScript's most important features is the use of classes and data types. Creating a new component is done with a structure similar to the one shown below - 'Props' and 'State' can be declared using the 'type' keyword.

```
type Props = { ... };
type State = { ... };
class MyComponent extends React.Component<Props, State> {
    ...
}
```

19. Do it with interfaces

Another way to solve the problem involves the use of interfaces. In this case, the syntax does not change significantly - Microsoft's favourite example demos it via a set of string parameters passed in via the 'HelloProps' object.

```
export interface HelloProps { compiler:
string; framework: string; }
export class Hello extends React.Component<HelloProps, {}> {
    render() {
        return <h1>Hello from {this.props.compiler} and {this.props.framework}!</h1>;
    }
}
```

20. Deploy the component

With that out of the way, the component can be integrated into our main window as if it were a normal React element.

```
import { Hello } from "./components/Hello";
```

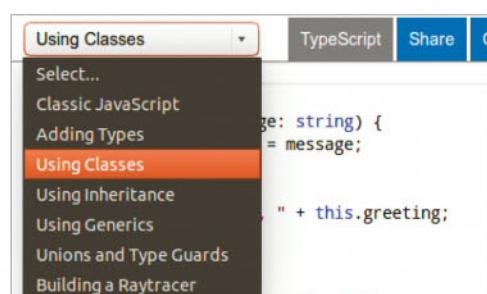
```
ReactDOM.render(
    <Hello compiler="TypeScript"
framework="React" />,
    document.getElementById("example")
);
```

21. Use advanced features

Simply introducing TypeScript to your code is not enough. Make sure to read the documentation carefully, as to enable your developers and yourself to use the most advanced features found in the language standard.

22. Try it out online

Should you ever feel like testing a bit of TypeScript, keep www.typescriptlang.org/play in mind. It contains a live version of the TypeScript compiler running in your browser. Simply enter code on the left-hand side, and see the JavaScript output on the right.



The screenshot shows the TypeScript playground interface. On the left, there is a code editor with the following TypeScript code:

```
class Greeter {
    greeting: string;
    constructor(message: string) {
        this.greeting = message;
    }
    greet() {
        return "Hello, " + this.greeting;
    }
}
let greeter = new Greeter("world");
```

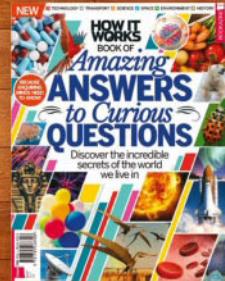
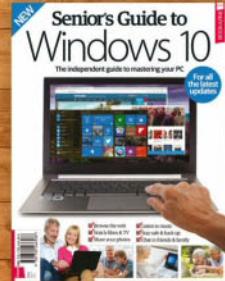
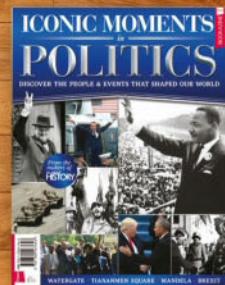
On the right, the JavaScript output is displayed:

```
var Greeter = /*#__ @class */ (function () {
    function Greeter(message) {
        this.greeting = message;
    }
    Greeter.prototype.greet = function () {
        return "Hello, " + this.greeting;
    };
})
;
return Greeter;
)();
var greeter = new Greeter("world");
var button = document.createElement('button');
button.textContent = "Say Hello";
button.onclick = function () {
    alert(greeter.greet());
};
document.body.appendChild(button);

```

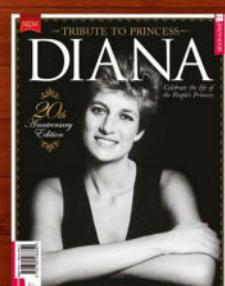
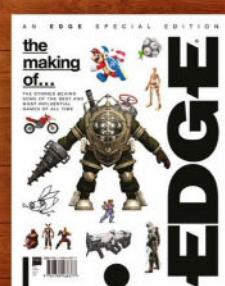
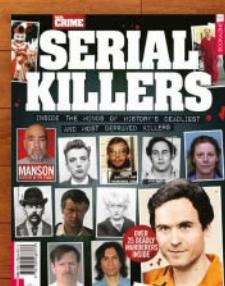
23. Load some examples

The TypeScript playground comes with a set of pre-loaded examples which can be loaded via the combination box on the left-hand side of your screen. In addition to various syntax samples, the product even contains an extremely interesting implementation of a ray tracer.



