



Introduction to the Web

INFO 6150
Fernando Augusto López Plascencia



What to expect from this course

- Being able to create a basic website or application, using current technologies
- Knowing what to focus on to create a great user experience
- Knowing what else to explore, and how



Requirements

Access to a computer and the internet

An IDE for Javascript. I recommend Visual Studio Code.

That's it!



In this lesson:

- Basic concepts: The internet, the web, and how that all works (roughly)
- Intro to web technologies: HTML, CSS, Javascript
- Frontend and backend; intro to APIs



What is the web?

- The web is not the Internet
 - But, it is a lot of things...
- If you're using your browser, you're in the web
- Today, the line between “app” and “website” is blurry



The Internet, conceptually

- Just a bunch of computers connected to each other, really.
- They talk the same languages: we call them **protocols**
- Also: they're always connected: the Internet has **resiliency**



A bit of history (I promise this makes sense)

- It all started with the military... and for good reason
- Bay Area was there almost from the start
 - Almost. UCLA was first!
- TCP/IP made it all better

Here comes the Web!

- Sir Tim Berners-Lee wanted hyperlinked documents
- ...and created [the first website](#)
- And the rest is history.

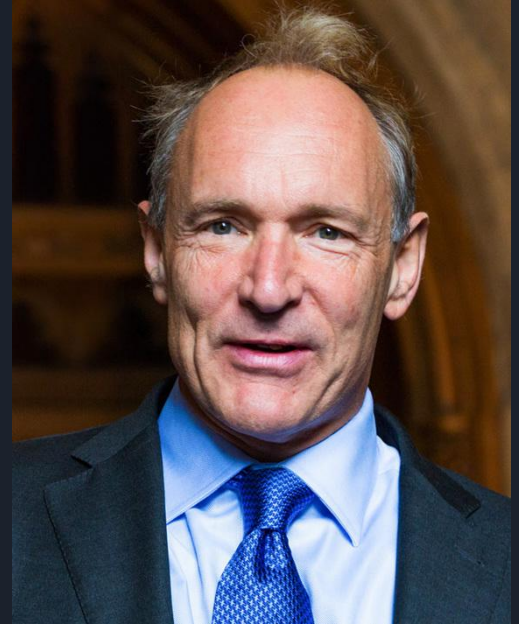


Photo from Wikipedia. CC BY-SA 4.0

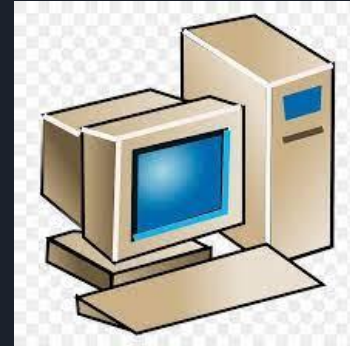


But what is the Web, then?

- What Berners-Lee created was:
 - **A language:** HTML
 - **A protocol:** HTTP
 - **A client program:** the first browser
 - **A server program:** the first web server

The client/server model

- Server: has data; can operate on it
- Client: wants data, asks the server for it

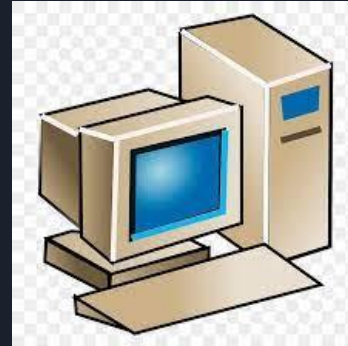


The client/server model

- Server: has data; can operate on it
- Client: wants data, asks the server for it



GET /index.html



www.wikipedia.org

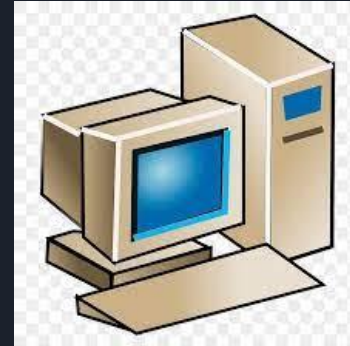
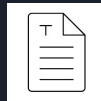
The client/server model

- Server: has data; can operate on it
- Client: wants data, asks the server for it



GET /index.html

200 OK



www.wikipedia.org



The HTTP protocol

- Allows servers and clients to send, receive, and otherwise work with HTML and other files
- `GET /index.html` is an HTTP “request”; `200 OK` is a “response”



Snoop your browser's HTTP calls!

