

HTML & CSS Level 1: Week 3

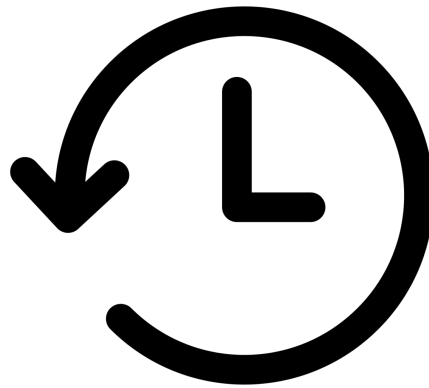
June 18 - July 23, 2014

Instructor: Devon Persing

This week: All CSS, all the time

- Review!
- Using **class** and **id** attributes for styles
- More background styles
- The CSS box model
- Resetting browser default styles

Review!



History designed by [Ema Dimitrova](#) from the [Noun Project](#)

<> Block and inline elements

- Block elements (ex.: **<div>**)
 - Take up space
 - Have height and width
 - Create line breaks
- Inline elements (ex.: ****)
 - Flow with content
 - Take up only as much space as they need
 - Don't create line breaks
- Inline-block elements (ex.: ****)
 - Two great tastes in one

{ } Anatomy of a CSS rule

selector { property: value; }

- **Selector** is the thing you want to style
- **Property** is what aspect you want to style
- **Value** is how you want to style it
- **Property + value = declaration**
- **Declarations** end in semicolons (;)

{ } CSS rule example

```
h1 { font-size: 2em; }
```

- **Selector** is `h1` (any `<h1>` on the page)
- **Property** is `font-size`
- **Value** is `2em`

{ } Type selectors

- **Type selectors** are single HTML element names that style all elements of that type on the page

```
h1 { font-size: 2em; }
```

{ } Descendent selectors

- **Descendent selectors** point to *children* of other selectors and are *more specific*

```
/* less specific type selector will style all <a>  
elements */
```

```
a { font-weight: bold; }
```

```
/* more specific type selector will style only <a>  
elements that are children of <p> elements */
```

```
p a { font-weight: normal; }
```


{ } Multiple selectors

- Rules can have **multiple selectors**
- Selectors can be of **mixed kinds**

```
/* single selector */
```

```
a { font-weight: bold; }
```

```
/* multiple selectors */
```

```
a, span { font-weight: bold; }
```

```
/* multiple selectors with a descendent selector */
```

```
p a, span { font-weight: bold; }
```

{ } Multiple declarations

- Rules can have **multiple declarations**

```
/* single declaration */
```

```
a { font-weight: bold; }
```

```
/* multiple declarations (and multiple selectors!) */
```

```
a, span {  
    font-weight: bold;  
    font-style: italic;  
}
```

{ } Common font & text properties

font-size

text-align

font-family

text-transform

font-style

text-decoration

font-weight

Lots of others!

line-height

letter-spacing

Lots of others!

{ } Colors

`color`

`background-color`

Values for `color` and
`background-color`:

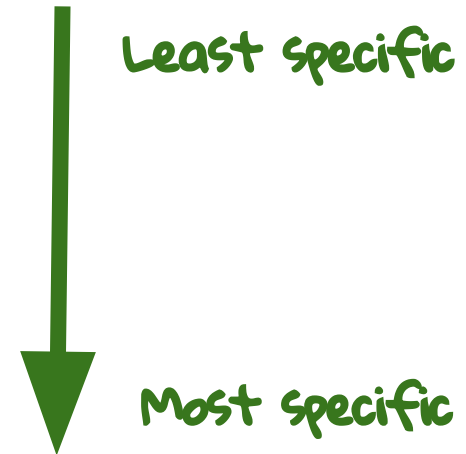
- Hex: `#ff0000`
- RGB: `rgb(255,0,0)`
- *And others!*

