Lecture 1: Introduction and Fundamentals

What is web development?

What is web development?

Simply put, web development is the the very broad field of creating and building websites and web applications.

There are many opinions out there on what the best technology is and what the best practices are: this course is not about opinions and proclaiming what the best technology, but rather arming you with the skillset needed to work in any web technology.

What’s special about web development?

There are many aspects of web development that are not different than non-web development

- You will break complex problems down to smaller, approachable issues
- There are a slew of programming languages and technologies to choose from

There are some unique problems, as a web developer, that you have to care greatly about

- In some way or another, your product is about transmitting information; you have to worry about the actual delivery of your information.
- Technology on the web moves **fast!** Change is constant!

Why is web development important?

Web development has allowed the internet to bring forth a new era for technology

- Lower barrier of entry; you can start without a compiler, all you need is a text editor and a browser!

Web development has ushered in a new era of communication, where ideas and content can be transmitted in new and exciting ways.

Allows information to be spread much more rapidly through many mediums.

Course Introduction

This week

Introductions

What are we going to cover in this course?

What technologies and tools will we use?

How does the web work?

About You

What’s your name, and what do you prefer being called?

What’s a fun fact about yourself?

What experiences have you had with programming, web or otherwise?

What’s something that you are interested in with regards to web programming and hope to learn in this class?

Grade Breakdown (1/2)

Homework Assignments: 25%

- There will be five homework assignments to practice the material covered in class.
- Assignments will be weighted equally

Labs: 15%

- There will be ten lab assignments given for the last half hour of lectures.
- Labs will be weighted equally

Class Participation: 5%

- Attendance is graded, and will account for 5% of your overall grade.

Grade Breakdown (2/2)

Final Project Proposal: 5%

- Students will form groups and propose a final project to work on throughout the semester.

Database Proposal: 5%

- Each group will submit a proposal for their database collections and schema

CEO Presentation: 10%

- Each group will give a non-technical presentation of their project, explaining what use case it solves, who the audience is, and why it's a worthwhile project.

Final Project Presentation: 10%

- You will give a technical presentation showcasing your project's features and technological decisions.

Final Project: 25%

- Each group will submit their project code and a database dump to be reviewed and graded based on what was promised in their proposal and delivered.

What will we be covering in this course?

In this course, we will be going through many fundamental web concepts and learning technologies related to them.

1. You will learn how to install and configure a modern web programming environment, from server to database. In our case, we will be using Node.js, MongoDB, and Express as our programming environment, database, and server respectively. You will also learn about many tools that you will be using as a web developer, such as Git.
2. You will learn how to do server-side programming. In our case, that means you will learn the JavaScript language, as well as good coding patterns in order to structure a web application. You will learn how to separate your code in logical ways that make sense and follow modern conventions.
3. You will learn how to use a modern database; in our case, this is MongoDB. You will learn what this database's strengths are, what its weaknesses are, and how to utilize it effectively.

What will we be covering in this course?

4. You will learn how to code for the client. Learning HTML and CSS will enable you to create a document that makes sense, both semantically and meaningfully to the human eye. You will learn how to use JavaScript to make your applications respond to your users' input and experience.
5. You will learn about web accessibility and the major hurdles that many people face using the web as it is today, how to identify issues that exist in a web page, as well as how to correct them.
6. You will learn advanced client-side programming techniques and how to leverage frontend tools that allow you to create incredibly dynamic web experiences.
7. You will learn about security issues in the web and how to minimize their risks.
8. You will create a market-ready database driven web-application, from start to finish, involving technical presentations

What will we *do* in this course?

In this course, you will:

- Complete a number of labs that will assess your understanding of the topics covered in class. These will be simple programming assignments that will serve to help you on your assignments.
- Complete several programming assignments over the course of one to two weeks.
- Form a group and come up with an idea for a full web application to use as your final project, which you will submit in the form of a final project proposal.
- Setup your database schema and submit a database proposal.
- Give a non technical presentation detailing your project idea.
- Give a technical presentation demonstrating your project result.
- Submit your final project code.

What are labs like?

There are ten labs, designed to make you practice the material that we will go over in class that week. The labs will give you a good foundation for your assignments and final projects.

Labs will begin at the end of lecture, and are designed to only take an hour; myself and the CAs will be available until the end of lecture. You will, however, be given 24 hours to complete the lab.

Labs are focused on small, approachable goals.

The majority of your labs will be incremental: they will build on the solutions from the previous week.

What are assignments like?

There are five assignments, designed to make you incorporate several lectures worth of material together. They will give a good foundation towards your final project, as they are more involved.

You will be given two weeks to complete each assignment. You will never have multiple assignments due at the same time, however there may be some weeks where a portion of the final project is due and an assignment is due at the same time.

What is the final project like?

There are five assignments, designed to make you incorporate several lectures worth of material together. They will give a good foundation towards your final project, as they are more involved.

