

*An Introduction to*  
Front End Web Development

ChickTech Virtual Workshop  
Summer 2020  
V. Lai



# What can we do online these days?

- Maybe a better question we should ask is:

What can't we do online these days?

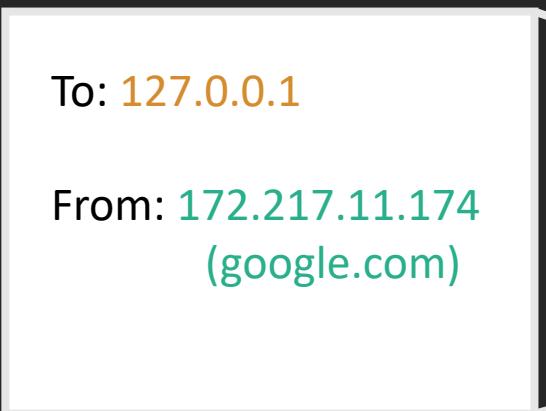
What is the internet?

# The 10,000 Foot View – The Internet

- The Internet is a connected set of independent networks (with physical wires) and everything connected to it has a unique IP address.
- The World Wide Web (the web) is the main tool used to interact on the Internet. The web is a collection of documents and resources identified by URLs (Uniform Resource Locators).

# The 10,000 Foot View – The Internet

- Data (documents & other resources) gets split up into packets of information which get routed through various network paths and then gets reassembled in the right order when all packages arrive.
- We're sending bits of information using electricity, light, and radio waves.



To: 127.0.0.1

From: 172.217.11.174  
(google.com)



Photo by Claudio Schwarz | @purzlbaum on Unsplash

# The 1,000 Foot View – Servers

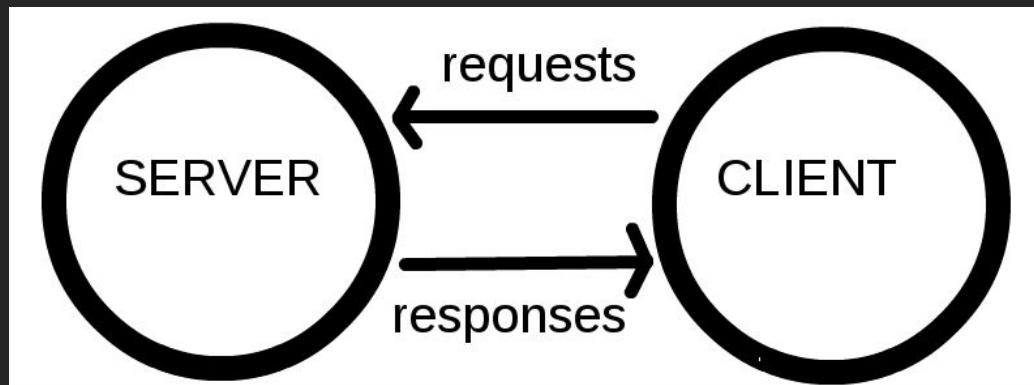
- Servers are computers connected directly to the Internet that hold files for web pages on their hard drives.
- The Domain Name System (DNS) associates names (ex: facebook.com or amazon.com) with its corresponding IP addresses. Your computer uses the DNS to get the correct IP address for the site you want to visit.



Photo from TechCrunch

# The 10 Foot View – Clients & Browsers

- Clients are computers that connect to an Internet Service Provider (ISP) which is connected to the Internet. With our browsers, we as clients make requests to servers with an IP address.



From MDN web docs



Photo by [NeONBRAND](#) on [Unsplash](#)

# Browsers

- Chrome, Edge, Firefox, Safari, and more!
- They are used to present content you want to see by requesting it from the server, interpreting the HTML, CSS, and JavaScript, and rendering it for you to see.
- We're going to focus on talking about what we see on our browsers
  - This is called client-side/front end, as opposed to server-side/back end which is more behind-the-scenes and data-centric). Think of the iceberg →



# Typical Parts of a Web Page

- These are assets/files stored on a server that typically get sent back to you when you request a page.
- HTML – HyperText Markup Language, the structure/“bones” of the page
- CSS – Cascading Style Sheet, the styling/“skin” of the page (if applicable)
- JS – JavaScript, the functionality/movement of the page (if applicable)
  - ex: clicking/hovering on a button triggers another behavior to show on the page

