Sass: CSS with superpowers

Ynon Perek

Why Sass?

- An extensible superset of CSS
- Used by Frameworks

What's Wrong with CSS?

```
body {
   background: #A86C58;
}

h1,h2 {
   color: #104A5D;
}

p:hover {
   background: #104A5D;
   color: #A86C58;
}
```

- CSS is full of repeating values and derived values

What's Wrong with CSS?

- CSS is repetitive by nature
- Colors are defined all over the place
- Sizes are defined and calculated everywhere
- There's no way to handle calculated properties
- => Hard to maintain and modify

Fixing CSS

Hello Sass

- Both seissmeister and codepen are online tools to convert scss to css
- http://sassmeister.com/
- http://codepen.io/
- Demo

Fixing Repetitive CSS

- The following scss code translates to your old repetitive CSS from the previous example

- Alternatively, use calculated properties: http://codepen.io/anon/pen/KdqdzW

```scss
$page-bg: #A86C58;
$page-fg: #104A5D;

body {
  background: $page-bg;
}

h1,h2 {
  color: $page-fg;
}

p:hover {
  background: $page-fg;
  color: $page-bg;
}
```

Sass Comments

- CSS comments are marked with /* ... */ and are part of the CSS file
- Sass comments are marked with // and are not part of the resulting CSS file

```
.my-button {
  // this line will not be written to the CSS
  /* but this line will */
  @include button(rem-calc(8));
}
```

Sass Nesting

- Sass tries to make CSS more compact by reducing repetetive code

- One common repetition case is that of selectors

```
nav {
  background: lightblue;
  padding: 5px;
}

nav ul {
  list-style:none;
}

nav li {
  padding: 5px;
}

nav p {
  color: darkblue;
}
```

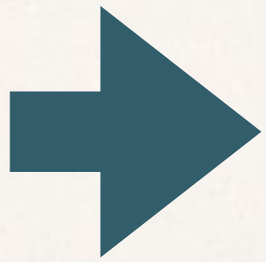
Sass Nesting

```
nav {
  background: lightblue;
  padding: 5px;

  ul {
    list-style:none;
  }

  li {
    padding: 5px;
  }

  p {
    color: darkblue;
  }
}
```



```
nav {
  background: lightblue;
  padding: 5px;
}

nav ul {
  list-style:none;
}

nav li {
  padding: 5px;
}

nav p {
  color: darkblue;
}
```

Sass Nesting: &

```sass
a {
  display:block;
  text-decoration:none;
  font-size: 24px;
  margin: 10px;
  color:blue;

  &:hover,&:focus {
    background: blue;
    color:white;
  }
}
```

```css
a {
  display: block;
  text-decoration: none;
  font-size: 24px;
  margin: 10px;
  color: blue;
}

a:hover, a:focus {
  background: blue;
  color: white;
}
```

Lab #1

- Use Sass to clean the following CSS:
  - Change comments so they're not part of the final CSS file
  - Use variables to give meaningful names to colors
- http://codepen.io/ynonp/pen/PPjQqa

Lab #2

- Use Sass to clean the following CSS:
  - Nest selectors
  - Use variables to give meaningful names values
- http://codepen.io/ynonp/pen/OygQNo

Q & A

Sass Variables

Defining and Using

```
$color: blue;

nav {
  a { color: $color };
}

// $color is blue
p {
  color: $color;
}
```

Scope

```scss
$color: blue;

nav {
  // $color is red
  $color: red;

  a { color: $color };
}

// $color is blue
p {
  color: $color;
}
```

Calculated Values

- Multiply unitless values with unit values to add the units
- `1px * 10 = 10px`
- `1s * 10 = 10s`
- Can also use +/-
- `10px + 10px = 20px`
- `4px + 1in = 100px`

```
$sz_a: 10;

.a {
  width:1px * $sz_a;
  height:1px * $sz_a;
}
```

Calculated Values

- Unit values can be used as-is
- Conversion table: https://github.com/sass/sass/blob/31c6c531fabc1538d37db334b2cf8b976e289c3b/lib/sass/script/value/number.rb#L455-L483

```
$sz_b: 10px;
$sz_c: 4rem;

.b {
  width:$sz_b;
  height:$sz_b;
}

.c {
  width:$sz_c;
  height:$sz_c;
}
```

Sass Functions

```
p {
  color: hsl(0, 100%, 50%);
}
```

Sass Functions

```
a {
  color: blue;

  &:hover,&:focus {
    color: darken(blue, 20);
  }
}
```

```
a {
  color: blue;
}
a:hover, a:focus {
  color: #000099;
}
```

Sass Functions: Colors

- `grayscale($color)`
- `complement($color)`
- `invert($color)`
- `lighten($color, $amount)`
- `darken($color, $amount)`
- `opacify($color, $amount)`
- `transparentize($color, $amount)`

Sass Functions: Numbers

- `round($number)`
- `ceil($number)`
- `floor($number)`
- `min($numbers…)`
- `max($numbers…)`
- `random([$limit])`

Sass Functions: Other

- `variable-exists($name)`
- `function-exists($name)`
- `mixin-exists($name)`
- `unit($number)`
- `unitless($number)`
- `comparable($number1, $number2)`
- `unique-id()`

Lab #1

- Use Sass to clean the following CSS of repetitive code:

- http://codepen.io/ynonp/pen/JYJGxR

Lab #2

- Create a project with .scss stylesheet
- Start with the colour: #670dd9
- Paint 5 boxes each is 20% lighter than the previous one

Sass Resources

- Quick reference for common functionality: http://sass-cheatsheet.brunoscopelliti.com/

- Colors: http://jackiebalzer.com/color

Q & A

Sass Import

layout.scss

modules.scss

reset.scss

ugly.scss

main.css

Working With Multiple Files

- The @import directive includes another file
- It's like a CSS @import, but much better:
  - No browser round-trip to get included file
  - Can use mixins and variables from included file
- Can still use old @import if file extension is css

Partials

- In a multiple file scenario, only one CSS file is required
- So we'll have:
  - One main.scss file
  - Multiple _partial.scss files

@import

main.scss

```
@import "layout";

body {
  background: $bg-color;
}
```

_layout.scss

```
.block-center {
  margin: 0 auto;
}

.inline-center {
  text-align: center;
}
```

Sass Mixins

- A Mixin is a collection of CSS rules tied together in a single name
- Useful in code re-use
- Can also take arguments to make them event more generic

Hello Mixins

```
@mixin large-text {
  font: {
    family: Arial;
    size: 20px;
    weight: bold;
  }
  color: #ff0000;
}
```

```
.page-title {
  @include large-text;
  padding: 4px;
  margin-top: 10px;
}
```

Mixins -> CSS

```
@mixin large-text {
  font: {
    family: Arial;
    size: 20px;
    weight: bold;
  }
  color: #ff0000;
}

.page-title {
  @include large-text;
  padding: 4px;
  margin-top: 10px;
}
```

```
.page-title {
  font-family: Arial;
  font-size: 20px;
  font-weight: bold;
  color: #ff0000;
  padding: 4px;
  margin-top: 10px;
}
```

Mixins can include other Mixins

```
@mixin compound {
  @include highlighted-background;
  @include header-text;
}

@mixin highlighted-background { background-color: #fc0; }
@mixin header-text { font-size: 20px; }
```

Mixins can have @content

```
@mixin bp-large {
  @media only screen and (max-width: 60em) {
    @content;
  }
}
```

```
@mixin bp-medium {
  @media only screen and (max-width: 40em) {
    @content;
  }
}

@mixin bp-small {
  @media only screen and (max-width: 30em) {
    @content;
  }
}
```

So you can use:

```
.sidebar {
  width: 60%;
  float: left;
  margin: 0 2% 0 0;
  @include bp-small {
    width: 100%;
    float: none;
    margin: 0;
  }
}
```

