



RICE[®]

Web Development

COMP 431 / COMP 531

Cascading Style Sheets

Scott E Pollack, PhD

January 21, 2016

Recap

- HTML
- Forms
- JavaScript
- Homework Assignment 1 (Simple Page)
 - **Now past due**
 - Two *html* files in the *hw1-simple* directory

CSS
Events
JavaScript
HTML5

Homework Assignment 2
(Dynamic Page)
Due Thursday 1/28

Homework Assignment 4
(JavaScript Game)
Due Thursday 2/18

Demo javascript calculation in debugger

- Create a page with internal javascript
- Open file in debugger and debug
- Cached scripts

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

```

```

```
<p><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.**








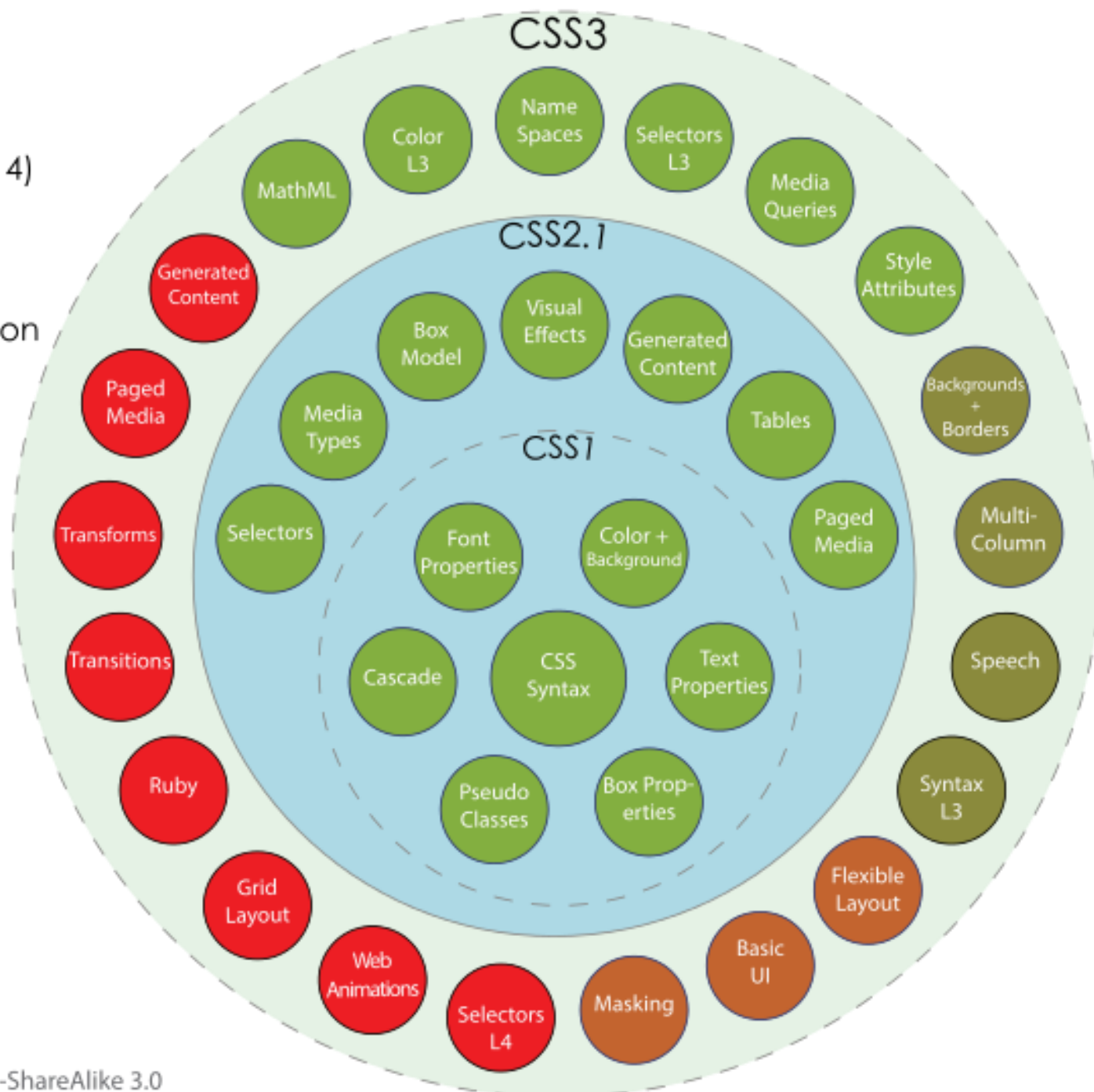
Some CSS history

- Aug 1996 – JavaScript Style Sheets (JSSS) from Netscape
