Introduction to CSS

- CSS is used alongside HTML and JavaScript to design web and mobile interfaces.
- It defines how HTML elements are displayed on screens, print, or other media.
- CSS enables consistent design across multiple pages.
- External stylesheets can be saved in separate .css files.

Why use CSS?

- Solves a big problem: avoids repeating font, color, and layout styles in each page.
- Developed by W3C to separate style formatting from HTML content.
- Saves time: one file change can update the entire website's look.
- Provides more styling options compared to plain HTML.

CSS Syntax

- A CSS rule set consists of:
- Selector → The HTML element to style (e.g., h1, p, div).
- Declaration block → One or more declarations separated by semicolons.
- Declaration → Property: value; pair (e.g., color: red;).

Example of CSS Syntax

```
h1 {
    color: blue;
    font-size: 20px;
}
```

Types of CSS

- Inline CSS → Applied directly in an HTML tag using the 'style' attribute.
- Internal CSS → Defined inside <style> within the <head> section.
- External CSS → Written in a separate .css file and linked using <link>.

Example of Inline CSS

-
- This is styled with inline CSS.
-

Example of Internal CSS

- <head>
- <style>
- p { color: red; font-size: 16px; }
- </style>
- </head>

Example of External CSS

- /* style.css */
- p {
- color: purple;
- font-size: 18px;
- }

- /* index.html */
- <link rel="stylesheet" href="style.css">

Exercise 1

- Create a paragraph and style it using inline CSS to:
- • Set text color to blue
- • Increase font size to 22px
- • Add background color yellow

Solution Exercise 1

-
- This is an inline CSS styled paragraph.
-

Exercise 2

- Use internal CSS to style headings:
- • h1 should be red with font size 30px
- • h2 should be green with font size 24px

Solution Exercise 2

- <head>
- <style>
- h1 { color: red; font-size: 30px; }
- h2 { color: green; font-size: 24px; }
- </style>
- </head>

Exercise 3

- Create an external CSS file and link it to HTML:
- • Paragraphs should be blue and centered
- • Body background should be light gray

Solution Exercise 3

- /* style.css */
- body { background-color: lightgray; }
- p { color: blue; text-align: center; }

- <!-- index.html -->
- <link rel="stylesheet" href="style.css">
- This is styled using external CSS.

Types of selectors:

- Element selector
- Class selector
- ID selector
- Universal selector (*)
- Grouping selectors

Element Selector

- Content:
- The element selector selects HTML elements by their tag name.
- It applies the same style to all occurrences of that element.
- Example:
- h1 {
- color: navy;
- text-align: center;
- }
- This will style all <h1> headings on the page.

Class Selector

- A class selector is defined with a dot (.).
- It is used to style multiple elements with the same class attribute.
- You can apply the same class to different HTML tags.
- Example:
- .intro {
- font-size: 18px;
- color: darkred;
- }
- Below code in index.html:
- This is an introductory paragraph.
- <h2 class=“intro”>This heading also uses the intro class.</h2>

ID Selector

- An ID selector is defined with a hash (#).
- It is used to style one unique element.
- Each ID must be unique within a webpage.
- Example:
- #main {
- background: yellow;
- padding: 10px;
- }
- Below code in index.html:
-
- This is the main section.
-

Universal Selector

- The universal selector (*) applies styles to all elements on the page.
- Useful for resetting or applying a common style everywhere.
- Example:
- * {
- margin: 0;
- padding: 0;
- }
- This removes default margins and paddings from all elements.

Grouping Selectors

- Grouping selectors saves time by styling multiple elements with the same rule.
- You list selectors separated by commas.
- Example:
- h2, h3 {
- color: green;
- text-transform: uppercase;
- }
- This will apply the same styles to all <h2> and <h3> elements.

Task

- Create a webpage with:
- A heading styled using an element selector.
- A paragraph styled using a class selector.
- A footer styled using an ID selector.