And take arguments

```
@mixin my-small-rounded-corners($r:5px) {
  -moz-border-radius: $r;
  -webkit-border-radius: $r;
  border-radius: $r;
}

.rounded {
  @include my-small-rounded-corners(8px);
}
```

Mixins use functions

```
@function topx($val) {
  @if unitless($val) {
    @return $val * 1px;
  }
  @else {
    @return $val;
  }
}


@mixin radius($val) {
  border-radius: topx($val);
}
```

Lab: Write the following Mixins

- `abs-pos(top,right,bottom,left)` should apply absolute positioning on an element with top, right, bottom and left values provided

- `text-truncate` mixin should truncate the text in the element using an ellipsis

- `link-color(normal,hover)` should define normal, hover and focus colours for a link

Q & A

Foundation

Ynon Perek. ynon@ynonperek.com

Foundation Provides

- Classes to build components / layout
- Mixins to integrate into your CSS

Foundation Grid

1	2	3	4	5	6	7	8	9	10	11	12

Header .small-12.columns

Main Content
.large-8.columns

Sidebar
.large-4.columns

Grid Layout

- Order screen in columns
- Stack columns automatically according to breakpoints
- demo

Grid + Sass

- With sass we can integrate Foundation styles into our CSS

```
.sidebar {
  @include grid-column(3);
}

.main {
  @include grid-column(9);
  background: orange;
}

.container {
  @include grid-row;
}
```

Grid + Sass + Flexbox

- Flexbox based grid provides many more options compared with traditional grid

- https://foundation.zurb.com/sites/docs/flex-grid.html

```
.flex {
  .sidebar {
    @include flex-grid-column(3);
  }

  .main {
    @include flex-grid-column;
    background: orange;
  }

  .container {
    @include flex-grid-row;
  }
}
```

Grid + Sass + Breakpoints

```
.container {
  @include grid-row;

  .sidebar {
    @include grid-column(3);

    @include breakpoint(small only) {
      @include grid-column(6);
    }
  }
  .main {
    @include grid-column(9);
    @include breakpoint(small only) {
      @include grid-column(6);
    }
    background: orange;
  }
}
```

Nested Grid

```
<div class="row text-center">
    <div class="small-4 columns">one-third</div>

    <div class="small-4 columns">
        <div class="row">
            <div class="small-6 columns">half of one-third</div>
            <div class="small-6 columns">half of one-third</div>
        </div>
    </div>

    <div class="small-4 columns">one-third</div>
</div>
```

http://codepen.io/ynonp/pen/OygzGZ

Visibility Classes

- The following classes can be used to hide/show items based on current screen size:

```
.show-for-small-only .show-for-medium-up .show-for-
medium-only .show-for-large-up .show-for-large-only
.show-for-xlarge-up .show-for-xlarge-only .show-
for-xxlarge-up .hide-for-small-only .hide-for-
medium-up .hide-for-medium-only .hide-for-large-
up .hide-for-large-only .hide-for-xlarge-up .hide-
for-xlarge-only .hide-for-xxlarge-up
```

- The following classes show/hide items based on touch capability:

```
.show-for-touch .hide-for-touch
```

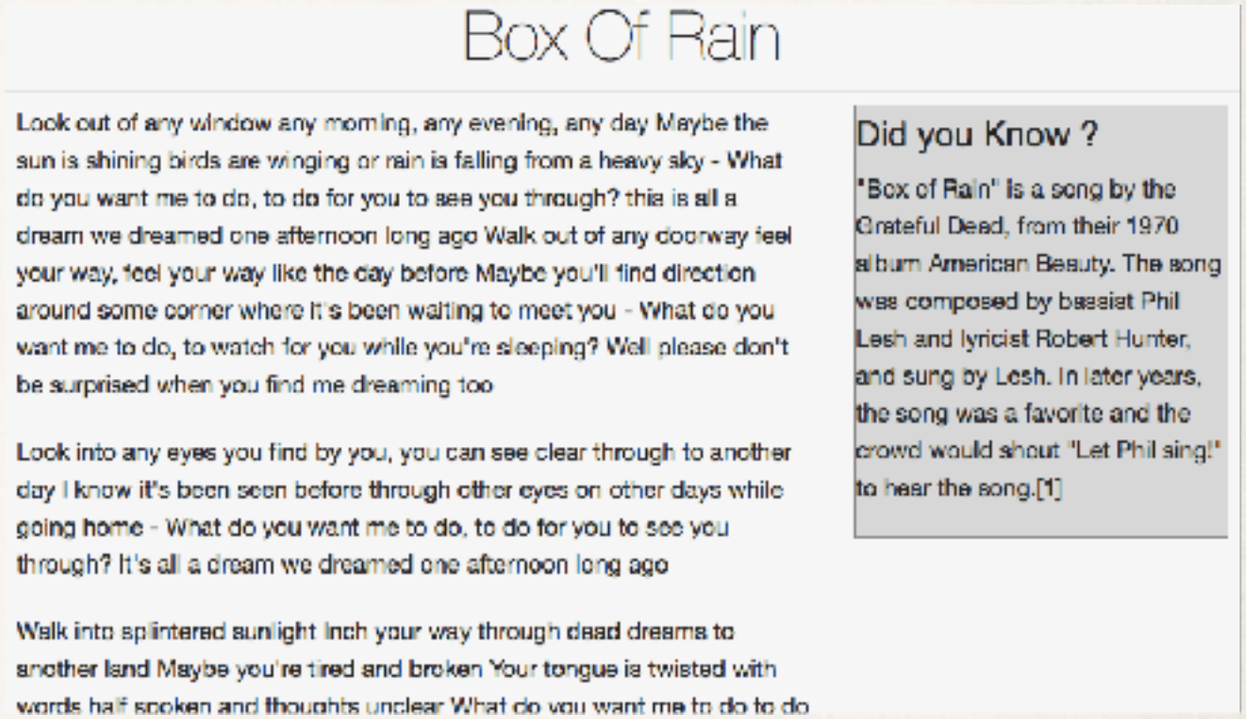
Block Grid

- Block grid “fills” the entire row with content

- Demo: http://codepen.io/ynonp/pen/bVvyzg

Grid Lab #1

- Implement a main content and sidebar layout
- For smaller screens place sidebar below content
- Starter: http://codepen.io/ynonp/pen/QjGrGL?editors=110



Box Of Rain

Look out of any window any morning, any evening, any day Maybe the sun is shining birds are winging or rain is falling from a heavy sky - What do you want me to do, to do for you to see you through? this is all a dream we dreamed one afternoon long ago Walk out of any doorway feel your way, feel your way like the day before Maybe you'll find direction around some corner where it's been waiting to meet you - What do you want me to do, to watch for you while you're sleeping? Well please don't be surprised when you find me dreaming too

Look into any eyes you find by you, you can see clear through to another day I know it's been seen before through other eyes on other days while going home - What do you want me to do, to do for you to see you through? It's all a dream we dreamed one afternoon long ago

Walk into splintered sunlight inch your way through dead dreams to another land Maybe you're tired and broken Your tongue is twisted with words half spoken and thoughts unclear What do you want me to do to do

Did you Know ?

