

Introduction to IT System

2nd Semester

UNIT 3: CSS (Cascading Style Sheets)



Unit – 3: CSS (Cascading Style Sheets)

CSS stands for Cascading Style Sheets. It is used to style and design HTML web pages. CSS controls the visual layout, colors, fonts, spacing, and more. Without CSS, a webpage would look plain and unstyled.

1. CSS Introduction

CSS (Cascading Style Sheets) is a stylesheet language used to describe the presentation (look and formatting) of a document written in HTML. CSS separates content (HTML) from design (CSS), making web pages easier to maintain.

Why Use CSS?

- Saves time – write once, apply to multiple pages
- Easy maintenance – change one file to update style everywhere
- Better control over layout and design
- Makes websites responsive and attractive

CSS Versions

Version	Description
CSS 1	Basic styling (1996)
CSS 2	Added positioning, z-index (1998)
CSS 3	Modern features: gradients, animations, flexbox (current)

2. CSS Syntax

A CSS rule consists of a selector and a declaration block. The selector points to the HTML element you want to style. The declaration block contains one or more declarations separated by semicolons.

Syntax:

```
selector {  
  property: value;  
  property: value;  
}
```

Example – Change paragraph text color and size:

INPUT:

```
/* HTML */  
<p>Welcome to CSS!</p>  
  
/* CSS */  
p {  
  color: blue;  
  font-size: 18px;  
}
```

OUTPUT:

The paragraph text "Welcome to CSS!" will appear in BLUE color with font size 18 pixels.

3. CSS Id & Class

CSS Class Selector

A class selector selects HTML elements with a specific class attribute. Use a dot (.) before the class name. Multiple elements can share the same class.

```
/* CSS */
.highlight {
  background-color: yellow;
  color: black;
}

/* HTML */
<p class="highlight">This text is highlighted.</p>
```

OUTPUT:

The paragraph will have a YELLOW background with BLACK text.

CSS ID Selector

An ID selector selects a single unique element. Use a hash (#) before the ID name. Only ONE element per page should use the same ID.

```
/* CSS */
#main-title {
  color: red;
  text-align: center;
}

/* HTML */
<h1 id="main-title">Hello World</h1>
```

OUTPUT:

The heading "Hello World" will appear in RED and centered on the page.

KEY DIFFERENCE: Class (.) = can be used on MANY elements. ID (#) = used on ONE unique element.

4. How to Apply CSS (Three Methods)

4.1 Inline CSS

Inline CSS is written directly inside an HTML element using the style attribute. It applies style to that single element only.

```
<!-- INPUT -->
<p style="color: green; font-size: 16px;">This is inline CSS.</p>

<!-- OUTPUT -->
Text appears in GREEN with 16px font size.
```

4.2 Internal CSS

Internal CSS is written inside a `<style>` tag in the `<head>` section of the HTML file. It applies to all elements on that page.

```
<!-- INPUT -->
<head>
  <style>
    body {
      background-color: lightblue;
    }
    h1 {
      color: navy;
    }
  </style>
</head>
<body>
  <h1>Hello!</h1>
</body>

<!-- OUTPUT -->
Page background = light blue. Heading = navy color.
```

4.3 External CSS

External CSS is written in a separate `.css` file and linked to the HTML file using the `<link>` tag. It is the **BEST** and most recommended method — one CSS file can style multiple HTML pages.

```
<!-- HTML File (index.html) -->
<link rel="stylesheet" href="styles.css">

/* CSS File (styles.css) */
p {
  font-family: Arial;
  color: darkgreen;
}

/* OUTPUT: All paragraphs use Arial font in dark green. */
```

Method	Best Used When
Inline CSS	Quick one-off styles on a single element
Internal CSS	Styling one specific page
External CSS	Styling multiple pages (recommended)

5. Styling Backgrounds

CSS provides several properties to style the background of elements.

Property	Description
<code>background-color</code>	Sets background color
<code>background-image</code>	Sets background image

background-repeat	Repeats or not the image (repeat/no-repeat)
background-position	Positions the background image
background-size	Sets size of background image

INPUT:

```
body {
  background-color: #f0e68c;      /* light yellow */
  background-image: url("bg.png");
  background-repeat: no-repeat;
  background-position: center top;
  background-size: cover;
}
```

OUTPUT:

The page background becomes light yellow with a non-repeating image centered at the top, covering the full page.

