

Class 1 - Web fundamentals and Intro to HTML

Course Curriculum

1. HTML CSS - 7 Class - Portfolio project
2. JavaScript and Machine Coding - 22 classes - Kanban board
3. React ~ 13 classes - Movies app
4. MERN module ~ 10 classes - BookMyShow clone

HTML / CSS

1. Foundation of Web Development:
 - a. HTML is the backbone of web development. It provides the basic structure and markup for creating web pages.
 - b. CSS complements HTML by allowing you to style and format the content, making it visually appealing.
 - c. HTML defines the structure of a webpage, like headings, paragraphs, lists, images, links, and more.
 - d. CSS controls the layout and presentation of these elements, enabling you to create aesthetically pleasing and user-friendly websites.

Javascript

1. Client-Side Scripting: JavaScript is the primary language for adding interactivity to websites. It allows you to create dynamic

and responsive user interfaces, enhancing the overall user experience.

2. Frameworks and Libraries: Many popular web development frameworks and libraries, such as React, Angular, and Vue.js, are built using JavaScript.

How Internet Works

1. What happens when you request for a webpage in a web browser on your computer or phone and it reaches a server and gives back response
2. What happens from the moment you enter scaler.com and get the page

1. Clients and Servers

- a. Computers connected to the internet are called clients and servers



b.

- c. Clients are the typical web user's internet-connected devices
- d. Servers are computers that store webpages, sites, or apps.

2. Internet Connection

- a. Internet - network of networks
- b. <https://www.submarinecablemap.com/>
- c.
- d. Fibre optic cables under the sea
 - i. The data transmission in these cables happen almost close to the speed of light
 - ii. Which is why you can connect to the other part of world almost instantly
- e. Starlink

3. TCP / IP

- a. Communication protocols that define how data should travel across the internet
- b. Rule about how traffic movement should happen

4. DNS

- a. Domain name system is like address book of websites
- b. Every website is mapped to an IP address
- c. The browser needs to find out which server the website lives on, so it can send HTTP messages to the right place

5. HTTP

- a. Browsers effectively only speak one language to request information: the HyperText Transfer Protocol (HTTP).

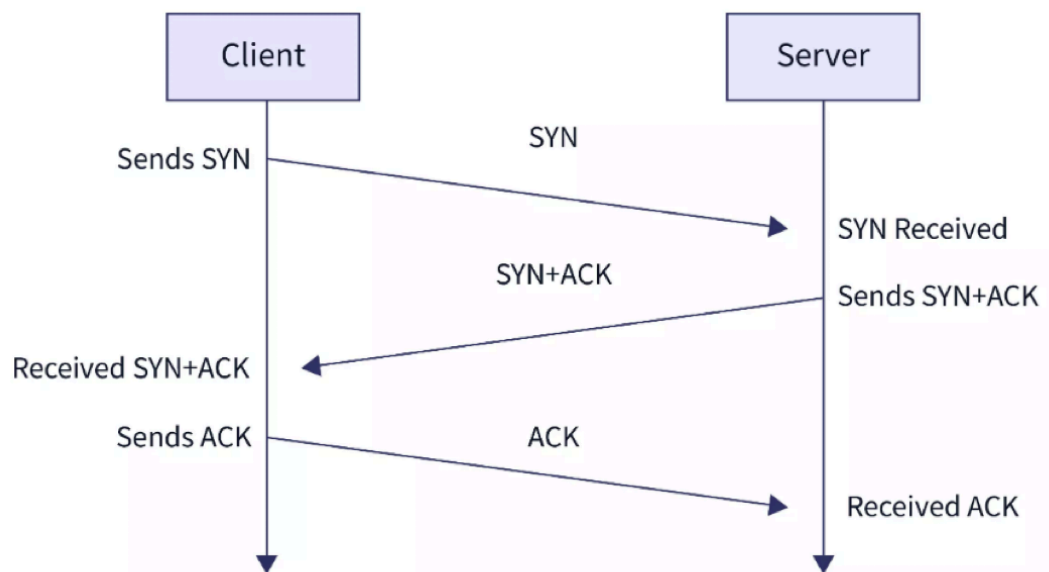
- b. This protocol defines a common language for client and server
- c. Tim Berners Lee considered father of world wide web . He was working at cern lab where he sent the first hyper text document
- d. Sir Tim formally submitted the proposal for what eventually became the world wide web
- e. <https://home.cern/science/computing/birth-web/short-history-web#:~:text=The%20document%20described%20a%20%22hypertext,at%20CERN%2C%20demonstrating%20his%20ideas.>