You will need Google Chrome for this.

- Open a new, blank window or tab.
- Click on the three dots at the top right of the window to open the menu.
- Select “More tools”; in the submenu select “Developer Tools”. A panel will open at the bottom of your window.
- In that panel, look for the “Network” tab and click on it. You may need to make the panel bigger to find it.
- Now, with the Network panel open, go to www.wikipedia.org. Observe how the Network tab of the Developer Tools fills with rows of data.

WIKIPEDIA

The Free Encyclopedia

English

6 715 000+ articles

Español

1 892 000+ artículos

日本語

1 387 000+ 記事

Русский

1 938 000+ статей

Deutsch

2 836 000+ Artikel

Français

2 553 000+ articles

Italiano

1 826 000+ voci

中文

1 377 000+ 条目 / 條目

Português

1 109 000+ artigos

العربية

مقالة 1 217 000+

 EN

Read Wikipedia in your language

We'll be quick:

If Wikipedia has given you \$3 worth of knowledge this year, take a minute to donate.

[Donate now](#)

Elements Console Recorder **Network** 1 1

☐ Preserve log ☐ Disable cache ☐ No throttling ☐ Invert ☐ Hide data URLs ☐ Hide extension URLs

All Fetch/XHR JS CSS Img Media Font Doc WS Wasm Manifest Other

☐ Blocked response cookies ☐ Blocked requests ☐ 3rd-party requests

50 ms 100 ms 150 ms 200 ms 250 ms 300 ms 350 ms 400 ms

Name	Status	Type	Initiator	Size	Time	Waterfall
www.wikipedia.org	304	docu...	Other	781 B	16 ms	
Wikipedia-logo-v2.png	200	png	(index):24	(me...	0 ms	
index-6852360a56.js	200	script	(index):854	(me...	0 ms	
gt-ie9-ce3fe8e88d.js	200	script	(index):856	(me...	0 ms	
sprite-8bb90067.svg	200	svg +...	(index):878	(me...	0 ms	
Wikinews-logo_sister...	200	png	(index):878	(me...	0 ms	
inter_Inter-Bold.woff	200	font	www.wikip...	143 kB	37 ms	
inter_Inter-Regular.woff	200	font	www.wikip...	135 kB	37 ms	
content.css	200	fetch	content.js:1	175 kB	21 ms	
content.js	200	script	import-co...	119 kB	8 ms	
sessions.bugsnap.com	(bloc...	xhr	content.js:1	0 B	7 ms	
wikipedia.ico	200	vnd...	Other	(disk...	15 ms	
actions-639cd34d.js	200	script	content.js:1	102 kB	18 ms	
helpers-10be1fb3.js	200	script	content.js:2	4.1 kB	29 ms	
globalStyles-22fbf6ab...	200	script	content.js:3	150 kB	23 ms	

15 requests | 829 kB transferred | 1.0 MB resources | Finish: 371 ms | DOMContentLoaded: 96 ms | Load:

Console What's New X Issues

Highlights from the Chrome 118 update



The languages of the web

- **HTML** (HyperText Markup Language) defines the **content**: that is, what is on the page.
- **CSS** (Cascading Style Sheets) defines the **look and feel**: how the content should be rendered.
- **JavaScript** provides **interactivity**: that is, any functionality beyond “writing on a page”.



HTML

The main unit is the **element**, represented by **tags**.

Tags can open and close, and contain other tags.

Tags can have attributes to specify details.

