



RICE[®]

Web Development

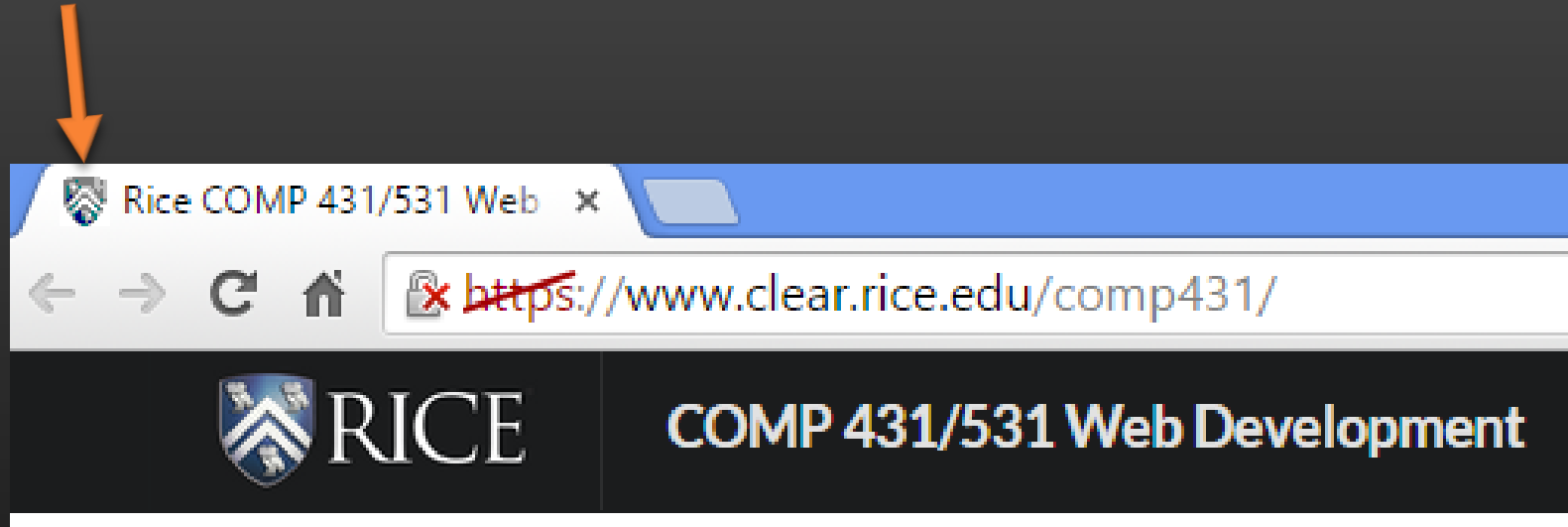
COMP 431 / COMP 531

Cascading Style Sheets

Scott E Pollack, PhD

Reference Slides

Favicon



```
<link rel="icon" href="/favicon.ico" type="image/x-icon">
```

Styling HTML

- We already have `` and ``
- What about changing the text color?
- Background color?
- Style of a table? E.g., border and spacing

Styling HTML

```
<font color="red">A &lt;font> tag can be  
used to change the "font" of some text.</font> This  
<em>still</em> works because browsers are backwards  
compatible and are forgiving. We can even change  
<font size="3" face="verdana">the font face</font>  
or <font size="1"> even the size</font>.  
But the <em>control</em> over these styles  
is made difficult because we have to search  
the DOM for the tags and then modify them.</p>
```

A `` tag can be used to change the "font" of some text. This *still* works because browsers are backwards compatible and are forgiving. We can even change the font face or even the size. But the *control* over these styles is made difficult because we have to search the DOM for the tags and then modify them.

Drive for Style

- We're constructing a web application
- Separation of content from presentation
- We want consistency between pages
- Our own "look" and "feel"
- Color scheme
- **FONT SELECTION**
- Background images or watermarks
- Ease of change
- Accessibility



Dynamic!

Cascading Style Sheets

- Separation of **style** (presentation) and **content**
 - For example, switching between desktop and mobile
- Parallel development with **content** stubs
 - **Styling** and **Control** (i.e., JavaScript) can then be developed in parallel
- Reuse of style sheets
 - Style once present everywhere
- “**Cascading**” meaning **one follows another**.
 - There’s an order of importance in style annotations
- **Users can selectively turn styles off or turn others on.**

Where does CSS go?

- Style can be inline

```
<span style="font-family: verdana; font-size: 14pt">the font face</span>
```

- Style can be defined internally

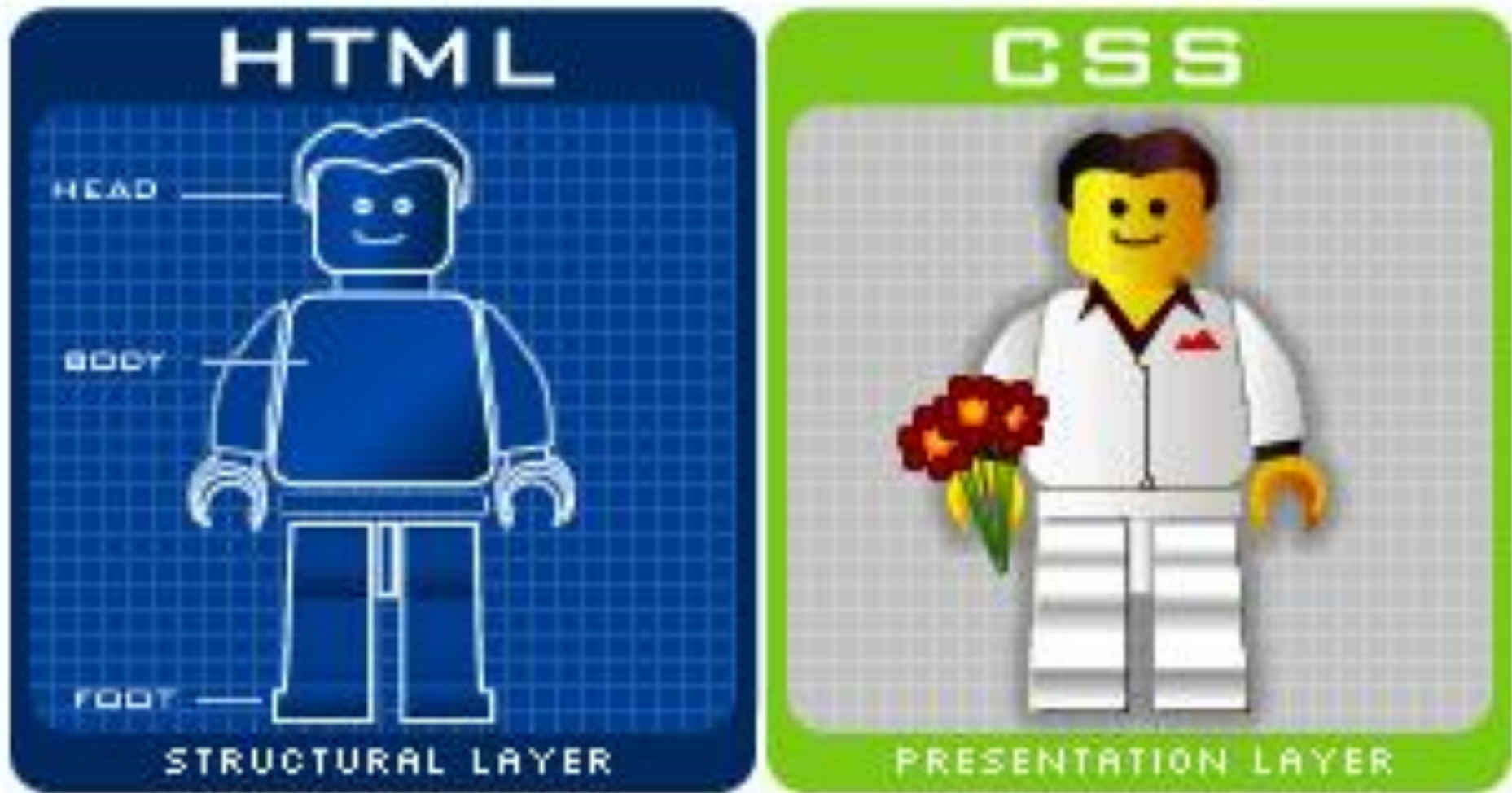
```
<style>
#red {
    color: red;
}
```

- Style can be defined externally

```
<link rel="stylesheet" href="mystyles.css">
```

We can use all of the above simultaneously.