# HTML - bones

<!DOCTYPE html> specifying what kind of document this is (html)  
<html lang="en"> lang attribute specifies language of document

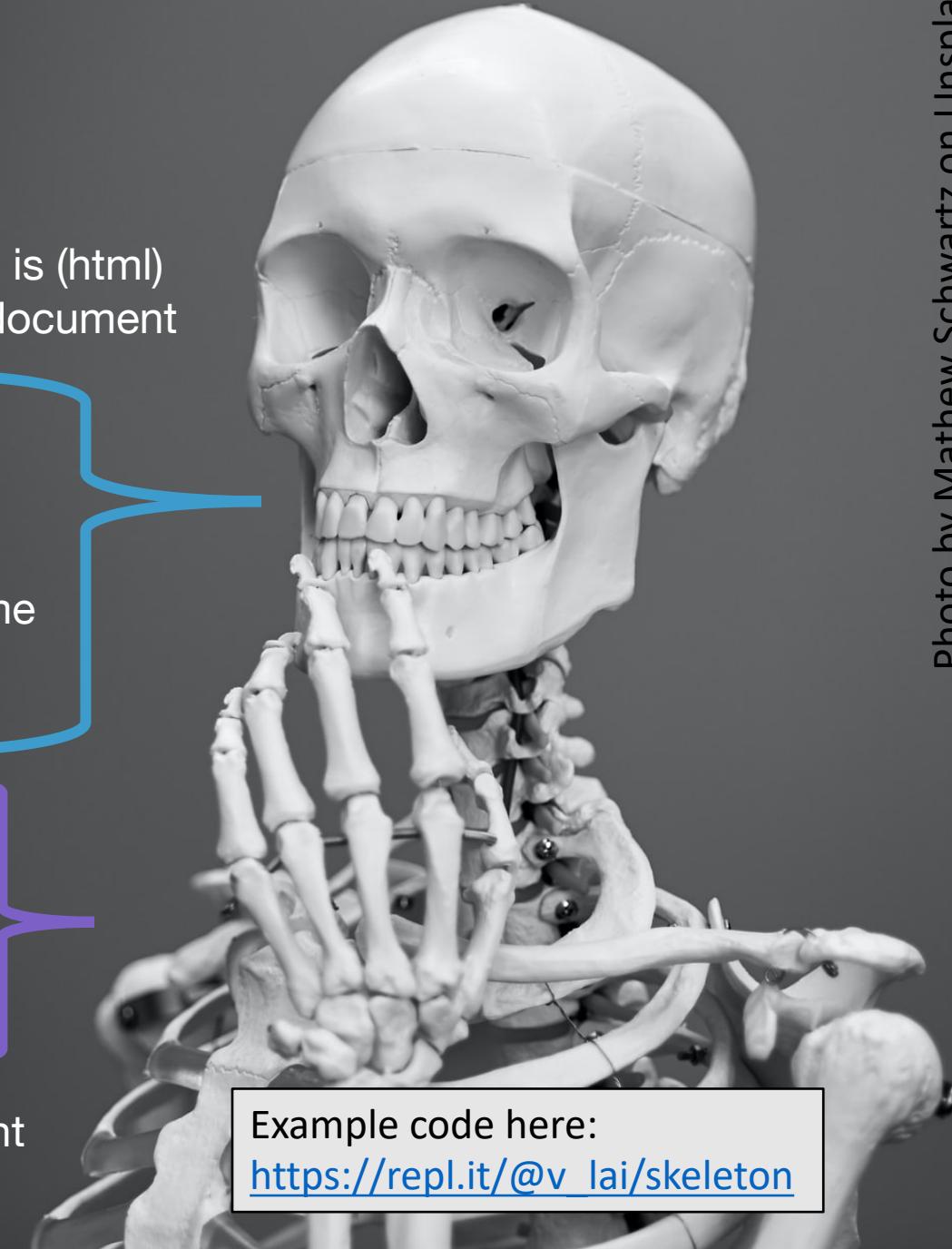
<head> this is the content mostly not shown to viewers

```
<meta charset="UTF-8"> meta data
<meta name="viewport" content="width=device-width,
initial-scale=1.0">
<title>HTML – bones</title> browser tab name
<link rel="stylesheet" href="style.css">
</head>
```

<body> this is the content shown to viewers

```
<h1 id="hello">Hello Skeleton!</h1>
<!-- more content here -->
<script src="script.js"></script>
</body>
```