"Box of Rain" is a song by the Grateful Dead, from their 1970 album American Beauty. The song was composed by bassist Phil Lesh and lyricist Robert Hunter, and sung by Lesh. In later years, the song was a favorite and the crowd would shout "Let Phil sing!" to hear the song.[1]

Grid Lab #2

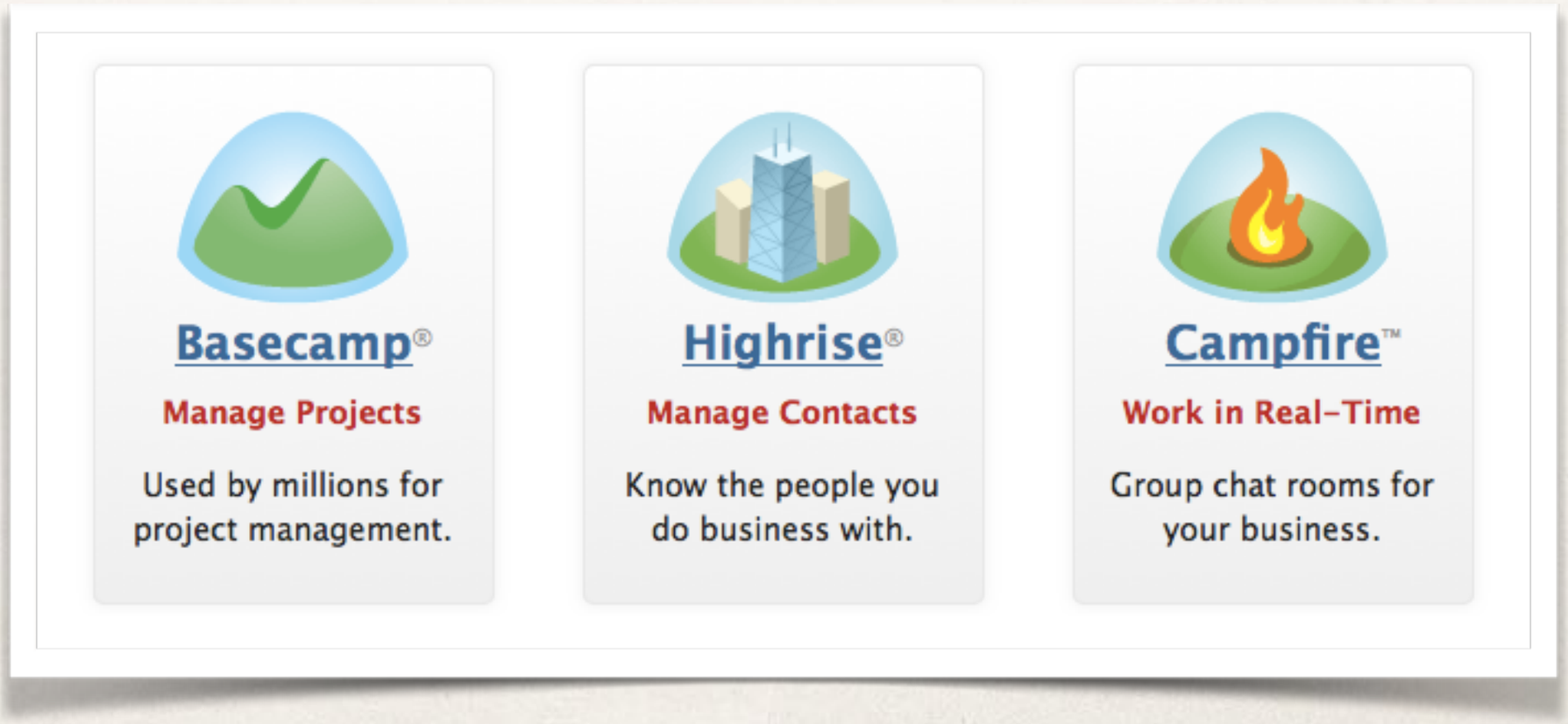
- Implement a list of images and titles.
- When more screen is available title should be placed next to the image
- Starter: http://codepen.io/ynonp/pen/NGbMbQ

small screen

medium screen

Grid Lab #3

- Implement HTML and CSS for the following using a block grid
- Starter: http://codepen.io/ynonp/pen/PPbemL



Q & A

Responsive Components

- Forms
- Images
- Navigation
- Video

Responsive Forms

- We need forms to work for everyone

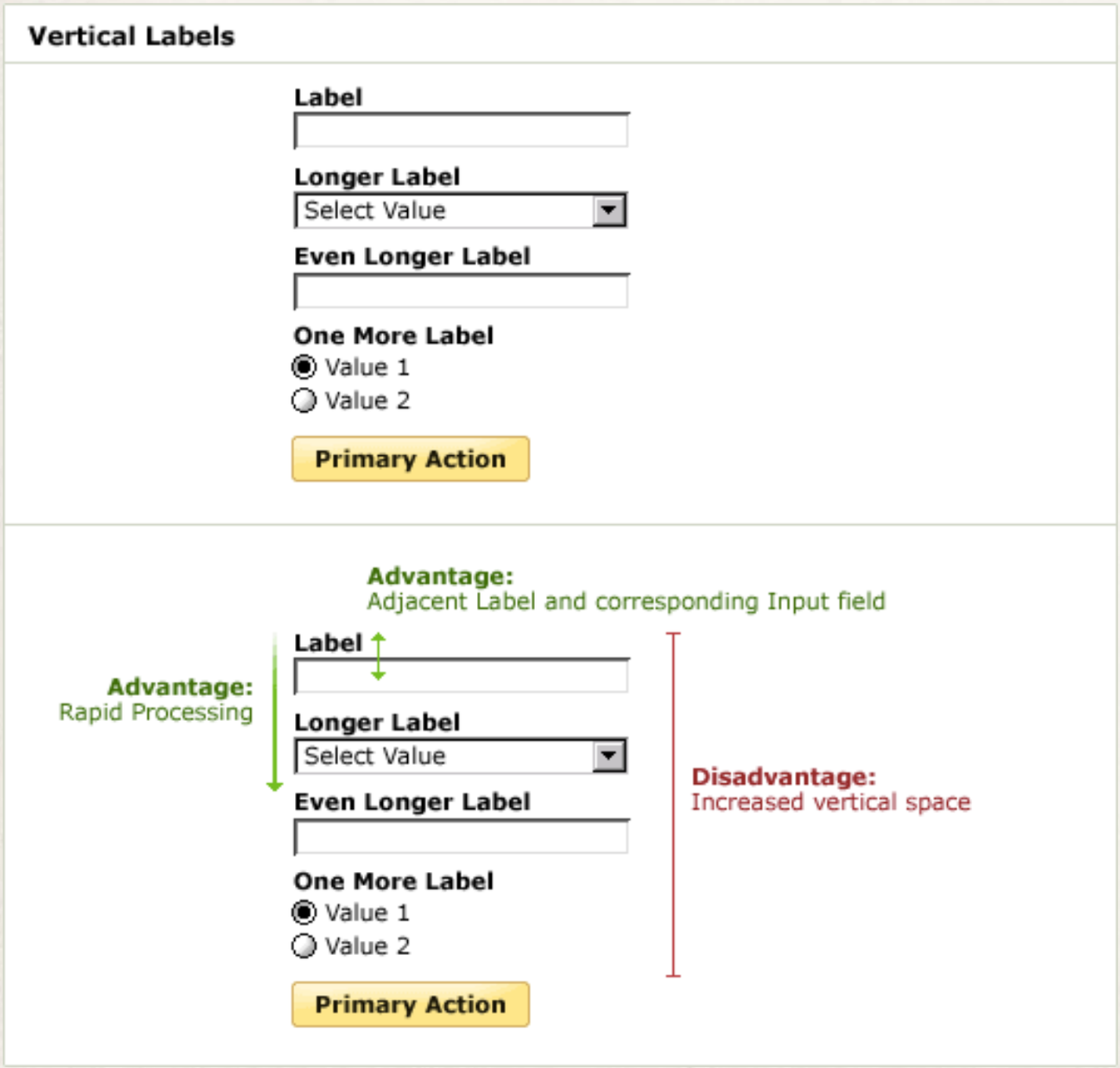
Responsive Forms

- We need forms to work for everyone
- Show on all devices
- Native form controls
- Client-side validation
- Easy to use

