What is Web Development?

Web Dev, Spring 2021

Web Dev in a sentence

Software development using the web as a deployment and execution platform

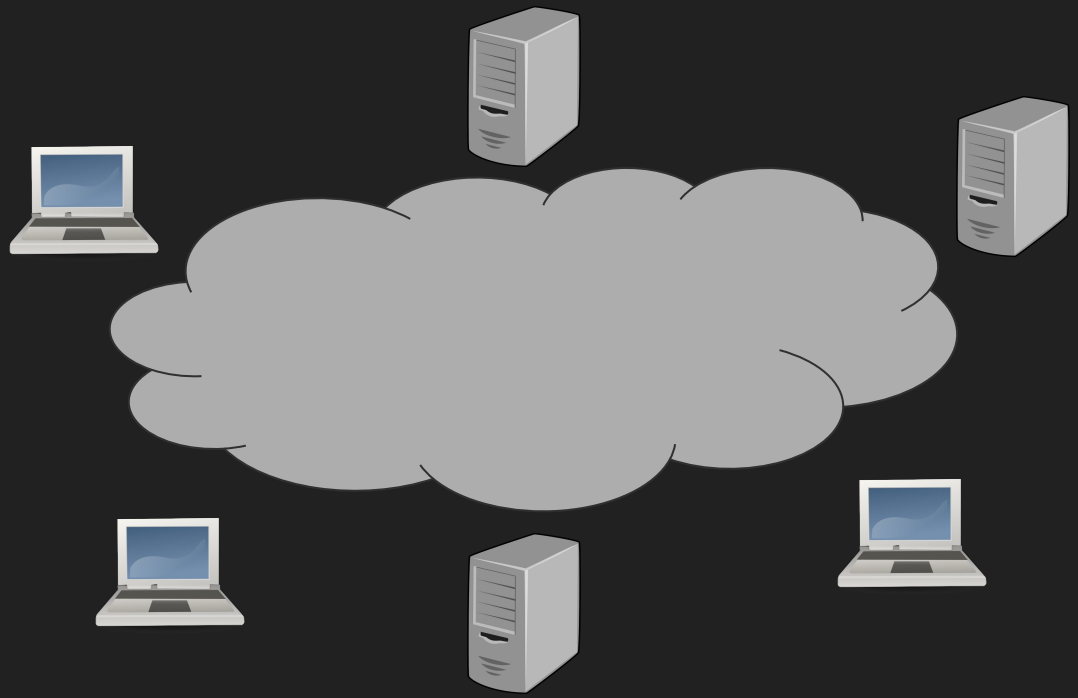
- Key: you use a web browser to interact with the created artifact