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

+44 (0)1225 687832

HOSTING LISTINGS



Featured host: tidyhosts

www.tidyhosts.com / 05603 674610

About us

Tidyhosts has become a leading cloud hosting provider throughout the world. It boasts a feature-rich selection of services including domain registration, shared web hosting, WordPress hosting, hosted exchange email, cloud virtual servers and media streaming. Its passion and drive for

success has gained it a highly reliable and trusted reputation from its customers, making it the number one for hosting services. Founded in 2004 by three developers who wanted to build hosting infrastructure for developers - years on, the company has expanded to offer much more.

What we offer

- **Domain names** - Simple domain registration with a large choice of TLDs
- **Shared web hosting**
 - Includes one-click application installers and a choice of Windows or Linux
- **Cloud virtual servers** - Take full control over your hosting, and install the software you need on your server
- **SHOUTcast hosting** - Start your own radio station with our easy to set up and use SHOUTcast hosting service

5 Tips from the pros

1. Ensure that you choose the right domain

When choosing your domain, make sure it's easy to remember and resembles what you are offering. Customers are more likely to come back if they can remember your URL.

2. Create clear, concise website content

Keep the content on your website clear, informative and, more importantly, relative! Avoid any duplication of content on different pages as this can affect your rankings on search engines.

3. Utilise SSL certificates to stay secure

More sites are moving to SSL security

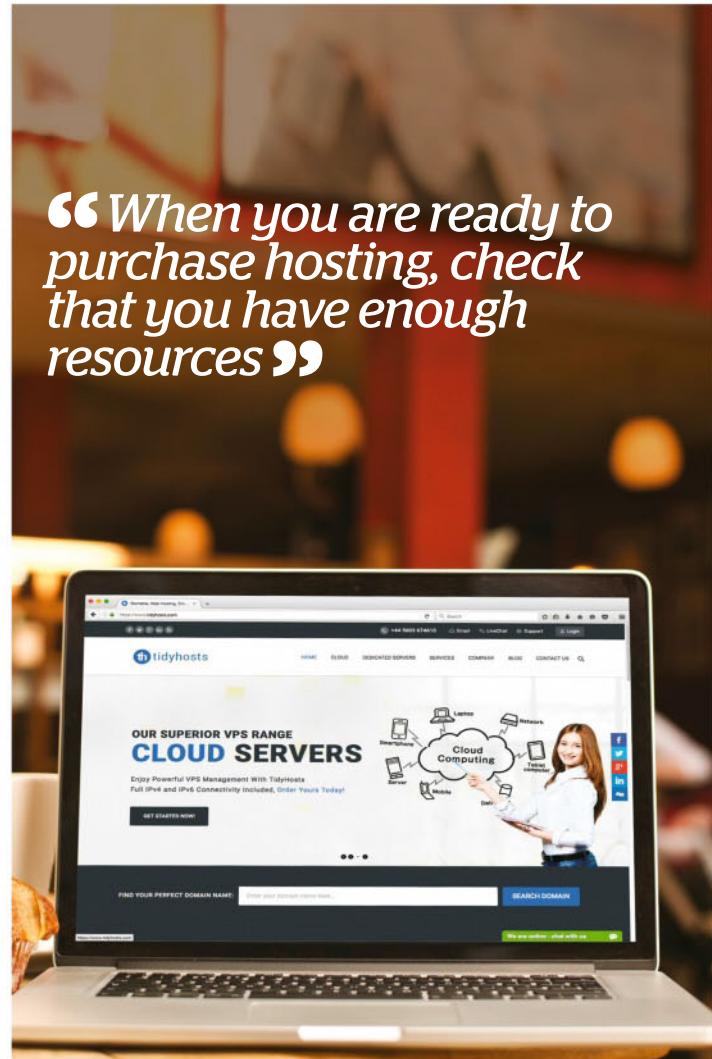
to protect their customers from the ever-increasing threats on the web. Search engines are now ranking sites with an SSL higher than those without.

