



FEWD - RESPONSIVE BASICS

NICO CASTRO

Web Developer at Red Badger

AGENDA

- Homework Solution
- Responsive Layouts
- Mobile First
- Media Queries
- CSS Specificity

HOMework SOLUTION



ROCK PAPER SCISSORS

RESPONSIVE LAYOUT

FIXED VS RESPONSIVE

Checkout this **Fixed** site

- [UPS.com](https://www.ups.com)

Checkout these **Responsive** Sites

- [Generalassembly.ly](https://generalassembly.ly)
- [Sweethatclub.com](https://sweethatclub.com)

FIXED LAYOUT

- Used up to this point
- Relies on a container of fixed width
- Usually 960px or 980px

RESPONSIVE LAYOUT

- Different styles(css) for different screen widths
- Uses an elastic/fluid layout (or a fixed layout designed for each screen width range)
- Sized in percentages, or fixed but takes all screen sizes into account
- Mobile-first

MOBILE FIRST

Why design and style the mobile-sized version of the website first?

MOBILE FIRST

<https://www.fortnumandmason.com/>

Note progression of complexity:

- image stacked on top of text
- image and text side-by-side
- 2-column (image stacked on top of text)
- 3-column
- wider 3-column

MEDIA QUERIES

MEDIA QUERIES

@media

Media-queries used to be pretty complicated. They involved multiple clauses separated with “and”

```
@media tv and (min-width: 700px) and (orientation: landscape) {  
  /* css goes here */  
}
```

MEDIA QUERIES

As the world moves towards websites that look good on ANY device type, differentiating between `tv` and `screen` and `print` no longer matters as much.

Neither does the `orientation` of the device, as there's now iPads so big that when they're held vertically, they're still larger than some "landscape" laptops.

MEDIA QUERIES

Now much simpler: the only thing that really matters is how much ROOM you have to play with:

```
@media (min-width: 650px) {  
  /* css goes here */  
}
```

STANDARD MEDIA QUERIES SIZES

- smallest screen is old iPhones at 320px
- wider mobiles at around 480px
- vertical tablets at about 650px
- horizontal tablets and smaller laptops: 940px
- larger devices: 1220px +

Keep these in mind, but do what's best for your design

MOBILE DISPLAY

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

Why necessary?

- Old mobile browser assumption of fixed layout of 980px (width=device-width overrides that).
- initial-scale=1 deals with the fact that some devices have twice as many pixels.

BASIC USAGE

```
.box {  
  display: block;  
  width: 100%;  
}  
  
@media (min-width: 650px) { /* when there's enough room, change to tw  
  .box {  
    display: inline-block;  
    width: 50%;  
  }  
}
```


CSS SPECIFICITY

<http://cssspecificity.com/>

CSS SPECIFICITY

- I nline styles:

```
<p style="background-color: red;">text</p>
```

- S tyle tag:

```
<head><style>div { background-color: red; }</style></head>
```

- I Ds:

```
#red { background-color: red; }
```

- C lasses:

```
.red { background-color: red; }
```

- E lements:

```
div { background-color: red; }
```

CSS SPECIFICITY

Avoid the headache:

- separate file, never mixed into the HTML
- use classes more than IDs or element styling
- bottom overrides top (cascade order)
- media queries add no specificity, but best to put them at bottom of page and use mobile-first approach (that way you're adding CSS complexity as the screen gets larger, and as the CSS file reads down)
- NEVER use `!important` (no matter what you might read online)



RESPONSIVE ARTICLES

OPTIONAL

CSS Transitions (make media query changes smoother)