ECS 162
WEB PROGRAMMING

4/1

Introductions

- Me: Prof. Nina Amenta
- TAs:
  - Erica Jurado
  - Uzair Inamdar
  - Nehal Agrawal
  - Anshi Agarwal
  - Jamie Oka (Web designer)

Welcome!

- Get to the point where you can build an application that runs over the Web.
- Example Web applications:
  - Banner
  - Amazon
  - Google
  - Facebook
  - The New York Times

Waitlist

- We are not going to expand the class
- There are 165 people total on the waitlist
- If you are on the waitlist, you are on a specific waitlist for one of 6 sections of 39
- So if you are number 10, 10 out of those 39 have to drop for you to get in. Chances are not great.

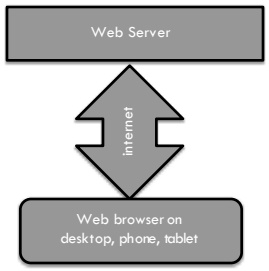
The Web

- What is the difference between the Web and the internet?

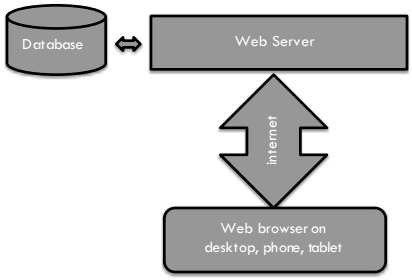
The Web

- What is the difference between the Web and the internet?
- The internet is the network of cables and wireless connections linking computers together, and the software that delivers messages across those connections.
- The World-Wide Web is one of many programs that use the internet.
- To learn about the internet take ECS 192A!

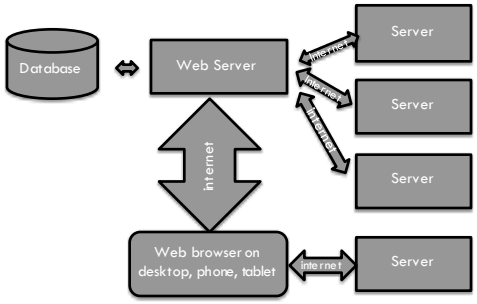
Big picture



Big picture



Big picture



Web application

- How it works
  - Collection of programs
  - Some are written by us, others are part of browser or server software
  - Some run in the Web browser, on the user's device
  - Some run on a Web server, or a collection of Web servers
  - Pass information back and forth via internet
  - Generate new Web pages or alter existing Web pages to interact with the user (register for course, buy a book, order a pizza, read the news….)

Many languages and frameworks

- Web pages – HTML and CSS
- Browser programming – Javascript, Jquery.
- Browser frameworks such as Angular, React, or D3 for visualization.
- Server-side programs – Lots of different Web app frameworks
  - PHP, Ruby on Rails,node.js (Javascript), Django (Python), ASP.net(Javascript/Perl/VBS),…
- Databases usually use SQL
- APIs for Web services such as Google maps or Facebook
- Authentication for apps and for users (internal, 3rd party, both)
- Any project has to choose some subset

Syllabus – choices for this course

- HTML5, CSS3
- Responsive front-end design with Flexbox, React
- Object-oriented Javascript
- AJAX, JSON and asynchronous software
- Server programming with Node.js
- A database (probably SQLite)
- Using an API
- Possibly:
  - Authentication, probably using the Passport module
  - Visualization with SVG and D3

Building complexity

- Web programming has the reputation of being easy.
- This is because people often mean simple "coding" HTML and CSS.
- We'll cover that this week.
- Later stuff builds on earlier stuff, so it gets harder as we go on. Do not be fooled by the first week.
- People earn C's and F's in this course, and we give them.

Drop deadline

- 10 day drop deadline
- Drop by Friday April 12

Prereqs

- This is an upper division CS course. You may use it as an elective for the major or minor in CS.
- ECS 30, 32B, 36B or equivalent programming experience is required. You need to have a solid programming background.
- You need to be comfortable in UNIX.

Discussion sections, office hours, labs

- Lab sessions rather than sections
- Will focus on doing the homework, hands-on examples of concepts.
- TAs and I will hold drop-in lab hours in CSIF, watch the Web site for where and when.
- Lab and office hours are not required
  - You are welcome to just do the readings and assignments, and ask questions via Piazza.

