



PCET's  
**Pimpri  
Chinchwad  
University**

Learn | Grow | Achieve

## **Module 1:- Introduction to Front End Development**

**Name:-** Jaykishan J Saharan

**Div:-** D

**Roll no.:-** 27

**Batch:-** D1

# Contents:-

- Introduction
- Short description of each module
- Advantages
- Disadvantages
- Applications
- Conclusion
- GitHub Link
- Certificate



# ● Introduction to Front End Development

1. Front-end development focuses on building the visual and interactive parts of a website or web application that users directly interact with. It involves coding the layout, design, and user experience using technologies like:
  - HTML (HyperText Markup Language) – Structures the content of a webpage.
  - CSS (Cascading Style Sheets) – Styles and enhances the appearance of the webpage.
  - JavaScript – Adds interactivity and dynamic features.



# Introduction to Front End Development

- Front-end developers also use frameworks and libraries such as React, Angular, and Vue.js to streamline development and improve efficiency. They ensure websites are responsive, meaning they work well on different screen sizes, and optimize performance for fast loading speeds.
- A front-end developer collaborates with designers and back-end developers to create seamless, user-friendly web experiences.



# HTML (Hyper text markup language)

## 1. Introduction to HTML:-

HTML (HyperText Markup Language) is the standard language for creating web pages. It structures web content using elements (tags) that define headings, paragraphs, links, images, tables, forms, and more. HTML works alongside CSS for styling and JavaScript for interactivity.

# HTML Syntax

An HTML document starts with a `<!DOCTYPE html>` declaration, followed by an `<html>` element that contains the `<head>` and `<body>` sections.

## Basic HTML Structure

```
<!DOCTYPE html>
<html>
<head>
  <title>My First HTML Page</title>
</head>
<body>
  <h1>Welcome to HTML</h1>
  <p>This is a simple HTML document.</p>
</body>
</html>
```



# Common HTML Tags:-

1. Headings: `<h1>` to `<h6>` for different heading levels.
2. Paragraph: `<p>` for paragraphs.
3. Links: `<a href="URL">Link Text</a>` for hyperlinks.
4. Images: `` for adding images.
5. Lists:  
Ordered list: `<ol><li>Item 1</li></ol>`  
Unordered list: `<ul><li>Item 1</li></ul>`
6. Tables: `<table><tr><td>Data</td></tr></table>` for tabular data.
7. Forms: `<form><input type="text"></form>` for user input.
8. Div & Span: `<div>` for block-level grouping, `<span>` for inline grouping.

# Multimedia

Multimedia comes in many different formats. It can be almost anything you can hear or see, like images, music, sound, videos, records, films, animations, and more. Web pages often contain multimedia elements of different types and formats.

## HTML Audio

The HTML `<audio>` element is used to play an audio file on a web page.

Example

```
<audio controls>
```

```
<source src="horse.ogg" type="audio/ogg">
```

```
<source src="horse.mp3" type="audio/mpeg">
```

Your browser does not support the audio element.

```
</audio>
```



# HTML Video

The HTML `<video>` element is used to show a video on a web page.

Example:-

Courtesy of Big Buck Bunny:

To show a video in HTML, use the `<video>` element:

Example

```
<video width="320" height="240" controls>  
<source src="movie.mp4" type="video/mp4">  
<source src="movie.ogg" type="video/ogg">  
Your browser does not support the video tag.  
</video>
```

# HTML Semantic Elements

Semantic elements = elements with a meaning

## Semantic Elements

A semantic element clearly describes its meaning to both the browser and the developer.

Examples of non-semantic elements: `<div>` and `<span>` - Tells nothing about its content.

