

- Mobile First,
A Flexible Mindset



Hello!

I'm Brad

Front-End Warlock at Vixlet



Why do we care?

Mobile Dominance in Today's Society

“

*39% of the world will have smartphones by the start of 2015, and over **301.3 million** were sold in both Q2 & Q3 of 2014.*

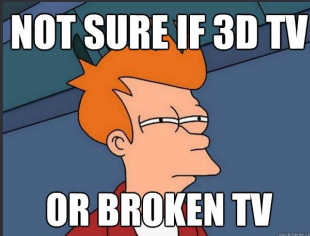
-IDC Research Analysis

- Programmers <3 Math

Let's think about this - today alone...



371,124 Babies Born



559,891 TVs Sold



165,042 Cars Produced



3,347,778 Mobile Devices were Sold

- **9x** over today's birthrate
- **6x** more than TVs sold today
- **20x** more than today's car production

● Mobile Web

○ Zillow

Recently stated that 45% of all traffic on their listings was from mobile devices.

Vixlet + MLB

In a recent contest we ran for World Series Tickets, the traffic average saw...

- 59% Mobile
- 17% Tablet
- 24% Desktop

2

So How Do We Start?

Starting Light = Ending Up With Less

● So How Do We Start?

- Limit projects to smaller processor & ram specs
- Limit plugin usage to essentials when possible
- Prioritize info so it's streamlined for implementation as well as user consumption.



3

Every Action Begets A Reaction
Choose your stylization with purpose

Even CSS can induce page rendering latency:



Compounded styles exponentially increase paint times.

4

Under The Hood

Technical Tidbits to Streamline The
Workflow

- Under The Hood

- **A Look at Your Thumbs**

Most elements for thumbs tend to be happiest at 44-50px.

- **Pseudo Classes Rock**

Whenever possible, use CSS instead of JS.
Comma divided lists for simple example:

```
.like-list a:not(:last-child):after {  
  content: “,”;  
}
```

Ex: Ally, Greg, Joe like this.

● CSS Columns

Responsive layouts can benefit greatly from columns as they help you collapse sections of text gracefully without needing many <p>'s

```
article {  
    /* 2 col, 200px wide min (otherwise stack) - Column-count & column-width */  
    -webkit-columns: 2 200px;  
    -moz-columns: 2 200px;  
    columns: 2 200px;  
  
    /* Gutter space between columns (Defaults to 1em) - Note: Can't do negatives */  
    -webkit-column-gap: 4em;  
    -moz-column-gap: 4em;  
    column-gap: 4em;  
  
    /* Sets the divider between columns should you desire one */  
    -webkit-column-rule: 1px dotted #ddd;  
    -moz-column-rule: 1px dotted #ddd;  
    column-rule: 1px dotted #ddd;  
}
```

Min & Max-width Are Your friends!

Setting up for tons of screen sizes will make you go insane, responsive & mobile first are basically twin brothers.

```
/* CONTAINERS - Keep container at 90% of boundary width, not always at 800px */  
.container {  
    max-width: 90%;  
    width: 800px;  
}
```

```
/* IMAGES - Keep responsive in containing divs as they collapse through sizes*/  
img {  
    max-width: 100%;  
    height: auto;  
}
```

```
/* NOTE : For ie8: add in width:auto or do... */  
@media /0screen {  
    img { width: auto; }  
}
```

```
/* INPUTS - Keep Inputs responsive, but give widely usable structure to them */  
input {  
    max-width: 500px;  
    min-width: 240px;  
    width: 100%;  
}
```

● Viewport Based Parameters are Helpful!

vw & vh are percentages based upon the viewport width and height respectively, and work well in responsive environments.

: SIMPLE EXAMPLE USAGES :

```
/* Set a div to be 50% the viewport width, and 25% its height */
```

```
.greg {  
    width: 50vw;  
    height: 25vh;  
}
```

```
/* Makes a border 1% the width of the viewport - say maybe its a  
vignette style wrap around the viewport perhaps */
```

```
.outerborder {  
    border: 1vw solid #0FF;  
}
```


Media Queries & SASS Work Well Together

We love SASS at Vixlet, and we all know about media queries, but when combined properly we can get some powerful voodoo going.