Assignments

- Check the Website for assignments; this week's is up already.
- It is due on Thur, 4/4.
- http://web.cs.ucdavis.edu/~amenta/s19/ecs162

Schedule

- (4/4) Hello world
- (4/11) Layout, Flexbox (Museum exhibit)
- (4/25) API, callbacks, CORS (Weather)
- (5/2) Server setup
- (5/6) Midterms
- Project…
- (6/13) Final

Exams

- Multiple choice and some code
- Open book, open notes, assigned seats

Grading

- Assignments and project – 35%
- Midterms – 30%
- Final – 35%

Academic dishonesty

- Major similarities on assignments will be sent to SJA.
- Lots of Web programming is very formulaic.
  - Eg. the command to request something from the server is complicated, and everyone copies and pastes it, and then edits the parameters.
- If you don't understand what you're doing, you're cheating.
- The level of your understanding shows up on the tests, especially when we ask you to write code.

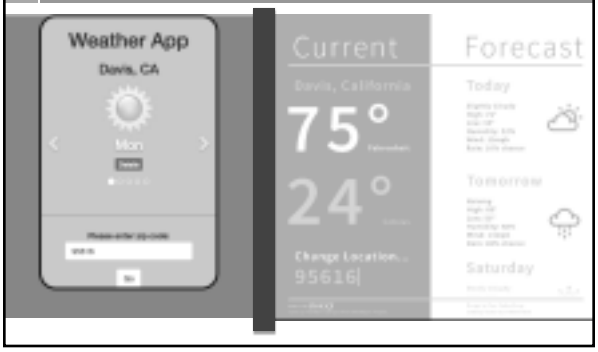
Design

- What is the difference between Web programming and Web design?

Design

- What is the difference between Web programming and Web design?
- Web design is a commercial art form using layout, color, fonts, images, and symbols to convey information and make navigation easy.
- Web programming implements Web design.
- Typically Web designers know some Web programming, but often not the other way around.
- In the workplace, they work as an interdisciplinary team.

Which one is by a designer?



Design

- Jamie is our designer. She will give us designs to code towards for all assignments.
- Part of your grade on the early assignments will be how close you get to implementing the design. This is surprisingly hard!
- She will provide extensive design documents and explain her design decisions in class.
- If you want to do your own designs for the project, you will need to submit equally detailed documents and explain your design decisions in writing.

HTML

- What does this stand for?

HTML

- Hyper-Text Markup Language
- What does this mean?

HTML

- Hyper-Text Markup Language
- Markup Language – tags to indicate the roles of different parts of the Web page. Tags are labels for bits of content, usually enclosed on angle braces: <>
- Hyper-text means it includes links.

Example!

Browsers are lenient!

- The browser will do it's best to render anything you send it.
- It won't produce (obvious) error messages for most mistakes.

A few principles

- Page elements (text, pictures, etc.) need to be nested inside tags, eg. <head> </head> and <body></body>.
- HTML tags should be nested.

GOOD

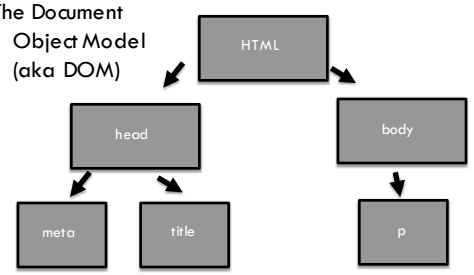
```
<body>
    <p></p>
</body>
```

BAD

```
<body> <p>
</body></p>
```

Tree structure of Web pages

The Document Object Model (aka DOM)



The DOM

- The DOM is the browser's internal data structure containing the Web page content.
- It is a tree. It is initialized by reading the HTML file.
- Does the browser construct the tree in depth-first or breadth-first order?

The DOM

- The DOM is the browser's internal data structure containing the Web page content.
- It is a tree. It is initialized by reading the HTML file.
- Does the browser construct the tree in depth-first or breadth-first order?
  - Depth-first. Each pair of matching opening and closing tags is completed before its sibling pair is started.

Collaboration with the browser

- All of our Web programming code is just little pieces that get plugged into larger systems and frameworks.
- Here we plug some HTML into a browser.
- Both are needed to produce Web pages.
- Understanding the larger systems and frameworks lets us know how to use them effectively.

Syntax details

- Start with <!doctype HTML>; this means HTML5.
- Whitespace is not meaningful except to separate words.
- Capitalization in tags or attributes never matters:
  <Img> <img> <IMG> <iMg> - all the same
- <head> usually includes a <title>. The title shows up in the tab, not on the page.
- <meta charset="utf-8"> is a "self-closing" tag. It just has a start tag, no end tag. The attribute charset defines the alphabet for the Web page. Attributes are inside the angle brackets of a tag.

First assignment

- Make a Web page for yourself.
- Found at:
  http://web.cs.ucdavis.edu/~amenta/s19/assn.html