You will be given two weeks per assignment, with an occasional break between

General Policies and Notes

You will have ample time to complete each assignment, so lateness will be harsly penalized:

- Late assignments will receive a 15 point reduction per day
- Late labs will receive a 25 point reduction per day
- There is no lateness tolerated on final project components; if you miss a deadline, you will receive a 0 for that component.
- **If some extenuating circumstance occurs that will cause you to be late, and you know in advance that it will cause you to be late, reach out to me as soon as possible.**

In lieu of a textbook, assignments will require you to research the topics in order to complete them. I will point you to resources for each assignment.

If you are having a group issue during the final project period, you must reach out to me **as soon as possible**; it is much easier for everyone to resolve issues early and amicably than let them destroy a group as the deadline approaches.

Technologies and Tools

Git: For Version Control

If you are unfamiliar with Git, it is a versioning control software.

Versioning control allows us to take periodic snapshots at code and save a reference to it at a certain point in time.

You can download Git here:

- https://git-scm.com/downloads

Node.js (Server-Side JavaScript)

Node.js is a JavaScript runtime environment that allows you to write JavaScript without a browser. It also exposes a number of system operations that allow you to manipulate files, make servers, etc.

Node.js has a huge community and large package repository, making it easy to build applications without having to re-engineer the wheel.

You can download Node.js here:

- https://nodejs.org/en/

MongoDB

MongoDB is a document based database.

You can download MongoDB here:

- https://www.mongodb.org/downloads#production

tota11y

The *tota11y* tool is an accessibility testing tool created by Khan Academy for the sake of identifying accessibility issues.

You can install it via a bookmarklet from the tota11y website

- http://khan.github.io/tota11y/

You may wonder why it's called *tota11y*; the phrase *a11y* is a condensed version of the term *accessibility*; there are 11 letters between the *a* and *y* in that word.

Being able to call yourself an accessibility *ally* also makes it quite a good phrase.

HTML and CSS

HTML and CSS are the markup and styling languages of the web, respectively.

HTML describes the format of a document, while CSS is a set of specifications as to how a document is styled.

You will write HTML and CSS to make web pages and applications.

JavaScript (Browser)

JavaScript originated as a programming language that was only run in a web browser. You will not only be writing JavaScript to run on Node.js, but you will also be writing it to run in your web browser.

In a browser environment, you will not have access to a user’s file system; instead, you will have access to a number of browser APIs, such as a limited use of a user’s history, their screen size, and so on. This will allow you to create robust web applications.

AngularJS

AngularJS is a JavaScript library that runs in a user's browser. It allows for the creation of powerful web applications.

There is no best frontend framework, however Angular forces developers to follow a number of opinions that are very relevant to modern web development.

What is the web, and how does it work?

The Core Process of the Web

At the end of the day, the web is all about the communication of ideas. Everything in the web can be seen as a request and a response (input and output).

- When you go to a news website, you're requesting news and receiving news in response.
- When you go to a shopping website, you're requesting product information and receiving relevant information.
- When your server receives input, it determines what to do with that input and outputs the proper response.

Your duty as a web developer is to make that communication possible. Your programs will get a request and give a response, and allow that communication to occur as smoothly as possible.

The Browser

There are many different browsers, but they each allow for the same fundamental actions to occur:

- They allow a user to navigate to a url
- They then submit a **request** on the user's behalf to the server at that url
- They receive a **response** back from the server
- They render these responses
- They execute any code that they may have received in these responses, for the lifetime of the user being on the web page.

What’s in a request?

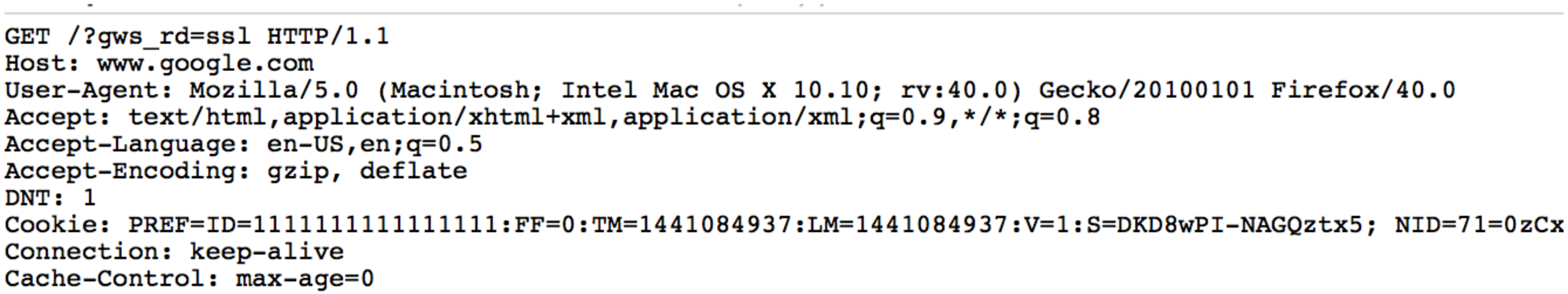
When the user navigates to a url, the browser sends data to the server at that url for the user.

You can see a good example at http://rve.org.uk/dumprequest

There are many different types of request that can happen. The two most common requests are:

- GET signifies a request to receive data from a server
- POST signifies a submission of data to a server.

