

WEEK 6

9.27.21

Agenda

THIS WEEK (6)

Discussion / Share Work

Review

HTML, CSS, JS, Tools, Workflow

Client Side vs Server Side

Libraries & Frameworks

Web Apps

Assignment 4: Student Directory

NEXT WEEK (7)

Building Web Apps

node.js, express

Working with data

JSON, visualizing data

API's

Working with API's

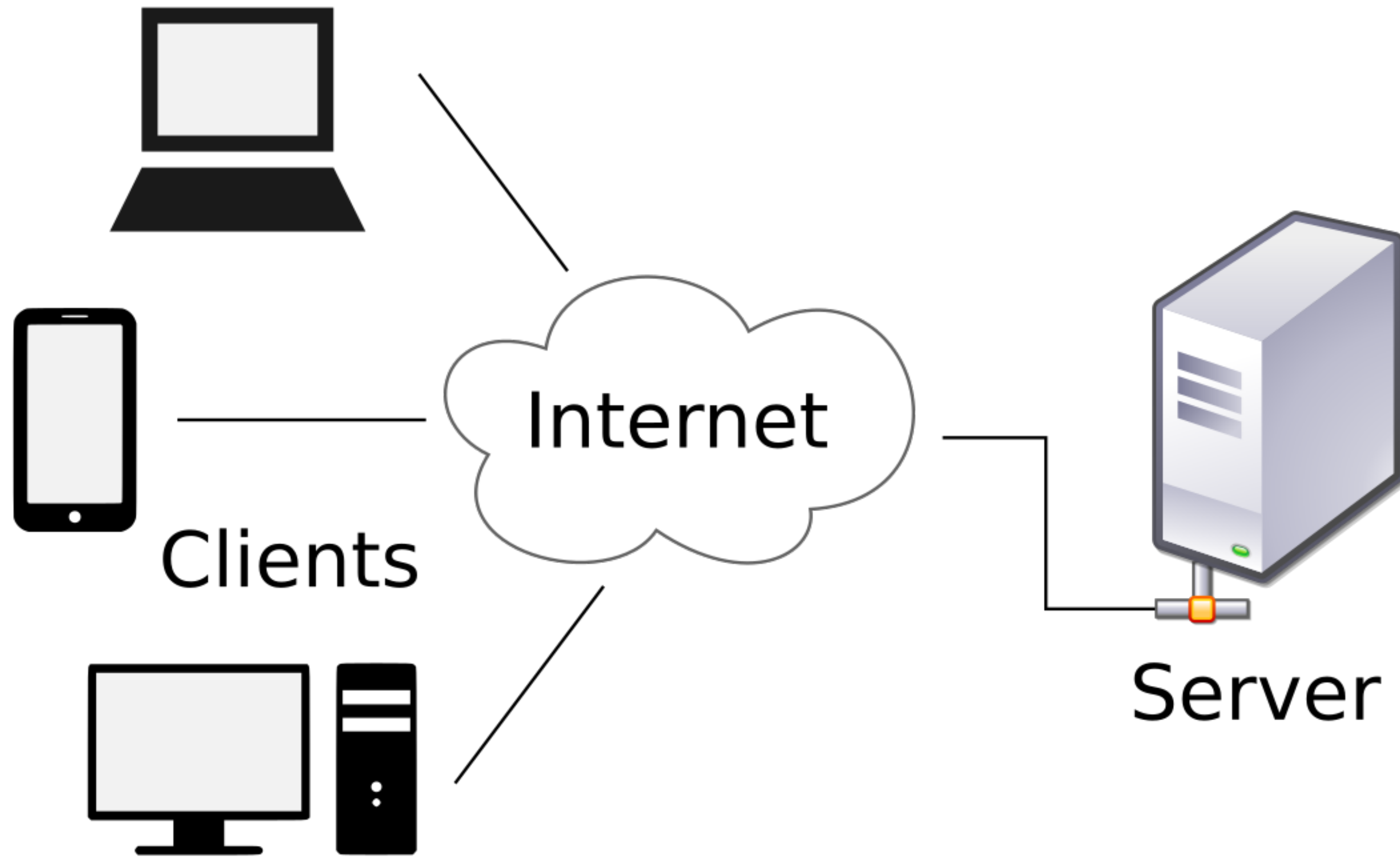
Building APIs?