{ } Overriding styles

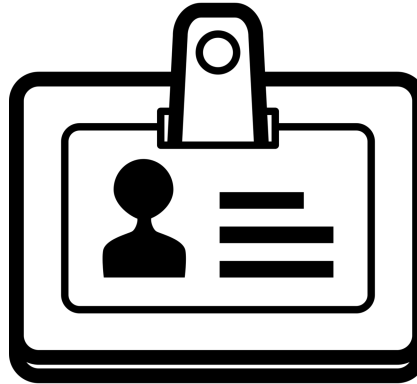
- Declarations that target children are more specific than rules children inherit from parents
- Declarations that come later override rules that come earlier
- **Hint:** Web Inspector will list the most specific styles on top and cross out overridden styles

{ } Stylesheet "location"

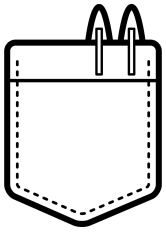
- Styles that are "closer" to the elements they style take precedence
 - Browser defaults
 - External styles (in a `.css` file)
 - Internal styles (in the `<head>`)
 - Inline styles (in an element)



Using class and id selectors



Identification Badge designed by [Michela Tannoia](#) from the [Noun Project](#)



Combining concepts

- **Week 1:** HTML elements can have attributes
- **Week 2:** HTML element names can be used as CSS selectors
- **Week 3:** HTML attributes can *also* be used as CSS selectors

</> / {} **class and id**

- **class** and **id** attributes can be added to any HTML element
- **Classes** are for multiple things on a page
- **IDs** are for individual things on a page
- You can make up whatever **class** and **id** values you want!

<> class attributes in HTML

- Classes can be shared by multiple elements on a page

```
<h1 class="kittens">...</h1>
```

```
<span class="kittens">...</span>
```

- Elements can have multiple classes

```
<div class="kittens puppies">...</div>
```

```
<div class="kittens puppies birds">...</div>
```

{ } class selectors in CSS

- Shown with a **period** (.)
- Can style any element with the class

```
.kittens { color: #000000; }
```

- Or can be used to style only a specific **type** of element with the class

```
h3.kittens { color: #000000; }
```

- More specific than an HTML type selector

<> id attributes

- IDs *cannot* be shared by multiple elements on a single page
- Elements *cannot* have multiple IDs

```
<div id="kittens">...</div>
```

```
<div id="puppies">...</div>
```

```
<div id="birds">...</div>
```

{ } id selectors in CSS

- Shown with a **hash/pound sign (#)**
- Can style the single element with the ID

```
#kittens { color: #000000; }
```

- More specific than a class selector

Mixing class and id attributes

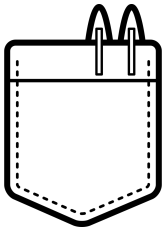
- Elements can have **id** and **class** attributes at the same time

```
<div id="kittens">...</div>
```

```
<div id="puppies" class="small floppy">...</div>
```

```
<div id="birds" class="small feathery">...</div>
```

- ID selector styles can be used to override class selector styles



Other reasons for `ids`

- Pre-HTML5, used to label major areas of the page

```
<div id="header">...</div>
```

```
<div id="article">...</div>
```

```
<div id="sidebar">...</div>
```

```
<div id="footer">...</div>
```

- Used to make on-page links

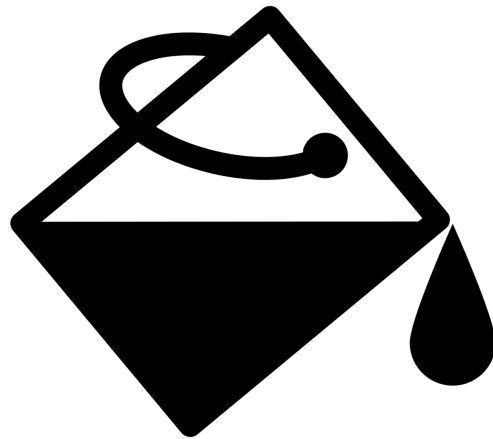
```
<a href="#header">Go back to the top</a>
```



Be thoughtful in your selectors

- Recommended order of attack:
 - a. Type selectors
 - b. Class selectors
 - c. Descendent selectors
 - d. ID selectors
- If you overuse IDs in your styles, **you're going to have a bad time**

More background styles



Paint Can designed by [Alex Valdivia](#) from the [Noun Project](#)