Exercise code

```
<head>
  <title>Exercise – External CSS</title>
  <!– Linking external CSS file →
  <link rel=“stylesheet” href=“style.css”>
</head>
<body>
  <!– Heading styled using element selector →
  <h1>Welcome to My Webpage</h1>

  <!– Paragraph styled using class selector →
  <p class=“text”>This is a paragraph styled with a class
selector.</p>

  <!– Footer styled using ID selector →
  <div id=“footer”>
    <p>Footer section styled with an ID selector.</p>
  </div>
</body>
```

```
/* Element selector */
h1 {
  color: blue;
  text-align: center;
  font-family: Arial, sans-serif;
}

/* Class selector */
.text {
  font-size: 18px;
  color: darkred;
  line-height: 1.5;
}

/* ID selector */
#footer {
  background-color: lightgray;
  text-align: center;
  padding: 15px;
  margin-top: 20px;
  font-style: italic;
}
```

Colors & Backgrounds

- Named colors, HEX, RGB, HSL
- Background color, image, repeat, position, attachment
- Code Example:
- body {
- background-color: #f0f0f0;
- background-image: url(“pattern.png”);
- background-repeat: no-repeat;
- background-size: cover;
- }
- Exercise: Design a webpage with a gradient background and a repeating pattern for the header.

Gradient

- A linear gradient in CSS is a way to create a smooth transition between two or more colors along a straight line (horizontal, vertical, diagonal, or any angle).
- Instead of using a single flat color, it blends multiple colors gradually.
- Syntax:
- background: linear-gradient(direction, color1, color2, ...);

Gradient

- Key Points:
- Direction (optional):
- to right → left to right
- to bottom → top to bottom
- 45deg → diagonal at 45 degrees
- Default is top to bottom
- Colors:
- You can use 2 or more colors.
- You can also specify percentage/stop points.

Example

```
/* Top to bottom blue to pink */
background: linear-gradient(blue, pink);

/* Left to right green to yellow */
background: linear-gradient(to right, green, yellow);

/* Diagonal 45 degrees */
background: linear-gradient(45deg, orange, purple);

/* With color stops */
background: linear-gradient(to right, red 20%, white 50%, blue 80%);
```

Repeat and size

- Background-repeat
- What: Controls tiling of a background image.
- Values: repeat (both axes), repeat-x, repeat-y, no-repeat, space, round.
- Example: body { background-repeat: no-repeat; }
- Tip: space and round are handy for patterns that must tile cleanly.

- Background-size
- What: Size of the background image.
- Values: auto, cover, contain, or specific sizes (100% 50%, 200px 100px).
- Example: section { background-size: cover; }
- Tip: Use cover to fill the container (may crop), contain to fully show image (may letterbox).

Text Styling

- Font families, sizes, weights
- Text alignment, decoration, transform, spacing
- Code Example:
- p {
- font-family: Arial, sans-serif;
- font-size: 16px;
- text-align: right;
- text-transform: uppercase;
- letter-spacing: 2px;
- }
- Exercise:
- Create a short article page with styled headings, justified paragraphs, and uppercase subheadings.