But the last is best.



HTML is the blueprint and CSS is the coloring on top.

Colors

- Change foreground (i.e., text) or background color
- Specify by

```
<style>
#red {
    color: red;
}
```

name, #ff55aa, rgb(255,156,0), rgba(0,10,55,0.5)
hsl(25,80%,15%) hsla(25,80%,15%,0.5)

http://www.w3schools.com/cssref/css_colornames.asp

http://www.w3schools.com/tags/ref_colorpicker.asp

Dev Tools

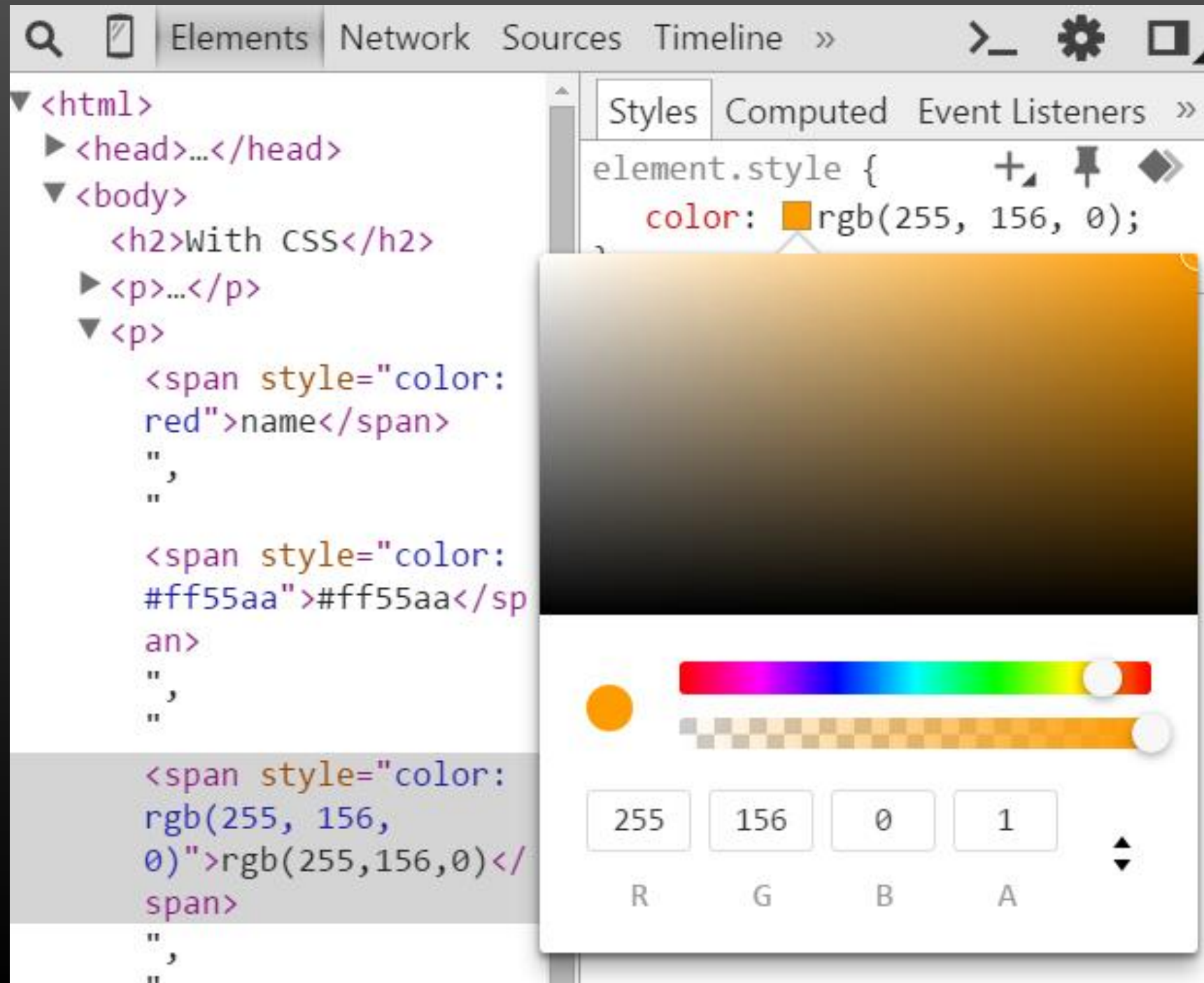
Live Editing

ctrl+shift+C

cmd+option+I

select an element to inspect

Update CSS styles in real time
Is much faster than edit and
refresh



Font “Face”

- *font-family*

- Comma separated list
- Browser picks the font depending on system availability

- Google Fonts

- Visual selection of various fonts.
- Create a special “collection” of fonts for your site

```
<link href='https://fonts.googleapis.com/css?family=Roboto'  
      rel='stylesheet' type='text/css'>
```

```
font-family: 'Roboto', sans-serif;
```

Don't go crazy with fonts, but it is good to have some variation to make your site look unique.

Decoration and Spacing

font-weight

bold

font-style

italic

text-decoration

~~strike-through~~

underline

```
<span style="text-decoration: overline">overlined text</span>  
<span style="text-decoration: underline">is better than u  
</p>  
<p>  
<span style="letter-spacing: 1em">we control the horizont  
<span style="word-spacing: 1.5em">Lorem ipsum dolor sit a  
elit, sed do eiusmod tempor incididunt ut labore et dolor  
</p>  
<p>  
<span style="line-height: 3em">We control the vertical!  
.
```

overlined text is better than underlined text

"letter-spacing: 1em"

w e c o n t r o l t h e h o r i z o n t a l Lorem
ipsum dolor sit amet, consectetur adipiscing elit, sed do
eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut
enim ad minim veniam, quis nostrud exercitation ullamco laboris
text-transform: uppercase modo consequat. DUIS AUTE IRURE
DOLOR IN REPREHENDERIT IN VOLUPTATE VELIT ESSE
CILLUM DOLORE EU FUGIAT NULLA PARIATUR. Excepteur
sint occaecat cupidatat non proident, sunt in culpa qui officia
deserunt mollit anim id est laborum **word-spacing: 1.5em**

We control the vertical! Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod
tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud
line-height: 3em
exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in

Size units

Absolute

Unit	Description
cm	centimeters
mm	millimeters
in	inches (1in = 96px = 2.54cm)
px *	pixels (1px = 1/96th of 1in)
pt	points (1pt = 1/72 of 1in)
pc	picas (1pc = 12 pt)

* px are viewing device relative.

Relative

Unit	Description
em	Relative to the font-size of the element (2em means 2 times the size of the current font)
ex	Relative to the x-height of the current font (rarely used)
ch	Relative to width of the "0" (zero)
rem	Relative to font-size of the root element
vw	Relative to 1% of the width of the viewport
vh	Relative to 1% of the height of the viewport
vmin	Relative to 1% of viewport's smaller dimension
vmax	Relative to 1% of viewport's larger dimension
%	