 - Dec 1996 – CSS level 1 [basic concepts and implementation]
 - Nov 1997 – CSS2
 - Mid-1998 – CSS3 begins specification
 - Mar 2000 – IE5 first fully-compliant CSS1 browser
-
- CSS3 is split into modules for independent improvement and implementation

CSS3

Taxonomy & Status (October 2014)

-  W3C Recommendation
-  Candidate Recommendation
-  Last Call
-  Working Draft
-  Obsolete or inactive



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.



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)

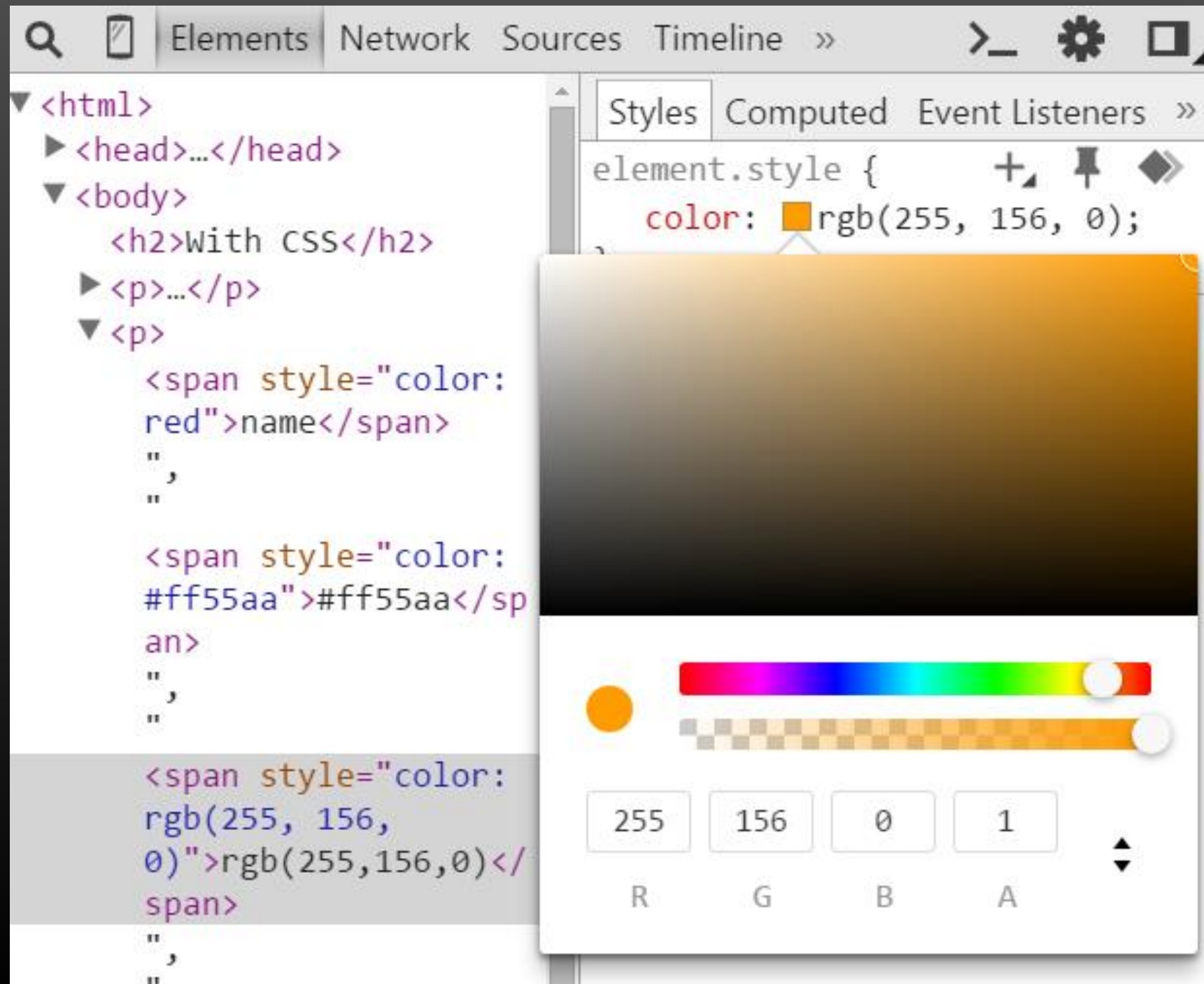
http://www.w3schools.com/cssref/css_colornames.asp

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

Dev Tools

Live Editing

ctrl+shift+C
select an element to inspect



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;
```