The flow

1. When we types in a website's name
2. The browser parses the information contained in the URL.
3. the protocol ("https")
4. the domain name
5. the resource ("/"). In this case, there isn't anything after the ".com" to indicate a specific resource, so the browser knows to retrieve just the main (index) page
6. Now when you send a letter, courier to anyone, do you just write their name
7. Here also we need the address, which is called as IP address
8. DNS Cache and lookup

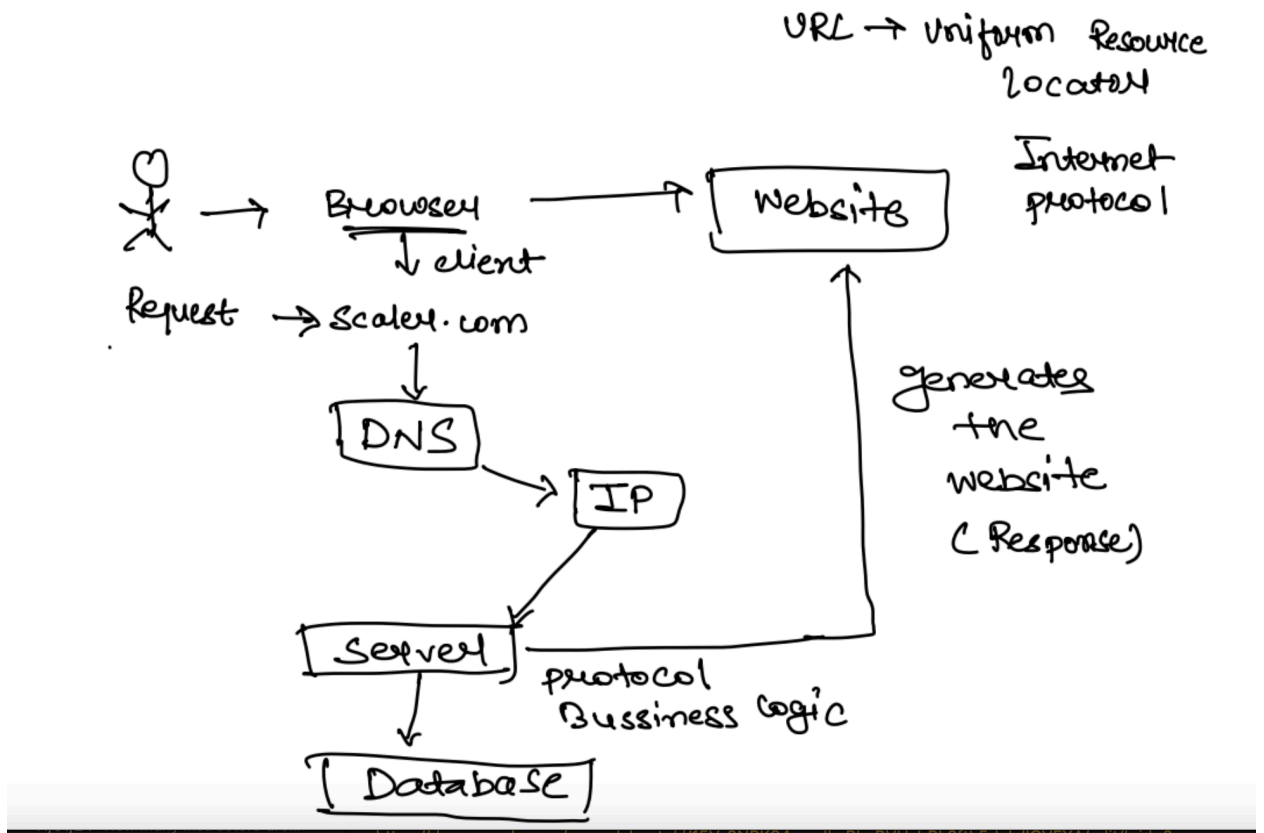
- a. IP address is retrieved. It can be from one of the cache as well
 - b. Browser cache - <chrome://net-internals/#dns>
 - c. OS cache
 - d. Router cache
 - e. ISP cache
 - f. DNS server
9. Now you have the IP address. Its like you have the number of the person you want to talk to
10. You calling a person by dialling in their number
11. The browser sends an **HTTP request message** to the server, asking it to send a copy of the website to the client
12. This message, and all other data sent between the client and the server, is sent across your internet connection using **TCP/IP**.
13. TCP / IP
- a. Any given interaction may be between two computer systems, or it may involve hundreds of systems.
 - b. For this to happen, the two computers need to know, ahead of time, how they are expected to communicate.
 - c. How do they start the conversation?
 - d. Whose turn is it to communicate?
 - e. How does each computer know its message was transmitted correctly?
 - f. How do they end the conversation?