font-size: 10<?>

width: 500<?>

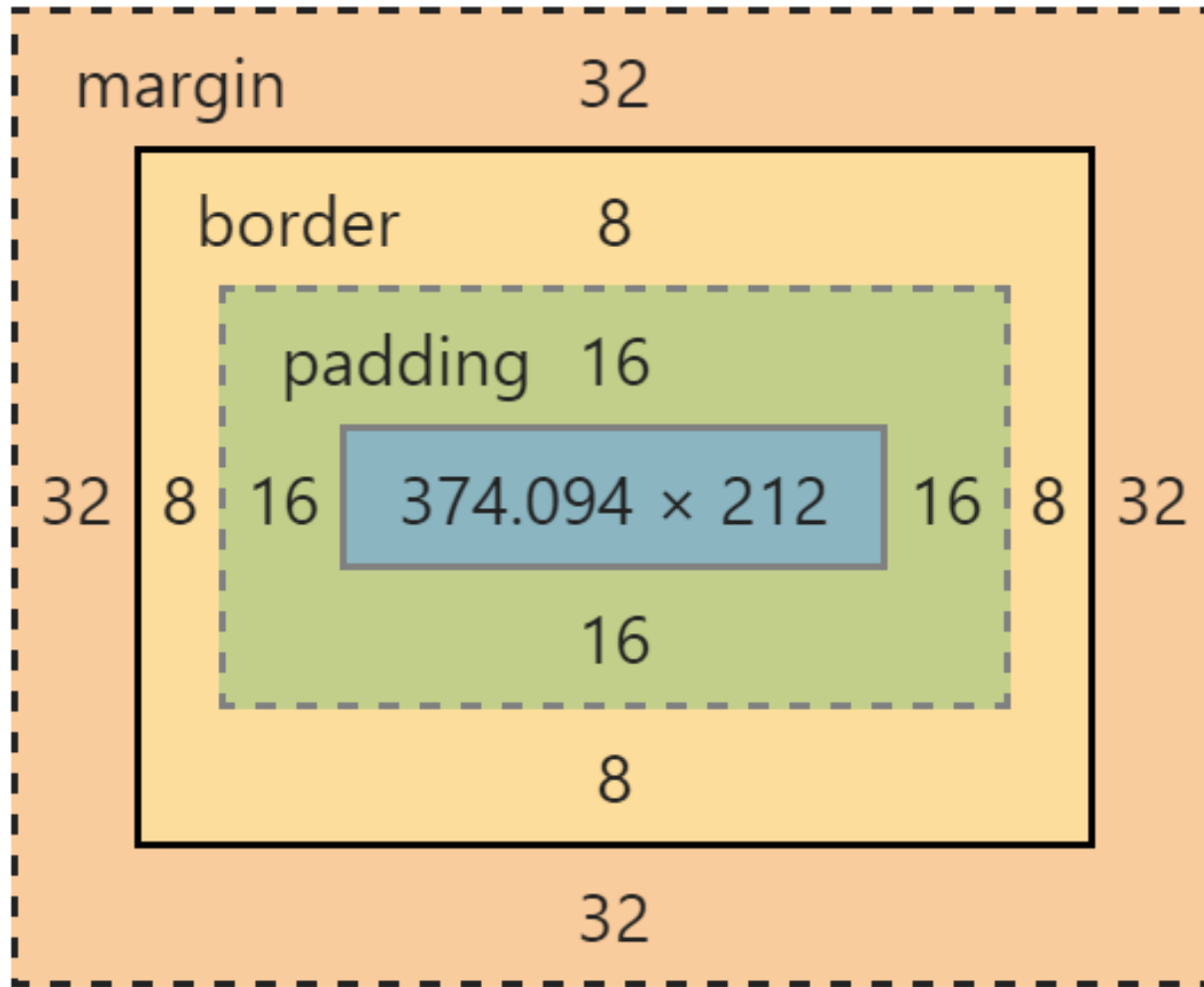
Use “em” whenever possible

	<code>body { font-size: 100%; }</code>	<code>body { font-size: 120%; }</code>
<code>font-size: 1em</code>	The quick brown fox	The quick brown
<code>font-size: 12pt</code>	The quick brown fox	The quick brown fox
<code>font-size: 16px</code>	The quick brown fox	The quick brown fox
<code>font-size: 100%</code>	The quick brown fox	The quick brown

© KyleSchaeffer.com

Devtools can show you all of the styles for an element

Box Model



Styles Computed Event Listeners DOM Breakpoints

☐ Show inherited properties

```
border-bottom-color: ■ rgb(0, 0, 255);
border-bottom-style: solid;
border-bottom-width: 8px;
border-image-outset: 0px;
border-image-repeat: stretch;
border-image-slice: 100%;
border-image-source: none;
border-image-width: 1;
border-left-color: ■ rgb(0, 0, 255);
border-left-style: solid;
```

```
border-top-color: ■ rgb(0, 0, 255);
border-top-style: solid;
border-top-width: 8px;
display: block;
float: right;
height: 212px;
margin-bottom: 32px;
margin-left: 32px;
margin-right: 32px;
margin-top: 32px;
padding-bottom: 16px;
padding-left: 16px;
padding-right: 16px;
padding-top: 16px;
width: 374.09375px;
```


Positioning

Visit the linked page to see how it functions with relation to resizing the screen and scrolling

Not so nice I think



Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.



Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.




float right
float right
float right
float right
float right

float right float right float
right float right float right
float right float right float
right float right float right

Positioning




 Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.



incidunt ut labore et dolore magna aliqua. Ut
exercitation ullamco laboris nisi ut aliquip ex
in reprehenderit in voluptate velit esse cillum
occaecat cupidatat non proident, sunt in
laborum.

Lorem ipsum dolor sit amet, consectetur
 adipiscing elit, sed d tempor
 incididunt ut labor magna aliqua. U
 exercitation ullam
 ea commodo consequatur. S
 in reprehenderit in voluptate velit esse cillum
 dolore eu fugiat nulla pariatur. Excepteur sint
 occaecat cupidatat non proident, sunt in culpa
 qui officia deserunt mollit anim id est



float right
float right
float right
float right
float right
float right float right float

[illegible]

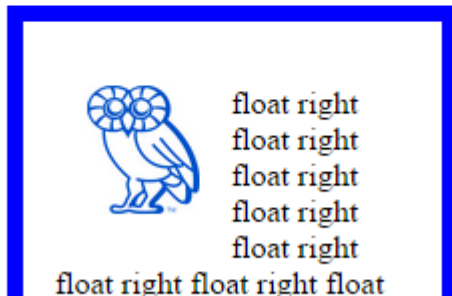
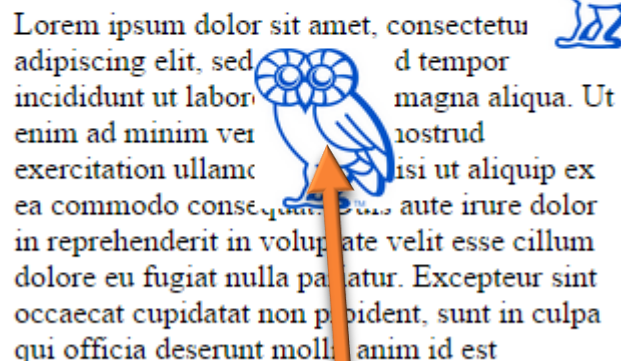
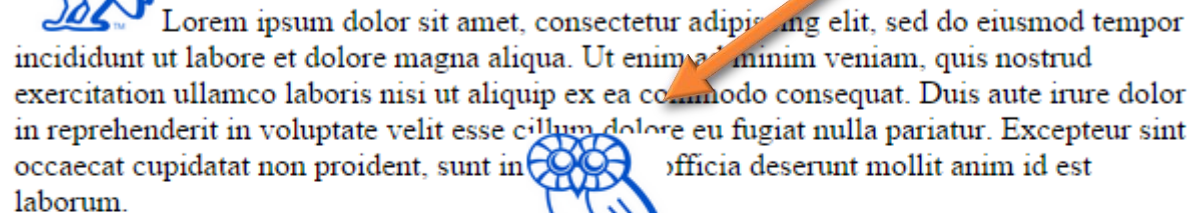
Resize the screen.

Absolute moves

```
<img style="height: 5em;  
position: absolute; left: 8em; top: 60em"
```

top/bottom, left/right

Scroll around, fixed stays fixed



float right
float right
float right
float right
float right

[illegible]

```
<img style="height: 5em;
position: absolute; left: 8em; top: 60em"
```

Positioning

```
<div style="position:fixed; left: 50%; top: 50%">  
  <img style="height: 5em; position:relative; left: -2.5em"  
</div>
```



```
<div style="float:right; width: 30%;  
border: 0.5em solid blue; padding: 1em;  
margin: 2em">  
    
  <p>  
    float right float right float right  
    float right float right float right  
    float right float right float right
```

```
<img style="height: 5em;  
position:absolute; left: 8em; top: 60em"
```

Recall: Inline vs Block

- We markup to control layout. Sometimes we want “blocks” of content that are positioned separately from other blocks.
- Other times we want inline layout or styling

Inline

``

``

`
`

``

Block

`<p>`

`<div>`

We can control using *display*

Display

```
tempor incididunt ut labore et dolore magna aliqua.  
</p>  
<p style="color: blue; display:inline">Ut enim ad  
minim veniam, quis nostrud exercitation ullamco
```

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint

occeaecat cupidatat non proident, ut labore et dolore magna aliqua. mollit anim id est

```
<span style="border: 1pt dashed red">  
<span style="visibility:hidden">Ut enim ad minim
```

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.



reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur.