## HTML5 Semantic Tags

HTML5 introduced several new tags called semantic tags. These tags were designed to communicate the function of blocks of content that were common on many web pages. Prior to HTML5, developers just used `<div>` tags for all blocks.



```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-
6  scale=1.0">
7      <title>My First HTML Page</title>
8      <style>
9          body {
10              font-family: Arial, sans-serif;
11              background-color: #f4f4f4;
12              text-align: center;
13              padding: 20px;
14          }
15          h1 {
16              color: #333;
17          }
18          p {
19              color: #666;
```

```
20         .button {
21             display: inline-block;
22             padding: 10px 20px;
23             font-size: 16px;
24             color: white;
25             background-color: blue;
26             text-decoration: none;
27             border-radius: 5px;
28         }
29     </style>
30 </head>
31 <body>
32
33     <h1>Welcome to My Website</h1>
34     <p>This is a simple HTML page with some styling.</p>
35     <a href="https://www.example.com" class="button">Click Me</a>
36
37 </body>
38 </html>
```

# OUTPUT

🌐 My First HTML Page >

## Welcome to My Website

This is a simple HTML page with some styling.

Click Me

# CSS (cascading style sheet)

CSS

What is CSS? Why use CSS?

CSS is the language we use to style a Web page.

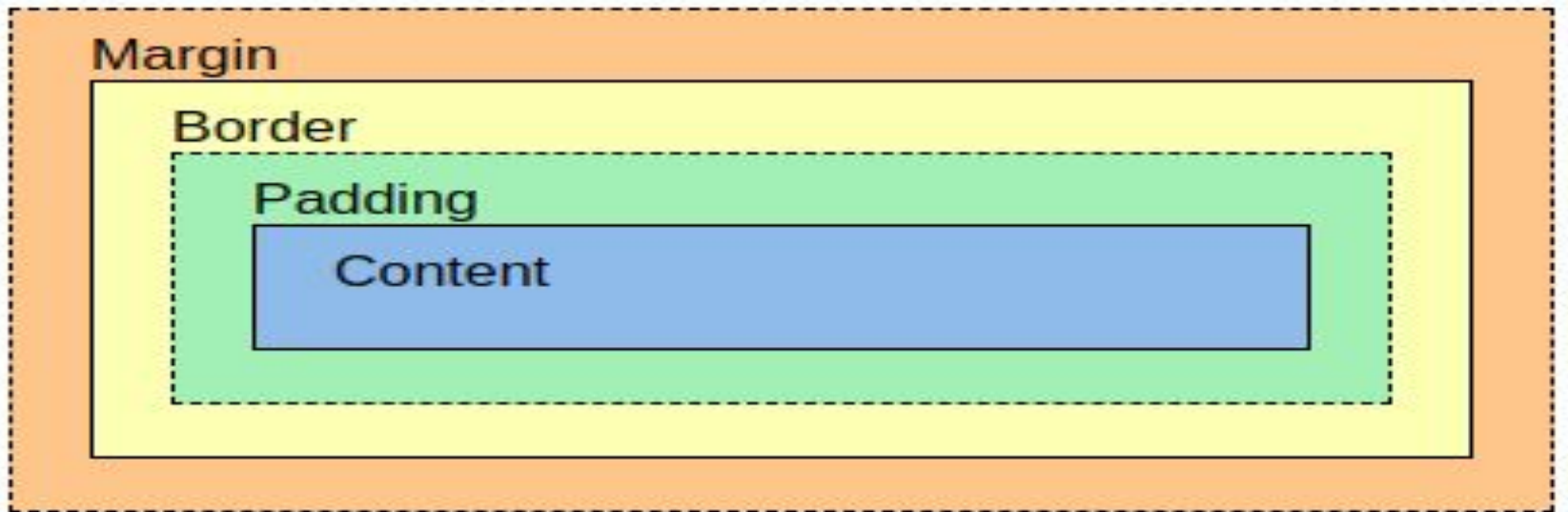
- CSS stands for Cascading Style Sheets
- CSS describes how HTML elements are to be displayed on screen, paper, or in other media
- CSS saves a lot of work. It can control the layout of multiple web pages all at once
- External stylesheets are stored in CSS files
- Why Use CSS?
- CSS is used to define styles for your web pages, including the design, layout and variations in display for different devices and screen sizes.
- CSS provides efficiency in webpage design
- CSS provides faster page download:
- CSS is easy to work:



CSS

# The Box Model

- When the browser draws an object on a page, it places it into an invisible rectangular space called a “bounding box.”
- You can specify the size, look, and feel of the margins, the padding, the border, and the content of the box.
- Internet Explorer interprets CSS box styles differently than most other web browsers.



