UVic Web Development and Design Club

Welcome all! 🎉

About Us 😄

- We're a community of students and developers who like learning about web development
- We meet every week
- We organize workshops on various web development and design topics
- We typically organize a yearly hackathon, sponsored by local tech companies

Hi, my name is Juan 👋

- 4th year Computer Science at UVic
- I've done co-ops at AbeBooks, Workday, and Microsoft
- Most of my industry experience has involved web technologies in some capacity
- I think the web is cool because it provides a **platform** for a wide variety of apps and services.

Who y'all? 

Intro to Web Development

Agenda 📋

1. Fundamental concepts 💡
    - Server-Client architecture
    - Request-Response communication scheme
2. What *is* web development? 🤔
    - "Frontend" v.s. "Backend" web development
3. Get hands-on experience 🔨
    - Run a simple web app and tinker with it!

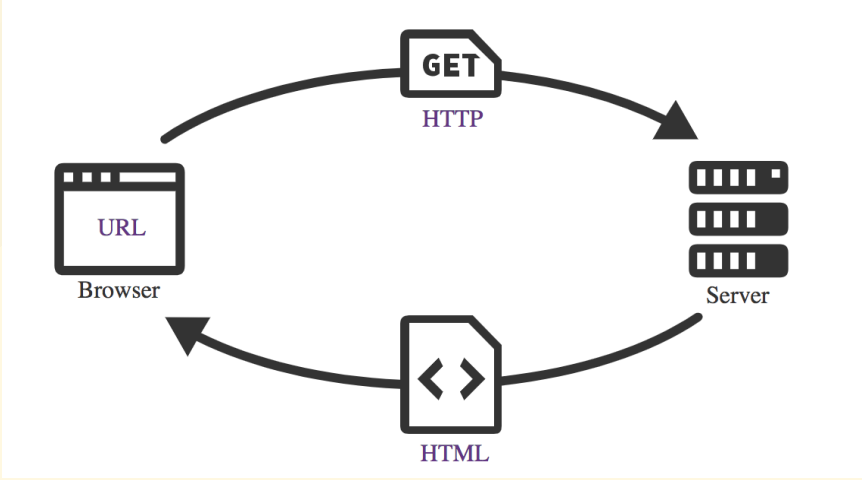
Prerequisites 💻

- A laptop with an internet connection
- A browser
- A text editor (e.g. VS Code, Atom)
- A terminal app (e.g. iTerm, Command Prompt)
- `node` installed on your laptop

In the beginning...

...there was HTTP ✉️

- A communication protocol
- A *uniform* way for computer A to get data from computer B via "the network"
- A **client** makes a **request** to a **server** for some data, the server **responds** with outcome status



Getting Data with HTTP ☎️

- Data is retrieved through an **HTTP GET request**
- With your browser, making HTTP requests is easy
  - Just type "wikipedia.com" + ENTER in URL bar
  - An **HTTP request** is sent to wikipedia.com
  - An **HTTP response** containing the wikipedia page is returned by the Wikipedia server

In this case, who's the client and who's the server?

Websites as Documents

- Notice: browser prepends `http://` to URLs
- Try: open an HTML file that lives on your computer, it will also open in the browser
  - But will the URL path start with `http://`?
- Many modern browsers have Developer Tools that let you inspect the underlying HTML page

Browsers? Noice. 👌

- Your browser provides lots of functionality:
  - Displays documents (`.html`, `.pdf`, `.txt`, etc.)
  - Makes HTTP requests on your behalf
  - Receives HTTP responses and presents the data

Among other things, your browser is an HTTP client and a document viewer.

Let's Get Crackin'!

What *are* servers? 🤔

- Really, just a program running on a computer connected to the internet
- The computer's OS handles incoming network messages and passes them on to the target app
- In our app, `express` provides a framework for receiving requests & sending responses

There's a lot going on under the hood! 💪

Frontend: the Client Side

- Responsible for user interface of a web app
  - Not just presentation, also **functionality**
- Inevitably involves JavaScript, HTML, & CSS

Backend: the Server Side

- Responsible for exposing functionality to clients through a well-defined API
- Typically involves querying databases and making requests to other services

Web Pages as Executables

- Web pages include JavaScript code, in addition to text, links, images, etc.
- The JavaScript code runs on the browser, which has a built-in JavaScript interpreter
- As mentioned earlier, HTTP requests are made on your behalf by the browser when you enter a URL or click on a link
- However, **the client JS code** makes many requests too (which run on the browser's interpreter)

Sending Data with HTTP ☎️

- Clients don't just get data, they also **send** it!
- Data is (typically) sent via **HTTP POST** requests
- In order to update a Wikipedia page, you must send the updates to the Wikipedia server
  - The updates are sent as the **body** of an HTTP request
  - The server responds with info about the success or failure of the update

Browsers. Real Noice. 👌

- Your browser provides lots of functionality:
  - Document display (`.html`, `.pdf`, `.txt`, etc.)
  - Make HTTP requests on your behalf
  - Receive HTTP responses and present the data
  - **Executes JavaScript code**

Your browser is an HTTP client, a document viewer, and a **runtime environment**.

Related Terms & Concepts 💡

Finding the Server

- Humans know URLs; routers know IP Addresses
  - **Universal Resource Locator** (URL)
- How to determine an IP address for a given URL?
  - **Domain Name Service** (DNS)

What's Under the Hood?

- How are HTTP requests sent?
  - It uses **Transmission Control Protocol** (TCP), which provides a *bidirectional* connection
  - **Port numbers** are *bound* to create a connection
- How does TCP create a bidirectional connection?
  - It uses **Internet Protocol** (IP) to create two 1-way connections

Advanced Topics

- Hosting and scaling web applications
  - Nowadays, likely using a cloud service provider like AWS, Google Cloud, Microsoft Azure
- Design patterns for web apps and services
  - e.g. REST Architecture, Model-View-Controller

Acknowledgements

- I used some content from workshops by awesome Web Dev Club members:
  - Amy Hanvoravongchai
  - Brynn Hawker
  - John Verwolf
- I wrote the slides using Marp