{ } background-color review

```
.block {  
    /* text is black and centered */  
    color: #000000;  
    text-align: center;  
    /* background is light mauve (#bc7384) */  
    background-color: rgb(188,115,132) ;  
}
```

**I feel useful. I'm having a
wonderful time!**

{ } Transparent background colors

```
body {
```

```
    /* background is electric blue */
```

```
    background-color: rgb(13,65,210);
```

```
}
```

```
.block {
```

```
    /* background is light mauve and 50% opaque */
```

```
    background-color: rgba(188,115,132,0.5);
```

```
}
```

**I feel transparent. I'm
having a wonderful time!**

{ } Styling a background image

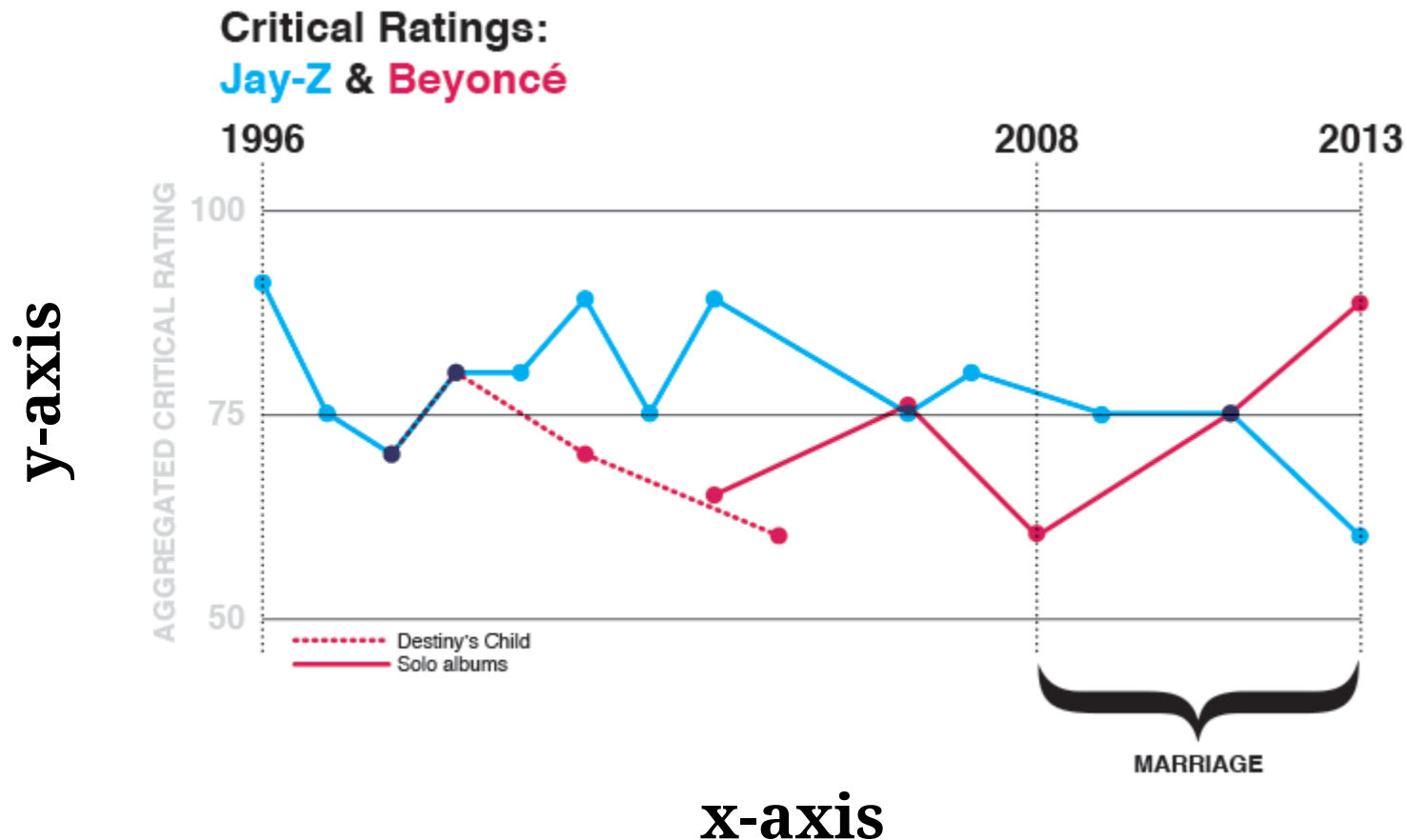
- The property is **background-image**
- The value is a **URL where an image lives**

```
.imagebg {  
    background-image: url("img/kittens.jpg");  
}
```