</html> <!-- index.html --> example of a comment



Example code here:  
[https://repl.it/@v\\_lai/skeleton](https://repl.it/@v_lai/skeleton)

# HTML – bones are made of elements

Opening tag (this is a heading)

Attribute (extra info)

```
<h1 id="hello">
```

Hello Skeleton! Content

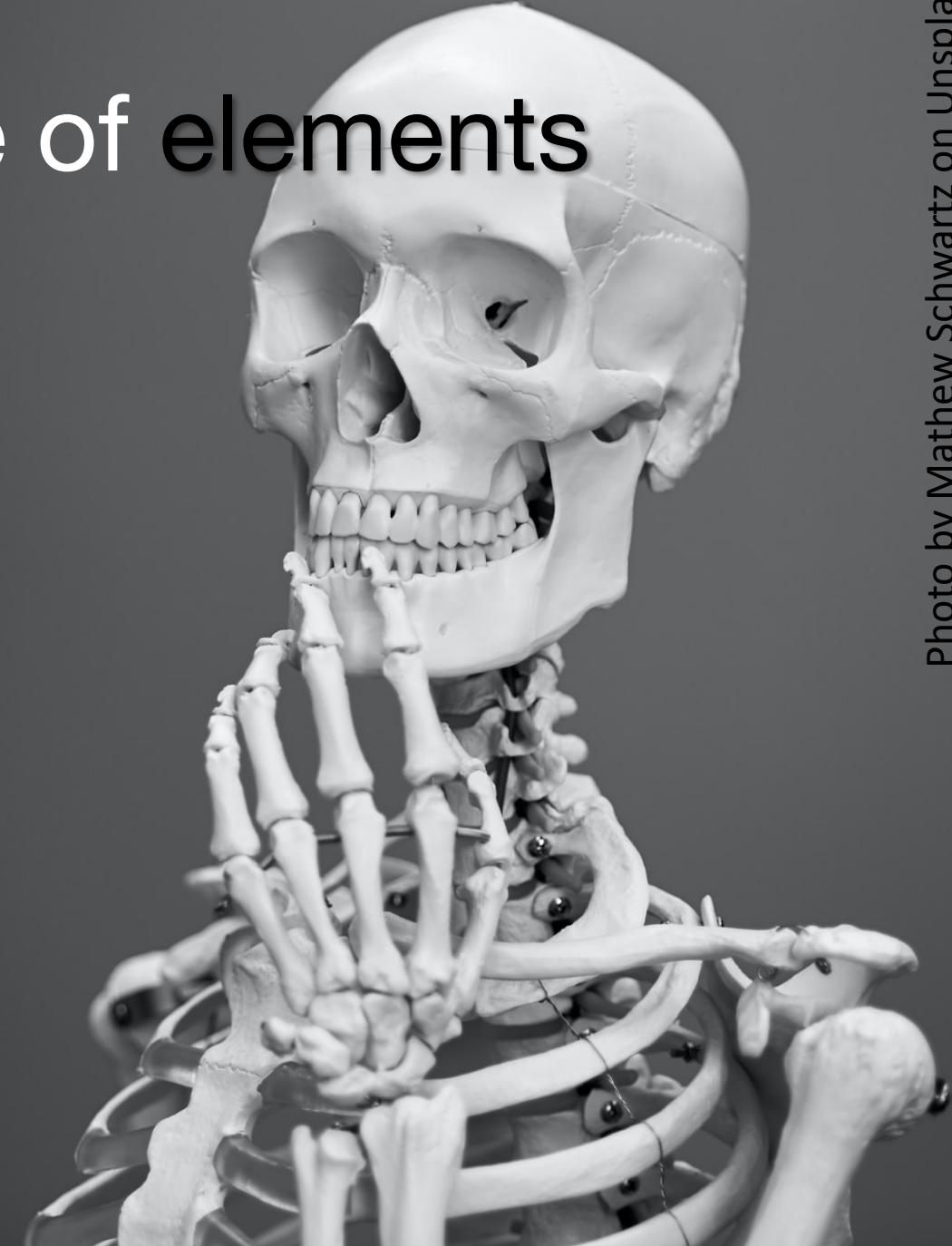
```
</h1>
```

Closing tag

Element

**Hello Skeleton**

seen in the browser



# CSS – skin

```
<h1 id="hello">  
  Hello Skeleton!  
</h1>
```

Previous HTML  
(index.html)

```
#hello { selected item to style  
  color: green;  
}
```