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</span>  
</p>  
<p>  
<span style="letter-spacing: 1em">we control the horizont</span>  
<span style="word-spacing: 1.5em">Lorem ipsum dolor sit a</span>  
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<?>

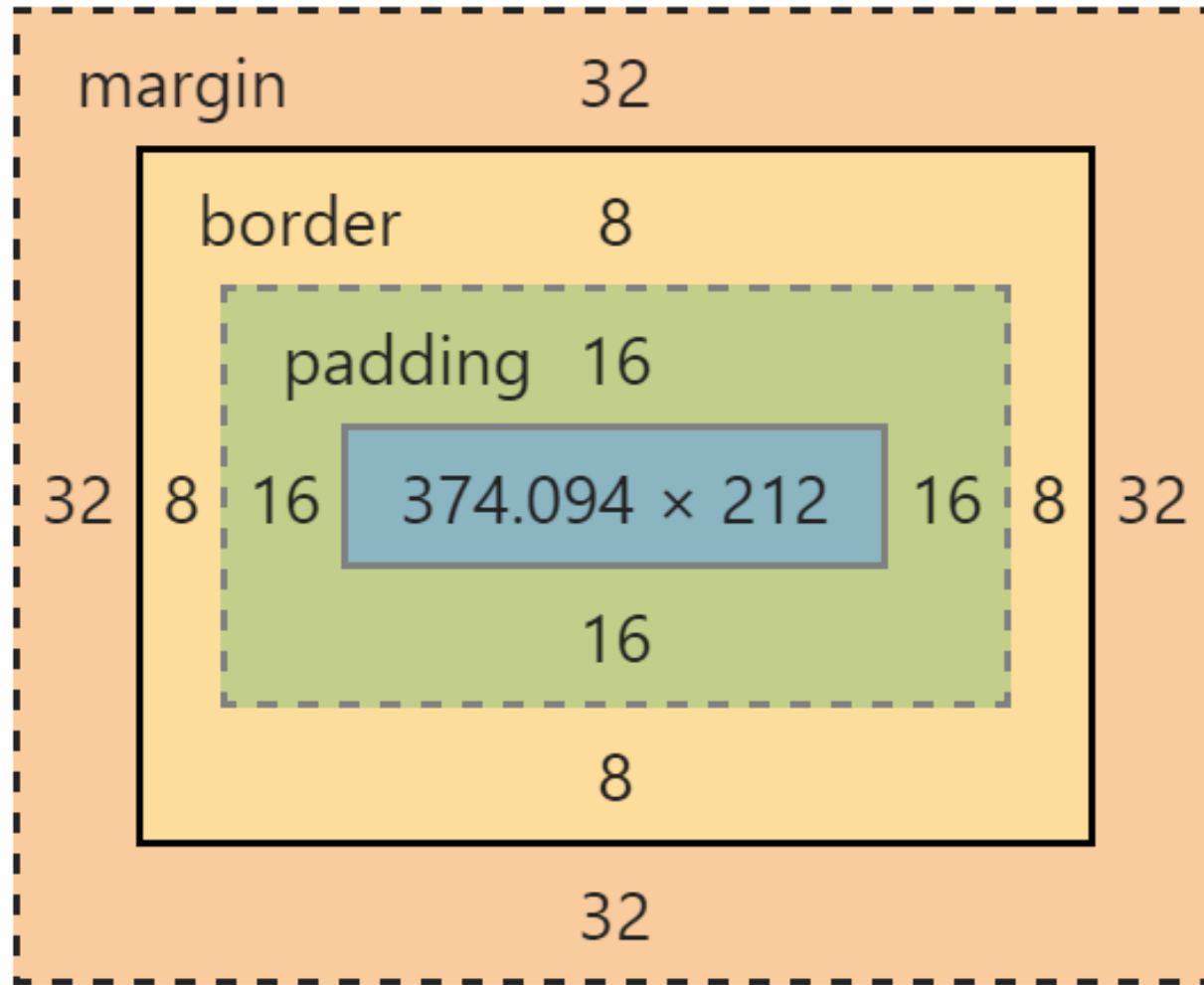
width: 500<?>

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

Box Model



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

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 ullamc nostrud
 ea commodo consequat nisi ut aliquip ex
 in reprehenderit in voluptate velit esse cillum
 dolore eu fugiat nulla paratur. Excepteur sint
 occaecat cupidatat non poident, sunt in culpa
 qui officia deserunt moll anim id est