6. Styling Text

Property	Description
color	Sets text color
text-align	Aligns text (left/right/center/justify)
text-decoration	Underline, overline, line-through
text-transform	Uppercase, lowercase, capitalize
text-indent	Indents the first line
letter-spacing	Space between letters
line-height	Space between lines
word-spacing	Space between words

INPUT:

```
/* HTML */
<p class="styled-text">CSS makes text beautiful!</p>

/* CSS */
.styled-text {
  color: #333333;
  text-align: center;
  text-transform: uppercase;
  text-decoration: underline;
  letter-spacing: 2px;
  line-height: 1.8;
}
```

OUTPUT:

Text shows as "CSS MAKES TEXT BEAUTIFUL!" in uppercase, centered, underlined, with 2px letter spacing.

7. Styling Fonts

Property	Description
font-family	Sets the font (e.g., Arial, Times New Roman)
font-size	Sets size (px, em, rem, %)
font-weight	bold, normal, 100–900
font-style	normal, italic, oblique
font-variant	small-caps

INPUT:

```
/* HTML */
<h2>Font Styling Example</h2>

/* CSS */
h2 {
  font-family: "Georgia", serif;
  font-size: 24px;
  font-weight: bold;
  font-style: italic;
  color: #2c3e50;
}
```

OUTPUT:

Heading shows in Georgia font, size 24px, bold and italic, in dark navy color.

8. Styling Links

CSS link states control how links look in different situations using pseudo-classes.

Pseudo-class	Description
a:link	Unvisited link
a:visited	Visited link
a:hover	Mouse over link
a:active	Clicked link

INPUT:

```
/* CSS */
a:link { color: blue; text-decoration: none; }
a:visited { color: purple; }
a:hover { color: red; text-decoration: underline; }
a:active { color: orange; }

/* HTML */
<a href="https://example.com">Visit Example</a>
```

OUTPUT:

Normal link = blue (no underline). Visited = purple. On hover = red with underline. On click = orange.

9. Styling Lists

Property	Description
list-style-type	disc, circle, square, decimal, none
list-style-image	Custom image as bullet
list-style-position	inside or outside

INPUT:

```
/* HTML */
<ul class='mylist'>
  <li>HTML</li>
  <li>CSS</li>
  <li>JavaScript</li>
</ul>

/* CSS */
.mylist {
  list-style-type: square;
  color: #2c3e50;
  font-size: 16px;
  line-height: 2;
}
```

OUTPUT:

A list with square bullets, dark text, 16px size, and double line spacing.

10. Styling Tables

Property	Description
border	Sets table borders
border-collapse	Collapse = single border between cells
width/height	Controls table dimensions
padding	Space inside cells
text-align	Aligns text in cells
background-color	Colors rows/cells
:nth-child(even)	Alternating row color (striped table)

INPUT:

```
/* CSS */
table {
  width: 100%;
  border-collapse: collapse;
}
th, td {
  border: 1px solid #ccc;
  padding: 10px;
}
```

```

    text-align: left;
  }
  th {
    background-color: #2E75B6;
    color: white;
  }
  tr:nth-child(even) {
    background-color: #f2f2f2;
  }

```

OUTPUT:

A styled table with blue header, 1px grey borders, 10px padding, and alternating grey/white row colors.

11. CSS Box Model

Every HTML element is a box. The CSS Box Model describes the layers around content: Content → Padding → Border → Margin.

```

[ MARGIN ]
[ BORDER ]
[ PADDING ]
[ CONTENT ]

```

Layer	Description
Content	The actual text or image inside the element
Padding	Space between content and the border (inside)
Border	The line surrounding the padding
Margin	Space outside the border (between elements)

INPUT:

```

/* CSS */
div {
  width: 300px;
  padding: 20px;
  border: 5px solid navy;
  margin: 30px;
  background-color: lightyellow;
}

/* HTML */
<div>Box Model Demo</div>

```

OUTPUT:

A 300px wide box with 20px inner padding, 5px navy border, and 30px outer space. Total rendered width = 300+40+10+60 = 410px.