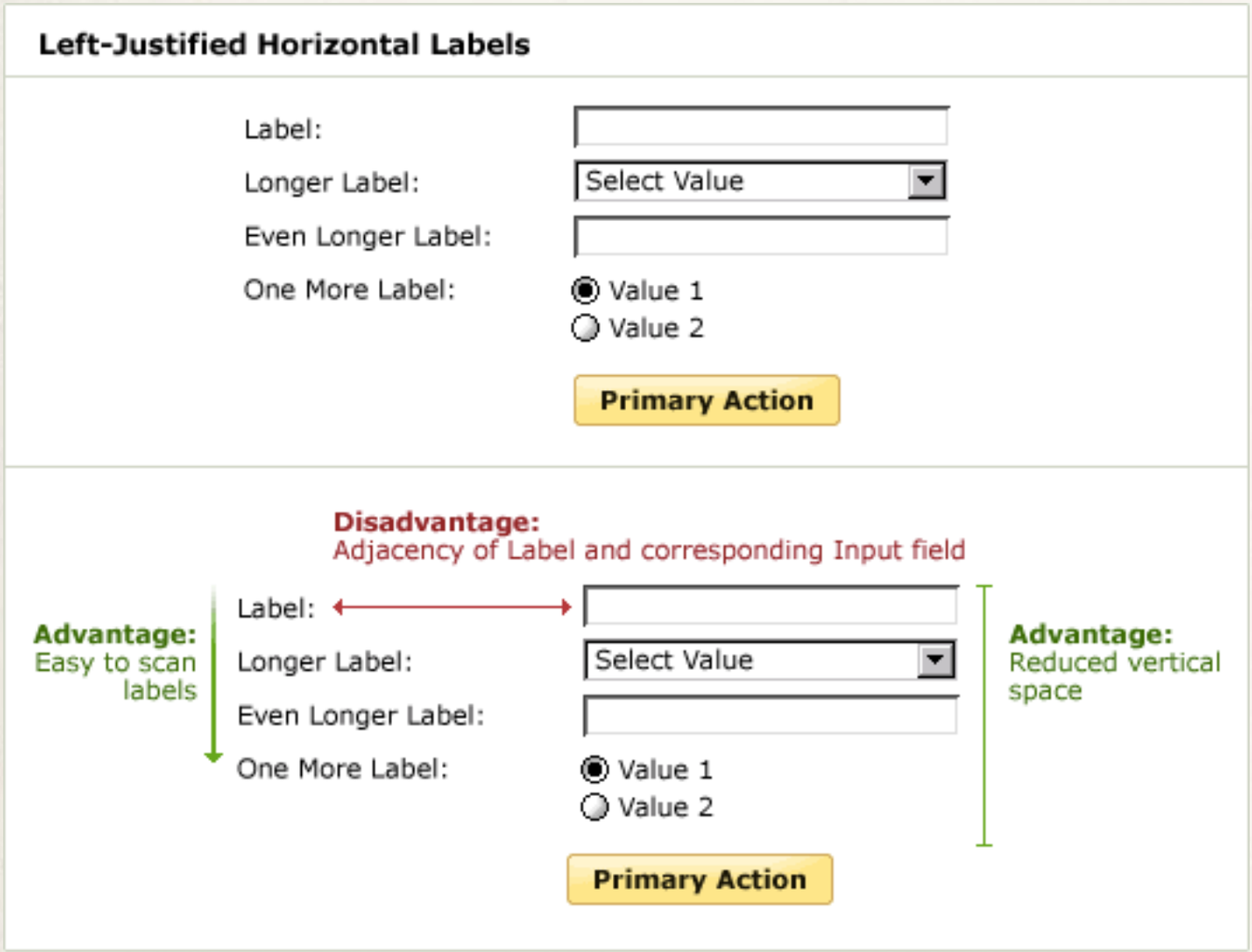
Responsive Forms: Foundation

- Style is determined by input[“type”] so usually no classes are required
- Use normal grid for layout