## CSS Selector

A CSS selector is a pattern used to select and apply styles to HTML elements. It targets HTML elements based on their attributes like ID, class, type, or other characteristics, and applies the defined style rules to those selected elements.

We can divide CSS selectors into five categories:

- Simple selectors (select elements based on name, id, class)
- Combinator selectors (select elements based on a specific relationship between them)
- Pseudo-class selectors (select elements based on a certain state)
- Pseudo-elements selectors (select and style a part of an element)
- Attribute selectors (select elements based on an attribute or attribute value)



# CSS properties

1. Layout properties:- Controls the structure and positioning of elements
  - display (e.g., block, flex, grid)
  - position (e.g., absolute, relative, fixed)
2. Box model properties:-
  - ~~ Defines sizes, spacing and borders of elements.
3. Typography properties:- Styles text and fonts
4. Color and background properties:-
  - ~~ Defines colors, gradient and images.

5. Flexbox properties:-

~~ (for flexible layouts)

6. Grid properties:-

~~ for grid based layouts

7. Animations and Transition properties:-

~~ Add motion effect

8. Visibility and interaction properties:-

Control element visibility and interaction

9. Responsive design properties:-

~~ media queries (@media (max-width:768px) { } )





# For Example!

< index.html ×

style.css ×

script.js ×

```
1  /* Apply styles to the body */
2  body {
3      font-family: Arial, sans-serif;
4      background-color: #f0f0f0;
5      text-align: center;
6      margin: 50px;
7  }
8
9  /* Style the heading */
10 h1 {
11     color: #333;
12 }
13
14 /* Style the paragraph */
15 p {
16     font-size: 18px;
17     color: #555;
18 }
19
```

```
20  /* Style the button */
21  button {
22      background-color: #007bff;
23      color: white;
24      border: none;
25      padding: 10px 20px;
26      font-size: 16px;
27      cursor: pointer;
28      border-radius: 5px;
29  }
30
31  /* Add hover effect to the button */
32  button:hover {
33      background-color: #0056b3;
34  }
```

# OUTPUT

**.. Welcome to  
.. My Website..**

**This is a simple exompese  
of CCSS styling.**

---

**Click me** •

**Click Me**

• **Click me**

# Java Script

## Introduction to Java script

JavaScript is a powerful and versatile programming language used to create dynamic and interactive web pages. It is one of the core technologies of web development, alongside HTML (HyperText Markup Language) and CSS (Cascading Style Sheets). JavaScript allows developers to add interactivity, manipulate the Document Object Model (DOM), handle events, and communicate with servers using APIs.

## Key Features of JavaScript:

1. Client-Side Execution: Runs directly in web browsers, reducing server load.
2. Lightweight and Fast: Designed for efficient performance.
3. Event-Driven: Responds to user interactions like clicks, key presses, and form submissions.
4. Cross-Platform Compatibility: Works on different operating systems and browsers.
5. Object-Oriented and Functional: Supports multiple programming paradigms.
6. Asynchronous Processing: Uses features like Promises and `async/await` for handling time-consuming tasks.

## Where is JavaScript Used?

- Web Development: Adds interactivity to websites (e.g., animations, form validation).
- Server-Side Development: With Node.js, JavaScript can be used to build backend applications.
- Mobile App Development:

Frameworks like React Native allow JavaScript to create mobile apps.

- Game Development: Used in  
browser-based game development.

