

Web Development

Introduction to CSS

Presented by Muhammad Munim

Introduction to CSS & Basic Styling

Learning Objectives

- By the end of this class, students will:
- Understand what CSS is and why it's important
- Learn different ways to apply CSS to HTML
- Understand CSS syntax and selectors
- Use basic text and color styling
- Practice styling a simple webpage layout



[linkedin.com/in/m-munim1](https://www.linkedin.com/in/m-munim1)



github.com/M-Munim



+92 330 2021926

What is CSS?

- CSS = **Cascading** Style Sheets
- Controls the look & layout of HTML elements
- Separates content (HTML) from presentation (CSS)

HTML is for structure → CSS is for style



[linkedin.com/in/m-munim1](https://www.linkedin.com/in/m-munim1)



github.com/M-Munim



[+92 330 2021926](https://wa.me/+923302021926)

Ways to Apply CSS

1. Inline CSS (not recommended)

```
<p style="color: red;">This is red text.</p>
```

2. Internal CSS (inside <style> </style> tag/element)

```
<head>
  <style>
    p { color: green; }
  </style>
</head>
```



[linkedin.com/in/m-munim1](https://www.linkedin.com/in/m-munim1)



github.com/M-Munim



+92 330 2021926

3. External CSS (best practice)

```
<link rel="stylesheet" href="styles.css">
```

- Use external CSS for large projects

CSS Syntax

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

Example:

```
h1 {  
    color: blue;  
    font-size: 24px;  
}
```

- **Selector:** targets the HTML element (like h1, p, .class, #id)
- **Property:** the style you want to change
- **Value:** the setting for the property



[linkedin.com/in/m-munim1](https://www.linkedin.com/in/m-munim1)



github.com/M-Munim



+92 330 2021926

Selectors in CSS

Selector	Example	Description
Type	p	Targets all <p> elements
Class	.box	Targets elements with class
ID	#main	Targets element with specific id
Grouping	h1, p	Targets multiple elements

Example:

```
<div class="box" id="main">Hello</div>
```

```
.box { color: red; }
#main { background: yellow; }
```



[linkedin.com/in/m-munim1](https://www.linkedin.com/in/m-munim1)



github.com/M-Munim



+92 330 2021926

Colors in CSS

Color Methods:

- Named colors: red, green, blue
- HEX: #ff5733
- RGB: rgb(255, 87, 51)
- RGBA: rgba(255, 87, 51, 0.7)

Example:

```
body {  
    background-color: #f4f4f4;  
    color: #333;  
}
```



linkedin.com/in/m-munim1



github.com/M-Munim



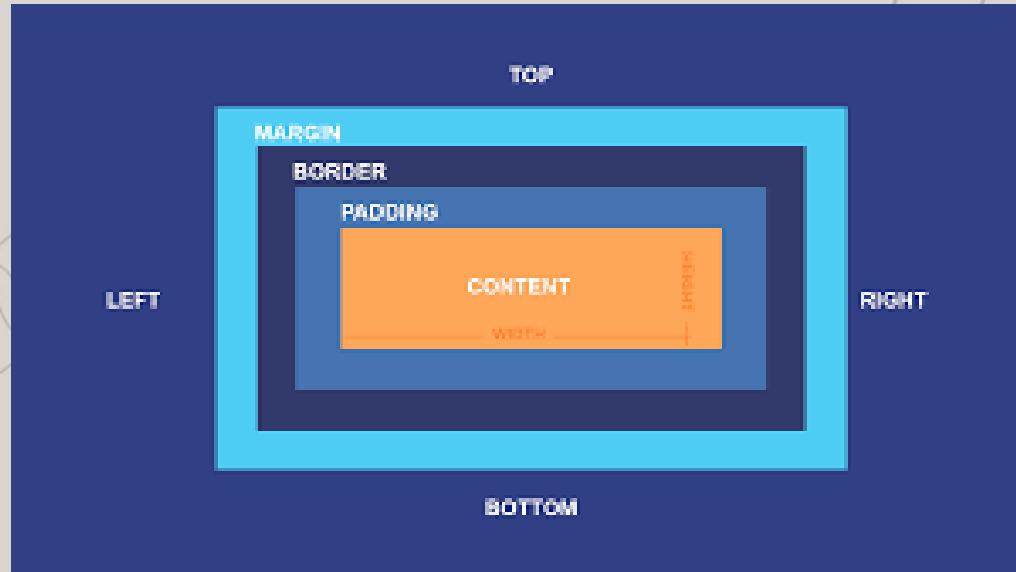
+92 330 2021926

Text Styling in CSS

Property	Example Value
color	red, #123456
font-size	16px, 2em, 1rem
font-family	'Arial', sans-serif
font-weight	normal, bold, 500
text-align	left, right, center
text-transform	uppercase, capitalize
line-height	1.5, 2



The Box Model (Intro)



- CSS Box Model = content + padding + border + margin
- We'll explore this in depth in Class 2.

