Class 3: Styling page with Basic CSS

SESSION OVERVIEW

By the end of this session, students will be able to:

- Understand what CSS is and its role in styling HTML content.
- Apply basic CSS properties to enhance the visual appearance of web pages.
- Use color, text styling, spacing, and borders to create structured and appealing designs.
- Style and position images effectively using CSS.
- Understand and apply the CSS Box Model to manage element dimensions, padding, borders, and margins.
- Develop a simple event invitation page using fundamental CSS techniques.
- Build a strong foundation in CSS for more advanced styling in web development.

Getting started with basic:

1. What is CSS?

Cascading Style Sheets (CSS) is a styling language used to control the presentation of HTML elements. CSS separates the content (HTML) from the presentation (styles), making it easier to maintain, update, and reuse code. The "cascading" aspect means that styles can be applied in layers, with rules from different sources combining to affect the final appearance. Key benefits include:

- **Separation of Concerns:** Keep content and design separate.
- **Reusability:** A single CSS file can style multiple pages.
- **Efficiency:** Faster page load times and easier maintenance.

---

2. Text Styling in CSS

Text styling involves modifying the appearance of text elements such as paragraphs, headings, and spans. Here are some common properties:

- **Color and Background**
  - `color`: Sets the text color.
  - `background-color`: Sets the background color behind text.
- **Font Properties**
  - `font-family`: Specifies a list of fonts to use.
  - `font-size`: Defines the size of the text.

  - **`font-weight`**: Controls the boldness (e.g., `normal`, `bold`, or numeric values like `700`).
  - **`font-style`**: Sets the text to normal, italic, or oblique.
- **Text Layout**
  - **`text-align`**: Aligns text (left, right, center, justify).
  - **`line-height`**: Sets the space between lines of text.
  - **`letter-spacing`**: Adjusts the space between characters.
  - **`text-transform`**: Controls capitalization (uppercase, lowercase, capitalize).
- **Text Decoration**
  - **`text-decoration`**: Adds underlines, overlines, or line-through effects.

Example:
HTML

```
<p class="styled-text">Hello, CSS World!</p>
```

CSS:

```
.styled-text {
  color: #333;
  font-family: 'Arial', sans-serif;
  font-size: 1.2rem;
  font-weight: 600;
  text-align: center;
  line-height: 1.5;
  text-decoration: underline;
}
```

---

3. How to Add CSS to HTML

There are three primary ways to include CSS in your HTML documents:

a. Inline CSS

You apply styles directly to an HTML element using the **`style`** attribute.

HTML

```
<p style="color: blue; font-size: 16px;">This is inline styled text.</p>
```

b. Internal CSS

CSS rules are placed inside a **`<style>`** block within the **`<head>`** section of your HTML.

html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Internal CSS Example</title>
  <style>
    p {
      color: green;
      font-size: 18px;
    }
  </style>
</head>
<body>
  <p>This paragraph uses internal CSS.</p>
</body>
</html>
```

c. External CSS

You write your CSS in a separate file (e.g., **`styles.css`**) and link to it using the `<link>` tag in the **`<head>`** section.

HTML

```
<!-- index.html -->
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>External CSS Example</title>
  <link rel="stylesheet" href="styles.css">
</head>
<body>
  <p>This paragraph uses external CSS.</p>
</body>
</html>
```

The **`rel`** attribute tells the browser the relationship between the HTML file and the linked file—in this case, that the file is a stylesheet.

CSS

```
/* styles.css */
p {
  color: red;
  font-size: 20px;
}
```

---

4. ID and Class Selectors

CSS selectors are used to target HTML elements for styling.

a. Class Selector

- **Usage:** Applies styles to one or more elements.
- **Syntax:** Precede the class name with a period (`.`).

HTML

```
<p class="highlight">This paragraph is highlighted.</p>
```

CSS

```
.highlight {
  background-color: yellow;
}
```

b. ID Selector

- **Usage:** Targets a single, unique element on the page.
- **Syntax:** Precede the ID name with a hash (#).

HTML

```
<p id="unique">This paragraph has a unique style.</p>
```

CSS

```
#unique {
  border: 2px solid blue;
}
```

Note: IDs should be unique within a page, while classes can be reused across multiple elements.

---

5. CSS Units

CSS supports various units to define lengths, sizes, and spacing. Here are some common ones:

a. Pixels (px)

- **Definition:** A fixed unit representing a single dot on the screen.
- **Usage:** Best used when a fixed size is required.

CSS