Responsive Forms: Vertical

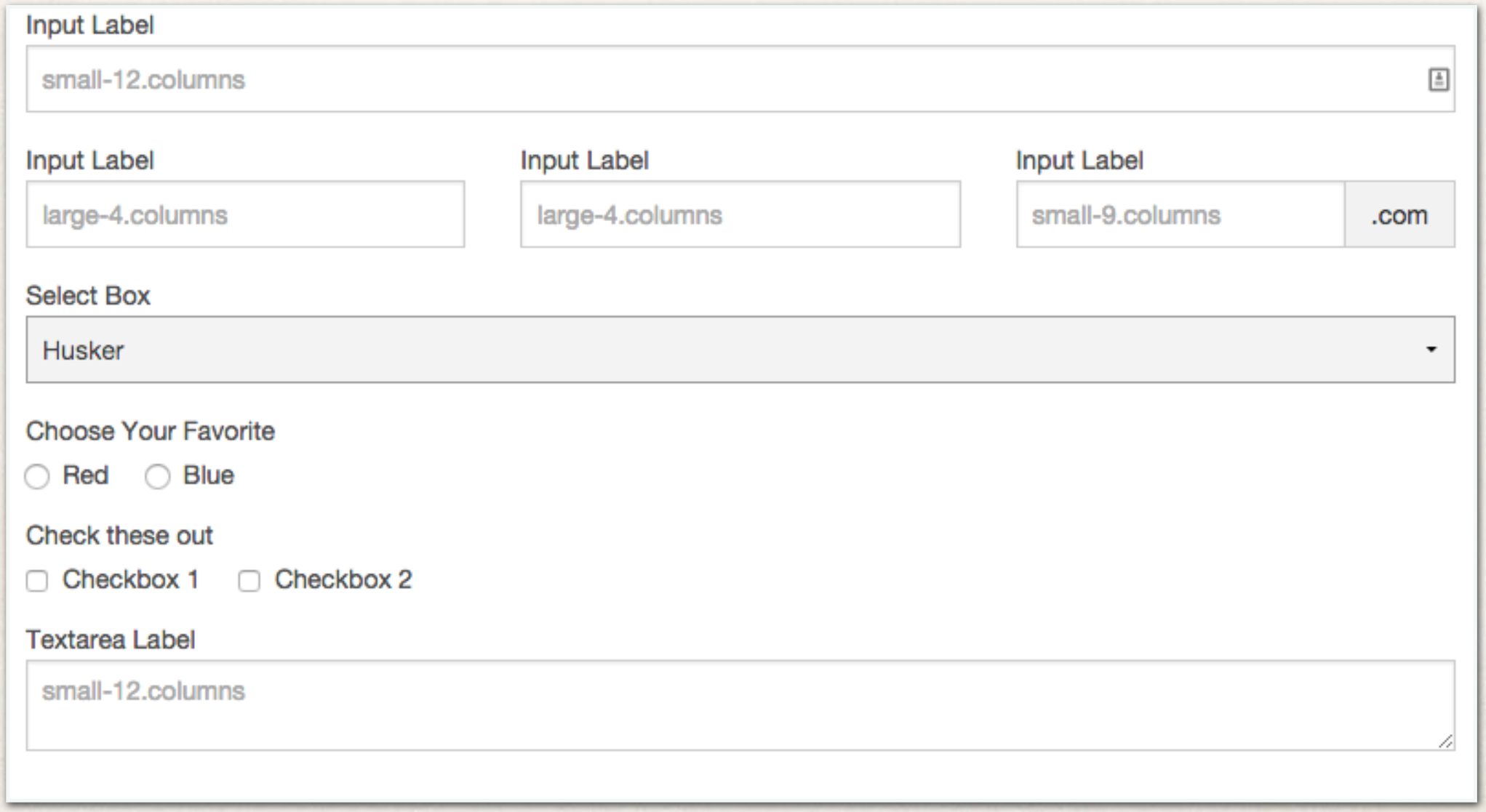


Responsive Forms: Horizontal



Forms: Demo

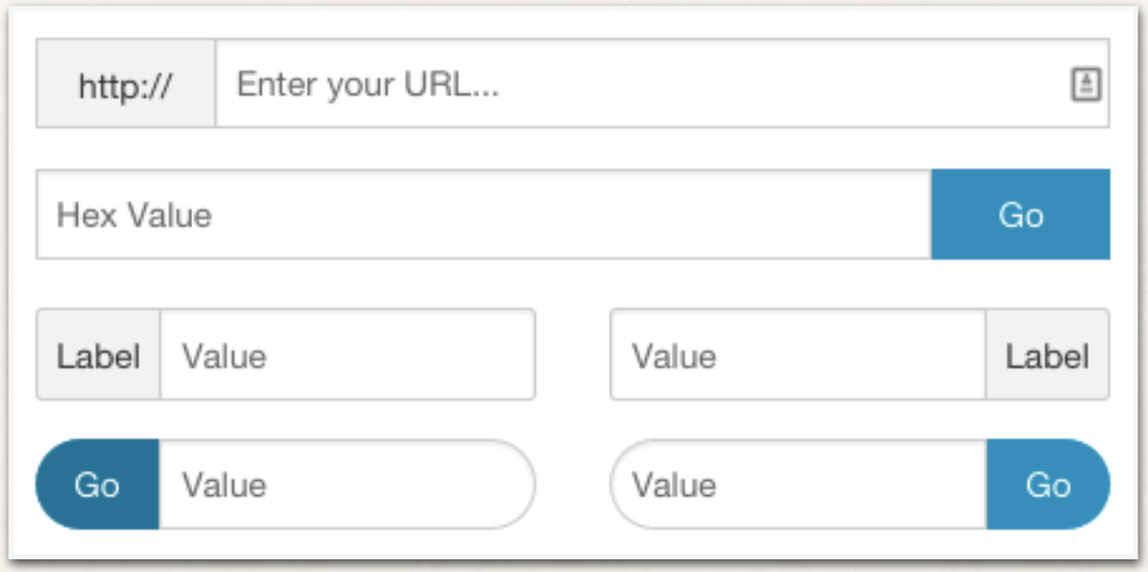
- Let's build the form below



Code: http://codepen.io/ynonp/pen/LpbrOo

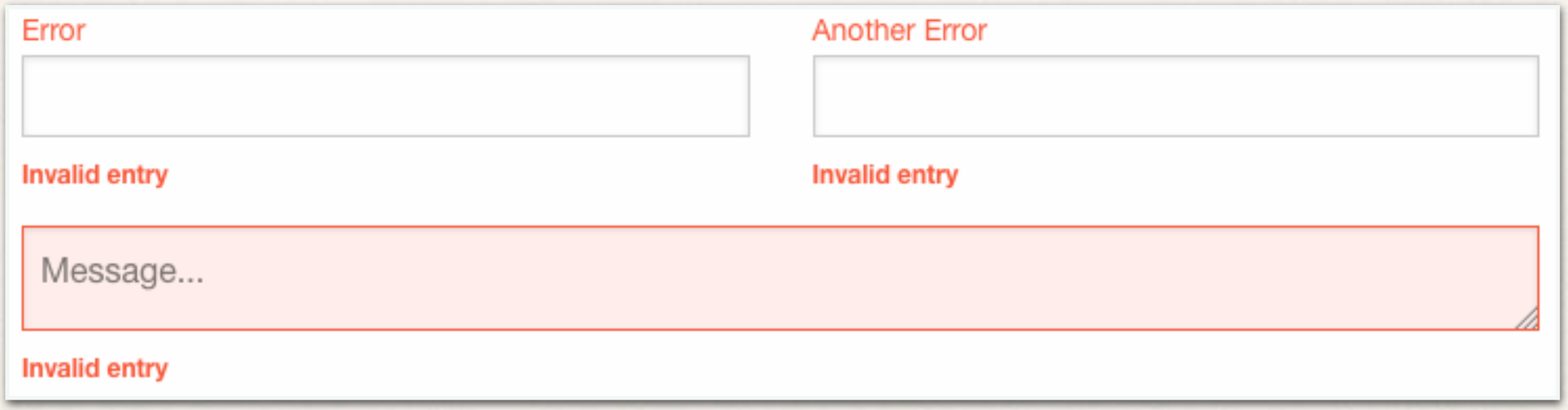
Other Things You Can Do

- Prefix / Postfix actions (by using an <a> with class="button")



Other Things You Can Do

- Error states (http://codepen.io/ynonp/pen/KdNeBQ)



Create on/off switches

```
<div class="switch">
  <input class="switch-input"
         id="exampleSwitch"
         type="checkbox"
         name="exampleSwitch">
  <label class="switch-paddle" for="exampleSwitch">
    <span class="show-for-sr">Download Kittens</span>
  </label>
</div>
```

Using Native Form Controls


```
<input type="tel" />
```

Using Native Form Controls


```
<input type="email" />
```

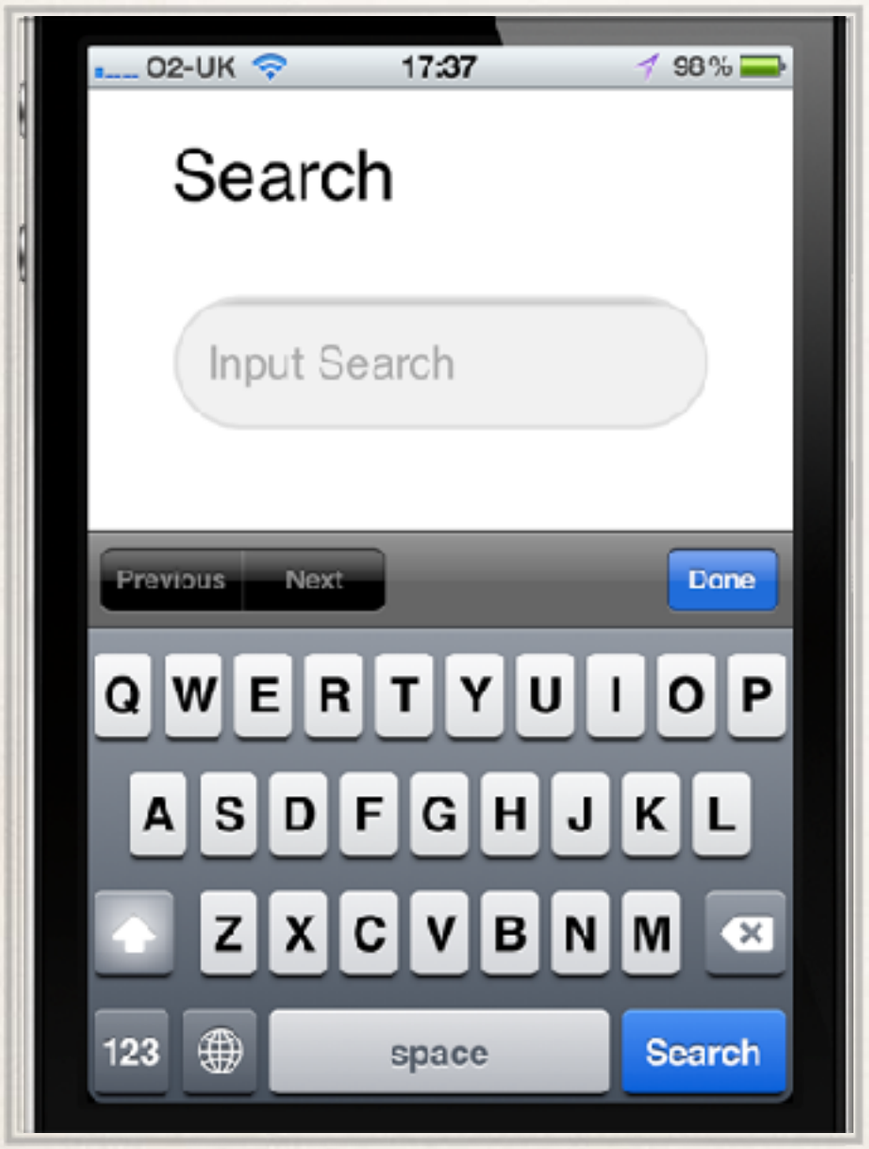
Using Native Form Controls


```
<input type="number" />
```