Web dev doesn't dictate the what, it dictates the HOW

To understand the HOW, we need some history

- Helps explain why web dev is the way it is today

The Internet

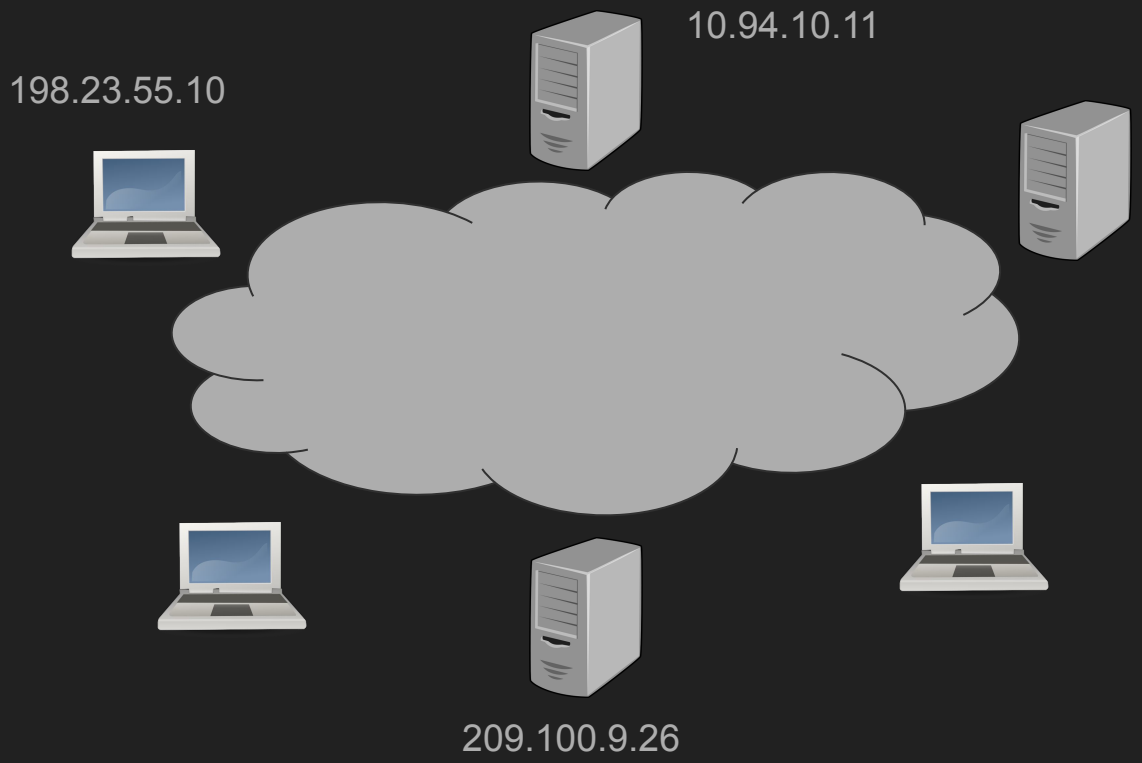


Set of computers connected via a network

Network lets computers send messages

Each computer has an address so that you can direct messages to it

The Internet

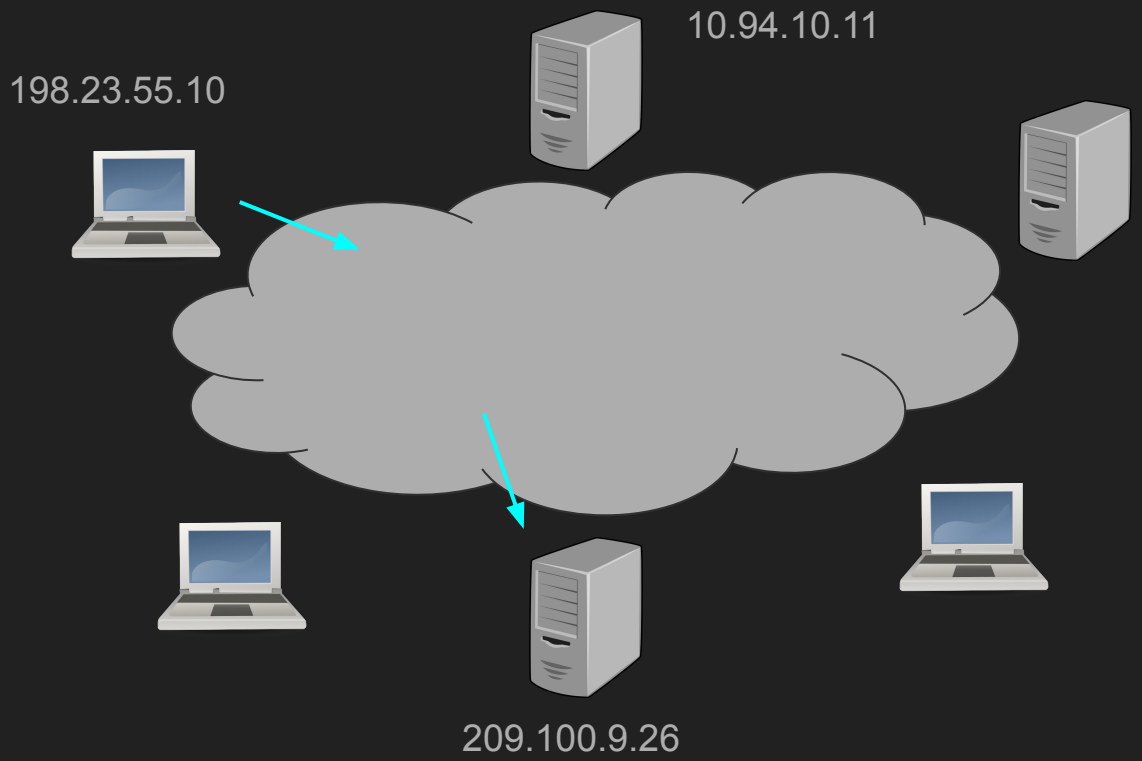


Set of computers connected via a network

Network lets computers send messages

Each computer has an address so that you can direct messages to it

The Internet

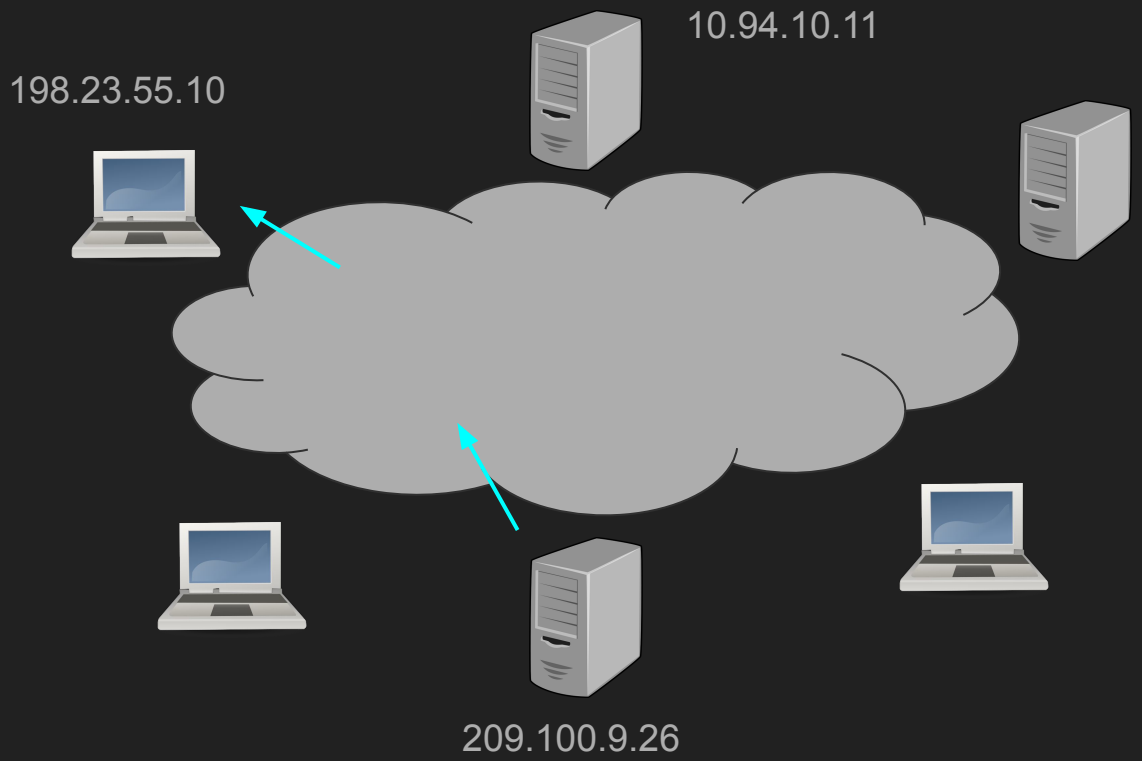


Set of computers connected via a network