CSS  
(style.css)



**Hello Skeleton**

seen in the browser



# JavaScript - movement

```
(function () { // hey, run this right after the script finishes loading
  var helloElement = document.getElementById('hello');
  function dance() {
    helloElement.onmouseover = function () {
      helloElement.innerText = 'Dancing!';
      helloElement.style.color = 'red';
    };
    helloElement.onmouseleave = function () {
      helloElement.innerText = 'Hello Skeleton Again!';
      helloElement.style.color = 'green';
    };
  }
  function sing() {
    helloElement.onclick = function () {
      helloElement.innerText = 'Singing!';
      helloElement.style.color = 'blue';
    };
  }
  dance(); // dancing won't happen unless you call it!
  sing(); // singing won't happen unless you call it!
})(); // in script.js
```



Photo by [Ahmad Odeh](#) on [Unsplash](#)



Photo by [Josh Rocklage](#) on [Unsplash](#)

# JavaScript - movement

**Hello Skeleton**

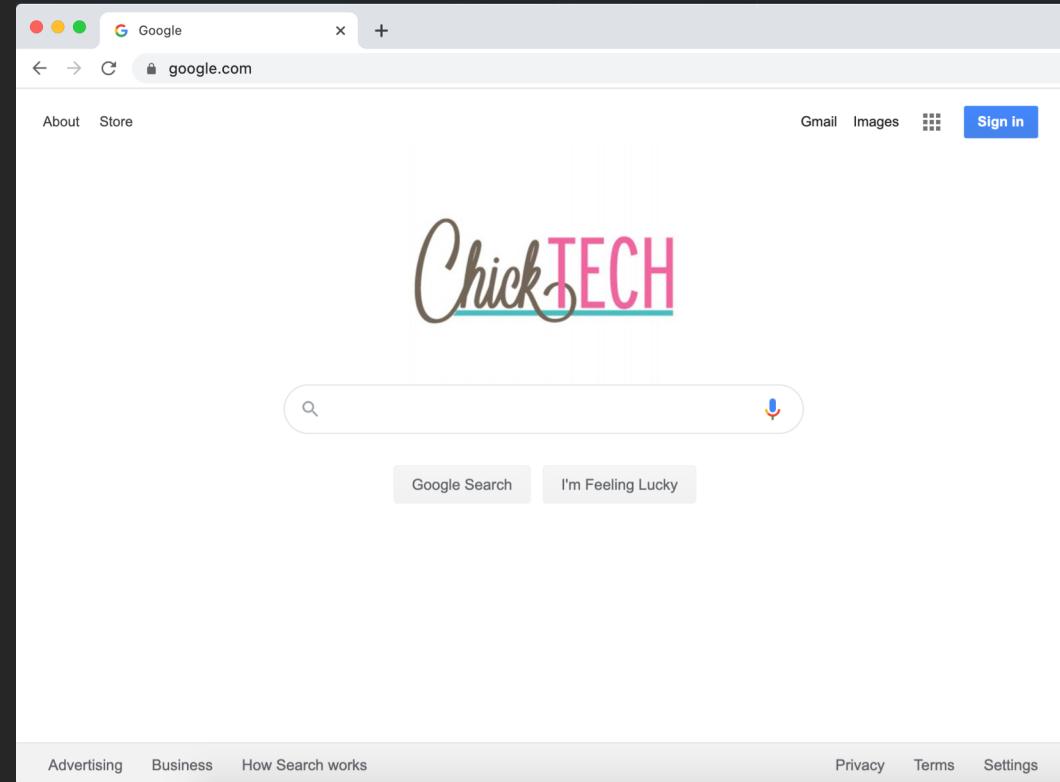
What are the typical parts of a web page?

# Activity 1

- Go to a web page that you often visit
- Open up your browser's developer tools
  - Chrome > View > Developer > Developer Tools  
or Right Click > Inspect
  - Firefox > Tools > Web Developer > Toggle Tools  
or Right Click > Inspect Element
- Notice where the HTML and CSS are in the browser
  - Where are they/what tab(s) are they on?
- Where is the JavaScript?