Using Native Form Controls


```
<input type="date" />
```

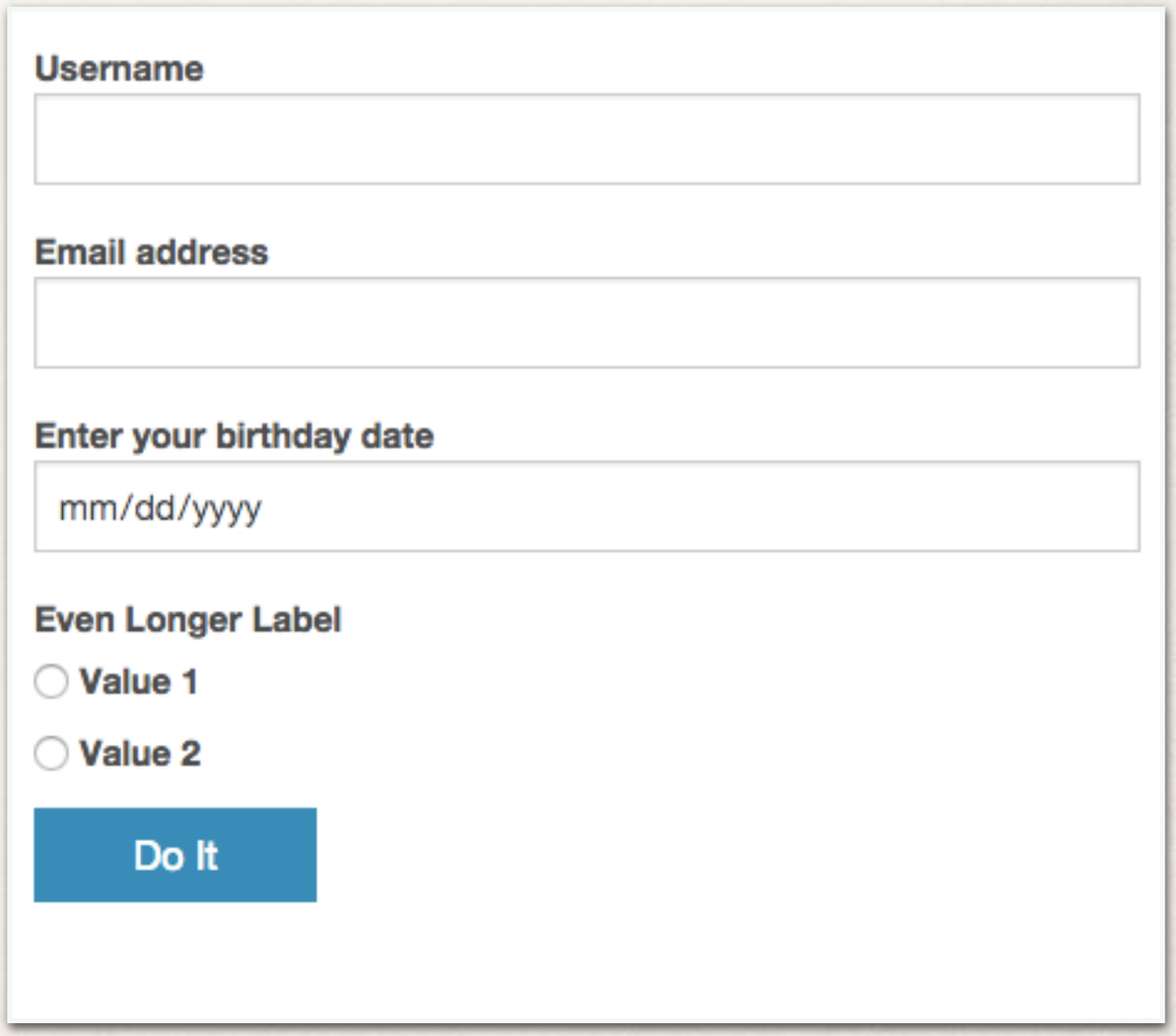
Using Native Form Controls




```
<input type="search" />
```

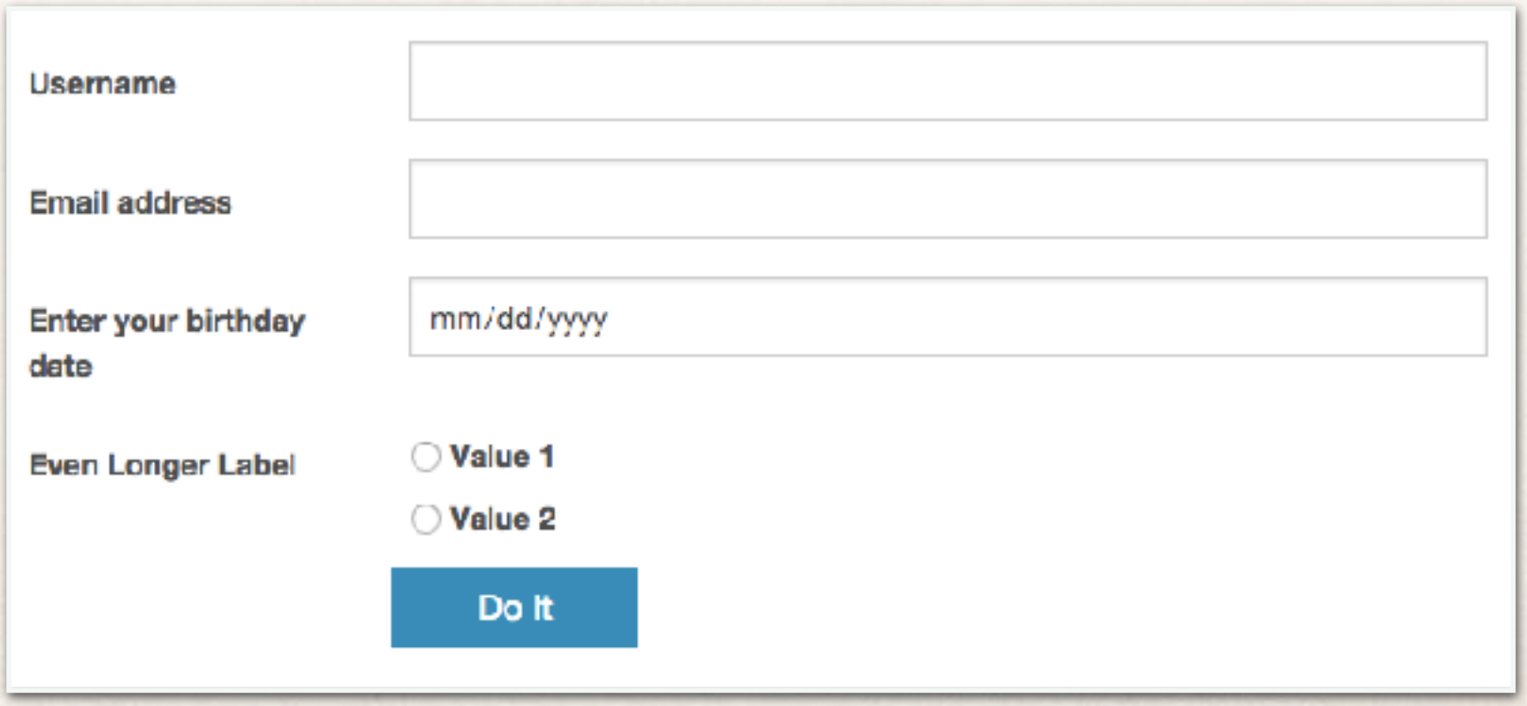
Forms Lab #1

- Create the following vertical form
- Starter: http://codepen.io/ynonp/pen/EVENLd



