# CSS & HTML: Week 3

Instructor: Devon Persing
TA: Becky Pezely

# Corrections!

# **`<meta>` attributes**

```
<meta name="author" content="
Your Name">

<meta name="description"
content="Page description.">
```

**Sorry!**

# More information!

# PNG-8 versus PNG-24

- PNG-8 is 8-bit, and PNG-24 is 24-bit (aka a bigger and better file)

- Only PNG-24 natively supports transparency, but Adobe products kind of break this rule

- Use PNG-24!

# `href versus src`

- **`href`** defines links or relationships between content or resources and downloads in parallel
  - `<a>` for links between pages
  - `<link>` for links to styles
- **`src`** embeds resources in the page and interrupts the current page download
  - `<img>` for image files
  - `<script>` for Javascript files

# Questions?

# This week: All CSS, all the time

- CSS abbreviations

- Using id and class attributes

- Styling backgrounds

- The CSS box model

- Resetting default browser styles

# CSS abbreviations

# CSS properties and values

`selector { property: value; }`

↑ the element being styled

↑ the aspect being styled

↑ the style

# Lots of related properties...

```css
h2 { font-family: helvetica,
sans-serif;
      font-size: 1.6em;
      font-style: italic;
      font-weight: bold;
      line-height: 2em;
}
```

**...can mean many lines of CSS to do only a few things.**

# Abbreviating CSS

- Some **hex color values** can be abbreviated to 3 characters from 6 characters

- Some **properties that are related to each other** can be grouped into a single property

- From here on out we'll go over non-abbreviated and abbreviated formats for each property we cover

# Abbreviated hex codes

```css
color: #333333;
/* becomes */
color: #333;


color: #aa0099;
/* becomes */
color: #a09;
```
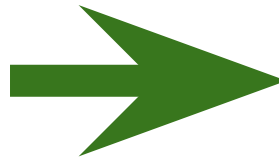
**Fun fact: These are called triplets!**

# Abbreviated font styles

```
font-style: italic;
font-variant: small-
caps;
font-weight: bold;
font-size: 1em;
line-height: 1.5em;
font-family:
Helvetica, sans-serif;
```

➡

```
font: italic
small-caps bold
1em/1.5em
Arial,
Helvetica,
sans-serif;
```

**Font size** and **font family** are required, and **order matters**.

# Abbreviated styles for `<ul>` & `<ol>`

```css
list-style-image: none;
list-style-position: outside;
list-style-type: disc;
```



```css
list-style: disc outside;
```

# Let's abbreviate some CSS.

# Using ids and classes

# New types of attributes

- Remember that:
  - HTML elements can have attributes
  - HTML elements can be used as CSS selectors

```
<a href="http://google.com">Go to Google</a>
```

```
a { color: #ff0000; text-decoration: none; }
```

# `class` attributes

- Can be applied to multiple elements to apply the same styles to them

- Are denoted in CSS with a period (`.`)

```css
.highlight { background: #ff0; }
```

```html
<h1 class="highlight">Here's my title!</h1>

<p class="highlight">Here's a paragraph.</p>
```

# `id` attributes

- An individual `id` can only be used once on an HTML page

- Is denoted in CSS with a hash (**#**)

`#title { background: #ff0; }`

`<h1 id="title">Here's my title!</h1>`

# `class` versus `id` attributes

- An element can have **both an `id` and a class**

- An element can have **more than one `class`**

- An element can have **only one `id`**

- Multiple elements on a single page **can't have the same `id`**

# class versus id examples

💚    `<h1 id="title" class="dog">`Page Title`</h1>`

💚    `<h1 id="title" class="dog cat">`Page Title`</h1>`

🚫    `<h1 id="title heading">`Page Title`</h1>`

🚫    `<h1 id="title">`Page Title`</h1>`
      `<h2 id="title">`Section Title`</h2>`

# `class` and `id` in the cascade

- An element with an `id` is more specific than...