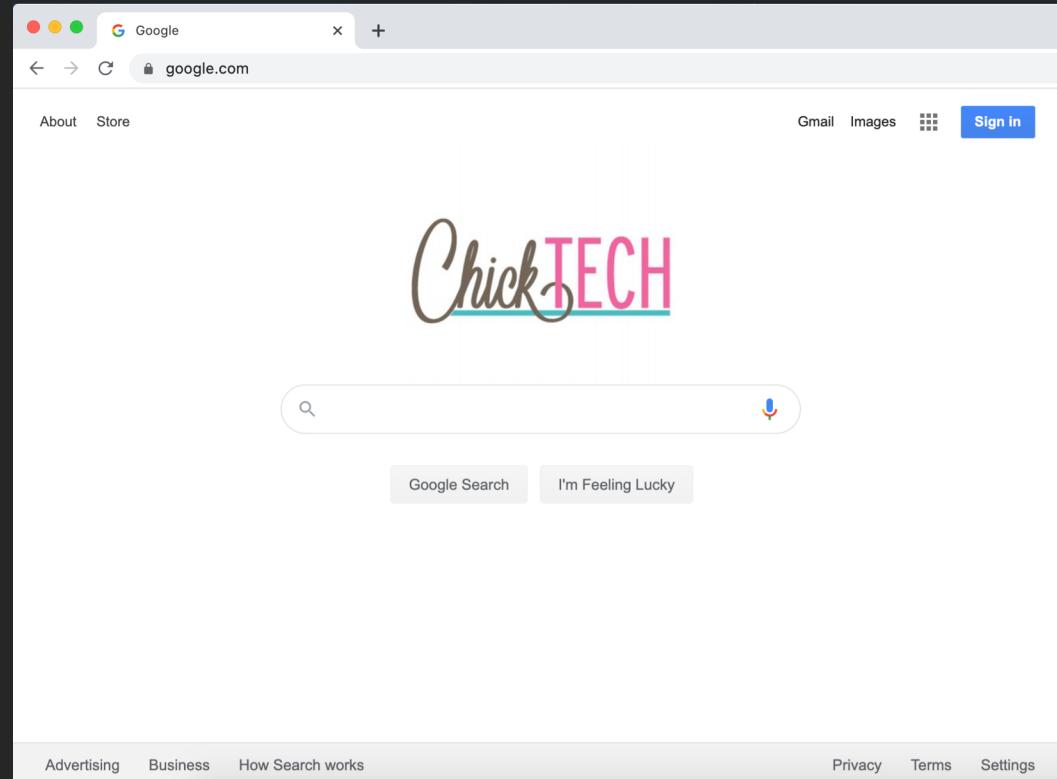
# Activity 2

- Find an image online that you like and try to change the Google logo image on google.com to the image that you've found using the developer tools in your browser. Try to get the image to the right size as well if it doesn't already fit nicely.