Excepteur sint occaecat
deserunt mollit anim id est

```
  
  
<p>Excepteur sint occaecat cupidatat non proident,
```

CONTAINER <DIV> ... </DIV>

Content before the float will appear here.

Left Floating Element
<div style="float: left">

Content after both floats will appear here. It flows around the floating elements.

Right Floating Element
<div style="float: right;">

We clear the floats here.
<div style="clear: both;">

Content after the clear will appear here.

More more more !!!

- Text-align: (left|center|right|justify)
- Vertical-align: (baseline|sub|super|top|text-top|middle|...)
- Text-shadow: 0.1em 0.1em #ff0022;
- Max-width / min-width / max-height / min-height
- Border-radius
- outline
- Box-shadow
- Overflow: (visible|hidden|scroll|auto|inherit)
- Positioning z-index
- Cursor: (crosshair|default|pointer|...)
- Transition
- Transform
- Opacity
- Filter
- gradients

Non-inline CSS: Selectors

```
/* CSS Comment */  
selector {  
    key: value;  
}
```

- No nesting
- Selectors are “rules”

```
div > tr.fancy #rule
```

Selector Rules (the easy ones)

- Tag

```
body {  
    background-color: #FFFFFF;  
}
```

- Class

```
.linkInverted {  
    color: #FFFF00;  
}
```

- Id

```
#riceLogo {  
    width: 6em;  
    margin-top: -1em;  
    margin-bottom: -1em;  
}
```

- Attribute

```
[name="fancy"] {  
    font-size: 2em;  
}
```

Selector Rules (some combinators)

- `div, p`
 - Select all `<div>` and all `<p>`
- `div p`
 - Select all `<p>` directly within a `<div>`
- `div > p`
 - Select all `<p>` with an eventual parent of `<div>`

(slightly edited)

cross-browser example from w3schools

```
ul {  
    list-style-type: none;  
    padding: 0px;  
    margin: 0px;  
}
```

```
ul li:hover {  
    background-color: #444444;  
}
```

```
ul li {  
    background-image: url(sqpurple.gif);  
    background-repeat: no-repeat;  
    background-position: 0px center;  
    padding-left: 15px;  
}
```

```
ul {  
    list-style-image: url('sqpurple.gif');  
}
```



- Coffee
- Tea
- Coca Cola

:hover is a “state selector”
:active, :focus, :link, :visited,

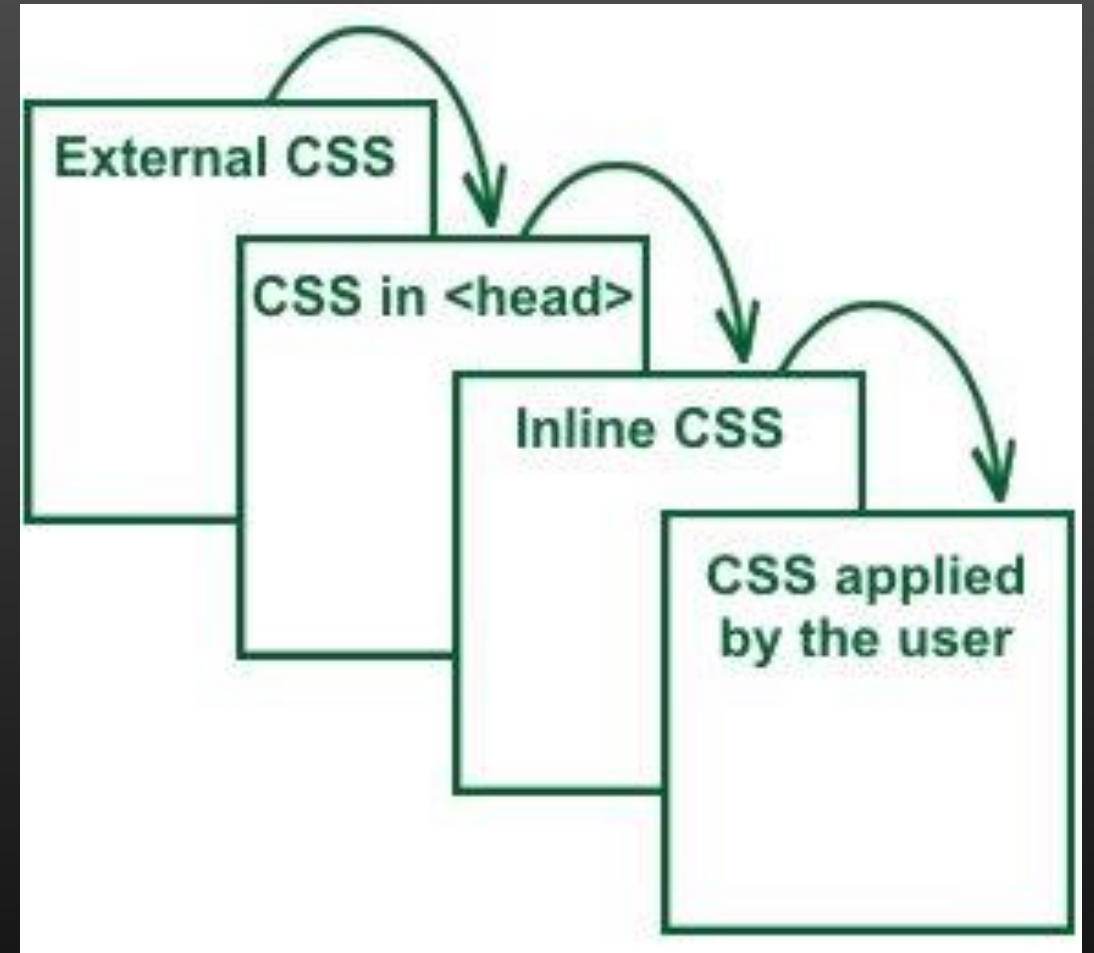
...

... spaces matter in selectors

- `div.myclass#myid`
 - Selects `<div class="myclass" id="myid">`
- `div .myclass #myid`
 - Selects some element with `id="myid"` that is contained within an element with `class="myclass"` that is contained within a `<div>`. E.g.,
 - `<div>`
 - ``
 - `<input id="myid" type="text">`

Cascading

- Style sheets cascade
 - Externally imported or linked sheets
 - In order
 - Followed by internal styles
 - Then inline
 - Finally overrides in the browser
-
- Not everything cascades. E.g., border doesn't.
`border: inherit`



```
@import url(http://fonts.googleapis.com/css?family=Open+Sans:400,300);
```

Style Cascade Order (*in reverse*)

Priority	CSS Source Type	Description
1	Importance	The 'important' annotation overwrites the previous priority types
2	Inline	A style applied to an HTML element via HTML 'style' attribute
3	Media Type	A property definition applies to all media types, unless a media specific CSS defined
4	User defined	Most browsers have the accessibility feature: a user defined CSS
5	Selector specificity	A specific contextual selector (#heading p) overwrites generic definition
6	Rule order	Last rule declaration has a higher priority
7	Parent inheritance	If a property is not specified, it is inherited from a parent element
8	CSS property definition in HTML document	CSS rule or CSS inline style overwrites a default browser value
9	Browser default	The lowest priority: browser default value is determined by W3C initial value specifications

Style Computation by Specificity

(style attribute),(id),(attributes+class),(element)

```
*          {}  /* specificity = 0,0,0,0 */
li         {}  /* specificity = 0,0,0,1 */
li:first-line {} /* specificity = 0,0,0,2 */
ul li      {}  /* specificity = 0,0,0,2 */
ul ol+li   {}  /* specificity = 0,0,0,3 */
h1 + *[rel=up] {} /* specificity = 0,0,1,1 */
ul ol li.red {} /* specificity = 0,0,1,3 */
li.red.level {} /* specificity = 0,0,2,1 */
#x34y      {}  /* specificity = 0,1,0,0 */
style=""   /* specificity = 1,0,0,0 */
```


Something to consider: Desktop vs Mobile

- Different sized viewports
- Text flow
- Layout
- Scaling and zooming

The Drive for Style

- We want awesome looking web pages
- Writing a bunch of CSS styling is... hard
- Especially having to do it multiple times for various devices
- Someone else has already done it for us
- Let's build on the shoulders of giants

(Responsive) Style Frameworks

- Wikipedia says:

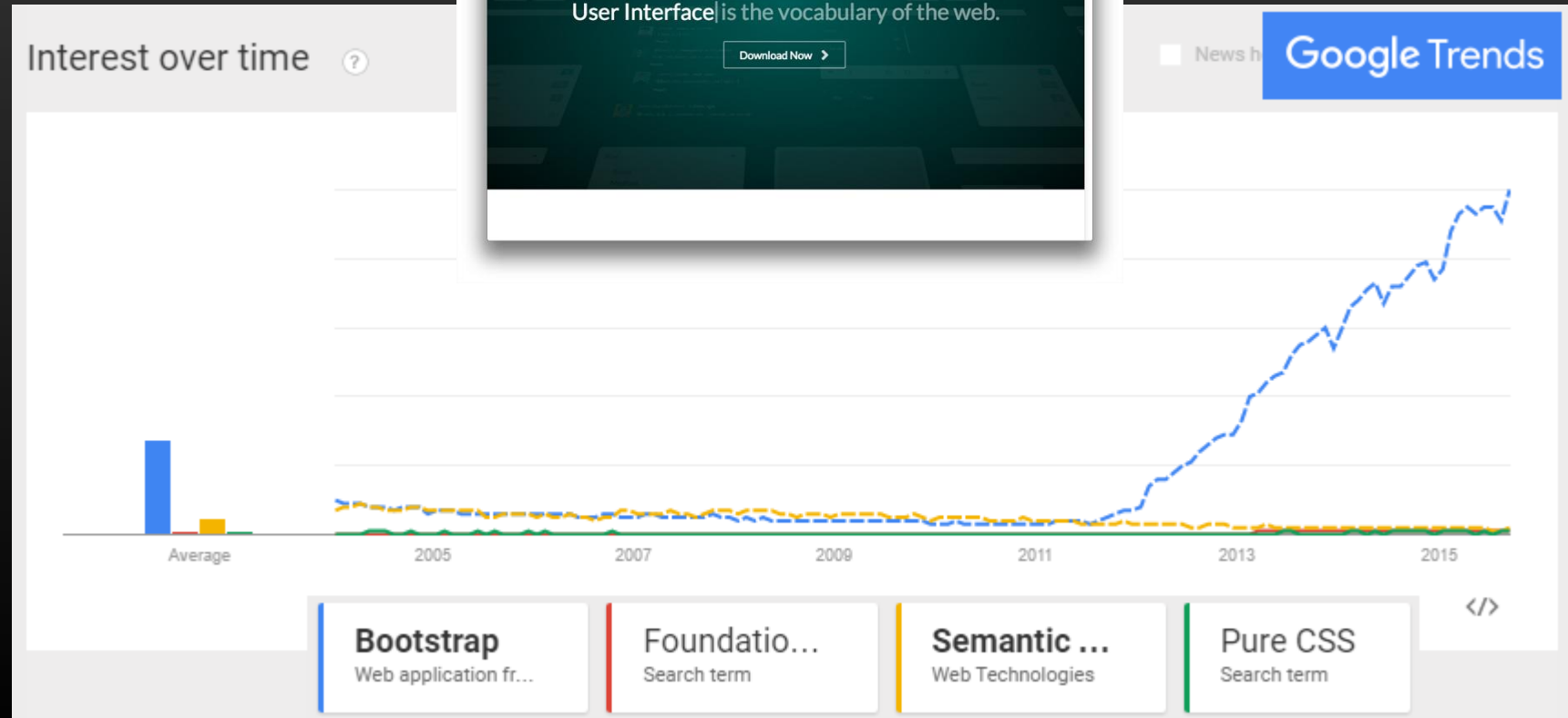
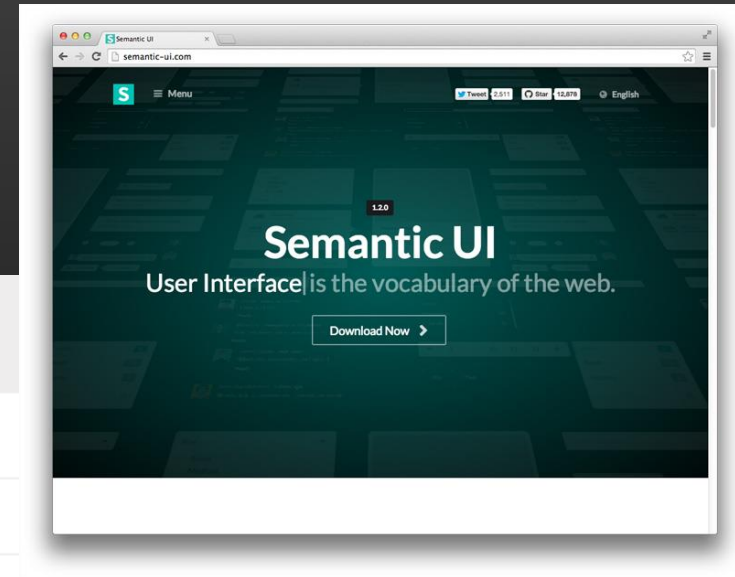
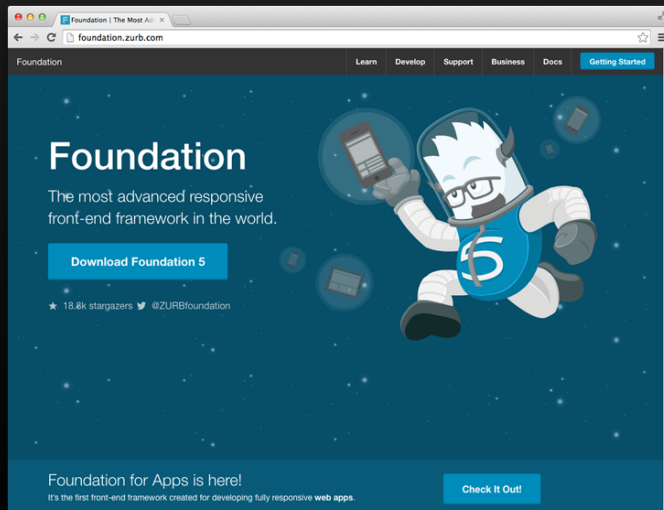
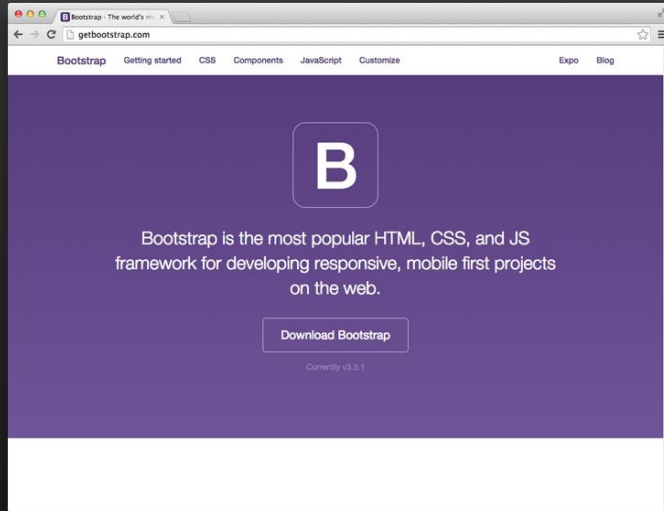
- **Frameworks** are pre-prepared [software frameworks](#) that are meant to allow for easier, more standards-compliant [web design](#) using CSS
- Most of these frameworks contain at least a [grid](#).
- More functional frameworks also come with more features and additional [JavaScript](#) based functions, but are mostly design oriented and [unobtrusive](#).
- This differentiates these from functional and full [JS frameworks](#).

Responsive Framework



The Top Contenders

(not rigorously determined...)