- An element with a `class`, which is more specific than…

- An element with neither an `id` nor a `class`

# class and id descendent selectors

```css
.sidebar .category { /* some styles */ }
#header .navigation { /* some styles */ }
.sidebar #about-bio { /* some styles */ }
#about #about-bio { /* some styles */ }
/* and */
#about a { /* some styles */ }
div .category { /* some styles */ }
/* etc. */

/* work just like */
p a { /* some styles */ }
div p a { /* some styles */ }
```

# styling to combined `class`/`id` selectors

```css
.sidebar.container { /* some styles */ }
#title.sub-page { /* some styles */ }
.page#title.sub-page { /* some styles */ }
div.page#title { /* some styles */ }
```

- String `class`/`id` selectors without spaces

- Style elements that have:

  - Two (or more) specific `class`es

  - A specific `class` and a specific `id`

  - Any combination of `class`es and `id`s

**Let's try some ids and classes.**

# CSS backgrounds

# CSS `background-color`

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

- Can be applied to the whole **body** or to any individual element
- Uses the same syntax as the **color** property for text

# CSS `background-image`

```css
body { background-image: url("../img/kitten.jpg"); }
```

- Can be applied to the whole **body** or to any individual element
- Can be added on top of a **background-color**
- Use a relative URL to the image file

# CSS `background`

```css
body { background: #eee url("../img/kitten.jpg"); }
```

- **background-image** and **background-color** can be abbreviated into **background**
- Color is always listed first, then the image

# Transparent backgrounds

```
body { background: rgba(200, 54, 54, 0.5); }
```

the color opacity

- **rgba** adds alpha transparency support to **rgb**
- Opacity is measured in decimals
- Older browsers will degrade to a solid color
- **rgba** can be used for the **color** property for text too!

# CSS `background-image` position

- By default, background-image repeats from the top left

- Properties can be used to:

  - Prevent repeating

  - Repeat in a specific direction

  - Display the image in a certain place in the background

# `background-image` options

```
/* repeating */
background-repeat: repeat-x; /* horizontal */
background-repeat: repeat-y; /* vertical */
background-repeat: no-repeat;/* none! */

/* position for no-repeat: vertical and
horizontal */
background-position: top right;
background-position: center center;
background-position: 20px 10px;
/* that is: */
/* 20px from the top (y-axis) */
/* 10px from the left (x-axis) */
```
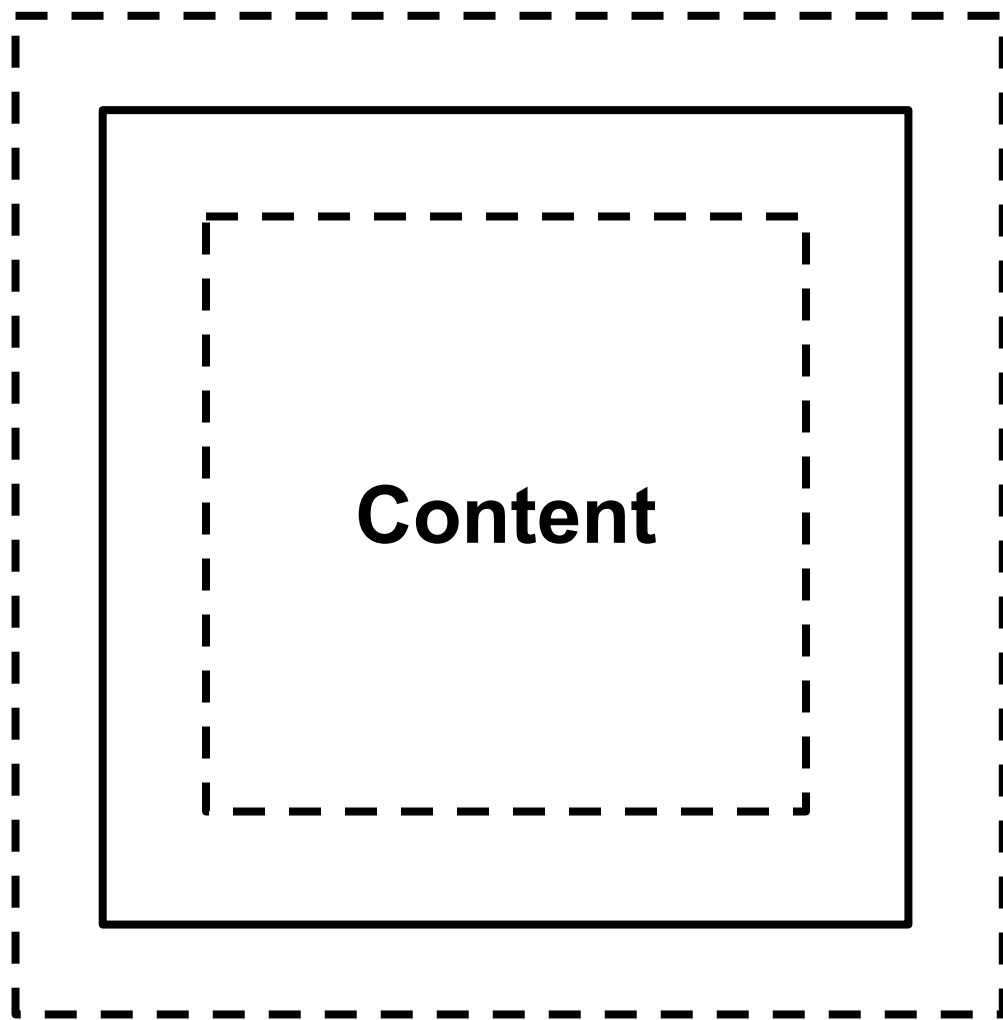
# Adding these to `background`

```css
/* a div with a colored background, and a non-repeated
background image positioned in the bottom right */

div {
   width: 400px;
   height: 400px;
   background: #eee url("../img/kitten.jpg")
no-repeat bottom right;
}
```