Text Decoration and Transform

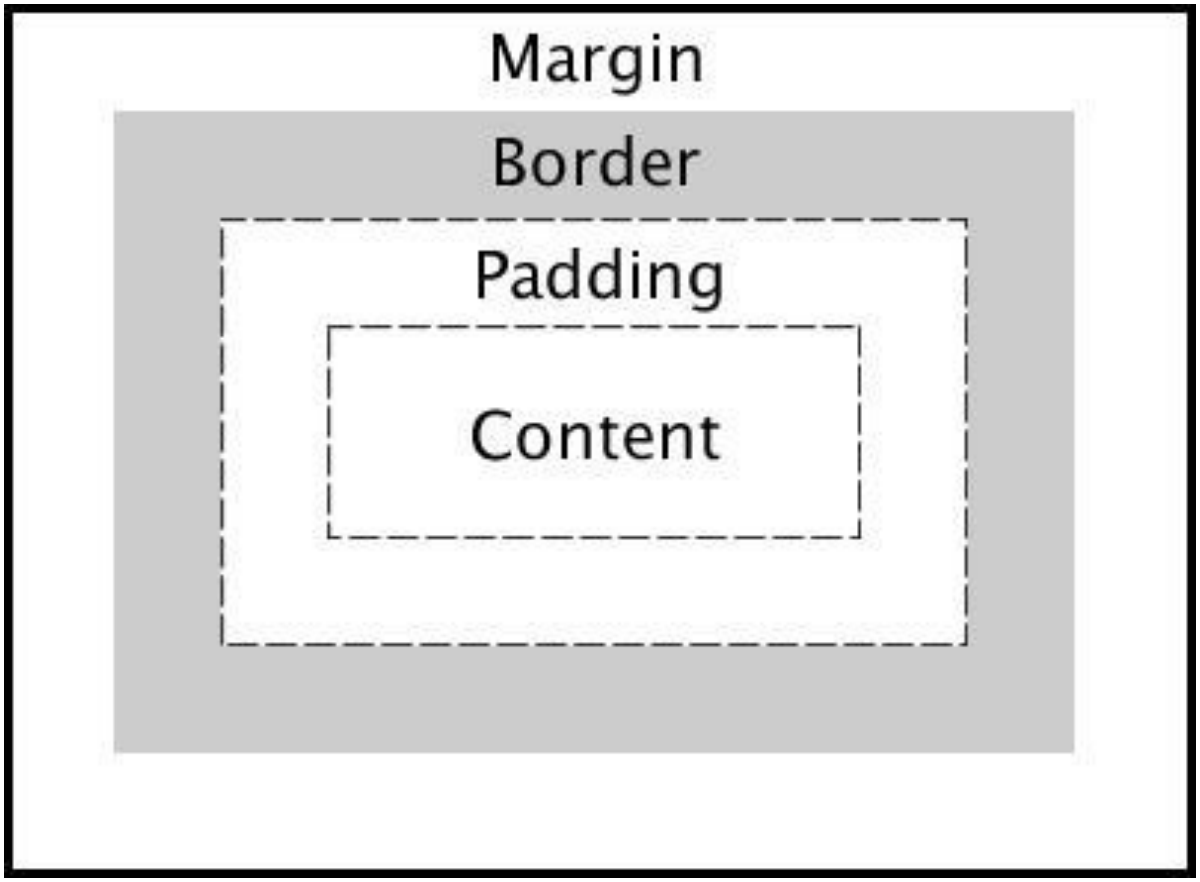
- Text-decoration
- What: Underlines, overlines, line-through, or none; also controls color and style in newer specs.
- Values: none, underline, overline, line-through, and longhand like text-decoration-line, text-decoration-color, text-decoration-style.
- Example: a { text-decoration: none; }
- Tip: For custom underline styles, use text-decoration-style: dotted;.

- Text-transform
- What: Alters case of text.
- Values: none, capitalize, uppercase, lowercase.
- Example: h2 { text-transform: uppercase; }
- Tip: Avoid using uppercase for large bodies of text for readability.