12. CSS Border

Property	Values / Description
----------	----------------------

border-style	solid, dotted, dashed, double, groove, ridge, none
border-width	px, em, thick, thin
border-color	Any CSS color
border-radius	Rounds corners (px or %)
border (shorthand)	width style color in one line

INPUT:

```
.box {
  border: 3px dashed #e74c3c; /* 3px dashed red border */
  border-radius: 10px;      /* rounded corners */
  padding: 15px;
}
```

OUTPUT:

A box with a 3px dashed red border and nicely rounded corners.

13. CSS Outline

An outline is a line drawn OUTSIDE the border. Unlike border, it does NOT affect layout (no space taken). Useful for highlighting elements (e.g., form inputs on focus).

```
input:focus {
  outline: 2px solid blue;
  outline-offset: 4px; /* gap between border and outline */
}
```

OUTPUT:

When a user clicks an input box, a blue outline appears 4px outside its border.

14. CSS Margin

Margin creates space OUTSIDE the element border, pushing other elements away.

```
/* Individual sides */
p {
  margin-top: 20px;
  margin-right: 30px;
  margin-bottom: 20px;
  margin-left: 30px;
}

/* Shorthand: top right bottom left */
p { margin: 20px 30px 20px 30px; }

/* Center an element horizontally */
div { width: 400px; margin: 0 auto; }
```

OUTPUT:

Paragraphs get 20px space top/bottom and 30px left/right. Using 'margin: 0 auto' on a div with a set width centers it on the page.

15. CSS Display

Value	Description
block	Takes full width, starts on new line (div, p, h1...)
inline	Takes only needed width, no new line (span, a, img...)
inline-block	Like inline but allows width/height
none	Element is hidden (removed from layout)
flex	Flexible layout (modern)
grid	Grid layout (modern)

INPUT:

```
/* HTML */
<span class="box">Item 1</span>
<span class="box">Item 2</span>

/* CSS */
.box {
  display: inline-block;
  width: 120px;
  height: 40px;
  background: #3498db;
  color: white;
  text-align: center;
  line-height: 40px;
}
```

OUTPUT:

Two blue boxes appear side by side (inline), each 120px wide and 40px tall with white centered text.

16. CSS Positioning

Value	Description
static	Default. Elements flow normally.
relative	Positioned relative to its normal position.
absolute	Positioned relative to nearest positioned parent.
fixed	Fixed to the browser window. Stays on scroll.
sticky	Sticks at a position when scrolling.

INPUT:

```
/* Fixed header that stays at top while scrolling */
header {
  position: fixed;
  top: 0;
  left: 0;
  width: 100%;
}
```

```
background: #2c3e50;
color: white;
padding: 10px;
}
```

OUTPUT:

A dark header stays fixed at the top of the page even when the user scrolls down.

17. CSS Floating

The float property places an element to the left or right, allowing other content to wrap around it. Used for image-text wrapping and layouts.

```
/* Image floated to the left, text wraps right */
img {
  float: left;
  margin-right: 15px;
  width: 150px;
}

/* Clear float so next element doesn't wrap */
.clearfix {
  clear: both;
}
```

OUTPUT:

Image appears on the left side; paragraph text wraps around the right of it. Adding clear:both after ensures the next section starts below.

18. CSS Navigation Bar

A navigation bar is a list of links styled with CSS. Use and elements, remove list styling, and use display:inline-block or flexbox.

INPUT:

```
/* HTML */
<nav>
  <ul>
    <li><a href="#">Home</a></li>
    <li><a href="#">About</a></li>
    <li><a href="#">Contact</a></li>
  </ul>
</nav>

/* CSS */
ul { list-style: none; margin: 0; padding: 0;
  background-color: #2c3e50; overflow: hidden; }
li { display: inline; }
li a { display: inline-block; color: white; padding: 14px 20px;
  text-decoration: none; }
li a:hover { background-color: #e74c3c; }
```

OUTPUT:

A horizontal dark navigation bar with white links. Each link turns red on mouse hover.