- g. TCP/IP breaks each message into packets, and those packets are then reassembled on the other end.
 - h. Puzzle game
 - i. <https://images.app.goo.gl/GRsSZDuHwEGvGvLz8>
14. TCP 3 Way handshake
- a. TCP establishes a secure and reliable connection between two devices.
 - b. The 3-Way handshake is a TCP/IP network connection mechanism that connects the server and client.
 - c. Before the real data communication process begins, both the client and server must exchange synchronization and acknowledgment packets.



- d.
15. Story continues
16. Client sends a Get/ Put / Post request to server

17. If the server approves the client's request, the server sends the response
 - a. What is in the response - HTML / CSS and JS
18. The browser assembles the small chunks of response into a complete web page and displays it
19. Summarising



IDE

1. Integrated development environment
2. It's a comprehensive tool used by developers to write and test software

3. We are going to use Visual Studio Code (VS Code) which is a popular IDE. It's a lightweight IDE developed by Microsoft, widely used for programming in various languages.
4. VS Code offers features like syntax highlighting, intelligent code completion, and other imp features making it a versatile tool for developers.
5. Download VS code

HTML

1. Under the folder, create index.html
2. Type exclamation and hit enter -> boilerplate code

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>

</body>
</html>
```

3. This boilerplate HTML code is like the skeleton of a webpage:
- 4.
5. <!DOCTYPE html>: It's like saying, "Hey, we're starting an HTML page!" (actually HTML5 which is the latest version of HTML)
6. <html lang="en">: This is the main container of your page, and lang="en" means it's in English.

7. Inside `<html>`, there are two parts:
8. `<head>`: Think of it as the brain of your page. Its not visible but contains imp info
9. Meta tags -
 - a. Meta tags in HTML are used to provide metadata about the web page. This data isn't displayed on the page itself but is important for browsers and search engines.
 - b. They're essentially instructions to the browser and search engines about how to handle and display your page.
10. `<meta charset="UTF-8">`: This makes sure your page can understand a wide range of characters from different languages.
 - a. When you're coding, you might use characters that are specific to certain languages, like accents in French or characters in Chinese.
 - b. The UTF-8 character set includes a huge array of these characters from different languages. Without specifying UTF-8, your browser might not correctly interpret and display these characters, leading to garbled text or question marks on your webpage.
11. `<meta name="viewport" ...>`: This helps your page look good on phones and tablets.
12. `<title>Document</title>`: This is your page's title. You'll see it in the browser tab.
13. `<body>`: This is where all the content you see on the webpage goes, like text and images.

Running the file

1. Write some text like hello world
2. Open the index.html from the file system and double click to run the file
3. Install live server extension

Case insensitive

1. HTML is designed to be case-insensitive for its tags, meaning it doesn't matter if you use uppercase, lowercase, or a mix of both for the tag names.
2. Whether you write <head>, <HEAD>, or <HeAd>, it will be interpreted the same way by the browser. This is part of the HTML specification to make the language more forgiving and easier to use, especially for beginners.
3. However, it's considered good practice to stick to lowercase for consistency and readability.

Tags

1. Headings - h1 to h6
2. Paragraph tags - for writing some descriptive content
 - a. Lorem - emmet abbreviations
3. HTML tags like <p> and <h1>-<h6> give meaning to the text, making it easier for both people and computers (screen readers) to understand the structure and importance of the content

4. Img tags - self enclosing tag

- a. It has additional information which describes the img which are called as attributes
- b. This is like adding adjectives to nouns in a sentence. Just like adjectives give more detail or characteristics to a noun, attributes provide additional information about an HTML tag.
- c. accessories or features of a car. Just as a car can have features like a sunroof, a specific color, or a navigation system, HTML tags can have attributes that add extra features or details.
- d. see the src and alt tags (alternate to image)
- e. Either break the url or disable images with extension to see the alt tags

5. Anchor tags

- a. <a> tags, commonly known as anchor tags, are used in HTML to create hyperlinks
- b. They allow users to click on a link and be taken to another webpage, document, or a specific section within the same page.
- c. This is fundamental for the interconnected nature of the World Wide Web, where resources are linked together through these hyperlinks.