Hands-On Example (Live Code)

```
<!DOCTYPE html>
<html>
<head>
    paste style here
</head>
<body>
    <h1>Hello, CSS!</h1>
    <p>This is styled with CSS.</p>
</body>
</html>
```

```
<style>
    h1 {
        color: navy;
        text-align: center;
    }
    p {
        font-family: Arial;
        font-size: 18px;
    }
</style>
```



Home Tasks for CSS Class 1

Create an HTML page with:

- 2 headings (h1, h2)
- 3 paragraphs of text

Create an external CSS file:

- Set background color of body
- Style headings with different colors & fonts
- Center one heading using text-align
- Use font-size, line-height for paragraphs

Try all 3 ways of writing CSS (inline, internal, external)



linkedin.com/in/m-munim1



github.com/M-Munim



[+92 330 2021926](https://wa.me/+923302021926)

Resources

Videos:

- CSS Crash Course – Traversy Media
- CSS Selectors Explained – Kevin Powell

Practice:

- CSS Diner Game – Learn Selectors
- CodePen – Try CSS Live



linkedin.com/in/m-munim1



github.com/M-Munim



[+92 330 2021926](https://wa.me/+923302021926)

Class: 02



[linkedin.com/in/m-munim1](https://www.linkedin.com/in/m-munim1)



github.com/M-Munim



[+92 330 2021926](https://wa.me/+923302021926)

Box Model, Spacing, Borders, Backgrounds, and Display

Learning Objectives

By the end of this class, students will:

- Master the Box Model: understand how every element is structured
- Apply and control spacing using margin and padding
- Add and customize borders around elements
- Style beautiful backgrounds with colors, gradients, and images
- Control element flow using the display property



Deep Dive into the Box Model

What is the Box Model?

Every HTML element is a rectangular box made up of:

- Content – The text or element itself
- Padding – Space inside the border, around content
- Border – A visible line around the padding
- Margin – Space outside the border (distance from other elements)

```
.box {  
    padding: 20px;  
    border: 2px solid black;  
    margin: 30px;  
}
```



linkedin.com/in/m-munim1



github.com/M-Munim



+92 330 2021926

Visual Representation:

[margin]

[border]

[padding]

[content]

All spacing (padding, margin) can be defined individually: top, right, bottom, left



[linkedin.com/in/m-munim1](https://www.linkedin.com/in/m-munim1)



github.com/M-Munim



[+92 330 2021926](https://wa.me/+923302021926)

Padding and Margin

- padding – Space between content and border

```
padding: 20px;      /* all sides */  
padding: 10px 20px; /* top-bottom, left-right */  
padding: 10px 15px 20px 25px; /* top, right, bottom, left */
```

- margin – Space outside the element

```
margin: 0 auto; /* center horizontally */
```

- Negative margins are allowed, but should be used carefully



[linkedin.com/in/m-munim1](https://www.linkedin.com/in/m-munim1)



github.com/M-Munim



+92 330 2021926

Border Styling

```
border: 3px solid red;
```

Breakdown:

- Width: 1px, 5px
- Style: solid, dashed, dotted, double, ridge, none
- Color: Any color value

Individual sides:

```
border-top: 2px solid blue;  
border-left: none;
```



[linkedin.com/in/m-munim1](https://www.linkedin.com/in/m-munim1)



github.com/M-Munim



+92 330 2021926

Border radius (rounded corners):

```
border-radius: 10px;
```

- Can be circular, elliptical
- Use percentages for perfect circles

Backgrounds – Color, Image, Gradient

- background-color

```
background-color: #f0f0f0;
```



[linkedin.com/in/m-munim1](https://www.linkedin.com/in/m-munim1)



github.com/M-Munim



[+92 330 2021926](https://wa.me/+923302021926)

Web Development

- background-image

```
background-image: url("pattern.jpg");  
background-repeat: no-repeat;  
background-size: cover;  
background-position: center;
```

- background-gradient

```
background: linear-gradient(to right, #ff6a00, #f9d423);
```

