RESPONSIVE

WITH GUY ROUTLEDGE

@GUYROUTLEDGE | #FEWD

OBJECTIVE

Learn the building blocks of responsive web design so we can make our projects look great on all devices

AGENDA

- Types of Layouts
- Relative units
- Media Queries
- Flexible Images
- Responsive Images

RESPONSIVE DESIGN OVERVIEW

- Fluid containers
- Media queries
- Flexible images

FIXED VS RESPONSIVE

Checkout these **Fixed** sites

- TyrrellsCrisps.co.uk
- CaterAllen.co.uk
- StevesLeaves.co.uk
- Google.com

FIXED VS RESPONSIVE

Checkout these **Responsive** Sites

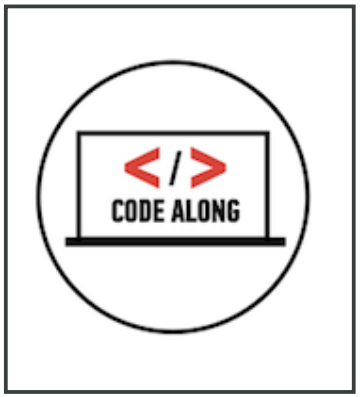
- General Assembly
- The Venture
- Kallo
- el Jimador

FIXED VS RESPONSIVE

The important takeaway is that a responsive site uses the same codebase to achieve optimised layout across multiple screens.

FIXED LAYOUT

- Simple and a good starting point
- Gives more control
- Easier to build
- Still works on mobile - pinch & zoom
- Relies on a container of fixed width



DESKTOP BOXES

http://codepen.io/guyroutledge/pen/QNBOqw

MOBILE BOXES

FLUID LAYOUT

- Sized in percentages
- Everything is relative
- It's all about proportions

RELATIVE UNITS

Further to sizing our containers in percentages, we can size other UI components (font-size, margin, padding, etc.) in relative units which cascade and build a system of proportions.

RELATIVE UNITS

Common units of measurement for these components are:

- `em`
- `rem`
- `vw` and `vh`

EM VS REM

EM

Sized based on the width of the letter "M"

http://alistapart.com/articles/howtosizetextincss

Size is relative to the parent

```
body { font-size:16px; }
h1 { font-size:2em; } /* font size is 32px */
```

EM VS REM

Rem

- Stands for "Root" em.
- Based on the font-size of html (root) element
- Browser support (IE9+) is not as good as `em` so a `px` fallback is needed.

MEDIA QUERIES

MEDIA TYPES

The two main media types are `print` and `screen`

This means we can write CSS for screen devices (laptops, tablets, mobiles) and different CSS for printing a web page out.

MEDIA QUERIES

```
@media only screen and (max-width: 800px) {
    /* styles only apply when viewport is up to 800px
}
```

MEDIA QUERIES

There are lots of things we can query about the media, these are the most common.

```
width | min-width | max-width
height | min-height | max-height
orientation
aspect-ratio | min-aspect-ratio | max-aspect-ratio
resolution | min-resolution | max-resolution
```

Separate multiple clauses with "and" and negate with "not"

MOBILE DISPLAY

```
<meta name="viewport"
      content="width=device-width, initial-scale=1" >
```

To make our media queries work we need to add this special **`meta`** tag to the **`head`** in our HTML.

USING MEDIA QUERIES

```
/* float boxes into columns */
.box {
    float:left;
    padding:20px;
}
@media only screen and (max-width:800px) {
    .box {
        float:none;
    }
}
```

MEDIA QUERIES - MOBILE FIRST

- Start with a single column
- Add media queries using `min-width`
- Performance benefits
- Slightly less code

MEDIA QUERIES - DESKTOP FIRST

- Start with the full-width version
- Add media queries using `max-width`
- Can be more intuitive
- Can cause trouble scaling complex layout to narrow screen

MEDIA QUERIES - BREAKPOINTS

When building a responsive site we often have some common breakpoints where we want major changes to occur. These are normally described as "mobile styles", "tablet styles" or "desktop styles".

Or small, medium or large screens.

MEDIA QUERIES - BREAKPOINTS

We want to use generic breakpoints that approximate most devices at these screen sizes so we don't design layouts for specific devices.

I tend to use `500px` to approximate mobile and `800px` to approximate tablets and `1200px` to approximate desktops

MEDIA QUERIES - TWEAKPOINTS

Having generic breakpoints is good but you will often need to tweak different styles to make the content or your layout look it's best.

We call these tweak points and they can take any pixel value to wrangle your layout into its best shape.

MEDIA QUERIES

You can either put all of your media queries in one place in your stylesheet (at the bottom to ensure no specificity issues).

Or you can have multiple `@media` declarations throughout your stylesheet, setting up responsive styles directly beneath your initial CSS.

RESPONSIVE BOXES

FLEXIBLE IMAGES

FLEXIBLE IMAGES

Images have fixed dimensions that can break our fluid layouts.
We can make them flexible by setting **`max-width`**

```
img {
    max-width:100%;
}
```

RESPONSIVE IMAGES

This is a hard problem to solve, but we're getting there. If you'd like to read up on it, start at the Responsive Images Community Group

TOOLS

TOOLS

Responsive Design testing can be tricky. Here are some tools to help

- Chrome Dev Tools
- iOS Simulator
- BrowserStack (https://www.browserstack.com/)
- GhostLab (http://vanamco.com/ghostlab/)
- Responsive Bookmarklet (http://responsive.victorcoulon.fr/)