Server-side programming

CSC309
Mark Kazakevich
So far..

- For the first half of the course, we’ve focused solely on **front-end** development
  - Everything that goes on in the browser
  - HTML, CSS, and JavaScript
Now..

- We are now going to move on to the other side of the coin: **back-end** programming
Back-end programming

- **Back-end** programming involves going beyond the browser
  - Communicating with servers, databases
    - Obtaining and manipulating web resources (like the ones we’ve made!)
  - Running the programs and logic that can’t easily be run on a client machine
    - Complex calculations
    - Fast computation
    - Data storage/persistence and retrieval
HTTP

- Back in the first week of class, we talked about the **client-server** relationship on the web
  - We use **HTTP**, to communicate at the **Application Layer**

```
"Ok, here is index.html."

"Give me index.html"

"Ok, got it."

Client

Server
```
Application layer for back-end?

- We now know all about the front-end application layer on the client machine.
  - The browser (HTML, CSS, JS)

- What about the server machine?
  - What does it use at the application layer?
What should the server machine have running?

- At the minimum, a **web server** application
  - The program that receives requests and sends responses

- In addition to a web server application, the machine can also have **back-end code** that is used to create appropriate responses, HTML files, etc.
For the **back-end code**, a server machine can use...anything!

- **The browser** can only understand a few things natively
  - HTML, CSS, JavaScript

- **The server** machine can use any **programming language** for its web work
  - It’s just an ordinary computer
    - ..but usually with nicer specs
Web Frameworks

• To help write back-end web code, a **web framework** usually exists for a programming language.
  ○ Set of tools for easily creating the back-end of web apps (including the web server) in that particular language

• Important: These back-end web libraries can be **separate** from the web server
  ○ They can however interact with the server in a streamlined way
Web Libraries

- Some languages have built-in web libraries
- Lots of 3rd-party web libraries out there in whatever language you like
  - Django, Pyramid, Flask (Python)
  - Rails (Ruby)
  - JSF (Java)
  - PHP has many
What are we using in this course?

- So that we don’t have to learn any new languages, we are going to use **JavaScript** as our back-end language.

- Recall - JS needs a runtime environment
  - So far, we’ve used the browser
  - Not efficient to use a browser as your backend runtime
  - Instead, we use **Node.js**
Node.js

- Node.js is a JS runtime environment
- Allows us to run JavaScript outside of the browser
- Implements ES
- Many modules have been written for it to help you write great server-side code
Node.js demo
npm

- Package manager
- Allows you to easily install 3rd-party modules
- Auto-installs everything given a config file