Network lets computers send messages

Each computer has an address so that you can direct messages to it

The Internet

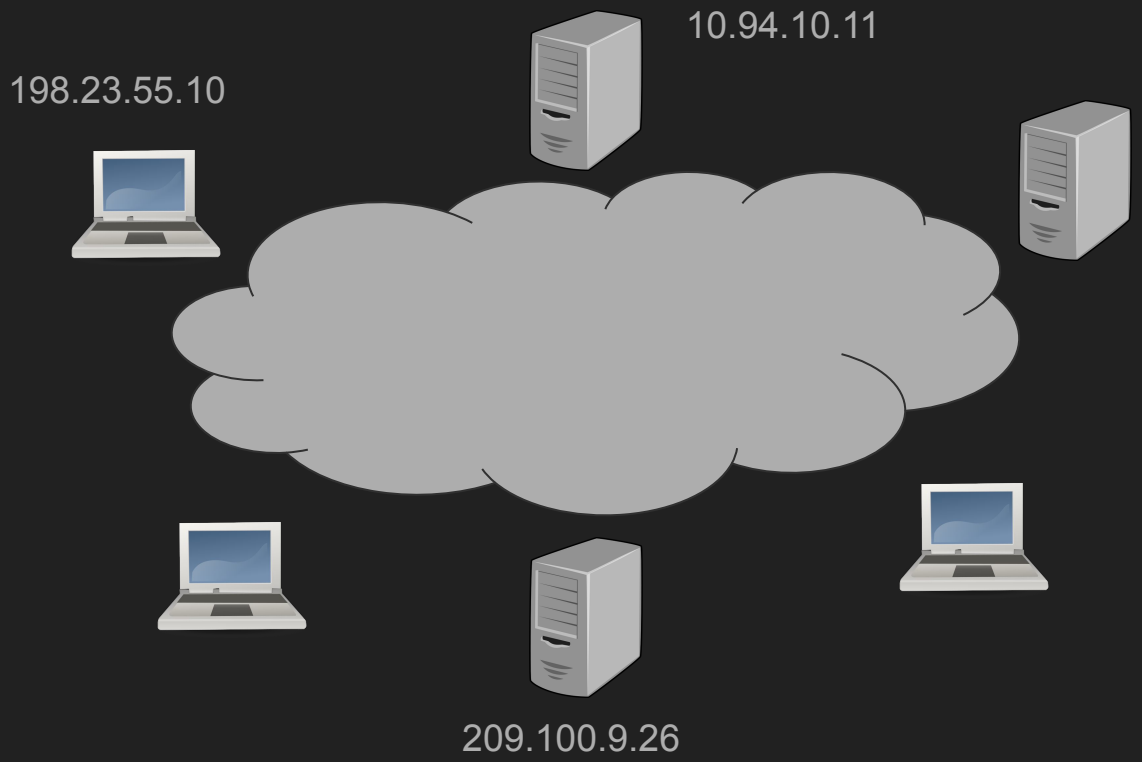


Set of computers connected via a network

Network lets computers send messages

Each computer has an address so that you can direct messages to it

The Internet



Many different kind of messages can be exchanged by various applications

Each defined by a protocol (message format, etc)

- ping
- FTP
- SSH
- VNC
- Web (= HTTP)

The World Wide Web

A grandiose name — the initial idea (1990s):

- A computer could make a set of files containing text + pictures available
- Those files could contain links to other files possibly on other computers
    - Hypertext - think Wikipedia

A *web server* makes those files available to everybody

A user would use a special application (*web browser*) to access those files using the address of the computer and the location of the file on that computer

- browser understand links and when you click on one, you could fetch the linked file and display it
- web browsers request files from web servers using HTTP

The World Wide Web

A grandiose name — the initial idea (1990s):

- A computer could make a set of files containing text + pictures available
- Those files could contain links to other files possibly on other computers
  - Hypertext - think Wikipedia