Forms Lab #2

- Create the following horizontal form
- For smaller screen, form should display as vertical
- Starter: http://codepen.io/ynonp/pen/avYBaP

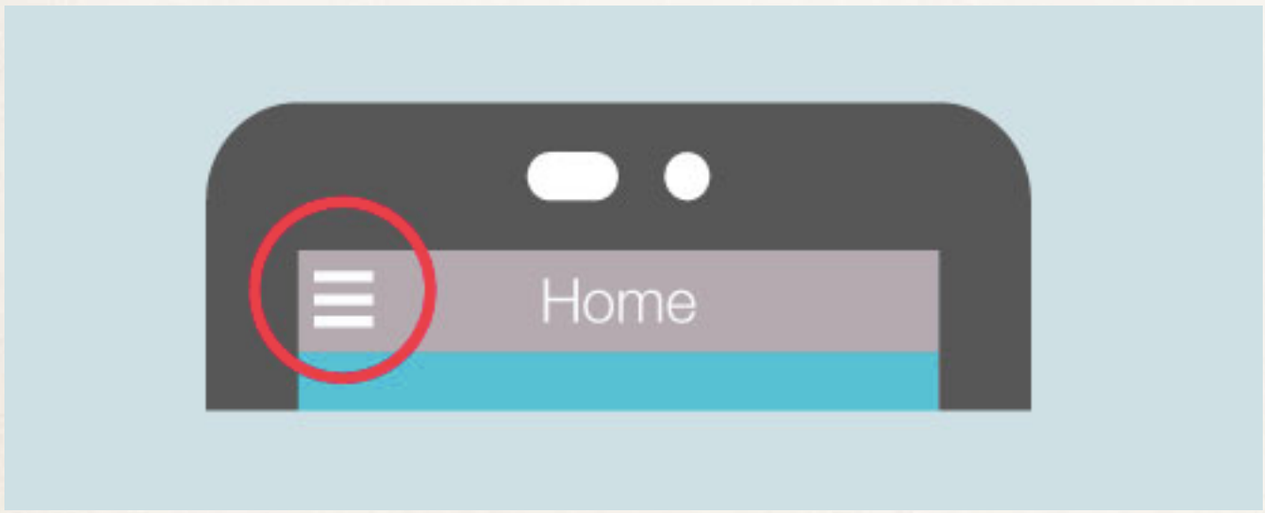


Responsive Navigation

foundation.zurb.com/docs/components/topbar.html

Top Nav What

- Full bar for large screen
- Hamburger menu for small

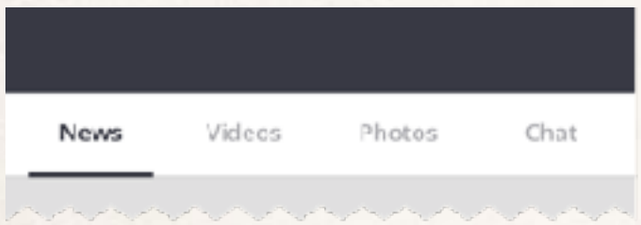


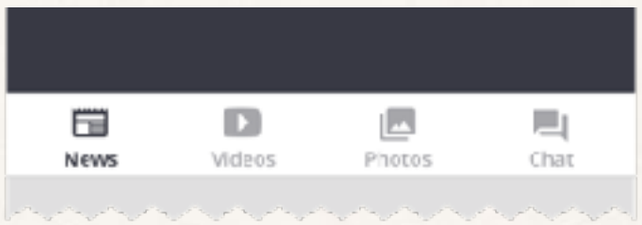
http://codepen.io/ynonp/pen/VvMXWd

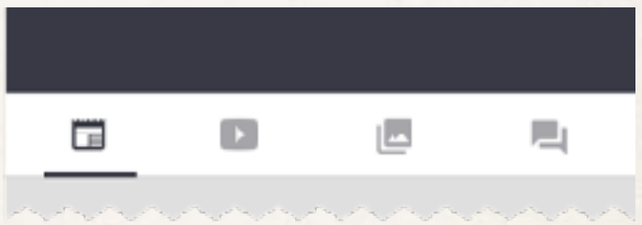
Top Bar Keep In Mind

- JS is required:
  - jQuery
  - Foundation

Tabs







- Use foundation menu + icons
- Demo: http://codepen.io/ynonp/pen/YpxoLj

Responsive Video

- The following CSS is enough to make your <video> responsive

```
video {
  width: 100%     !important;
  height: auto    !important;
}
```

Responsive Video

- However it won't work for iFrames (YouTube embeds etc.)
- Here's a pattern that works for them too

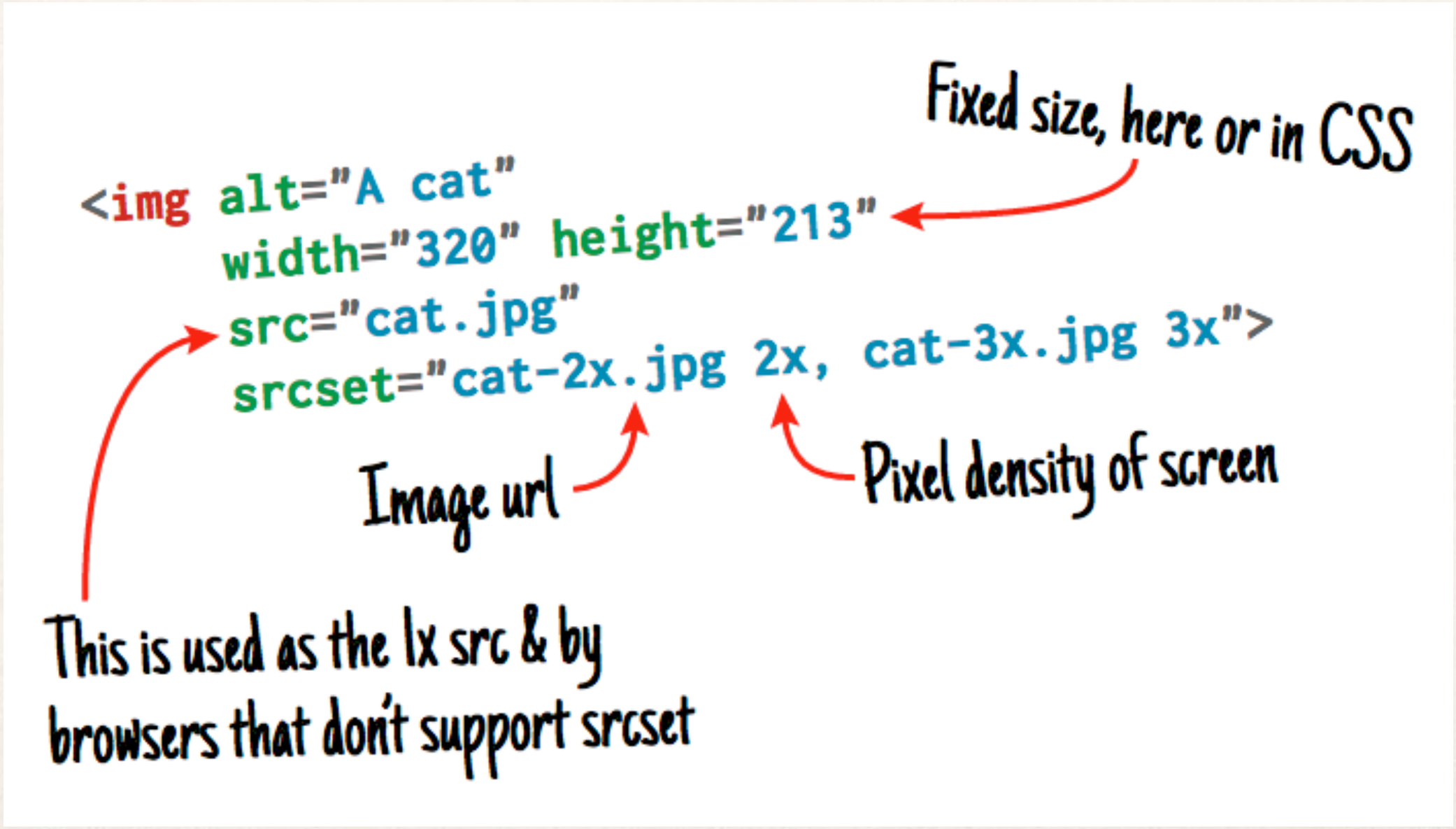
```
.videoWrapper {
    position: relative;
    padding-bottom: 56.25%; /* 16:9 */
    padding-top: 25px;
    height: 0;
}
.videoWrapper iframe {
    position: absolute;
    top: 0;
    left: 0;
    width: 100%;
    height: 100%;
}
```

```
<div class="videoWrapper">
    <!-- Copy & Pasted from
YouTube -->
    <iframe width="560"
height="349" src="http://
www.youtube.com/embed/n_dZNLr2cME?
rel=0&hd=1" frameborder="0"
allowfullscreen></iframe>
</div>
```