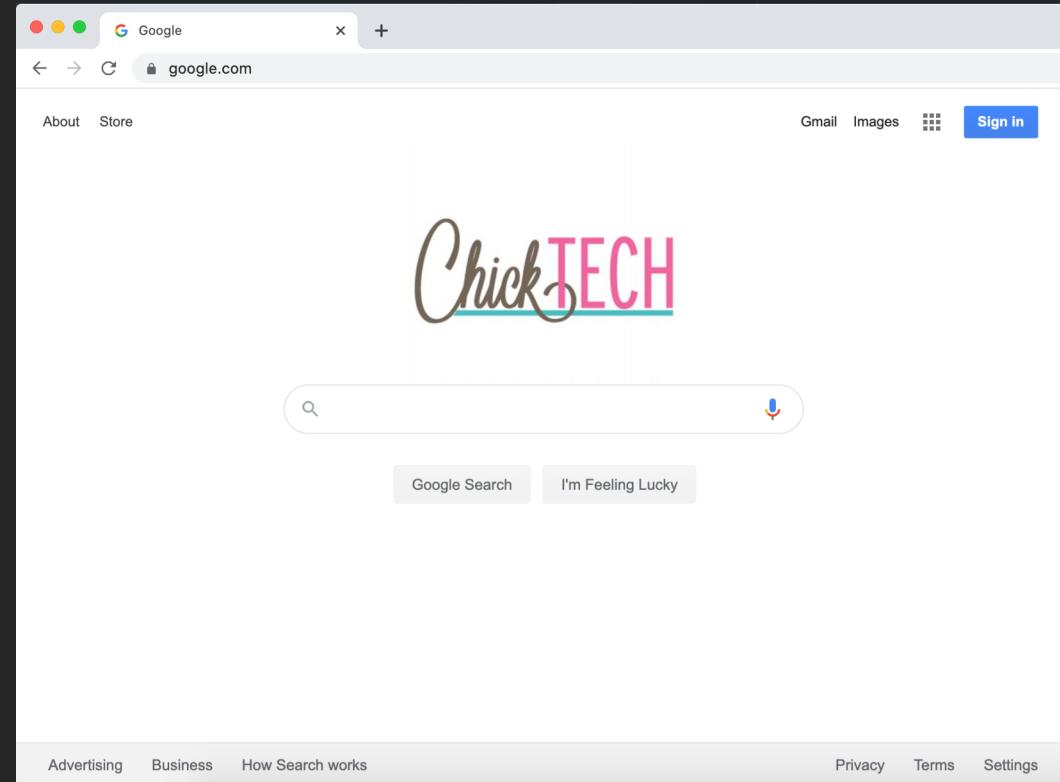
# Activity 2

- Step 0: Don't panic.
- Step 1: What information do we know and what information do we want?  
(clarify the inputs and outputs; make sure we understand the problem)

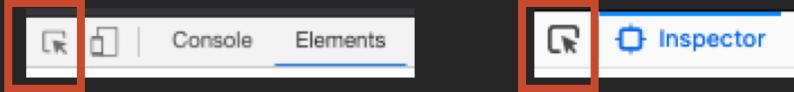


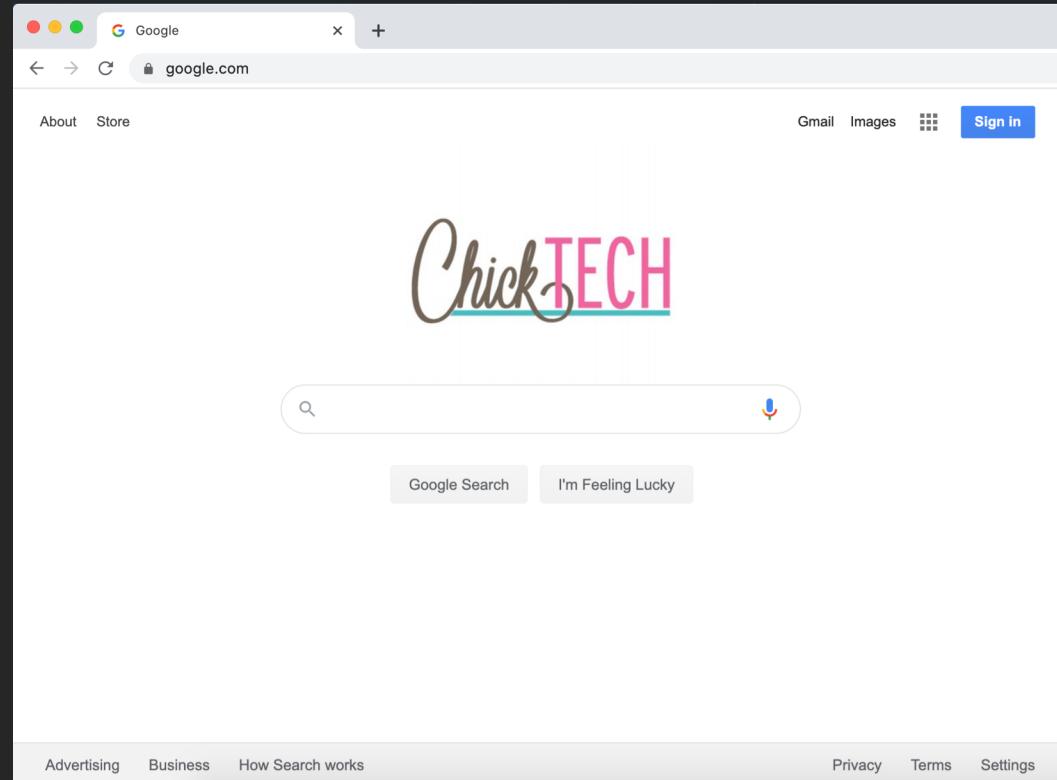
# Activity 2

- Step 2: What information would be helpful to know or what can I ask to get me closer to a solution?  
(What are things that can help me plan a solution?)
  - What should/could we be using to solve this?
    - HTML, CSS, or JS?