Installing Bootstrap

Bootstrap CDN

The folks over at [MaxCDN](#) graciously provide CDN support for Bootstrap's CSS and JavaScript. Just use these [Bootstrap CDN links](#).

```
<!-- Latest compiled and minified CSS -->
<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.5/css/bootstrap.min.css">

<!-- Optional theme -->
<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.5/css/bootstrap-theme.min.css">

<!-- Latest compiled and minified JavaScript -->
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.5/js/bootstrap.min.js">
</script>
```

Copy

<http://getbootstrap.com/>

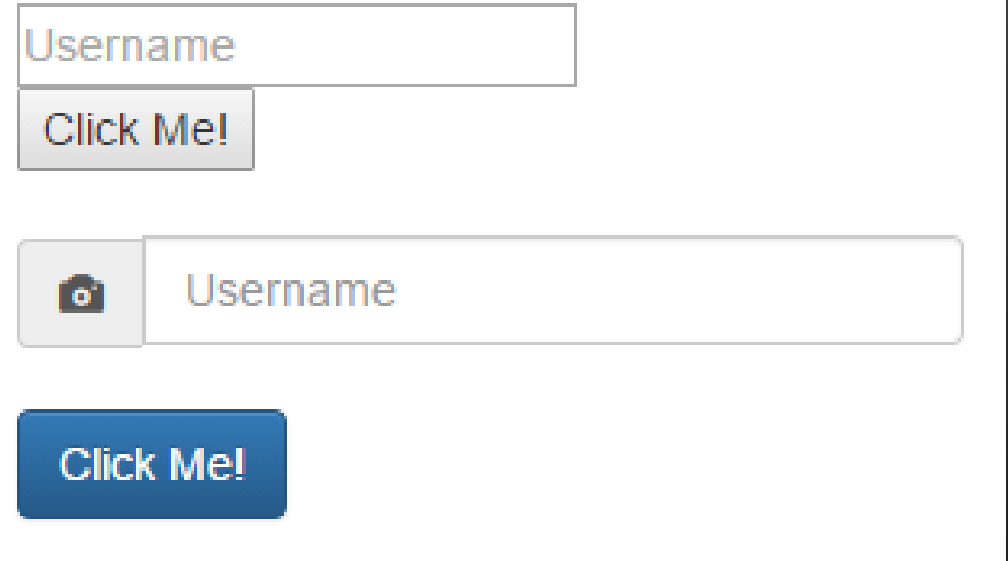
<http://www.w3schools.com/bootstrap/>

Using Bootstrap

```
<input type="text" placeholder="Username">
<br>
<input type="button" value="Click Me!">

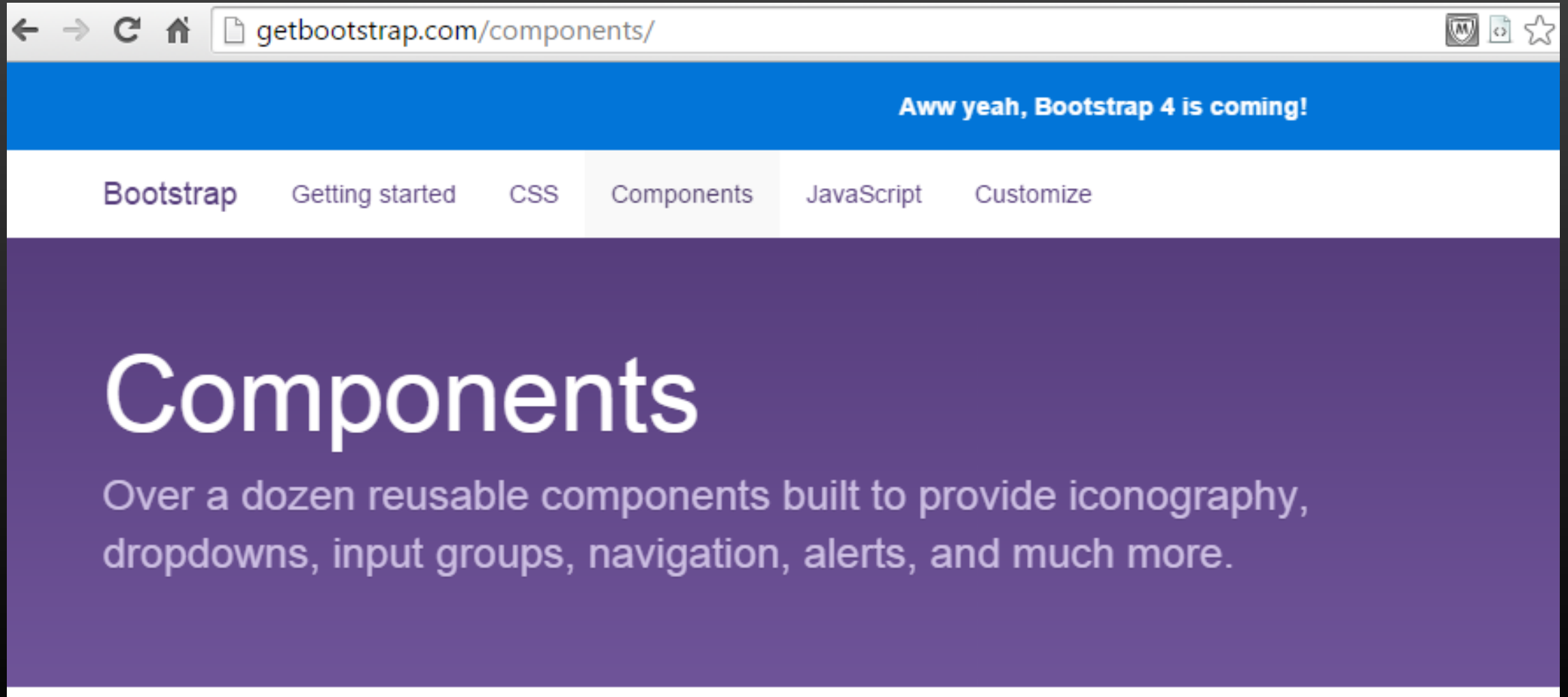
<br><br>

<div class="row">
  <div class="col-xs-6 col-md-3">
    <div class="input-group">
      <span class="input-group-addon glyphicon glyphicon-camera">
      </span>
      <input type="text" class="form-control" placeholder="Username">
    </div>
    <br><input type="button" class="btn btn-primary" value="Click Me!">
  </div>
</div>
```



The image displays a Bootstrap form layout. At the top, there is a simple text input field with the placeholder text "Username" and a "Click Me!" button below it. Below this, a more complex Bootstrap-styled form is shown. It features a camera icon (glyphicon-camera) inside a light gray box, followed by a text input field with the placeholder "Username". Below the input field is a blue "Click Me!" button, which is a primary button in Bootstrap styling.

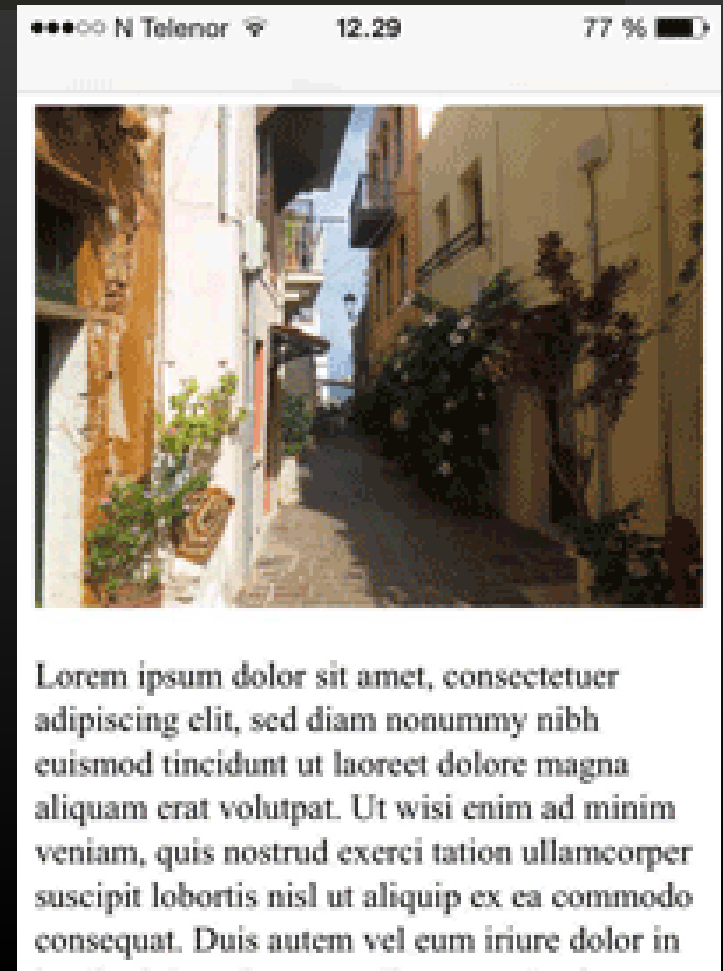
More Bootstrap



Viewport

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

- User “viewable” area
- Set the width to the device-width
- Set the initial scaling