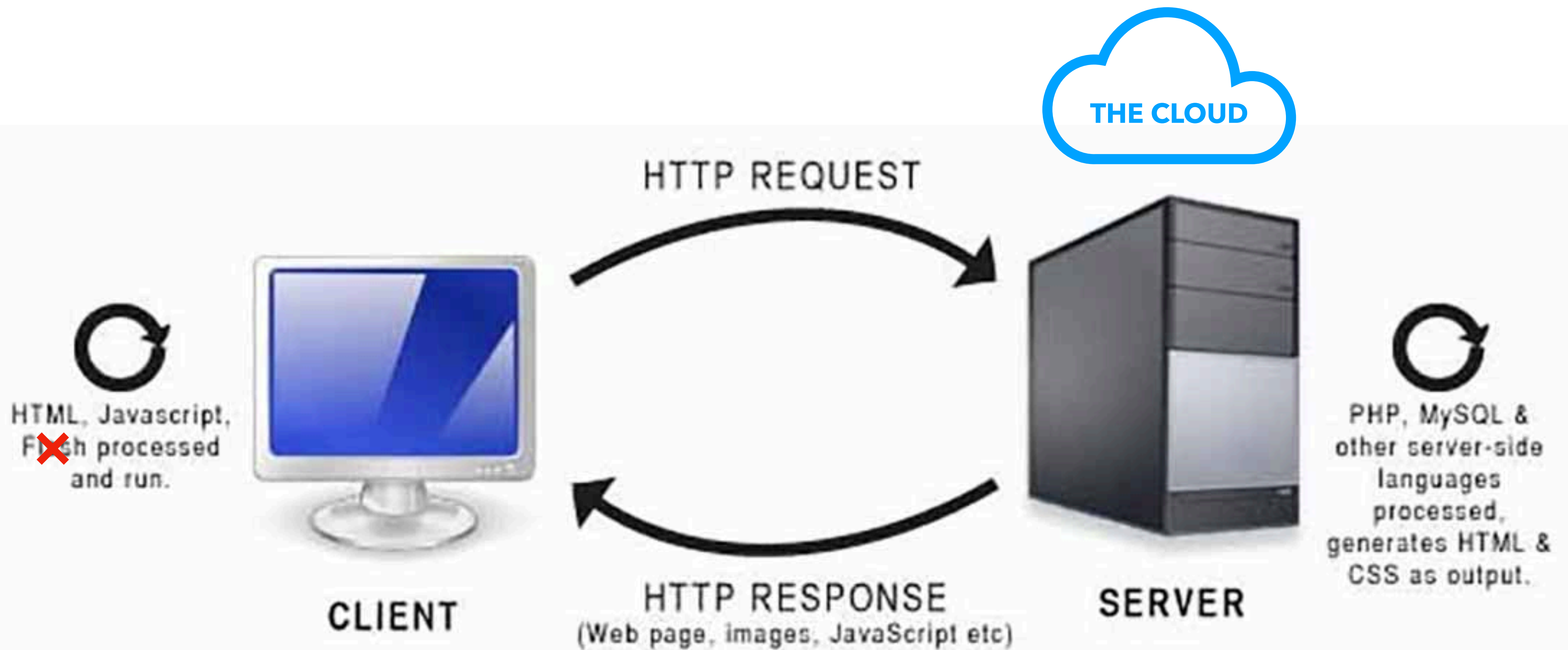
SHARE

Objectives

- Review the client / server relationship
- Understand what web apps are and how they work
- Understand what libraries and frameworks are in the context of web development
- Understand how to use a front-end library in a static website

CLIENTS
& SERVERS





The cloud refers to software and services that run on the Internet, instead of locally on your computer.