Tags (elements) give **meaning** and **structure** to the page.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8" />
    <title>A title</title>
  </head>
  <body>
    Some text
  </body>
</html>
```



Create your first page

- Copy the example text in Notepad (Windows), TextEdit (Mac), or any other simple text editor.
- Save the file in plain text mode as “index.html” on your desktop.
 - If using TextEdit, select “ Web Page (.html)” as the file format.
- Double-click on the file. It should open in your default browser.

You just created a web page!



Keep exploring HTML:

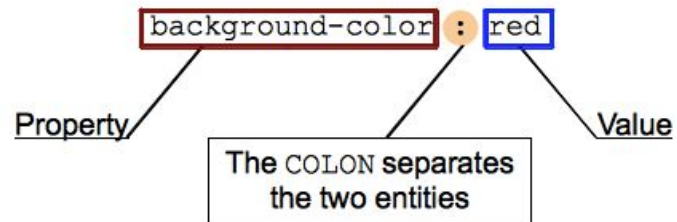
Explore other common tags here: <https://learnxinyminutes.com/docs/html/> and continue adding content to your page. Try any (or all) of the following:

- Write a self-introduction in a few paragraphs (use `<p>`).
- Write a list of hobbies (use `` or ``; use ``).
- Add an image downloaded from the web (use ``).
- Try marking some text with `` or `` and see what happens.
- Use headers (`<h1>`, `<h2>`...).

CSS

CSS stands for “Cascading Style Sheets”. It uses **declarations** to tell the browser what each HTML **element** should look like.

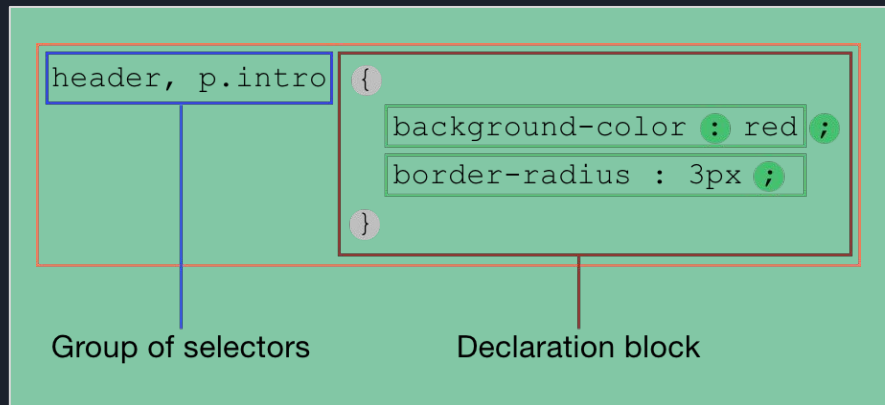
A CSS declaration :



CSS

Each declaration contains a **property** and a **value**.

Declarations go inside **blocks** - curly braces `{ }`. Before each block, we specify **selectors**, which refer to HTML elements.





CSS

In an HTML page, use the `<link>` tag inside `<head>` to link it to a CSS file:

```
<link rel="stylesheet" href="style.css" />
```



Add CSS to your page

Created by HideMaru
from Noun Project

- Create a new file called “style.css” in the same folder as your HTML file and add the following:

```
h1 {  
  color: blue;  
}
```

- Link the CSS file to the HTML file, using <link>.
- Make sure you have a <h1> tag in your HTML, such as:

```
<h1>  
  This should be blue
```

```
</h1>
```

- Save both files, then reload your webpage in your browser. Your header should show up in blue letters.



Javascript on the web

- A full fledged programming language
- Used to be much smaller
- Runs in your browser, but also everywhere else



Add Javascript to your page

Created by HideMaru
from Noun Project

- Add a `<script>` tag to the `<head>` of your page, as follows:

```
<!-- import the webpage's javascript file -->
```

```
<script src="script.js" defer></script>
```

- Create a new file called “script.js” in the same folder as the other two files.
- Add the following to your Javascript file:

```
console.log("Hello, world!");
```

- Reload the HTML page in your browser.
- Open the Developer Tools panel in your browser, then click on the “Console” tab. You should see the message “Hello, world!” there.



Front end and Back end

- Frontend: what happens in the **client** program (the browser)
 - What we just talked about
- Backend: what happens in **server** programs
 - Not just a web server. Many programs (services) collaborate for big applications
 - Can be written in many languages



Web APIs

API: how two programs communicate; or: how a programming interface is structured



Web APIs

- **Front-end (Client-side) Web APIs:** The standard libraries available within the browser for Javascript to manipulate a page, and the browser itself.
- **Back-end (Server-side) Web APIs:** The protocols for communication between a Javascript program running in a browser and a back-end program, running on a server.



Common Server-side APIs

- **REST** - uses HTML over HTTP; vanilla web. The most robust and popular. Standard
- **SOAP** - uses XML. Useful for very big, secure applications. Also standard
- **GraphQL** - a query language on top of HTTP; fast, but not standard-based
- **gRPC** - call functions remotely; binary data. Open source. Oh so complex