VIEWPORT SIZES

viewportsizes.com

?

iPhone

Device Name ▲	Platform	OS Version	Portrait Width	Landscape Width	Release Date
iPhone	iOS	3.1.3	320	480	2007-06
iPhone 3G	iOS	4.2.1	320	480	2008-07
iPhone 3GS	iOS	6.0a2	320	480	2009-06
iPhone 4	iOS	5.1.1	320	480	2010-06
iPhone 4S	iOS	4.3.5	320	480	2011-10

Viewport <meta> vs CSS

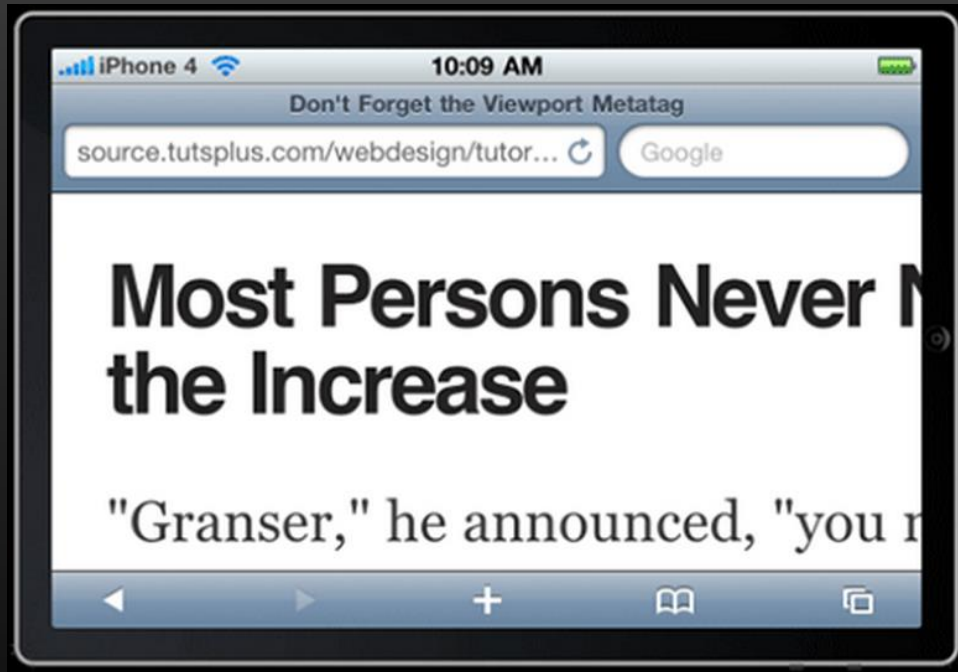
```
<meta name="viewport" content="width=device-width, initial-scale=1">
```

```
@viewport{  
  zoom: 1.0;  
  width: device-width;  
}
```

Recommendation:
Do both.

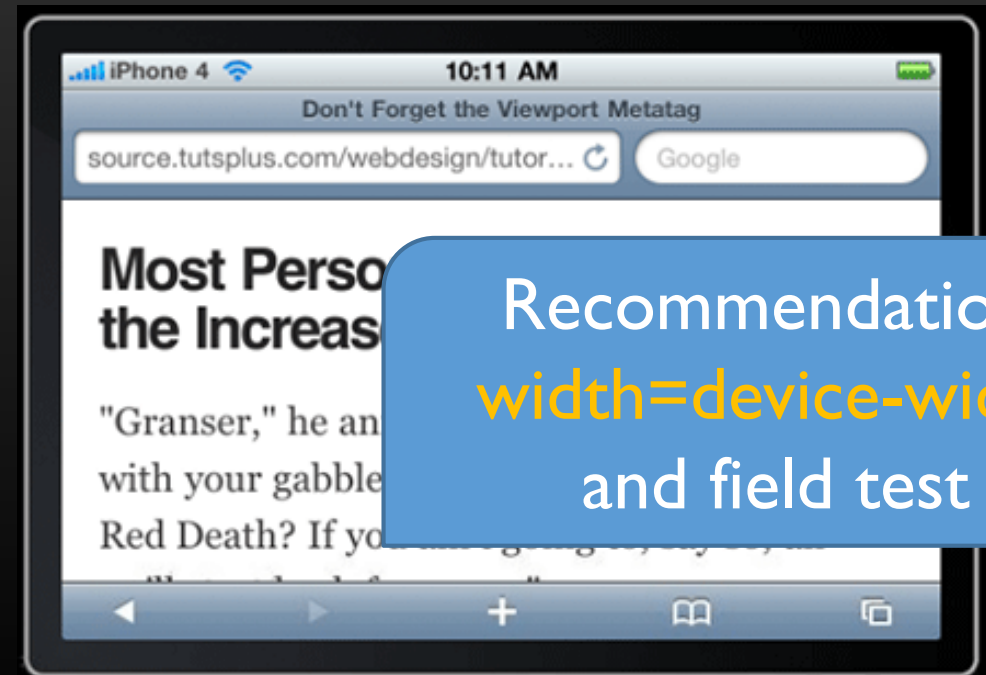
Viewing in Landscape

Don't set width? only scale/zoom?



width=device-height

```
@viewport{  
  zoom: 1.0;  
  width: extend-to-zoom;  
}
```



Recommendation:
width=device-width
and field test

CSS @media queries

```
@media screen and (max-width: 300px) {  
    body {  
        background-color: lightblue;  
    }  
}
```

```
<link rel="stylesheet" type="text/css" href="style.css" media="screen,  
handheld" />  
<link rel="stylesheet" type="text/css" href="enhanced.css"  
media="screen and (min-width: 40.5em)" />  
<!--[if (lt IE 9)&(!IEMobile)]>  
<link rel="stylesheet" type="text/css" href="enhanced.css" />  
<![endif]-->
```

... bad code is everywhere

useless comments

```
//Check if Mobile  
function checkMobile() {  
    if(sw > breakpoint) {  
        mobile = false; //Not Mobile  
    } else {  
        mobile = true; //Mobile  
    }  
}
```

Just use the Boolean!

```
if (!mobile) { //If Not Mobile  
    loadAux();  
}
```

Not descriptive

```
//Check if Mobile  
function checkMobile() {  
    mobile = (sw > breakpoint) ? false : true;  
    if (!mobile) { //If Not Mobile  
        loadAux();  
    }  
}
```

Better Responsive Images

```

```

```

```

```
<picture>
  <source
    media="(min-width: 650px)"
    srcset="images/kitten-stretching.png">
  <source
    media="(min-width: 465px)"
    srcset="images/kitten-sitting.png">
  
</picture>
```

Better Responsive Images



By [Pearl Chen](#)

Published: September 11th, 2014

Updated: September 11th, 2014

Comments: [42](#)

Introducing the <picture> element

Don't use <picture> (most of the time)