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

[illegible]


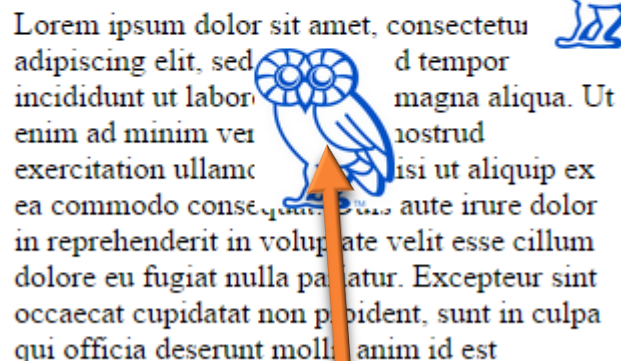
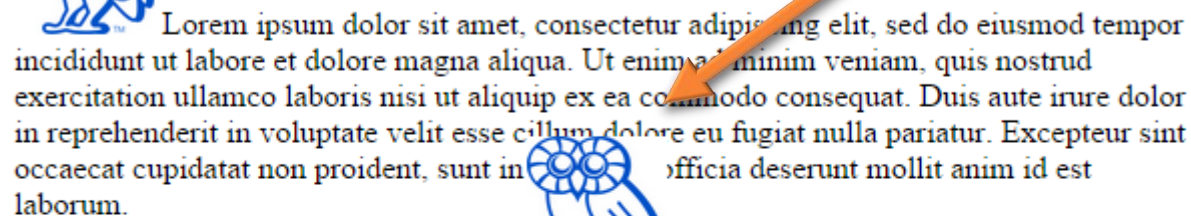
Resize the screen.

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
float right float right float



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>
```



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

```
<div style="float:right; width: 30%;  
border: 0.5em solid blue; padding: 1em;  
margin: 2em">  
    
  <p>  
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, sunt in culpa qui officia deserunt mollit anim id est

lal

```
<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

Excepteur sint oc
deserunt mollit a

pariatur.