Use gradients for hero sections, CTA buttons



linkedin.com/in/m-munim1



github.com/M-Munim



+92 330 2021926

Width, Height & Overflow

```
width: 300px;  
height: 200px;
```

- Use %, px, em, rem, vw, vh
- Auto vs fixed size

Overflow Control:

```
overflow: auto; /* scrolls when content exceeds */  
overflow: hidden; /* cuts off content */  
overflow: scroll; /* always scrollbars */
```



[linkedin.com/in/m-munim1](https://www.linkedin.com/in/m-munim1)



github.com/M-Munim



+92 330 2021926

display Property – Block vs Inline vs Inline-block

block (default for div, p, h1)

- Starts on a new line
- Takes full width available

inline (default for span, a)

- Flows within the line
- Cannot set width/height

inline-block

- Like inline, but you can set width and height

Example:

```
.box {  
    display: inline-block;  
    width: 100px;  
    height: 100px;  
}
```



linkedin.com/in/m-munim1



github.com/M-Munim



+92 330 2021926

display: none vs visibility: hidden

- display: none – Hides element and removes it from layout
- visibility: hidden – Hides element but still takes up space

box-sizing Property

```
box-sizing: border-box;
```

- Makes element's width include padding and border
- Helps avoid layout issues when combining box model layers

Always use it in resets:

```
* { box-sizing: border-box; }
```



[linkedin.com/in/m-munim1](https://www.linkedin.com/in/m-munim1)



github.com/M-Munim



+92 330 2021926

Hands-On Code Demo

Let students build a styled card using:

- Margin, padding, border-radius
- Background color or image
- Centered box using margin: auto
- Text inside a padded div

```
<div class="card">  
  <h2>Learn CSS</h2>  
  <p>Style your website beautifully!</p>  
</div>
```

```
.card {  
  width: 300px;  
  margin: 30px auto;  
  padding: 20px;  
  border: 2px solid #444;  
  border-radius: 10px;  
  background: #f7f7f7;  
}
```



[linkedin.com/in/m-munim1](https://www.linkedin.com/in/m-munim1)



github.com/M-Munim



+92 330 2021926

Home Tasks for CSS Class 2

1. Create a box card layout with:
 - Fixed width/height
 - Padding & border
 - Rounded corners

2. Add a background image to your entire page
 - Use cover and center for styling

3. Create 3 boxes side-by-side using inline-block
 - Each with different background gradients

4. Experiment with:
 - overflow: hidden inside a fixed-height box
 - box-sizing: border-box in your CSS reset



[linkedin.com/in/m-munim1](https://www.linkedin.com/in/m-munim1)



github.com/M-Munim



+92 330 2021926

Resources

Videos:

- Box Model in CSS – Kevin Powell
- CSS Display Explained – Web Dev Simplified

Practice Tools:

- Flexbox Froggy Game (for next class)
- CSS Gradient Generator
- CSS Box Model Visual Tool



[linkedin.com/in/m-munim1](https://www.linkedin.com/in/m-munim1)



github.com/M-Munim



[+92 330 2021926](https://wa.me/+923302021926)

Class: 03



[linkedin.com/in/m-munim1](https://www.linkedin.com/in/m-munim1)



github.com/M-Munim



[+92 330 2021926](https://wa.me/+923302021926)

Flexbox, Positioning, and Advanced Layouts

Learning Objectives

- By the end of this class, students will:
- Understand Flexbox and why it's used for modern layouts
- Apply Flexbox properties for alignment, spacing, and direction
- Learn positioning types (static, relative, absolute, fixed, sticky)
- Combine layout techniques for building real UI components
- Create responsive and adaptive page sections



[linkedin.com/in/m-munim1](https://www.linkedin.com/in/m-munim1)



github.com/M-Munim



[+92 330 2021926](https://wa.me/+923302021926)

Intro to Flexbox

Why Flexbox?

- Makes it easier to align, center, and distribute space
- Ideal for 1D layouts (row or column)

Apply **display: flex** to a parent container to activate Flexbox on its children.

```
.container {  
  display: flex;  
}
```



linkedin.com/in/m-munim1



github.com/M-Munim



+92 330 2021926

Flex Direction

```
flex-direction: row;      /* default */  
flex-direction: row-reverse;  
flex-direction: column;  
flex-direction: column-reverse;
```

Use this to control the main axis (horizontal or vertical layout)

- Row = horizontal (left to right)
- Column = vertical (top to bottom)



[linkedin.com/in/m-munim1](https://www.linkedin.com/in/m-munim1)



github.com/M-Munim



[+92 330 2021926](https://wa.me/+923302021926)

Justify Content (Main Axis)