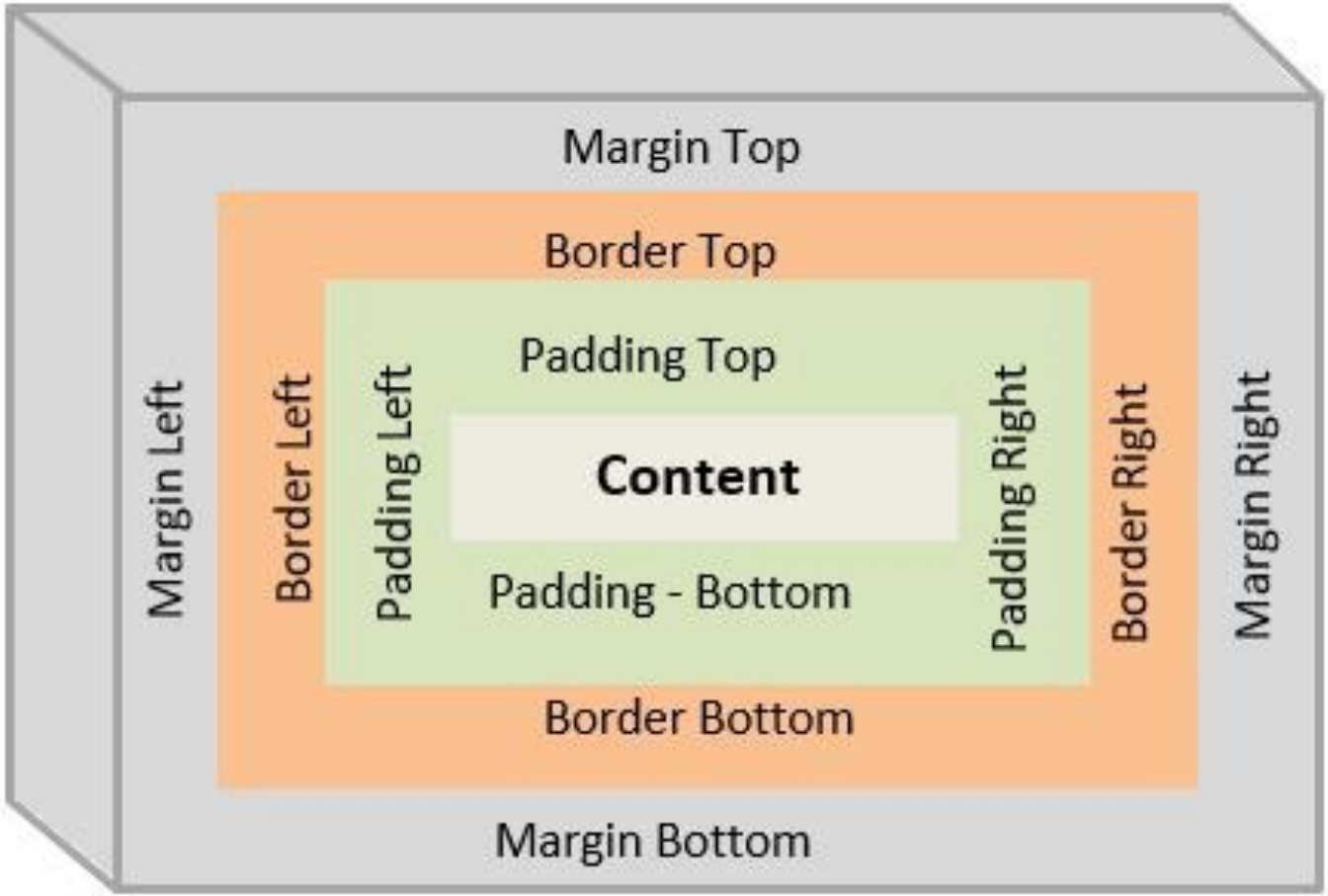
CSS Box Model – Introduction

- Every HTML element is a rectangular box.
- The CSS Box Model describes how the size and spacing of elements are calculated.
- It consists of 4 parts:
- Content – the actual text, image, or element inside.
- Padding – space between content and border.
- Border – wraps around padding and content.
- Margin – space outside the border, separating elements.
- Key Idea: Total element size = content + padding + border + margin

Box model



Box model


Margin Top
Border Top
Padding Top
Margin Left
Border Left
Padding Left
Content
Padding Right
Border Right
Margin Right
Padding - Bottom
Border Bottom
Margin Bottom

Box model properties

- 1. Content
- The actual area where text or images are displayed.
- Controlled by: width, height
- 2. Padding
- Transparent space around content.
- Property: padding (can be top, right, bottom, left).
- 3. Border
- Surrounds the padding and content.
- Properties: border-width, border-style, border-color.
- 4. Margin
- Space outside the border.
- Property: margin (can be top, right, bottom, left).

Example code

```
<head>
  <style>
    .box {
      width: 200px;
      height: 100px;
      background-color: lightblue; /* Content area */
      padding: 20px;              /* Space inside */
      border: 5px solid blue;     /* Border */
      margin: 15px;               /* Space outside */
    }
  </style>
</head>
<body>
  <div class="box">This is a box model example</div>
</body>
```

Exercise

- Exercise 1 – Change padding:
- Start with the example box.
- Increase padding to 50px and observe changes.
- Exercise 2 – Add margin:
- Add margin: 50px; and check distance from page edge.
- Exercise 3 – Modify border:
- Change border to 10px dashed red.
- Exercise 4 – Try box-sizing:
- Apply box-sizing: border-box; and see how total width changes.
- Exercise 5 – Multiple boxes:
- Create 3 boxes with different margins and borders.
- Align them and note how spacing differs.