```

```

```

```

```
<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,

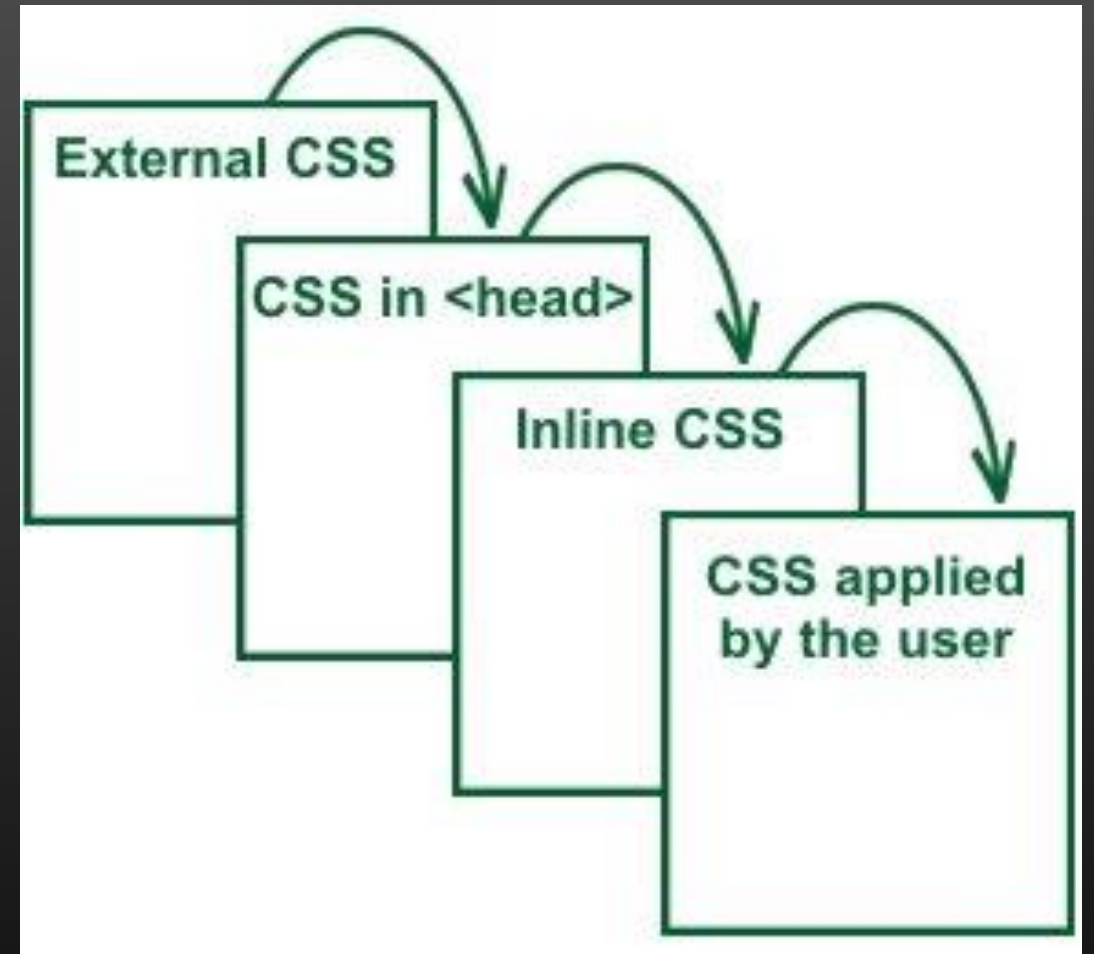
...