```
.element {
  width: 200px;
}
```

b. Rem vs. Em

rem (Root EM)

- **Definition:** Relative to the root (HTML) element's **`font-size`.**

- **Usage:** Consistent scaling based on the root font size, making it easier to maintain across the document.

CSS

```
html {
  font-size: 16px; /* 1rem = 16px */
}
.element {
  font-size: 1.5rem; /* 24px */
}
```

em

- **Definition:** Relative to the font-size of its parent element.
- **Usage:** Useful when you want elements to scale based on their parent, but can lead to compounding effects if nested.

CSS

```
.parent {
  font-size: 16px;
}
.child {
  font-size: 1.5em; /* 24px relative to .parent's font size */
}
```

c. Viewport Height (vh)

- **Definition:**1vh equals 1% of the height of your browser's visible area.

  For example, if your browser window is 1000px tall, then 1vh is 10px.

- **Usage:** Useful for responsive designs where elements need to scale with the viewport.

CSS

```
.full-height {
  height: 100vh; /* Full height of the viewport */
}
```

6. Background Properties

CSS background properties allow you to control the appearance of an element's background.

a. Background Color

- **Property:** `background-color`
- **Usage:** Sets the background color of an element.

CSS

```
.box {
  background-color: #f0f0f0;
}
```

b. Background Image

- **Property:** `background-image`
- **Usage:** Sets an image as the background of an element.

CSS

```
.banner {
  background-image: url('banner.jpg');
}
```

c. Background Repeat

- **Property:** `background-repeat`
- **Usage:** Determines if/how the background image repeats.
    - `repeat` (default): Tiles the image both horizontally and vertically.
    - `repeat-x`: Tiles only horizontally.
    - `repeat-y`: Tiles only vertically.
    - `no-repeat`: Does not tile the image.

CSS

```css
.banner {
  background-repeat: no-repeat;
}
```

d. Background Position

- **Property:** `background-position`
- **Usage:** Specifies the starting position of a background image (e.g., `center`, `top right`, `50% 50%`).

CSS

```css
.banner {
  background-position: center;
}
```

e. Background Size

- **Property:** `background-size`
- **Usage:** Defines the size of the background image.
  - `cover`: Scales the image to cover the entire element.
  - `contain`: Scales the image to be fully visible inside the element.

CSS

```css
.banner {
  background-size: cover;
}
```

f. Shorthand for Background

- **Property:** `background`
- **Usage:** A shorthand property that can set multiple background properties at once.

CSS

```
.banner {
  background: #000 url('banner.jpg') no-repeat center/cover;
}
```

CSS Integration to Event Invitation Page:

Let's integrate CSS into an event invitation page to enhance its visual appeal and divide it into 4 sections: Body, Header, Main, Footer

Body Styling:

```
body {
   font-family: 'Arial', sans-serif;
   line-height: 1.6;
   background-color: #f0f4f8;
   color: #333;
   margin: 0;
   padding: 0;
}
```

Explanation:

font-family: Uses Arial as the primary font, with a generic sans-serif fallback.
line-height: Sets the spacing between lines to 1.6 times the font size for better readability.
background-color: Applies a light, subtle background color (#f0f4f8) to the entire page.
color: Sets the text color to a dark gray (#333) for clear visibility.
margin & padding: Resets default spacing around the page by setting both to 0.

Header Section:

```
h1, h2, h3, h4 {
   font-family: 'Georgia', serif;
   color: #2c3e50;
}
header {
   background : url('Place your Url here') ;
   color: white;
   text-align: center;
   background-size: cover;
   background-repeat: no-repeat;
   padding: 30px 0;

}

header h1 {
   font-size: 3em;
   margin: 0;
   text-transform: uppercase;
   font-weight: bold;
}

header p {
   font-size: 1.3em;
   margin-top: 10px;
}
```

Explanation:

`background: url('Place your Url here');`
Sets a background image for the header (replace the placeholder with an actual image URL).
`color: white;`
Applies white color to any text inside the header.
`text-align: center;`
Centers the text horizontally within the header.
`background-size: cover;`
Scales the background image to cover the entire header area, ensuring it fills the space without distortion.
`background-repeat: no-repeat;`
Prevents the background image from repeating.
`padding: 30px 0;`
Adds 30px of padding to the top and bottom of the header, with no padding on the left and right.

Main Section:

```
/* Main Section */
main {
   width: 80%;
   margin: 20px auto;
   padding: 30px;
   background-color: #fff;
   border-radius: 15px;

   background: linear-gradient(to bottom, #ffffff, #f2f4f7);
}

section {
   margin-bottom: 30px;
}

section img {
   width: 100%;
   max-width: 600px;
   height: auto;
   display: block;
   margin: 20px auto;
   border-radius: 8px;
}

/* Event Details */
h2, h3, h4 {
   color: #2980b9;
   font-size: 2em;
   text-transform: capitalize;
   margin-bottom: 15px;
   text-align: center;
}

p {
   font-size: 1.1em;
   margin-bottom: 12px;
   line-height: 1.8;
}

strong {
   color: #e74c3c;
   font-weight: bold;
```

```
}

/* Video Section */
video {
   width: 100%;
   max-width: 600px;
   display: block;
   margin: 20px auto;
   border-radius: 8px;

}

/* Google Map */
iframe {
   width: 100%;
   max-width: 600px;
   border: none;
   display: block;
   margin: 20px auto;
   border-radius: 8px;
}

/* RSVP Form */
form {
   width: 80%;
   margin: 0 auto;
   background-color: #f5f5f5;
   padding: 20px;
   border-radius: 8px;
   border: 1px solid #ccc;
}

form label {
   font-size: 1em;
   color: #333;
   margin-bottom: 6px;
   display: block;
}

form input,
form select {
   width: 100%;
```

```
    padding: 10px;
    margin-bottom: 12px;
    border: 1px solid #ccc;
    border-radius: 5px;
    font-size: 1em;
    background-color: #fff;
}

form input[type="submit"] {
    background-color: #2980b9;
    color: white;
    border: none;
    padding: 10px;
    font-size: 1em;
    border-radius: 5px;
    cursor: pointer;
    width: 100%;
}
```

Explanation:

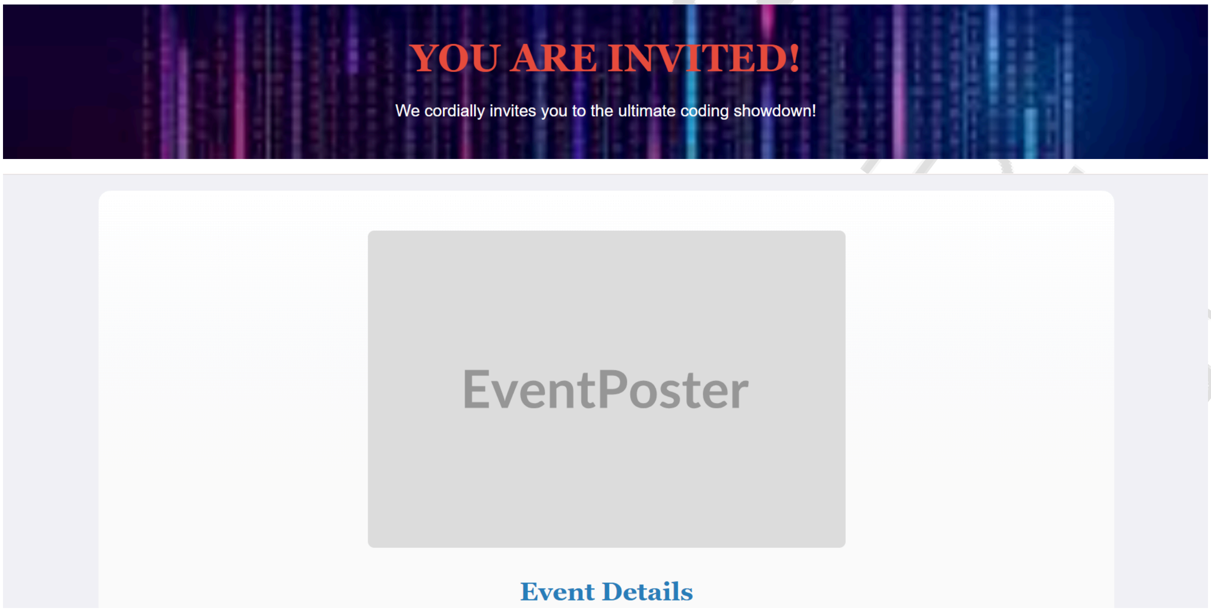
Footer Section:

```
footer {
    background-color:black ;
    color: white;
    text-align: center;
    padding: 20px 0;
    margin-top: 40px;