Client Side vs Server Side

CLIENT SIDE *(Front-End)*

Laptop, phone, tablet,
watch, tv, etc

Built to engage with user

Makes request to Server
for static assets

Built with HTML, CSS, JS

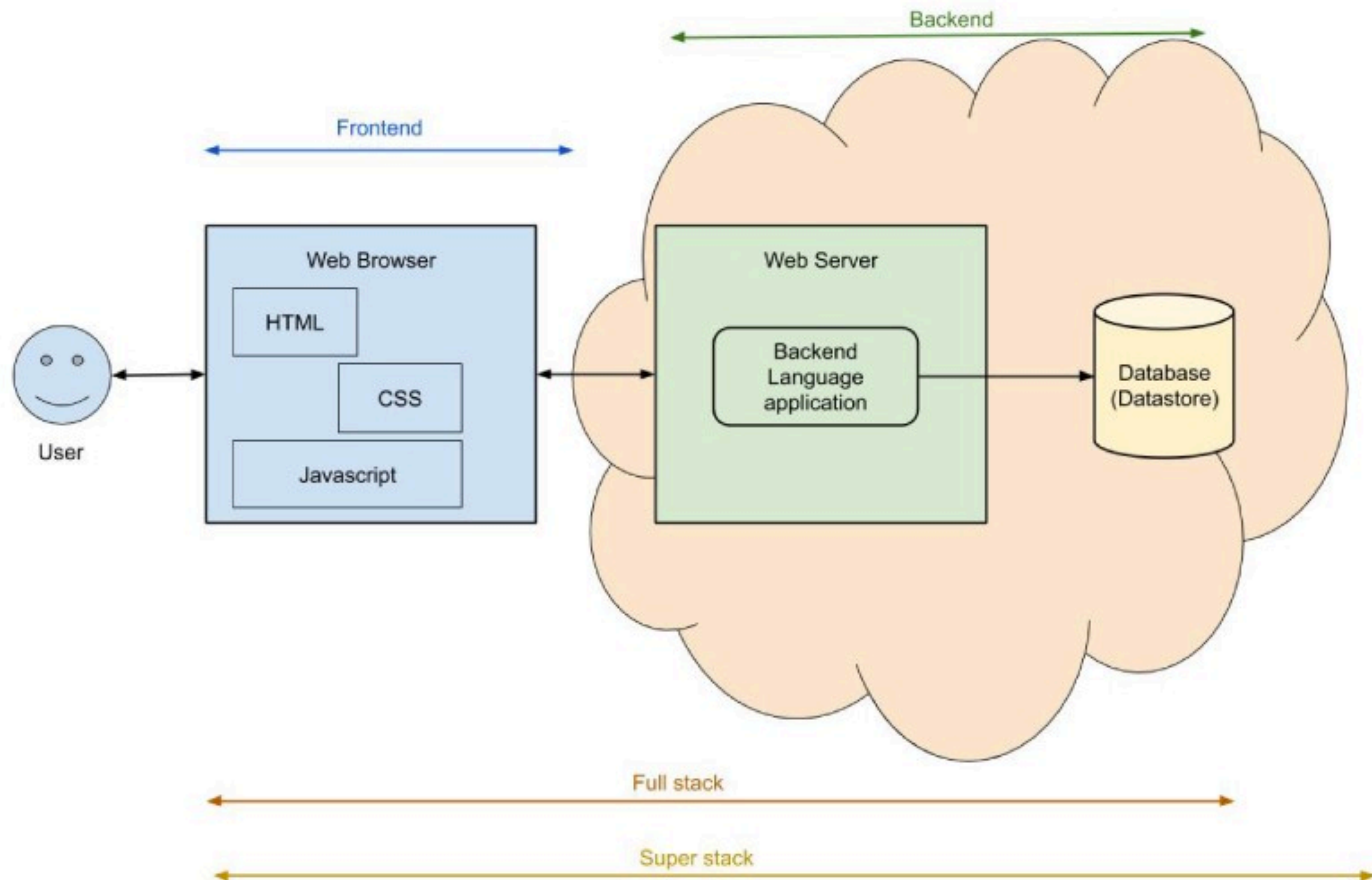
SERVER SIDE *(Back-End)*

Any computer configured to be
a server

Built to manage data and
handle requests from clients

Sends response to Client

Built with PHP, JS, .NET, ruby,
java, and many other languages



WEB APPS

What is a web app?