19. CSS Image Gallery

An image gallery shows multiple images in a grid layout using CSS float or flexbox.

```
/* HTML */
<div class="gallery">
  
  
  
</div>

/* CSS */
.gallery {
  display: flex;
  flex-wrap: wrap;
  gap: 10px;
  justify-content: center;
}
.gallery img {
  width: 200px;
  height: 150px;
  object-fit: cover;
  border: 3px solid #ddd;
  border-radius: 5px;
}
.gallery img:hover {
  transform: scale(1.05);
  border-color: #2E75B6;
}
```

OUTPUT:

Three images appear side by side in a flex row, each 200x150px with grey borders. On hover, an image slightly zooms in and border turns blue.

20. CSS Image Opacity

The opacity property controls transparency. Value ranges from 0 (fully transparent) to 1 (fully visible).

```
/* Normal image */
img { opacity: 1; }

/* Semi-transparent */
img { opacity: 0.5; }

/* Hover effect: transparent then full */
img {
  opacity: 0.6;
```

```
    transition: opacity 0.3s ease;
  }
  img:hover {
    opacity: 1.0;
  }
}
```

OUTPUT:

Images appear at 60% opacity by default. On mouse hover, they smoothly fade to full visibility over 0.3 seconds.

21. CSS Align

Horizontal Alignment

Method	Usage
text-align: center	Centers inline content (text, inline elements)
margin: 0 auto	Centers a block element (must have set width)
display: flex; justify-content: center	Centers flex children horizontally

Vertical Alignment

Method	Usage
line-height = height	Vertically centers single-line text in a box
display: flex; align-items: center	Centers flex children vertically
vertical-align: middle	For inline/table-cell elements

INPUT – Center a div horizontally and vertically:

```
/* HTML */
<div class="center-box">Centered!</div>

/* CSS */
body {
  display: flex;
  justify-content: center; /* horizontal */
  align-items: center; /* vertical */
  height: 100vh;
}
.center-box {
  background: #3498db;
  color: white;
  padding: 20px 40px;
  border-radius: 8px;
}
```

OUTPUT:

A blue button-like box labeled "Centered!" appears perfectly centered in the middle of the page both horizontally and vertically.

IMPORTANT FOR EXAM

- ★ CSS stands for Cascading Style Sheets
- ★ Three ways to apply CSS: Inline, Internal, External
- ★ CSS Syntax: selector { property: value; }
- ★ Class selector uses DOT (.) | ID selector uses HASH (#)
- ★ Box Model: Content → Padding → Border → Margin
- ★ position: fixed — stays on screen during scroll
- ★ float: left/right — wraps text around elements
- ★ display: none — hides element completely
- ★ opacity: 0 to 1 (0=invisible, 1=fully visible)
- ★ margin: 0 auto — centers a block element

Common Viva Questions

1. What does CSS stand for?

CSS stands for Cascading Style Sheets.

2. What are the three types of CSS?

Inline CSS, Internal CSS, and External CSS.

3. What is the difference between Class and ID in CSS?

Class (.) can be applied to multiple elements; ID (#) is unique to one element per page.

4. What is the CSS Box Model?

It is the layered model around every element: Content, Padding, Border, and Margin.

5. What is the difference between border and outline?

Border takes up space and affects layout; Outline is drawn outside the border and does NOT affect layout.

6. What does 'float' do in CSS?

Float moves an element to the left or right so other content can wrap around it.

7. How do you center a block element in CSS?

By setting margin: 0 auto and giving the element a defined width.

8. What is the opacity property?

opacity controls transparency: 0 = invisible, 1 = fully visible.

9. What is the difference between display:none and visibility:hidden?

display:none removes element from layout; visibility:hidden hides it but keeps its space.

10. What does position:fixed do?

The element stays fixed in the browser window and does not move when the user scrolls.

Quick Revision Points

CSS = Cascading Style Sheets | Adds design to HTML
syntax: selector { property: value; }
Inline = style attribute | Internal = <style> tag | External = .css file
Class = . (many elements) | ID = # (one element)

Box Model: Content > Padding > Border > Margin
border-radius = rounded corners
text-align: center/left/right/justify
float: left/right | clear: both
position: static/relative/absolute/fixed/sticky
display: block/inline/inline-block/none/flex/grid
opacity: 0-1 | visibility: hidden/visible
margin: 0 auto = horizontal centering of block element
External CSS = most recommended method for large projects

End of Unit 3 – CSS Notes