- Machine Learning & AI: Libraries like TensorFlow.js enable AI applications in the browser.



```
// Display an alert message  
alert("Hello, JavaScript!");  
  
// Print to the console  
console.log("Welcome to  
JavaScript!");  
  
// Declare a variable  
let name = "John";  
console.log("Hello, " + name);
```



# Advantages and disadvantages

## Advantages & Disadvantages Of HTML



### Advantages

- All browsers supports HTML.
- It is easy to learn and use.
- Useful for beginners in web design.
- The software is available for free.
- It is widely used.



### Disadvantages

- It takes more time.
- We need to create a lot of code for a simple webpage.
- It is not completely secured.
- Long codes becomes complex.
- It can only be used for static and plain page.





# CSS Advantages & Disadvantages

Advantages	Disadvantages
<ul style="list-style-type: none"><li>• Better Website Speed</li><li>• Easier to Maintain</li><li>• Consistent Design</li><li>• Better Device Compatibility</li></ul>	<ul style="list-style-type: none"><li>• Cross-Browser Issues</li><li>• Security Issues</li><li>• Extra Work for Developers</li><li>• Confusion due to many CSS levels</li></ul>

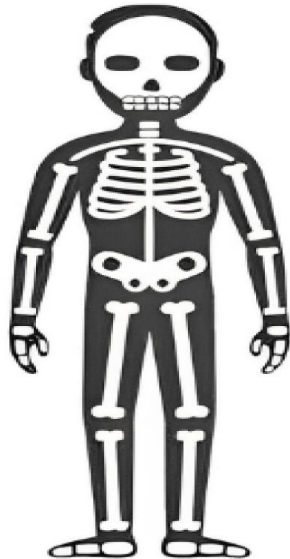
# Advantages & Disadvantages Of JavaScript

Advantages	Disadvantages
<ul style="list-style-type: none"><li>• Speed &amp; Simplicity</li><li>• Regular Updates</li><li>• Inter-operability</li><li>• Asynchronous</li><li>• Reduced Server Load</li></ul>	<ul style="list-style-type: none"><li>• Code Visibility</li><li>• Debugging Issues</li><li>• Single Inheritance</li><li>• SEO Challenges</li><li>• Slower DOM</li></ul>

# HTML



**HTML** the Skeleton



# CSS



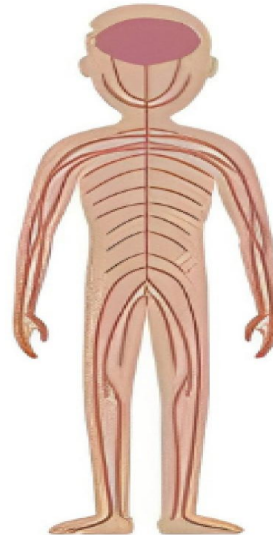
**CSS** the Skin



# JS



**Javascript** the Brain



---

# Application of Front End Developed



- Website Design: Creating static and dynamic web pages.
- Web Applications: Developing interactive tools like online calculators, games, and dashboards.
- Mobile Web Development: Building responsive websites that work well on mobile devices.
- E-Commerce: Designing online stores with interactive product listings and carts.



# Conclusion:-

## CONCLUSION

In conclusion, web development is a dynamic and essential field that powers the modern internet. It involves front-end and back-end technologies, combining coding, design, and functionality to create interactive websites and applications. As technology evolves, web development continues to advance, offering endless opportunities for innovation and growth. Whether you're a beginner or an experienced developer, mastering web development opens doors to creative problem-solving and career possibilities in the digital world.



# GitHub



<https://github.com/jaykishan1saharan/WP-D1-27>



# CERTIFICATE



Feb 25, 2025

Jaykishan Saharan

has successfully completed

Introduction to Front-End Development

an online non-credit course authorized by Meta and offered through Coursera



COURSE  
CERTIFICATE



Verify at:  
<https://coursera.org/verify/HRPV4A7E6B6M>

Coursera has confirmed the identity of this individual and  
their participation in the course.