The term "web app" or "web application" is used to describe dynamic websites with similar functionality to a desktop or mobile app. These are some common types of web apps and websites.

- Static vs Dynamic web sites
- Single-Page Application (SPA)
- Progressive Web Application (PWA)
- Hybrid Applications

Static vs Dynamic web sites

- Static web page (or web site)
 - A static web page and resources will be sent to the users web browser from the server upon request. The resource is loaded from the server file system based on the URL and sent to the client.
 - For example, <https://rjduran.github.io/hello-world/>.
- Dynamic web page (or web site)
 - A dynamic web page shows different content to different users based on their account, choices, or selection. The page is assembled on the server from the data provided in the URL and send to the client.
 - For example, when you log in to Amazon to purchase products, the website will recommend products based on browse and purchase history.

Single-Page App (SPA)

- Application-like because they reject the typical web paradigm of moving between separate pages with different URLs.
- Use frameworks that include all libraries needed for desktop and mobile, speeding development, enhancing the user experience, and reducing load times.
- Examples: Gmail, Facebook, Slack, Instagram, Twitter, Google Drive

Progressive Web App (PWA)

- Applications that load like regular websites but offer user functionality such as working offline, push notifications, and device hardware access traditionally available only to native mobile apps. Sometimes referred as a “hybrid of website and mobile application”.
- Examples: Uber, Starbucks, Spotify PWA, Telegram, Wikipedia

Hybrid Applications

- Hybrid apps embed a website inside a native app, possibly using a hybrid framework. This allows development using web technologies (HTML, CSS, JS) while also retaining some advantages of native apps (ie. hardware access, offline operation)
- Examples: Apps like Slack, built with Electron, React Native

Resources

The following resources are great for taking the next step into web apps, web development, languages, libraries, and frameworks.

- [Web Development in 2021 - A Practical Guide by Traversy Media](#)
This video is very comprehensive. Highly recommend. ~1 hr long
- [Basic concepts of web apps, how they work and the HTTP protocol](#)
This video describes how clients and servers work.
- [Websites vs Web Apps](#)
This video explains the difference between static and dynamic webpages and web apps.
- [Web Application Architecture](#)
Summary of the components found in various kinds of web apps.