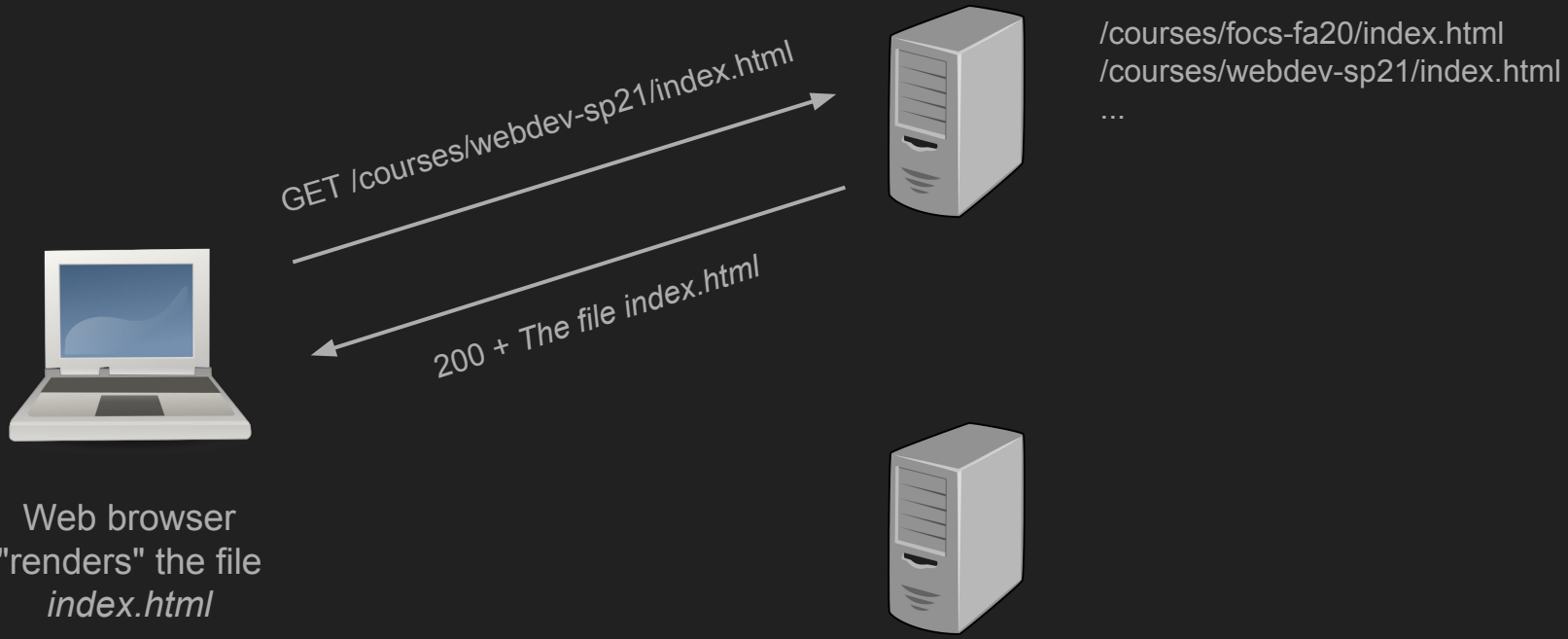
A *web server* makes those files available to everybody

A user would use a special application (*web browser*) to access those files using the address of the computer and the location of the file on

- browser understand links and when you click on one, fetch the linked file and display it
- web browsers request files from web servers using HTTP

HyperText Transfer Protocol

The World Wide Web



Hypertext

The web was initially designed to exchange hypertext

Web browsers are applications that can display hypertext files smartly

- format of hypertext = HTML
- basically, structured text
- got a lot more complicated since initial v1.0
- structured text broken down into a tree of elements
- browser "renders" the tree using a fairly complicated algorithm that only a few people at Mozilla and Google genuinely understand

Hypertext

The web wa

Web brows

- format
- basical
- got a lo
- structu
- browse
  people

```
<html>
  <body>
    <p> This is a paragraph with a <b>bold</b> word </p>
    <ul>
      <li> Item 1 </li>
      <li> Item 2 </li>
    </ul>
    <p> This is a <a href="http://google.com">link</a> </p>
  </body>
</html>
```

Great — then what?

That was good enough for a short while

- web browsers get and render HTML files from web servers = everybody happy

People wanted more "dynamic" behavior — that's where things get fun

Two distinct ways to add more dynamic behavior:

(1) make the HTML file sent to the browser more dynamic
(2) make the generation of the HTML page on the server more dynamic

Dynamicity on the browser side

A web browser gets an HTML file

- transforms it into a tree
- renders the tree

Idea: create a scripting language for the browser

- add code to the web page to, e.g., react to events (clicks, hovers, keypresses) and do something to the tree in response

That scripting language evolved into the JavaScript of today

- An HTML file creates a tree that becomes an "active" artifact
- A web browser is an HTML renderer + a JavaScript interpreter

Dynamicity on the browser side

A web browser gets an HTML fi

- transforms it into a tree
- renders the tree

Idea: create a scripting languag

- add code to the web page keypresses) and do somet

That scripting language evolve

- An HTML file creates a tre
- A web browser is an HTML

Extreme picture:

Send an "empty" HTML file, with code to create the tree programmatically on the browser!