The program on the server will decide how requests are handled and how the server will respond.



```
GET /?gws_rd=ssl HTTP/1.1
Host: www.google.com
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.10; rv:40.0) Gecko/20100101 Firefox/40.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
DNT: 1
Cookie: PREF=ID=1111111111111111:FF=0:TM=1441084937:LM=1441084937:V=1:S=DKD8wPI-NAGQztx5; NID=71=0zCx
Connection: keep-alive
Cache-Control: max-age=0
```

What does the server do with that info?

The server reads that request and determines what needs to be done in order to generate a response that makes sense for the data that the server has been given.

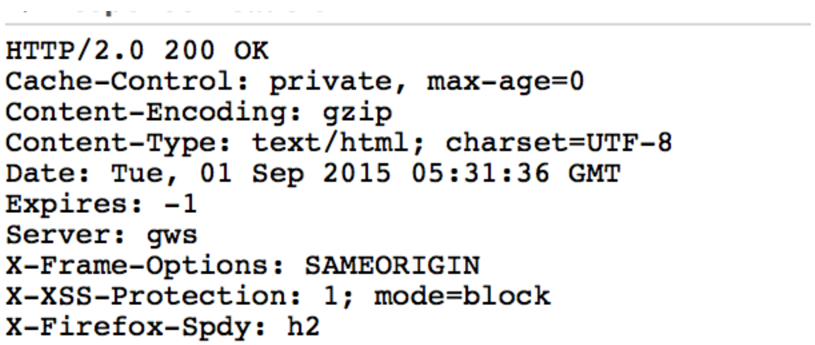
The server can use a ton of data from a request in order to generate a response. Some common types of data in a request that the server could use are:

- Querystring params
- Headers
- Cookies
- Request body

What should the server send back?

The server sends back a response that is similar to the request. It contains:

- A status code
- A set of headers
  - Cookies, data about the response such as content type
  - This is often "meta-data" about the response.
- Some form of response body:
  - An HTML Document
  - A JSON response
  - A File Stream
  - Plain Text
  - Etc.



```
HTTP/2.0 200 OK
Cache-Control: private, max-age=0
Content-Encoding: gzip
Content-Type: text/html; charset=UTF-8
Date: Tue, 01 Sep 2015 05:31:36 GMT
Expires: -1
Server: gws
X-Frame-Options: SAMEORIGIN
X-XSS-Protection: 1; mode=block
X-Firefox-Spdy: h2
```

A Document Response

When a server responds with a document, it means that the server has responded with data that is formatted in a document

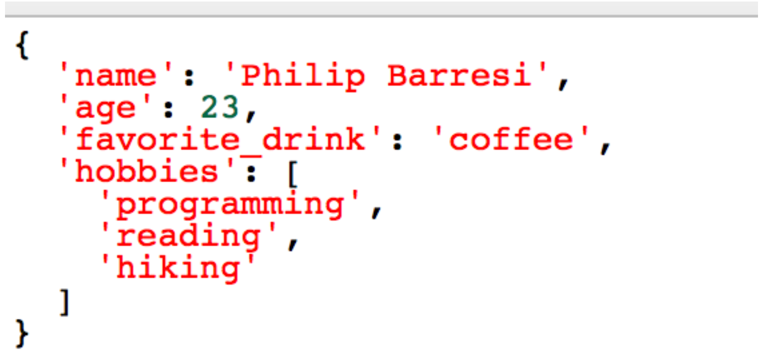
An “API Response”

A server can send back a response to a request that’s not a document to be styled and rendered, but rather some form of data. These responses can be in a variety of formats, from plain text to “**JSON**”, which we will use in this course.

JSON?

JSON is JavaScript Object Notation, a very popular way for representing data. Many technologies in the modern era can easily communicate with each other by using JSON to represent data between them.

JSON will become more useful to us once we really start programming, but here’s a preview of a JSON object describing myself:



```
{
  'name': 'Philip Barresi',
  'age': 23,
  'favorite_drink': 'coffee',
  'hobbies': [
    'programming',
    'reading',
    'hiking'
  ]
}
```

Preparing for Next Week

Bring your laptop!!!

Next week, we will be setting up our programming environments.

You will need to bring your laptop into class each week.

Install Node.js, npm, and MongoDB

You will need Node.js next week

- While installing, it may ask you if you want npm; **you do**.

You will not need MongoDB for a few weeks, but it does well to install it and test it now.

Check out some Text Editors and IDEs.

There are many options for text editors to program for this course. Some common text editors people like for frontend programming are:

- Sublime Text
- Notepad++

IDEs

- Webstorm
- Visual Studio Code

Download a developer-friendly browser

While you can use any browser you want on a day-to-day basis, you're going to want to develop a browser that has a great developer tool panel:

- Mozilla Firefox
  - https://www.mozilla.org/en-US/firefox/new/
  - http://getfirebug.com/
- Google Chrome
  - http://www.google.com/chrome/

Read up on the Fundamental JavaScript

Mozilla, the maintainers of the Firefox browser, are excellent resources for all things frontend; for now, you can start by looking at JavaScript basics.

- https://developer.mozilla.org/en-US/Learn/Getting_started_with_the_web/JavaScript_basic

Questions?