4. Make sure you choose the right plan

When you are ready to purchase hosting, check that you have enough resources, especially if you expect your website to grow quickly. Seek advice from tidyhosts if you are unsure.

5. Use one-click installers when building a website

If you are new to building a website then we have a number of useful one-click application installers. These help you get up and running, including the popular WordPress system.



Testimonials

Kelly Underwood

“ I've been with a number of web host providers in the past which have at some point let me down. I now realise the importance of using a host that is well known ”

John Corey

“ We have our email hosted with tidyhosts, which serves a small number of users in our office. We have found this solution is much more cost-effective ”

Jenny Brice

“ A great host with fantastic knowledge. I have only had to use the support channels a few times, but the replies I have had helped me instantly, so well worth it ”



Get your listing in our directory

To advertise here contact *Chris*

chris.mitchell@futurenet.com

+44(0)1225 687832

Supreme hosting



www.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.

- Co-location hosting
- VPS
- 100% Network uptime

UK-based hosting



www.cyberhostpro.com
0845 5279 345

Cyber Host Pro are committed to providing the best cloud server hosting in the UK; they are obsessed with automation. If you're looking for a hosting provider who will provide you with the quality you need to help your business grow, then look no further than Cyber Host Pro.

- Cloud VPS servers
- Reseller hosting
- Dedicated servers

Cluster web hosting



www.fasthosts.co.uk
0808 1686 777

UK based and operating 24/7 from dedicated UK data centres. Fasthosts keep over one million domains running smoothly and safely each day. Services can be self-managed through the Fasthosts Control Panel.

- Dedicated servers
- Cloud servers
- Hosted email



Budget hosting



www.hetzner.com
+49 (0)9831 505-0

Hetzner Online is a professional web hosting provider and experienced data centre operator. Since 1997, the company has provided private and business clients

with high-performance hosting products as well as the infrastructure for the efficient operation of sites. A combination of stable technology, attractive pricing, flexible support and services has enabled Hetzner Online to strengthen its market position both nationally & internationally.

- Dedicated/shared hosting
- Colocation racks
- SSL certificates



All-inclusive hosting



www.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



www.bargainhost.co.uk
0843 289 2681

Since 2001, Bargain Host have campaigned to offer the lowest possible priced hosting in the UK. They have achieved this goal successfully and built up a large client database which includes many repeat customers. They have also won several awards for providing an outstanding hosting service.

- Shared hosting
- Cloud servers
- Domain names

Value Linux hosting



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
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, email and 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