Dynamicity on the server side

Initial picture:

- Web server delivers existing HTML files from the file system
- Good for static information, not so much when information changes
- E.g., student records
  - one HTML file per student?
  - make a change manually or pragmatically to a file when something changes? Urgh…

Better solution:

- Store student records in a database
- Generate the HTML file in response to a browser request

Dynamicity on the server side

Classical Web Servers (Apache, NGINX)

- use an external program to generate HTML files dynamically (CGI)
  - this includes PHP

Web Application Servers (templates, or generate in code):

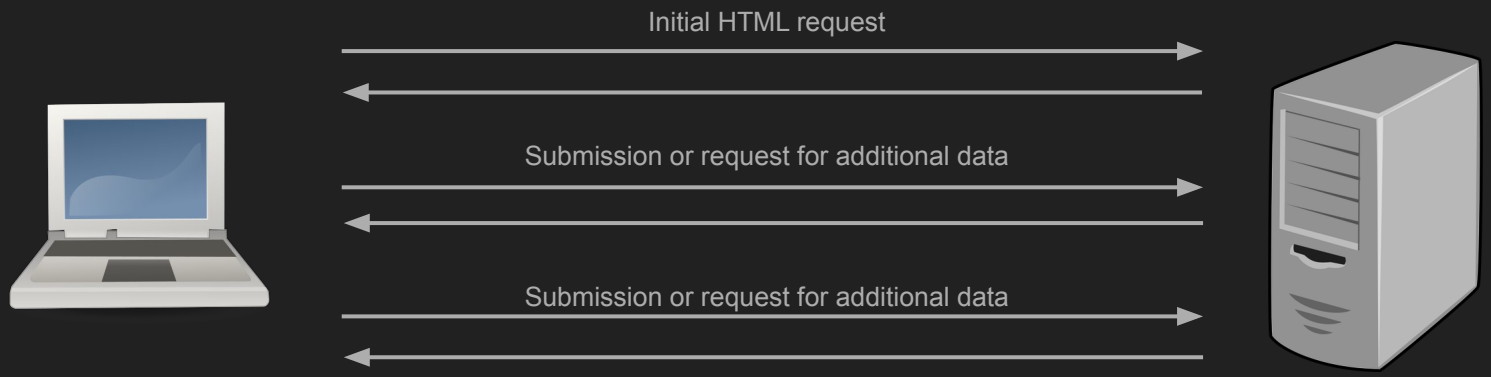
- Ruby / Rails
- Python / Django, Python / flask
- Java / Spring
- .NET / ASP
- JavaScript / Express

Dynamicity über Alles

Web Application servers let you send more general data than just HTML files

Javascript can make calls to a Web Application server after an HTML file has rendered

⇒ Modern web app (including SPAs)



Learning path

We're going to look at all the models above, and gets our hands dirty with pretty much all of them

- HTML / CSS
- JavaScript
- Frameworks for coding in JavaScript more reasonably (React, ...)
- Web application servers (Python / Flask, ...)
- Data persistence (databases)

Course work

Regular homeworks

- mostly getting our hands dirty with what we've seen
- idea for an ongoing web app that we'll be tinkering with as the course progresses

Final project (2nd half of semester)

- implement something you find interesting

Teams of 2 (or 3 if in a bind — I will expect more/better from teams of 3)

Your first homework

Send me an individual email with:

- your name
- your background in webdev:
  - do you know HTML/CSS?
  - any javascript experience?
  - any server-side experience (and if so, what frameworks)?
  - example of something you've done (if applicable)

There are no wrong answers — I just want to know roughly where everybody is