: Example #1 : Simple Extends in Media Breakdowns

SASS:

```
.errorOne {    font-size: 1.2em;    }  
.errorTwo {    font-size: 1.5em;    }
```

```
@media only screen and (max-width: 300px){  
  .errorTwo {  
    @extend .errorOne;  
    color: blue;  
  }  
}
```

Media Query Compiles to:

```
@media only screen and (max-width: 300px){  
  .errorTwo {  
    font-size: 1.2em;  
    color: blue;  
  }  
}
```

: Example #2 : Simple Mixins with Variables & Media Queries

//SASS:

\$break-medium 480px;

\$break-large: 1024px;

```
.profile-pic {  
  float: left;  
  width: 250px;  
  @media screen and (min-width: $break-medium){  
    width: 350px;  
    float: none;  
  }  
  @media screen and (min-width: $break-large){  
    float: right;  
  }  
}
```

//COMPILES TO:

```
.profile-pic {  
  float: left;  
  width: 250px;  
}  
@media screen and (min-width: 480px){  
  .profile-pic {  
    width: 100px;  
    float: none;  
  }  
}  
@media screen and (min-width: 1024px){  
  .profile-pic {  
    float: right;  
  }  
}
```

● : Example #3 : Whole Query as a SASS Variable

//SASS:

\$info-phone: “only screen and (max-width:320px)”;

```
@media #{$info-phone} {  
  .greg {  
    background: #0ff;  
  }  
}
```

//COMPILES TO:

```
@media only screen and (max-width: 320px){  
  .greg {  
    background: #0ff;  
  }  
}
```

: Example #4 : Let's Step It Up

//SASS:

\$break-small: 320px;

\$break-large: 1024px;

```
@mixin respond-to ($media) {  
  @if $media == handhelds {  
    @media only screen and (max-width: $break-small) { @content; }  
  }  
  @else if $media == medium-screens {  
    @media only screen and  
      (min-width: $break-small + 1) and  
      (max-width: $break-large - 1) { @content; }  
  }  
  @else if $media == wide-screens {  
    @media only screen and (min-width: $break-large) { @content; }  
  }  
}  
  
.profile-pic {  
  float: left;  
  width: 250px;  
  @include respond-to(handhelds) { width: 100%; }  
  @include respond-to(medium-screens) { width: 125px; }  
  @include respond-to(wide-screens) { float: none; }  
}
```

● : Example #4 : Let's Step It Up

//COMPILES TO:

```
.profile-pic {  
  float: left;  
  width: 250px;  
}
```

```
@media only screen and (max-width: 320px) {  
  .profile-pic {  
    width: 100%;  
  }  
}
```

```
@media only screen and (min-width: 321px) and (max-width: 1023px) {  
  .profile-pic {  
    width: 125px;  
  }  
}
```

```
@media only screen and (min-width: 1024px) {  
  .profile-pic {  
    float: none;  
  }  
}
```

Pixel Density & Images

For mobile, retina and serving things in various pixel densities has become commonplace. This is a simplified example of the usage.

: Example : Retina Usage with single query

```
/* Regular Resolution image setup */
```

```
.my-image {  
    background-image: url('/img/non-retina.jpg');  
}
```

```
/* Retina screens, sometimes its better support to do 1.5 as this hits certain android  
screens as well */
```

```
@media
```

```
(-webkit-min-device-pixel-ratio: 2),
```

```
(min-resolution: 192dpi){
```

```
    .my-image {  
        background-image: url('/img/retina@2x.jpg');  
    }
```

```
}
```

● SVG is Easy

○ SVG is basically a vector on a webpage, able to easily scale to any size and lets you avoid fooling with retina images altogether.

- IE 9+, Android 2.3+ support
- Color svg using styling even
- Directly embed code or use as img
- Save directly from Illustrator easily now

: Example : SVG as background image

```
.main-header {  
    background:url(logo.svg) top left / contain no-repeat;  
}  
  
.no-svg .main-header {  
    background-image: url(logo.jpg);  
}
```

Resources for Covered Topics



Media Queries

Good site for breakdowns of media queries should you seek some example stuff on them.

<http://cssmediaqueries.com/overview.html>



More on Paint Times

This is where a lot of the timings for the css items paint times came from, great article full of info.

<http://www.html5rocks.com/en/tutorials/speed/css-paint-times/>



SASS Mastery

Thesassway.com is a great site for learning SASS in an intuitive and easy fashion, with many examples borrowed from here.

<http://thesassway.com/intermediate/responsive-web-design-in-sass-using-media-queries-in-sass-32>



Boston Globe

Good example of collapsed responsive, the weather item collapses in a unique, but intuitive manner, first showing up in the navbar in a nice

<http://brolik.com/blog/responsive-web-design-examples-with-css-tips-and-tricks/>



Using SVG

Good resource on the utilization and implementation of SVG Graphics

<http://css-tricks.com/using-svg/>



Discover Mobile

Good walkthrough of a solid design process start to finish, showcases the whole design flow.

<http://kathrynjamesfaulkner.com/mobile-first/index.html>

Thanks!

ANY QUESTIONS?

brad@vixlet.com