+44(0)1225 687832

COURSE LISTINGS



Featured: **Northcoders**

www.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

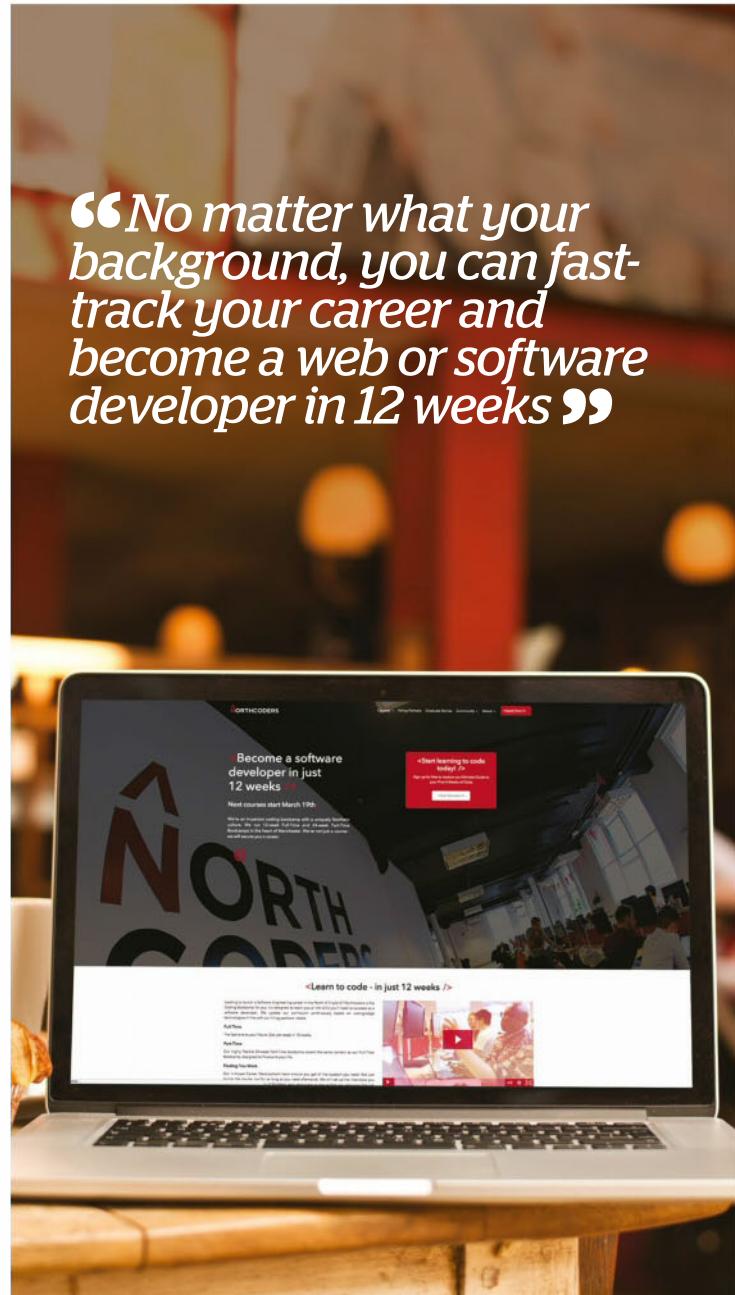
Once you've decided it's right



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



Get your listing in our directory

To advertise here contact *Chris*

chris.mitchell@futurenet.com

+44 (0)1225 687832



UDEMY

www.udemy.com

Twitter: [@udemy](https://twitter.com/udemy)

Facebook: [udemy](https://facebook.com/udemy)

The inspiration for Udemy began in a small village in Turkey, where founder Eren Bali grew up frustrated by the limitations of being taught in a one-room school house. Realising the potential of learning on the internet he set out to make quality education more accessible. Udemy is now a global marketplace for learning and teaching online. Students can master new skills by choosing from an extensive library of over 40,000 courses including HTML, CSS, UX, JavaScript and web development.

40,000+ courses: There is a course for every designer and dev

Self-paced learning: Learn how to code at your own pace



THE IRON YARD

www.theironyard.com

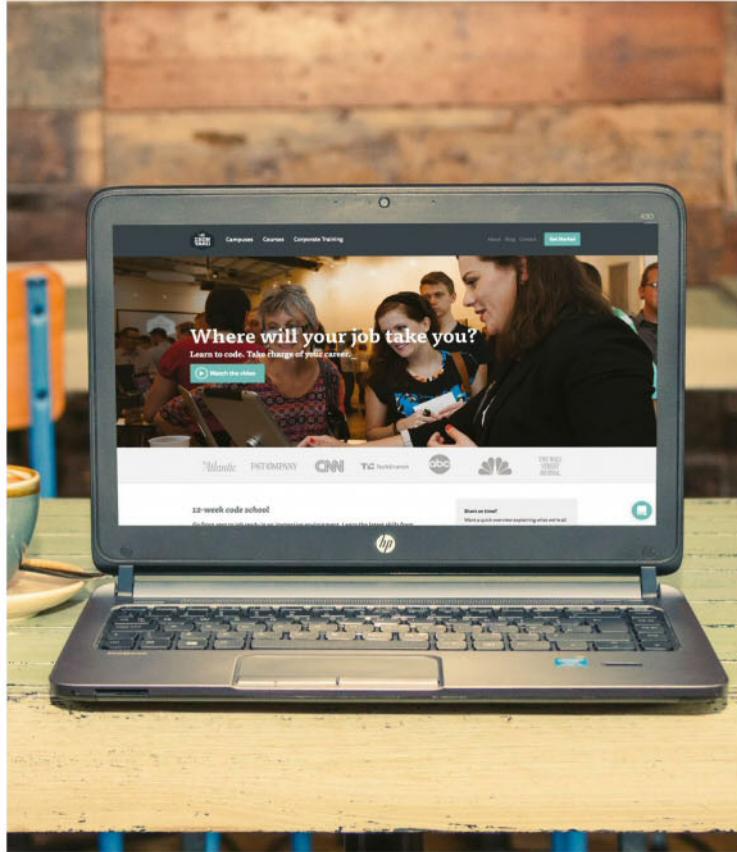
Twitter: [@TheIronYard](https://twitter.com/TheIronYard)