Responsive Images

- Use srcset to define multiple sources for different screen sizes
- Use sizes to help browser pick correct one
- Demo: http://codepen.io/ynonp/pen/NqGrro?editors=100

Responsive Images



Responsive Images #2

```
<img alt="A red panda eating leaves"
     src="panda-689.jpg"
     srcset="panda-689.jpg 689w,
             panda-1378.jpg 1378w,
             panda-500.jpg 500w,
             panda-1000.jpg 1000w"
 sizes="(min-width: 1066px) 689px,
        (min-width: 800px) calc(75vw - 137px),
        (min-width: 530px) calc(100vw - 96px),
        100vw">
```

Only for browsers that don't support srcset

Image url

Width of the image data

Fallback width, when no media conditions match

Width of the window

Width of the img element when the condition matches

Responsive Images #3

- Use picture when you need different images for different devices

```
<picture>
  <source
    media="(min-width: 768em)"
    srcset="quilt_2/detail/large.jpg  1920w,
            quilt_2/detail/medium.jpg  960w,
            quilt_2/detail/small.jpg   480w" />

  <source
    srcset="quilt_2/square/large.jpg  822w,
            quilt_2/square/medium.jpg 640w,
            quilt_2/square/small.jpg  320w" />
  <img
    src="quilt_2/detail/medium.jpg"
    alt="Detail of the above quilt, highlighting the embroidery and exotic
stitchwork." />
</picture>
```

Q & A

Illustrations from:
https://jakearchibald.com/2015/anatomy-of-responsive-images/

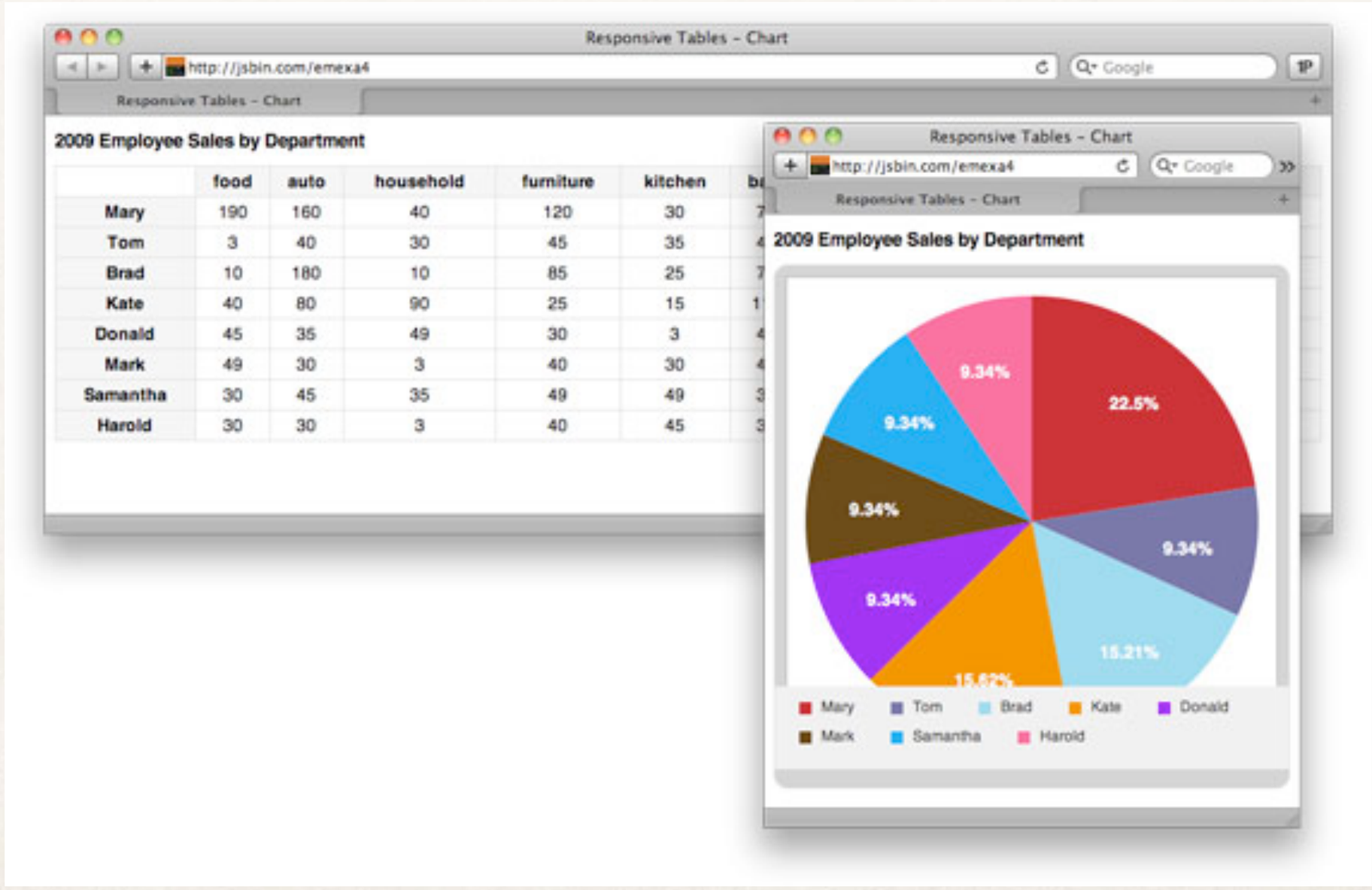
Responsive Data Tables

Ideas?

Option #1: Break the table

- Show separated “record” on smaller screen
- https://css-tricks.com/examples/ResponsiveTables/responsive.php

Option #2: Visualize Differently



Option #3: Inner Scrolls

- Demo: http://zurb.com/playground/projects/responsive-tables/index.html

- Plugin source: https://github.com/zurb/responsive-tables

Responsive Components Lab

- Build a responsive image slider:
  - Show a single image
  - Clicking on the image moves forward to the next one
  - Image resolution is selected according to screen size and slider size

Q & A

Thanks For Listening

- Read more: www.tocode.co.il
- Talk to me: ynon@tocode.co.il
- Photos from: http://123rf.com
- Slides at: http://1drv.ms/1q5grsi