Used for spacing items **horizontally** (in row direction):

```
justify-content: flex-start;  
justify-content: center;  
justify-content: space-between;  
justify-content: space-around;  
justify-content: space-evenly;
```



linkedin.com/in/m-munim1



github.com/M-Munim



+92 330 2021926

Align Items (Cross Axis)

Used to align items vertically (when flex-direction: row):

```
align-items: flex-start;  
align-items: center;  
align-items: flex-end;  
align-items: stretch;
```

stretch (default) makes items fill the height of the container



linkedin.com/in/m-munim1



github.com/M-Munim



+92 330 2021926

Flex Wrap & Gap

- flex-wrap

```
flex-wrap: wrap;
```

Allows items to break to the next line if they overflow

- gap

```
gap: 20px;
```

Adds space between Flexbox children



[linkedin.com/in/m-munim1](https://www.linkedin.com/in/m-munim1)



github.com/M-Munim



+92 330 2021926

Positioning in CSS

- position values:

Value	Description
static	Default, normal flow
relative	Moves relative to its normal spot
absolute	Positioned relative to nearest positioned ancestor
fixed	Stuck to viewport (even on scroll)
sticky	Scrolls until a threshold, then sticks



Example:

```
.box {  
    position: absolute;  
    top: 20px;  
    left: 50px;  
}
```

2021926



[linkedin.com/in/m-munim1](https://www.linkedin.com/in/m-munim1)



github.com/M-Munim



+92 330 2021926

Z-Index

Controls stack order (which element appears “on top”)

```
.modal {  
  position: absolute;  
  z-index: 1000;  
}
```

Only works on positioned elements (relative, absolute, fixed)



linkedin.com/in/m-munim1



github.com/M-Munim



+92 330 2021926

Combining Flex + Position (Real Layout)

Build a layout:

- Flexbox for navigation bar
- Fixed button on bottom-right (position: fixed)
- Sticky header
- Responsive box rows with flex-wrap

Hands-On Layout Challenge

Build this layout:

- A nav bar with items spaced using justify-content: space-between
- A hero section with text aligned center using align-items: center
- A floating "Back to Top" button in bottom-right
- A responsive grid using:

```
flex-wrap: wrap;  
width: 100%;  
max-width: 1200px;  
margin: auto;
```



linkedin.com/in/m-munim1



github.com/M-Munim



+92 330 2021926

Home Tasks for CSS Class 3

Task 1: Flex Card Layout

- Create a section with 3 boxes side-by-side
- Use Flexbox for layout and spacing
- Make it wrap on small screens

Task 2: Sticky Nav Bar

- Create a sticky navigation bar using position: sticky

Task 3: Overlay Layout

Create a box with:

- A background image
- An absolutely positioned <h1> on top
- A z-index overlay with semi-transparent background



[linkedin.com/in/m-munim1](https://www.linkedin.com/in/m-munim1)



github.com/M-Munim



+92 330 2021926

Resources

Videos:

- Flexbox in 100 Seconds – Fireship
- Flexbox Full Course – freeCodeCamp

Interactive Games:

- Flexbox Froggy (Game)
- Flexbox Zombies (Interactive)

Docs & Demos:

- MDN Flexbox Guide
- CSS Tricks Flexbox Cheatsheet



linkedin.com/in/m-munim1



github.com/M-Munim



[+92 330 2021926](https://wa.me/+923302021926)

Class: 04



[linkedin.com/in/m-munim1](https://www.linkedin.com/in/m-munim1)



github.com/M-Munim



[+92 330 2021926](https://wa.me/+923302021926)

Responsive Design, Transitions, Animations, Pseudo-elements

Learning Objectives

By the end of this class, students will:

- Create responsive layouts using media queries
- Add animations and transitions to elements
- Use pseudo-classes and pseudo-elements for enhanced UI interactions
- Apply shadows and hover effects
- Follow CSS organization and naming conventions



linkedin.com/in/m-munim1



github.com/M-Munim



+92 330 2021926

Responsive Design with Media Queries

Why Media Queries?

Make your website look good on all screen sizes (mobile, tablet, desktop).

Syntax:

```
@media (max-width: 768px) {  
  body {  
    background-color: lightblue;  
  }  
}
```



[linkedin.com/in/m-munim1](https://www.linkedin.com/in/m-munim1)



github.com/M-Munim



+92 330 2021926

Common Breakpoints:

```
/* Small phones */  
@media (max-width: 480px) { ... }  
  
/* Tablets */  
@media (max-width: 768px) { ... }  
  
/* Laptops */  
@media (max-width: 1024px) { ... }
```

Combine with flex-wrap and % widths for best results



linkedin.com/in/m-munim1



github.com/M-Munim



+92 330 2021926

Transitions – Smooth Changes

Syntax:

```
.button {  
    background: blue;  
    transition: background 0.3s ease-in-out;  
}  
.button:hover {  
    background: darkblue;  
}
```

- transition-property
- transition-duration
- transition-timing-function
- transition-delay

Use for hover effects, focus, theme changes, etc.