Facebook: [TheIronYard](https://facebook.com/TheIronYard)

The Iron Yard is one of the world's largest and fastest-growing in-person code schools. It offers full-time and part-time programs in back-end engineering, front-end engineering, mobile engineering and design. The Iron Yard exists to create real, lasting change for people, their companies and communities through technology education. The in-person, immersive format of The Iron Yard's 12-week courses helps people learn to code and be prepared with the skills needed to start a career as junior-level software developers.

12-week code school: Learn the latest skills from industry pros

Free crash courses: One-night courses, the perfect way to learn



WE GOT CODERS



[WE GOT CODERS](http://www.wegotcoders.com)

www.wegotcoders.com

hello@wegotcoders.com

We Got Coders is a consultancy that provides experts in agile web development, working with startups, agencies and government. Take one of their 12-week training 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



www.futurelearn.com

feedback@futurelearn.com

Choose from hundreds of free online courses, from Language & Culture to Business & Management; Science & Technology to Health & Psychology.

Learn from the experts. Meet educators from top universities who'll share their experience through videos, articles, quizzes and discussions.

- Learn from experts
- Free courses
- All-device access

GYMNASIUM



www.thegymnasium.com

help@thegymnasium.com

Gymnasium offers free online courses designed to teach creative professionals in-demand skills.

Courses are all self-paced and taught by experienced practitioners with a passion for sharing practical lessons from the design trenches.

- Gain real-world skills
- Get expert instruction
- Career opportunities

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
- Over 3 hours of expert JavaScript video from Pluralsight (www.pluralsight.com)
- 150 Nostalgia Lightroom presets and Dot grid Photoshop actions (www.sparklestock.com)

→ Log in to 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



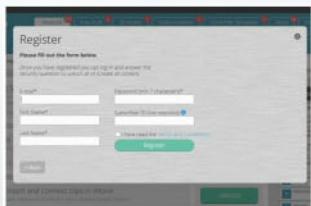


The home of great downloads – exclusive to your favourite magazines from Future!

- 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

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

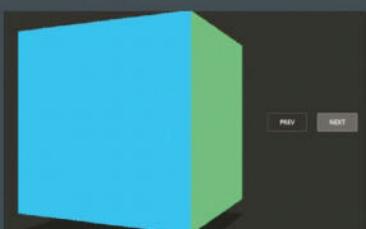
Over 60 hours of video guides

Let the experts teach you to create and code



More than 400 tutorials

Get the code you need to get creative



Over 210 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 eight-digit alphanumeric code printed above your address details on the mailing label of your subscription copies – also found on any renewal letters.

More than 900 reasons to subscribe

+
More added every issue

NEXT MONTH

CREATE CUSTOM HTML

DISCOVER THE POWER OF WEB COMPONENTS
AND HOW TO BUILD WITH GOOGLE'S
POLYMER LIBRARY

FASTER WEB ANIMATIONS

Learn a host of techniques to improve the performance of web animations

CONVERSION RATE OPTIMISATION

Discover how to increase site visitor numbers and turn them into customers

CODE WITH GOOGLE & TWITTER APIs

Learn how to show tweets and pictures from a selected location on a map

Visit the **WEB DESIGNER** online shop at



myfavouritemagazines

myfavouritemagazines.co.uk

for the latest issue, back issues and specials

ALL IN YOUR NEXT
WEB DESIGNER
Issue 272 on sale
Thursday 8th March 2018

moqups



Mockup, wireframe, diagram, prototype
& collaborate with one flexible solution.

moqups.com

OFFER EXPIRES
MARCH 1ST 2018

Save an extra 30% on all new Moqups plans
Discount code: **NETMAG30**

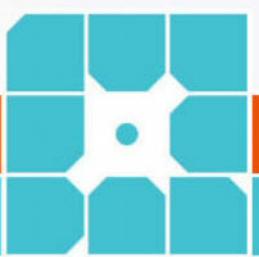
**SAVE
AN EXTRA 30%**





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

WP engine®