# Let's try some backgrounds.

# CSS box model

**Content**

# CSS box model components

- **Content** that defines the default width and height of an element's "box"

- **Border** that goes around the edge of the box

- **Padding** that adds space between the content and border

- **Margin** that adds space outside the border

- Edges are referenced clockwise

# Box model content

- Determines the default **width** and **height** of the element's box

- Defaults can be overridden with CSS

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

# Box model `border`

- Goes around the outside edge of the element's box

- Default `width` is `0` for most elements

- Borders can have **color** and **style** as well

```
border-width: 20px;
border-style: solid;
border-color: #ff0000;
border-bottom-width: 4px; /* overrides
border-width for the bottom edge */
```
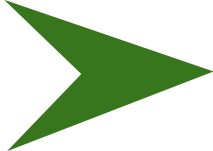
# Box model `border` abbreviated

**border-width**: 4px;
**border-style**: solid;
**border-color**: #a00;

> **border**: 4px solid #a00;

**border-top-width**: 4px;
**border-right-width**: 2em;
**border-bottom-width**: 3px;
**border-left-width**: 5px;
**border-style**: solid;
**border-color**: #a00;

> **border**: 4px 2em 3px 5px solid #a00;

# Box model `margin`

● Goes outside the `border`

● Creates space between elements for layout

● Size can be specified with CSS

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

margin: 20px 2em; /* shortcut! */
```

# Box model `padding`

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

- Size can be specified with CSS

```
padding: 20px; /* or em or % */
padding-right: 2em;
padding-left: 40px;
/* etc. */

padding: 20px 2em 20px 40px;
```

# Box model `padding` and `background`

- **`background`** styles fill the content and the **`padding`** of elements

