

ECS 162 WEB PROGRAMMING

4/3

Book

- Reading on HTML and CSS in:



To see the book for free from off campus, install VPN client (see Library Web page)

Start up VPN, use link on to book on class Web page.

You might have to login to O'Reilly's Safari system.

Verifiers

- Why do we have to upload our HTML and CSS files to a verifier?

Verifiers and Browsers

- Why do we upload HTML and CSS to a verifier?
 - Because Web browsers accept almost anything, and they don't give (obvious) error messages.
- If it works, why do we care if it's wrong?

Verifiers and Browsers

- Why do we upload HTML and CSS to a verifier?
 - Because Web browsers accept almost anything, and they don't give (obvious) error messages.
- If it works, why do we care if it's wrong?
 - Cross-browser compatibility.
 - Hidden bugs might lead to trouble down the road
- Why don't browsers give error messages on bad HTML?

Verifiers and Browsers

- Why do we upload HTML and CSS to a verifier?
 - Because Web browsers accept almost anything, and they don't give (obvious) error messages.
- If it works, why do we care if it's wrong?
 - Cross-browser compatibility.
 - Hidden bugs might lead to trouble down the road
- Why don't browsers give error messages on bad HTML?
 - A good browser displays as many Web pages as it can nicely

CSS vs HTML

- HTML tags assign meaning to items; CSS is used to control their appearance and behavior
- For historical reasons, many element style attributes can be specified either in HTML or in CSS.

We will put content in HTML, style in CSS.

- CSS can go at top HTML file, inline anywhere with the `<style>` tag, or in separate file.

We'll always put CSS in a separate file.

Learn these HTML tags

- h1, h2, h3 - headings
- img – image
- div, header, main, footer, nav –no default effect, semantic meanings used by screen readers for the visually impaired
- ul, ol, li – lists
- button
- a – is for “anchor”. A link.
- em, strong – often styled in css as italics, bold
- span – also no default effect

Adding a CSS file

- To attach a css file, put into the HTML `<head>`:
`<link rel="stylesheet" type="text/css" href="poultry.css">`
- Can be a relative or absolute link. Which is this?
 - Relative; css file lies in same directory as html
- Most browsers have some default css so that raw html does not look totally awful. Get rid of it by attaching a “reset.css” file. There is one on the course assignments Web page.

Font stacks

- The browser picks the first font that it has available.
- Some typical choices (pick ONE):

font-family: "Lucida Sans Unicode", "Lucida Grande", sans-serif;

font-family: "Baskerville", "Times New Roman", Times, serif;

serif sans-serif

Google fonts

- Google has a collection of fonts that a Web page can download and use. The HTML has to do the download:
`<link href="https://fonts.googleapis.com/css?family=Montserrat:200,500" rel="stylesheet">`

font-family: montserrat, sans-serif; font-weight: 200;
font-style: normal;

Tip

- Start a CSS session by giving everything a distinguished background color
- You can get rid of them at the end

Selectors

- Attach style attributes to HTML elements using CSS selectors
- Select all elements of a given type:

```
<h2>
  Exotic Poultry
</h2>
```
- And a CSS selector modifying it:

```
h2 {
  font-size: xx-large;
  background-color: paleturquoise;
}
```

Selectors

- ```
h1 {
 color: blue; /* text color */
 text-align: center; /* centered */
 background-color: white;
}
```
- CSS syntax is similar to what?
  - color, text-align, background-color are properties
  - How do you find the possible color names?

## Class selector

```
<h1 class="myHeading">Working Title</h1>
```

```
.myHeading {
 color: blue;
}
```

- Notice the leading period (indicates class name)
- You make up the class names; all one word, case sensitive.
- There can be many elements, of different types, belonging to class "myHeading"

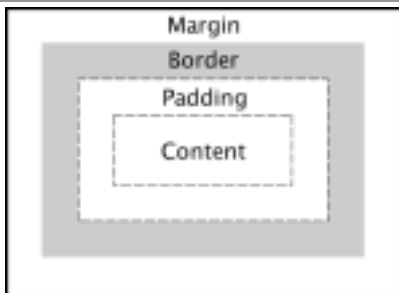
## Id Selector

```
<h1 id="mainHeading">Working Title</h1>
```

```
#mainHeading {
 color: blue;
}
```

- Note the hash symbol.
- You make up the ID names. There should be only one element with this ID in the document.
- There are more selector methods.....

## Margin, border, padding



## Inheritance in CSS

- Some CSS properties are inherited by child DOM elements (eg. font color, font), some are not.
- Mainly you have to experiment, and/or look it up
- But a property specified for a child element overrides the property inherited from an ancestor.

## The Cascade

- A page can have many style sheets attached, contradicting each other!
- Basic rule:
  - More specific specifiers override less specific.
  - If equally specific, later override earlier.
- For example, here's some HTML:

```
<p class="bird" id="peacock">
All about the peacock.
</p>
```

## Quiz

```
#peacock { background-color: green; }
p.bird { background-color: blue; }
p { background-color: pink; }
```

- What color is the paragraph?

## Quiz

```
#peacock { background-color: green; }
p.bird { background-color: blue; }
p { background-color: pink; }
```

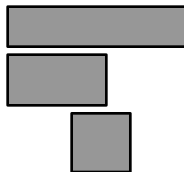
- What color is the paragraph?  
Green, because an ID specifier is more specific than a class specifier is more specific than an element-type specifier.

## Default layout

- How does the browser display the Web page in the absence of any instructions from CSS?
- There are two default display modes.
- Every element has a default mode.

## Block

- The beginning of a new block displayed element, such as a paragraph, stacks the new element under the existing layout.



## Inline

- Elements with the inline display property are arranged within their block from left to right.



- Text is inline (but paragraph is block). em, strong, span, img are inline.

## Example

- `<img>`, `<button>`, `<span>` elements are inline by default. So if we move our images into the paragraphs...

Are `<div>` elements block or inline?



## Example

- `<img>`, `<button>`, `<span>` elements are inline by default. So if we move our images into the paragraphs...

Are `<div>` elements block or inline?

Block! We see the two divs stacked on top of each other.



## Next assignment

jan shrem and maria manetti shrem museum of art [Home](#) | [About](#) | [Contact](#)



### Bruce Nauman

*Blue and Yellow Corridor*

**Extended through April 14th**  
Guest Curator: Ted Mann

Coinciding with the opening of Bruce Nauman's (MFA '66, LIC Davis) retrospective at The Museum of Modern Art, New York, this exhibition features the first realization of a participatory environment Nauman conceived in 1970. The work is a narrow passageway that wraps around an existing room, combining colored fluorescent light and closed-circuit video to manipulate the viewer's perceptual experience. An adjacent gallery includes artworks that situate the corridor within the artist's career.

[View Media](#)