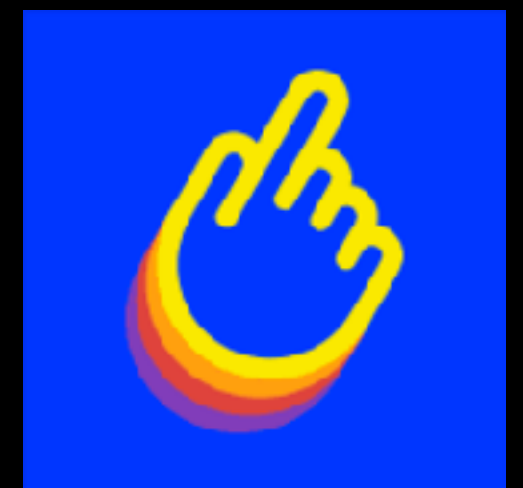
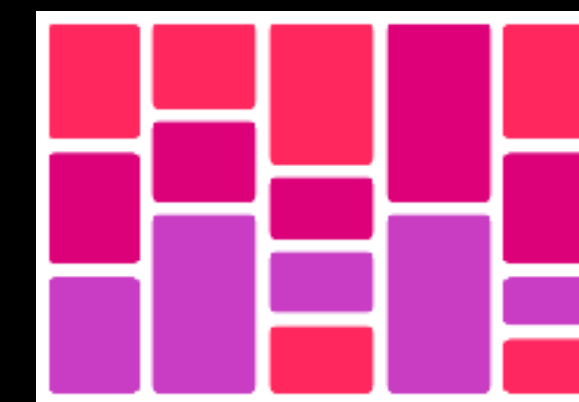
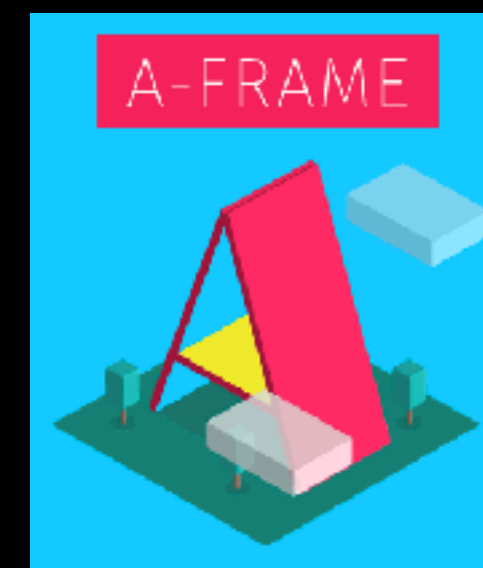
Libraries

A library is a collection of code containing objects, functions, and methods that you can use in an application or project. It can be written in CSS, Javascript, or some combination of both.



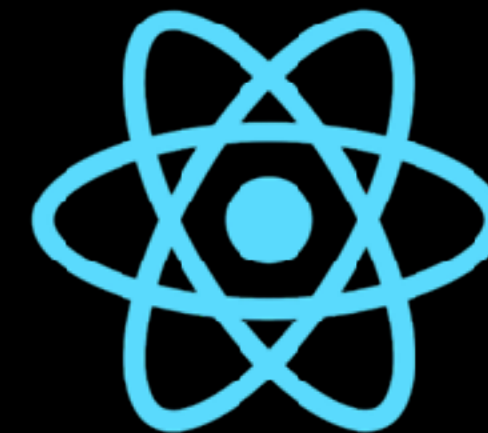
Animate.css

p5.js



Frameworks

A framework contains front-end and/or back-end libraries that you need to build a functional web app. You can think about a framework as something that is *mostly* ready to use out of the box.



Evaluating a library or framework

There are several questions you need ask yourself when choosing to use a library or framework. You typically start by looking at the documentation to evaluate features and functionality. Many times from the design perspective.

- What does it do? Does it solve the problem I have? Does it mostly meet all or some of the design goals?
- What's the benefit of using it? Does it solve a common problem? Does it make design/development easier?
- How do you use it? Does it have good examples to learn from?
- How large is the community using it? This means there will be lots of ways to get help.

Resources

The following resources are great for diving deeper into the many toolsets used by developers.

- Front-End Case Studies

A great collection of case studies about development tools and practices across the industry.

- State of JS | 2020 | 2019

A great resource showing the state of various JS based frameworks, languages, and toolsets.

- The 2019 Web Developer Roadmap (Github Repo)

Has some great flowcharts showing how different technologies relate. Look at the maps in github repo to see the big picture.

BREAK

FRONT-END DEV
WORKSHOP

How to use a front-end library

Let's start by looking at how to use Bootstrap in a basic static web page.

1. Open up VSCode and create a new project folder with index.html file inside. Use HTML template from documentation if available.
 2. Include required stylesheets and scripts. Look for CDN links.
 3. Fire up a local development server to run or use a sandbox.
 4. Refer to documentation for example code snippets.
- Tip for designers: Utilize component libraries for your visual design workflow in Sketch, Figma, Invision, Adobe Xd, etc to help you create high quality wireframes.

ASSIGNMENT 4:
STUDENT DIRECTORY

QUESTIONS?