```
padding: 20px;
background: url("kitten.jpg");
border: 10px solid #000;
```
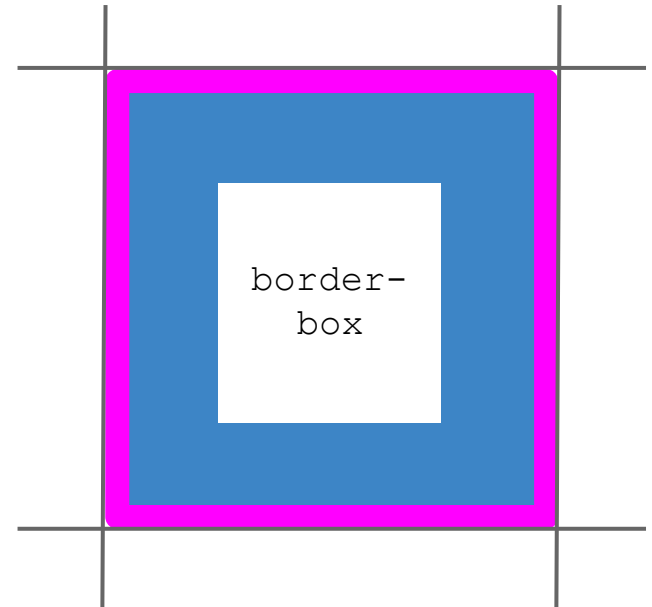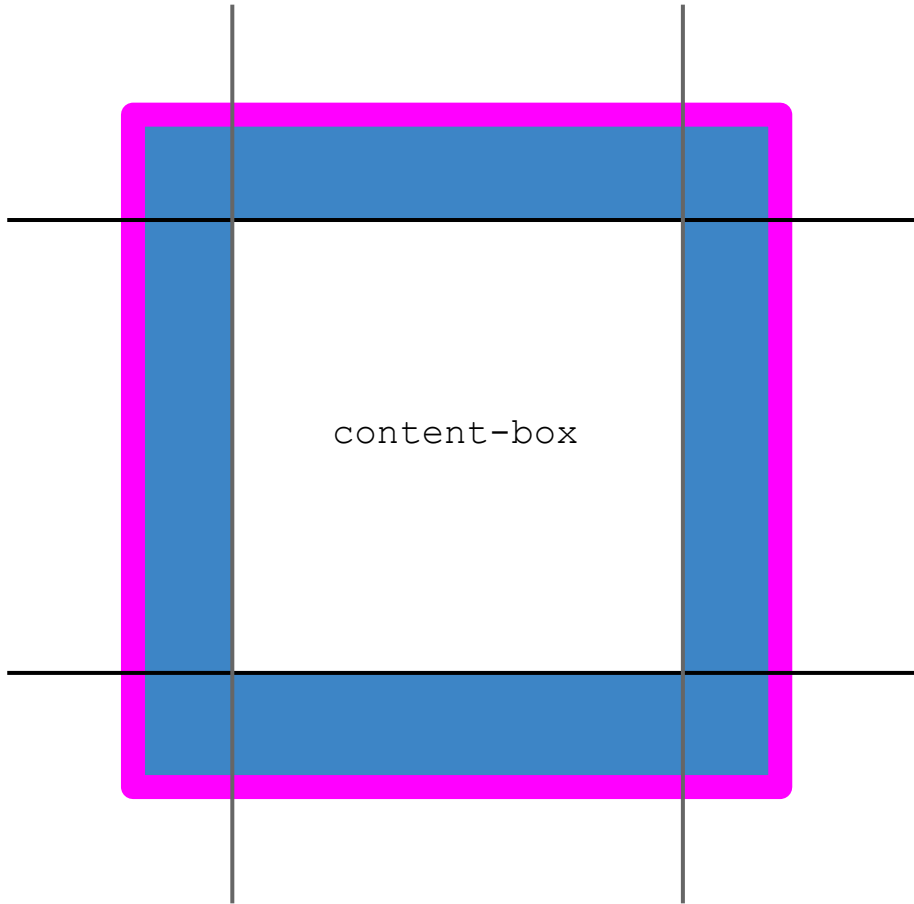
# Let's try some box model stuff.

# `box-sizing` to the rescue

- The **box-sizing** property affects how containers are rendered on the page, with `content-box` as the default value

- The `border-box` value renders border and padding inside the container instead of outside

# content-box to border-box

# Using `box-sizing`

```css
div {
    /* Firefox vendor prefix */
    -moz-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 (shiny! new!) feature, and Mozilla/Firefox still requires the prefix

- Check out **Can I Use…** to see what CSS features require vendor prefixes

Let's try using `border-box`.

# Browser default reset

# Browser defaults can be a pain

- Every browser has slightly different styles

- Different types of elements get different amounts of padding, different font sizes, 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:**

Copy and paste reset styles **into the top** of your existing stylesheet.

**Method two:**

Copy and paste reset styles into **their own .css file** and load it before your existing stylesheet.

# Method one: 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

# Method two: Separate reset styles

1. Copy reset styles

2. Paste into a blank document in your text editor

3. Save the blank document as a different .css file (e.g., **reset.css**)

4. Add a link to the reset stylesheet in `<head>`, **before** your existing styles

# For next week

- Add **browser reset styles** to your project and tidy up the results

- Give your page more style with:

    - the box model

    - backgrounds

    - ids and classes

- *HTML and CSS*: read ch. 13-15

# **Next time**

- New HTML5 container elements

- HTML data tables

- Positioning content with CSS

- Multi-column and grid layouts!

- CSS pseudo-classes

# Questions?

- Visit [http://dpersing.github.io/svc](http://dpersing.github.io/svc)

  - Class slides

  - Code examples from class

  - Additional general and class-specific resources

- Email me at [dep@dpersing.com](mailto:dep@dpersing.com)