```
<a>take me to google</a>
```

- d. As you can see, it does nothing
- e. What we need are some attributes for anchor tags

```
<a href="http://www.google.com">take me to  
google</a>
```

- f. What if we want to open in a new tab

```
<a href="http://www.google.com"  
target="_blank">take me to google</a>
```

6. Div tags

- a. Think of <div> tags as a way to divide the page into different areas, making it easier to design and manage the layout.
- b. <div> tags in HTML are like invisible boxes that help organize and group content on a webpage.
- c. like a header, footer, some content area, buttons area
- d. Let us divide our code into different divisions

```
<!DOCTYPE html>  
<html lang="en">  
  <head>  
    <meta charset="UTF-8" />  
    <meta name="viewport" content="width=device-width,  
initial-scale=1.0" />  
    <title>Document</title>  
  </head>  
  <body>  
    <div>
```

```

        <h1>heading 1</h1>
        <h2>heading 1</h2>
        <h3>heading 2 or 3</h3>
        <h8>checking thus</h8>
        <h11>comparing this</h11>
    </div>
    <div>
        <p>Lorem, ipsum dolor sit amet consectetur
adipisicing elit. Magnam voluptatibus a veniam
doloribus omnis laborum ipsum, dolore incidunt earum
delectus est? Culpa, aut voluptas voluptatem provident
exercitationem magni accusantium! Laudantium.</p>
    </div>
    <div>
        <!-- images are self closing tags -->
        <img src="" alt="cute dog image"/>
        <a href="http://www.google.com"
target="_blank">take me to google</a>
    </div>
    <div>
        <p>some detial</p>
        <div>
            <p>some more detail</p>
        </div>
    </div>

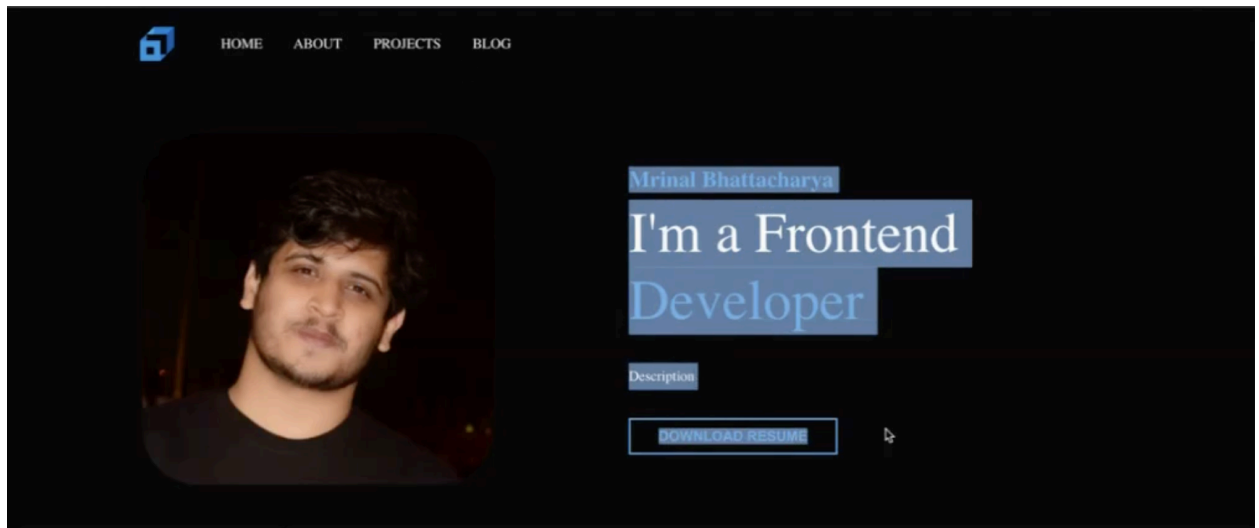
</body>
</html>

```

1.

Semantic tags

1. Semantic tags in HTML provide meaning to the structure of web pages, helping browsers, search engines, and assistive technologies understand the content better.
2. They were needed to improve web accessibility and search engine optimization, moving away from generic tags like `<div>` and `` which don't convey specific meanings.
3. <https://www.semrush.com/blog/semantic-html5-guide/>
4. Semantic tags were introduced with HTML5,
5. Create a new semantic.html file and copy the assets folder for portfolio



- 6.
7. First we will talk about headers
 - a. This has this big image, navbar, some big content to capture attention and a call to action

