Client-Side programming	• User inputs
Server-Side programming	• Initial HTML + JS + CSS fetch • HTML/CSS/JS + data

Web 1.0	• Static web pages
	  ○ Because status pages cannot change the data, it is basically read-only
	  ○ Example: YouTube like/subscribe buttons are not functional
	• Static web pages hosting services

Web 2.0	• Everything is centralized • Centralized app hosting platforms • Dynamic read/write web apps ○ Example: Facebook, Amazon, Blackboard

Web 3.0	• Peer-to-Peer (decentralized providers)
	• User account is not stored in a single server (Yahoo, Amazon, etcetera), images are not in the same location – everything is distributed
	• How does it work? How does the content get distributed across the Internet?
	  ○ Get code, data, and images (etcetera) from different servers
	  ○ Bitcoin – decentralized; makes multiple copies of transactions across multiple computers
	• Majority vote used for verification process
	• Verified by blockchain technology

Web 2.0 Apps -> Web 3.0 DApps (distributed apps)	

HTML & CSS

History of HTML	• 1989 Tim Berners-Lee, CERN Lab (Switzerland) ○ How to share research papers
The ultimate first website: http://info.cern.ch/ (.ch = TLD for Switzerland)	
Once code is "uglified," everything is compressed. This is important because the amount of traffic is reduced when transferring from the server to the browser.	
HTML	• Hypertext markup Language • Not a programming language, but a *presentation* language • Code for **structuring** and **displaying** a web document

VSCode

- Create a new folder; not a file
- Control + Shift + P OR F1
- liveServer