[linkedin.com/in/m-munim1](https://www.linkedin.com/in/m-munim1)



github.com/M-Munim



+92 330 2021926

Animations in CSS

Keyframes:

```
@keyframes slideIn {  
 0% { transform: translateX(-100%); }  
 100% { transform: translateX(0); }  
}
```

Apply Animation:

```
.box {  
  animation: slideIn 1s ease-out forwards;  
}
```

Add effects to banners, modals, loaders, entrances



[linkedin.com/in/m-munim1](https://www.linkedin.com/in/m-munim1)



github.com/M-Munim



+92 330 2021926

Pseudo-Classes – Interactivity without JS

Selector

:hover

:focus

:active

:nth-child()

:not()

Description

On mouse hover

When element is focused (e.g. input)

When element is clicked

Targets specific children

Targets everything except



[linkedin.com/in/m-munim1](https://www.linkedin.com/in/m-munim1)



github.com/M-Munim



[+92 330 2021926](https://wa.me/+923302021926)

Example:

```
button:hover {  
    background: green;  
}  
input:focus {  
    border-color: blue;  
}  
li:nth-child(odd) {  
    background: #eee;  
}
```



[linkedin.com/in/m-munim1](https://www.linkedin.com/in/m-munim1)



github.com/M-Munim



+92 330 2021926

Pseudo-Elements – Style Parts of Elements

Selector

::before

::after

::first-letter

::first-line

::selection

Purpose

Adds content before element

Adds content after element

Styles first letter

Styles first line

Styles highlighted text



[linkedin.com/in/m-munim1](https://www.linkedin.com/in/m-munim1)



github.com/M-Munim



[+92 330 2021926](https://wa.me/+923302021926)

Example:

```
.card::before {  
    content: "💡 ";  
}  
  
p::first-letter {  
    font-size: 200%;  
    color: red;  
}
```



[linkedin.com/in/m-munim1](https://www.linkedin.com/in/m-munim1)



github.com/M-Munim



+92 330 2021926

Box Shadows & Text Shadows

Box Shadow:

```
.card {  
  box-shadow: 0 4px 8px rgba(0,0,0,0.2);  
}
```

Text Shadow:

```
h1 {  
  text-shadow: 2px 2px 5px rgba(0,0,0,0.3);  
}
```

Combine with transitions for hover effects



[linkedin.com/in/m-munim1](https://www.linkedin.com/in/m-munim1)



github.com/M-Munim



+92 330 2021926

CSS Naming Conventions & Clean Code

Best Practices:

- Use BEM naming (block__element--modifier)
- Group related styles together
- Keep units consistent
- Avoid !important unless absolutely necessary



linkedin.com/in/m-munim1



github.com/M-Munim



+92 330 2021926

Home Tasks for CSS Class 4

Task 1: Responsive Card Layout

- Create a layout with 3 cards in a row on desktop
- Cards stack vertically on screens < 768px

Task 2: Button Animation

- Animate a button using @keyframes
- Change color smoothly with :hover

Task 3: Media Query Experiment

- Create a section that changes background, font size, and padding based on screen size



linkedin.com/in/m-munim1



github.com/M-Munim



+92 330 2021926

Resources

Videos:

- Responsive CSS with Media Queries – Kevin Powell
- CSS Animations Explained – Web Dev Simplified

Tools & Practice:

- Animation Generator – Animista
- CSS Clip Path Generator
- Can I Use (browser support checker)



linkedin.com/in/m-munim1



github.com/M-Munim



+92 330 2021926

Web Development

Submission Notes

Capture Evidence:

- Take screenshots or record a short video for each task.
- Ensure your evidence clearly shows the completed task.

Upload Tasks:

- Upload each task as a separate folder or file in your GitHub repository.

LinkedIn Post:

- Create a short post summarizing what you learned and how you completed the tasks.
- Highlight your key takeaways.
- Mention the following account in your post:
- **Muhammad Munim**



linkedin.com/in/m-munim1



github.com/M-Munim



[+92 330 2021926](https://wa.me/+923302021926)

Thank You



[linkedin.com/in/m-munim1](https://www.linkedin.com/in/m-munim1)



github.com/M-Munim



+92 330 2021926