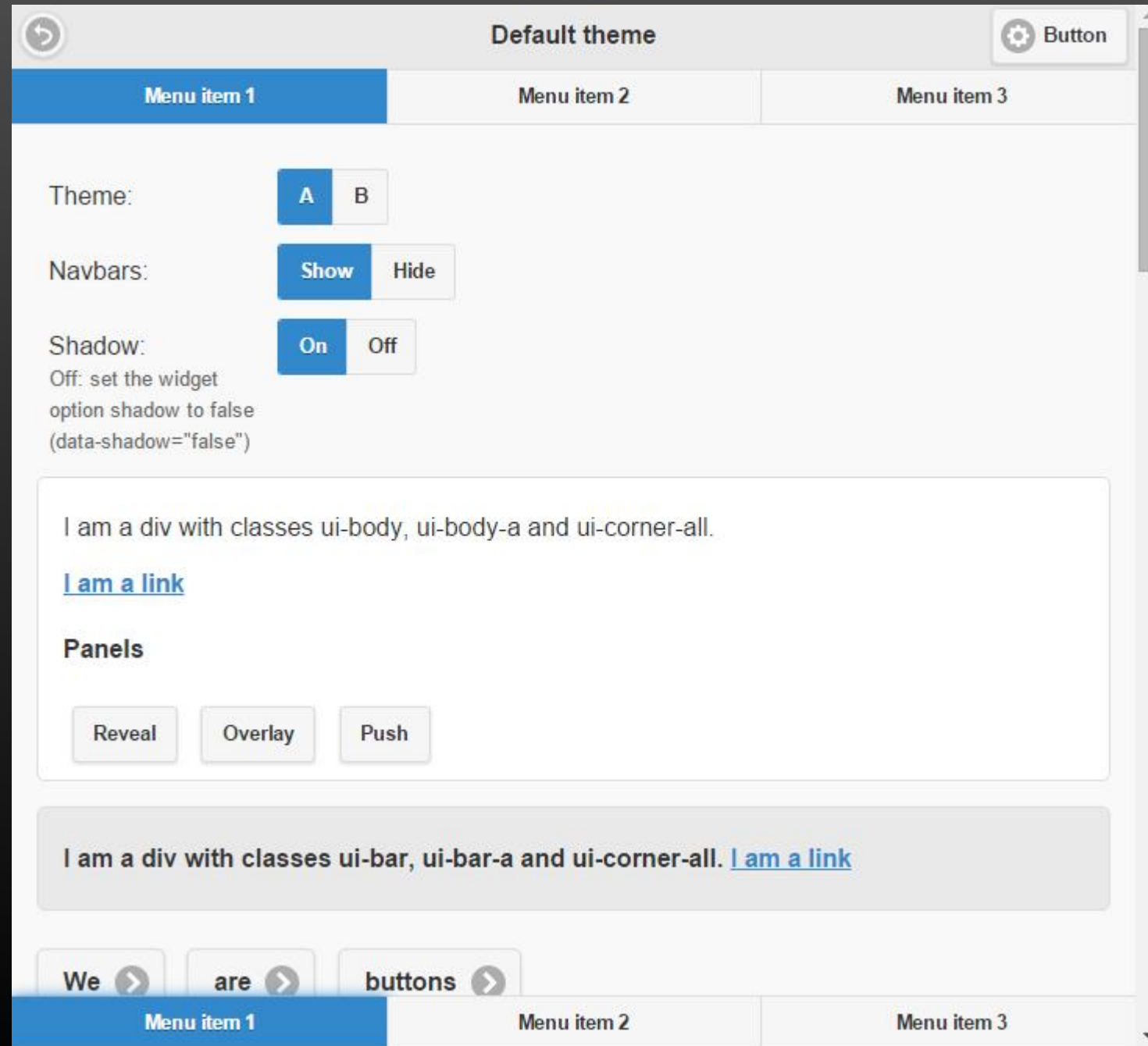
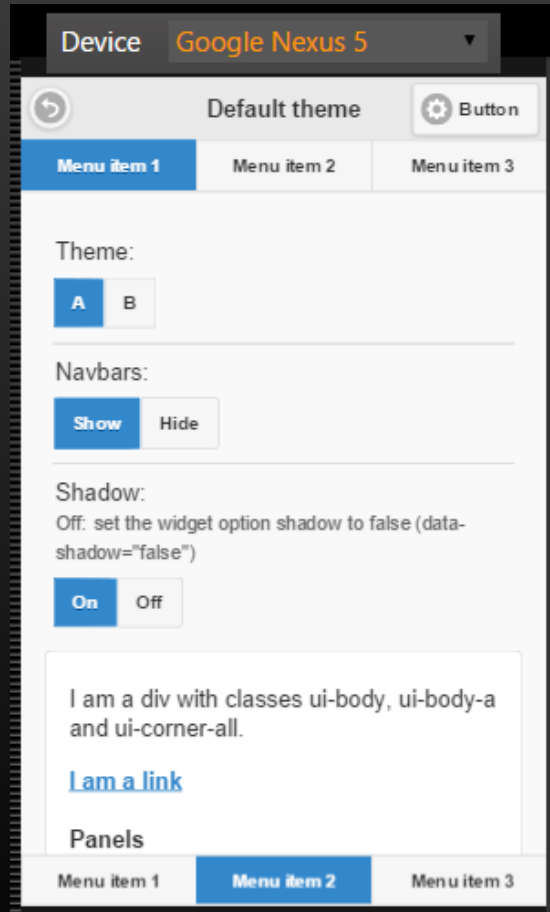
Jason Grigsby

September 22, 2014

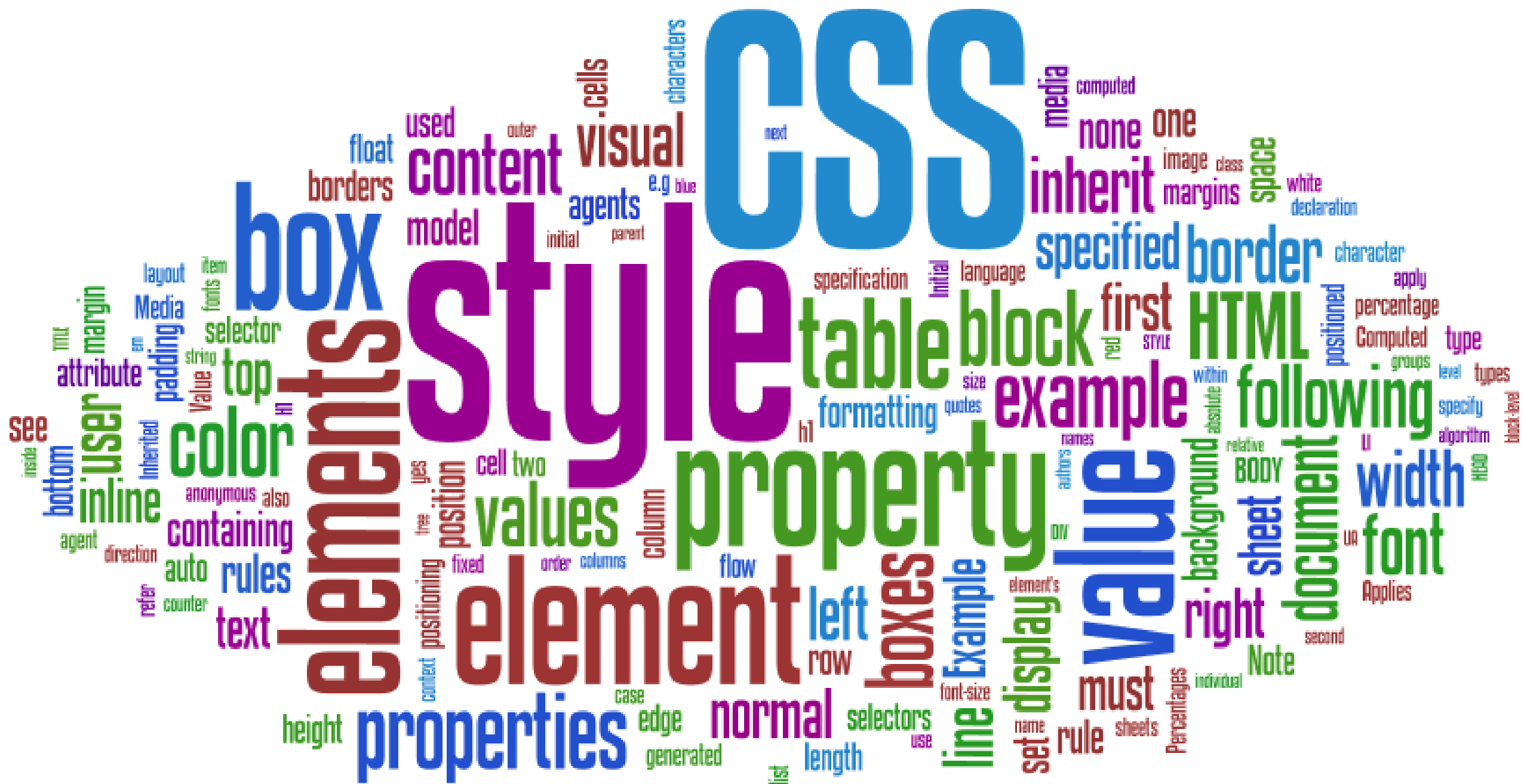
Browser support for the picture specification is landing and as [Marcos](#)

[Cáceres](#) said, it is time to “go forth and <picture> all the things!”

Mobile First with jQuery Mobile



<https://jquerymobile.com/>



References

- w3schools
- CSS Zen garden
- CSS-tricks (Chris Coyler)