{ } Styling a background image

- **background-repeat** says:
 - Repeat horizontally
 - Repeat vertically
 - Don't repeat at all
- **background-position** says:
 - Start at the left or right
 - Start at the top or bottom
 - Whether the image is centered vertically or horizontally (or both)!

{ } Left & right, top & bottom



{ } Repeating a background

```
/* repeat the background horizontally */
```

```
background-repeat: repeat-x;
```

```
/* repeat the background vertically */
```

```
background-repeat: repeat-y;
```

```
/* don't repeat the background */
```

```
background-repeat: no-repeat;
```

{ } Positioning a background

- **background-position** values include both the x-axis and y-axis
- x-axis first, y-axis second
- Can be **left/right top/bottom** *or* any measurement (pixels, %, ems, etc.)

```
/* position a background in the left top corner */
```

```
background-position: left top;
```


{ } Positioning a background ex.

```
/* position a background in the left top corner */
```

```
background-position: left top;
```

```
/* positioning a background in the right bottom corner */
```

```
background-position: right bottom;
```

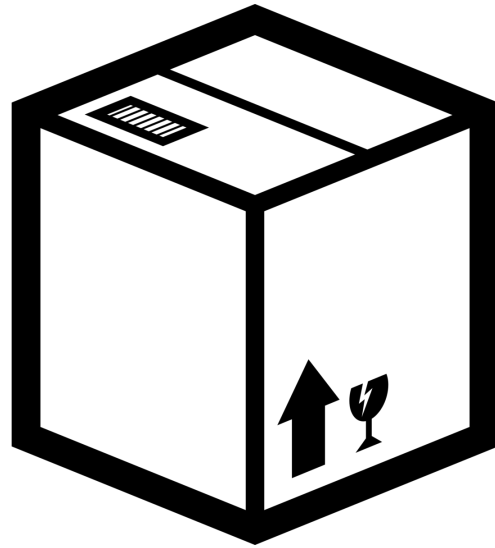
```
/* position a background on the left, centered vertically  
*/
```

```
background-position: left center;
```

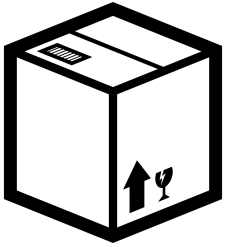
```
/* position an image completely centered */
```

```
background-position: center;
```

The CSS box model



Box designed by [Cornelius Danger](#) from the [Noun Project](#)



A CSS box model metaphor

- **Content:** stuff in the box
- **Padding:** bubble wrap & packing peanuts
- **Border:** the sides of the box
- **Margin:** space between multiple boxes

Margin

PADDING

Place sugar cube in old fashioned glass and saturate with bitters, add a dash of plain water.

Muddle until dissolved.

Fill the glass with ice cubes and add whiskey.

Garnish with orange slice, and a cocktail cherry.

PADDING

PADDING

Margin

Margin

Margin

{ } Box model content

- By default, helps determines the default **width** and **height** of the element's box
- Defaults can be overridden with CSS with the width and height properties

```
div {  
    width: 100px; /* or em or % */  
    height: 200px; /* or em or % */  
}
```

{ } Box model padding

- Creates space between content and the **border** for readability

```
/* around all sides */
```

```
padding: 20px;
```

```
/* around some sides */
```

```
padding-top: 20px;
```

```
padding-right: 2em;
```

```
padding-bottom: 20px;
```

```
padding-left: 40px;
```

{ } Box model border

- Goes around the edge of the element
- Default **width** is 0 for most elements
- Borders can have **color** and **style** too

```
border-width: 20px;  
border-style: solid;  
border-color: #ff0000;  
/* border-width for the bottom edge only */  
border-bottom-width: 4px;  
/* border-color for the left edge only */  
border-left-color: #ff0000;
```

{ } **Box model margin**

- Goes outside the **border**
- Creates space for layouts

```
margin: 20px; /* or em or % etc. */  
margin-right: 2em;  
margin-left: 2em;  
/* etc. */
```


{ } With background-image

- **background-image** styles fill both the content and the **padding** of elements
- Use with **background-position** and the box model properties to arrange your background images!