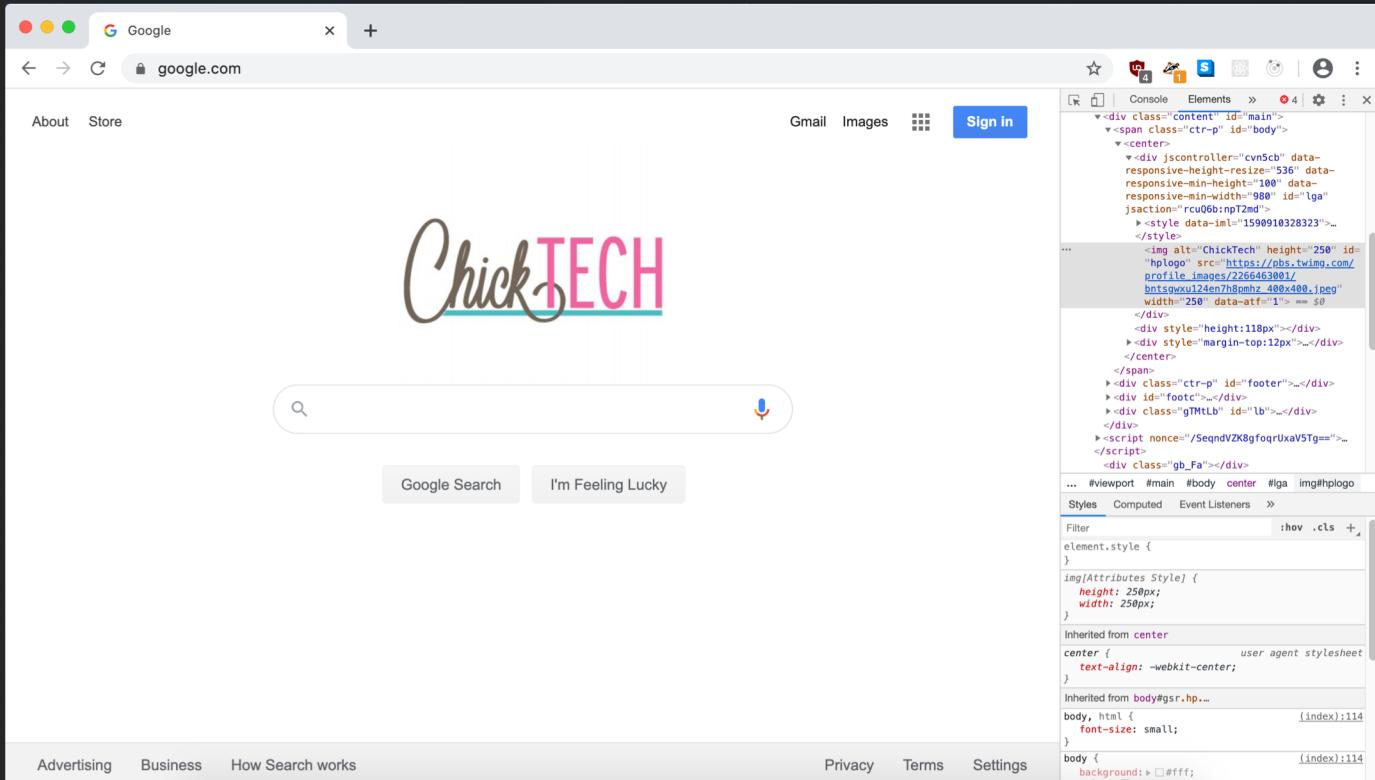
# Activity 2

- Step 3: Put the information together (implement the solution)
  - If we're using HTML, what element am I looking for?
    - Look through HTML manually or use the Element selector with Dev Tools
  - How do I modify that element?
    - Look up that tag in MDN Web Docs on how to use it
  - Do I have the image I want to replace ready?
    - This means I need to copy the link of the image I want over to be used



# Activity 2

- Solved! 🙌🙌🙌
- Step 4: Recap/reflect on what we did: **img** tag's **src** attribute as well as **height** and/or **width** may have been modified.
  - The **alt** attribute was modified to hold a text description of the image – useful if the image doesn't load, as well as for accessibility



# Activity 3

Activity code: [https://repl.it/@v\\_lai/pizzagenerator#script.js](https://repl.it/@v_lai/pizzagenerator#script.js)

What is something you learned today?

Q & A