```
<body>  
  <header>
```

```

        <div>
            <nav>
                <div>
                    
                </div>
            </nav>
        </div>
    </header>

</body>

```

- b. Let start with the header tag
- c. Inside the header tag, we can have the divs to separate the nav section and the content part
- d. Inside the nav, again we have the logo and the menu
- e. Try the image without the width
- f. With the width, the browser also adjusts the height to maintain the aspect ratio
- g. Add another divs for menu

```

<div>
    
</div>
<div>
    <a href="#">Home</a>
    <a href="#">About</a>
    <a href="#">Projects</a>
    <a href="#">Blogs</a>
</div>

```

- h. Now we come out of the nav's division and below where we have the image and some text
- i. Again this can be thought of as two divisions
- j. Creating a div for the image

```
<div> ...  
</div>  
<div>  
    
</div>
```

- k. Adding a div for intro

```
<div>  
    
</div>  
<div>  
  <h1>Mrinal</h1>  
  <h2>I am a Full Stack Developer</h2>  
  <p>A simple guy</p>  
</div>
```

- l. Add a button to download resume

```
<p>A simple guy</p>  
  <button>Download Resume</button>
```

8. About me section

```
<section>
```



```

  <div>
    <h1>About me</h1>
    <h3>Hello I am Mr.X</h3>
    <p>Lorem ipsum dolor sit amet consectetur
adipisicing elit. Tempora, sapiente.</p>
  </div>
</section>
```

9. Homework to make the project section

Github account

2. Signup and create a account if not done
3. GitHub is like a big online library for computer code.
4. Imagine every project as a book in this library. People from all over the world can store their code projects here, make changes, and work together. It keeps track of all the changes, so you can always see who changed what and when. It's a place where coders collaborate and share their work, making it easier to create and improve software.

Creating an account on GitHub

5. Create a personal account to get started with GitHub.
6. About your personal account on GitHub.com

7. To get started with GitHub, you'll need to create a free personal account on GitHub.com and verify your email address. Every person who uses GitHub.com signs in to a personal account.
8. Your personal account is your identity on GitHub.com and has a username and profile. For example, see [@octocat's](#) profile.
9. If your GitHub account has been created for you by your company, you can skip this article and continue to "[Hello World.](#)"

Signing up for a new personal account

10. Navigate to <https://github.com/>.
11. Click Sign up.
12. Follow the prompts to create your personal account. During sign up, you'll be asked to verify your email address. Without a verified email address, you won't be able to complete some basic GitHub tasks, such as creating a repository. If you're having problems verifying your email address, there are some troubleshooting steps you can take. For more information, see "[Verifying your email address.](#)"

Next steps

13. Now that you've created your personal account, we'll start to explore the basics of GitHub. In the following tutorial, "[Hello World,](#)" you'll learn about repositories and how to create one, and

you'll be introduced to concepts such as branching, commits, and pull requests.

14. We strongly recommend that you configure 2FA for your account. 2FA is an extra layer of security that can help keep your account secure. For more information, see [“Configuring two-factor authentication.”](#)