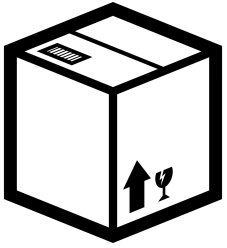


{ } What is up with my boxes?

- How containers take up space is dictated by the **box-sizing** property
- The default value for **box-sizing** is **content-box**

```
* { box-sizing: content-box; }
```

- This means that **width** and **height** include only the content by default

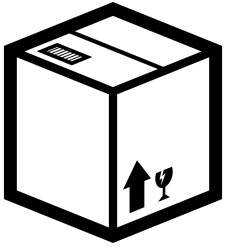


content-box multiple choice

Q: Imagine you're moving to a new place.

You decide you need to get boxes to pack up your worldly belongings so everything is secure and organized for the move.

Do you:



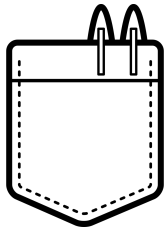
content-box multiple choice

- A. Take everything you own, put it in piles on the floor, measure around the piles, and custom cut and build packing materials for the piles?
- B.** Make some educated guesses about the different sizes of boxes you probably need, and pack your stuff into the boxes?

{ } border-box to the rescue

- Changing **content-box** to **border-box** makes it so that the width and height include the border and padding

```
* { /* this goes on everything! */  
    /* Webkit (aka Chrome and Safari) vendor prefix */  
    -webkit-box-sizing: border-box;  
    box-sizing: border-box;  
}
```



Vendor prefixes

- Vendor prefixes are an artifact of how browsers test out new CSS features
- **border-box** is a CSS3 (still shiny! new!) feature, and older versions of Webkit still require the prefix
- Check out [Can I Use...](#) to see what CSS features require vendor prefixes

Resetting browser default styles





Browser defaults can be a pain

- Every browser has slightly different styles
- Different types of elements get different font sizes, different amounts of padding , different margins, etc.
- Creating styles that work across "all" browsers can be daunting/crazy-making



A blank slate

- Reset styles strip out browser default and let us make our own defaults
- We'll use the canonical reset stylesheet:
<http://meyerweb.com/eric/tools/css/reset/>



Two ways to add reset styles

- **Method one:** Put reset styles into **their own .css file** and load it before your existing stylesheet

```
<link href="css/reset.css" rel="stylesheet">
```

```
<link href="css/styles.css" rel="stylesheet">
```

- **Method two:** Put reset styles **into the top** of your existing stylesheet



Method one: Separate styles

1. Copy reset styles
2. Paste into a new blank document in your text editor
3. Save your styles as a different .css file (e. g., **reset.css**) in your css folder
4. Add a link to the reset stylesheet in **<head>**, *before* your existing styles



Method two: One stylesheet

1. Copy reset styles
2. Paste at the very top of your existing stylesheet (e.g., **styles.css**) so they load first
3. Save your stylesheet

$(\neg \wedge \neg) \neg \neg$

"Homework" for next week

1. Reset your styles
2. Practice with image backgrounds
3. Practice with classes and IDs
4. Practice your border box

Optional: HTML and CSS ch. 11, 13, 16

Next time

- Questions and review from Week 3
- HTML data tables
- New HTML5 container elements
- CSS abbreviations
- Positioning and layouts with CSS

Questions? Comments?

- Visit dpersing.github.io/svc for:
 - Class slides
 - Code samples
 - Resources
- Email me: dep@dpersing.com
- Tweet at me: [@devonpersing](https://twitter.com/devonpersing)