... 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

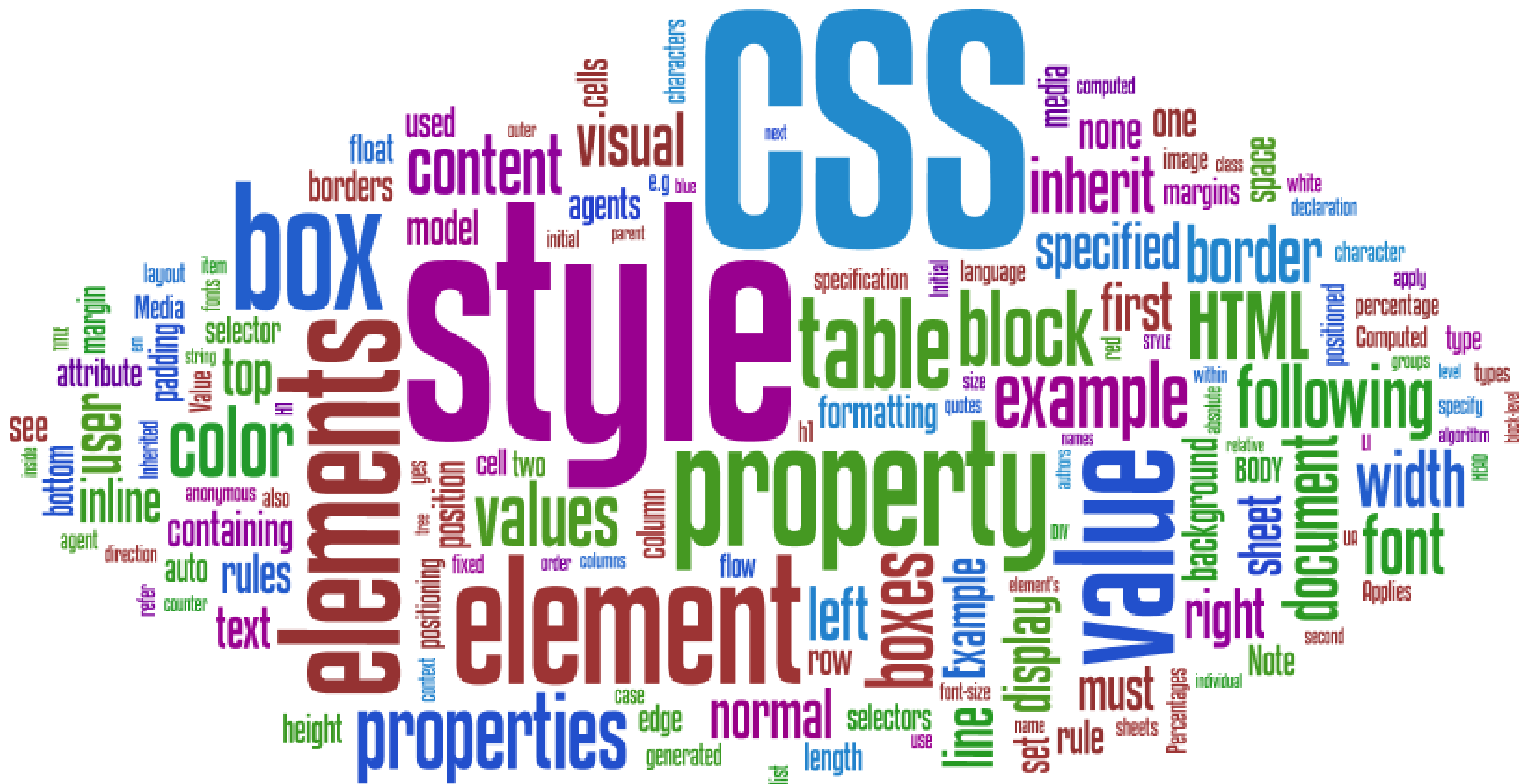
Style Computation by Specificity

(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



References

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

<https://www.clear.rice.edu/comp431/sample/hello-styled.html>
<https://www.clear.rice.edu/comp431/sample/main.html>

In-Class Exercise: CSS

Start with *main.html*

1. Add your choice of styling to the cards/posts, background, etc
2. When the user enters their username and presses “login”
 1. “login” changes to “logout”
 2. The username input becomes hidden
 3. The hidden “share” text field and button become visible
3. When a user presses “logout” the reverse happens
4. Add styling to the “share” text field and “post” button
5. Add a “navigation” area fixed to the top of the screen
the area contains the site name, username, and log(in|out) button

Turnin main.html to COMP431-S16:inclass-4