}

footer p {
    margin: 10px 0;
}

footer p a {
    color: white;
    text-decoration: none;
    font-weight: bold;
}
```

Explanation:

`background-color: black;`
Sets the background of the footer to black.
`color: white;`
Applies white color to all text within the footer, ensuring good contrast against the black background.
`text-align: center;`
Centers the content horizontally within the footer.
`padding: 20px 0;`
Adds 20 pixels of padding to the top and bottom, providing space inside the footer without affecting the left and right sides.
`margin-top: 40px;`
Adds 40 pixels of space above the footer, separating it from the content above.

Final Output:



Event Name: CodeXtravaganza 2025 The Ultimate Coding Showdown

Date: 14th March 2025

Time: 10:00 AM - 5:00 PM

Venue: Unit 007-008, GF, Tower-A, Unitech Cyber Park, Durga Colony, Sector 39, Gurugram, Haryana

Why Attend?

Join us for an electrifying event where top coders, tech enthusiasts, and problem solvers come together to battle in competitive programming and innovation challenges!

No Team? No Problem!

Participate solo or find teammates at our networking zone. Win exciting prizes, internships, and coding swag!

Event Rules

Registration: Mandatory

Eligibility: Open for students & professionals

Rounds: Algorithm challenge, Debugging round, Hackathon sprint

Strictly follow the event rules for a fair competition.

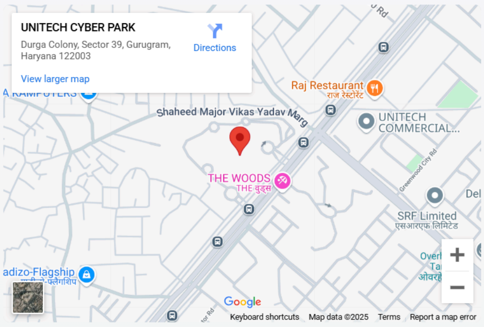
Glimpses From Past Events

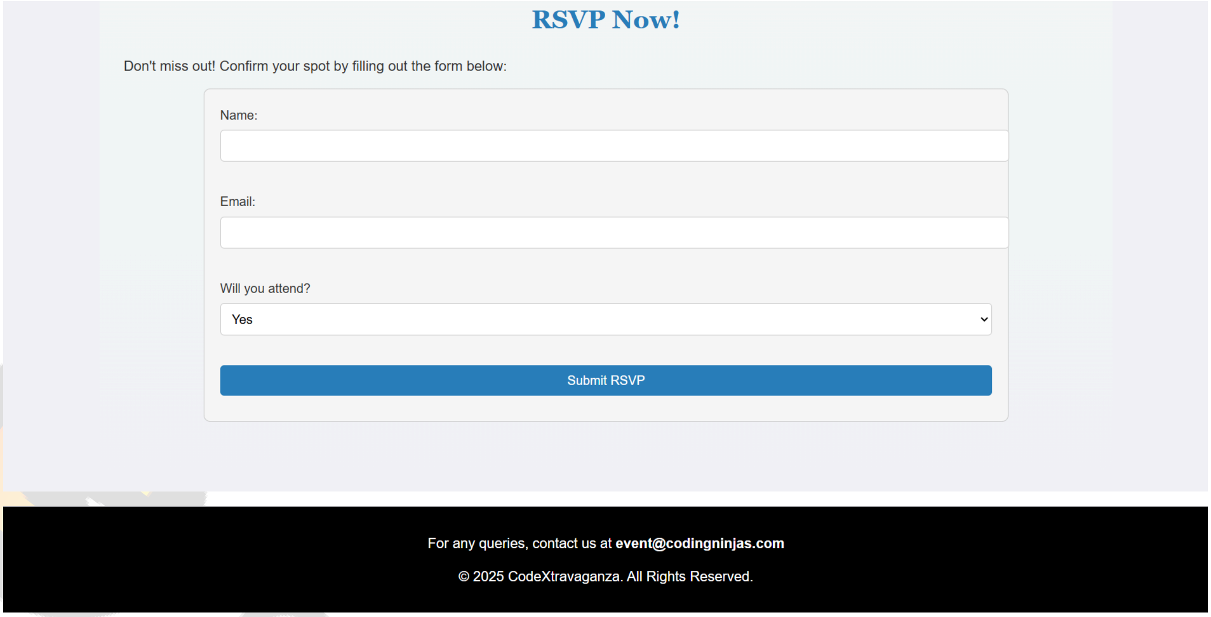
Still wondering how epic this event is? Check out some moments from our previous coding battles!

Watch The Action Unfold!

Wondering Where To Come?

Lost in the matrix of locations? No worries! Just follow the map below and let Google navigate you straight to the battle arena!





RSVP Now!

Don't miss out! Confirm your spot by filling out the form below:

Name:

Email:

Will you attend?

Yes

For any queries, contact us at **event@codingninjas.com**

© 2025 CodeXtravaganza. All Rights Reserved.

Concept In-depths:

Understanding the CSS Box Model

The CSS Box Model is the cornerstone of web layout design. It describes how every HTML element is structured as a rectangular box, controlling the spacing, sizing, and overall layout of elements on a webpage.

1. What Is the CSS Box Model?

In CSS, every element is represented as a box that consists of several layers:

- **Content:** The area where your text, images, or other media are displayed.
- **Padding:** The space between the content and the border, adding internal spacing.
- **Border:** The line surrounding the padding and content, visually separating the element.
- **Margin:** The space outside the border, which separates the element from neighboring elements.

Understanding how these parts work together is key to controlling the layout and spacing of elements on your page.

---

2. Components of the Box Model

a. Content

- **Definition:**
  The core of the box where your actual data (text, images, etc.) is displayed.

b. Padding

- **Definition:**
  The transparent space between the content and the border.
- **Usage:**
  Padding increases the clickable or readable area inside the element without affecting the content directly.

Example:

```
.box {
  padding: 20px;
}
```

- *This adds 20 pixels of space on all sides between the content and the border.*

c. Border

- **Definition:**
  The edge surrounding the padding and content.
- **Customization:**
  You can set the border's width, style (solid, dotted, dashed, etc.), and color.

Example:

```
.box {
  border: 2px solid #333;
}
```

- *This creates a 2-pixel solid border with a dark gray color around the box.*

d. Margin

- **Definition:**
  The transparent space outside the border.
- **Usage:**
  Margins create separation between different elements on a page.

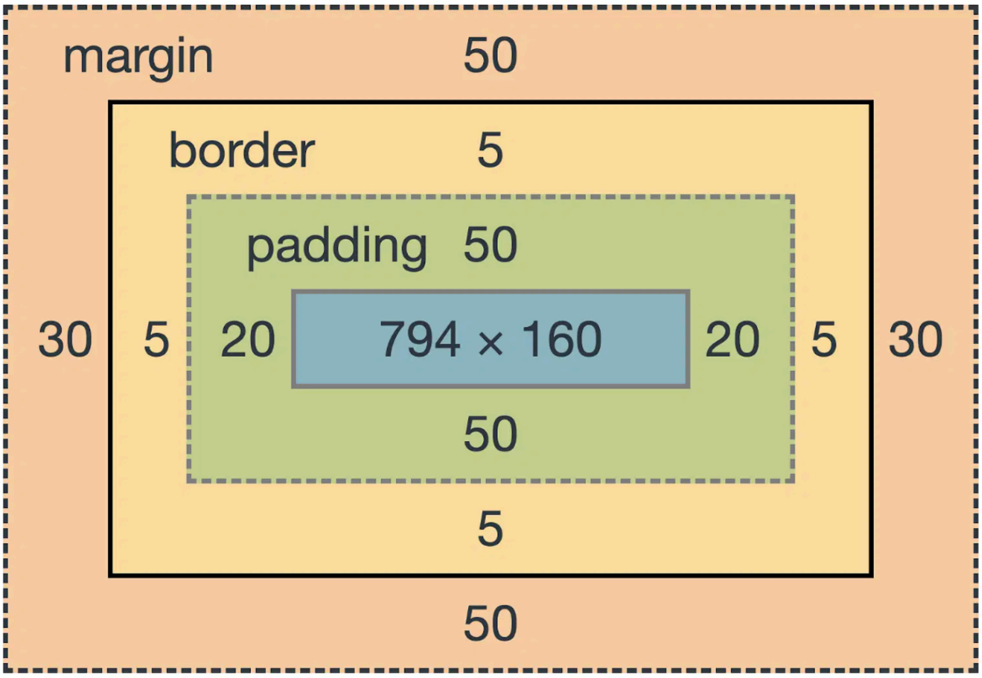
Example:

```
.box {
  margin: 15px;
}
```

- *This adds a 15-pixel space around the entire box, separating it from adjacent elements.*

---

3. Visualizing the Box Model



Each layer adds to the overall size of the element, affecting its final appearance and spacing on the page.

---

4. Impact on Layout Design

a. Spacing and Alignment

- **Control:**
  Using padding and margins effectively allows you to control the spacing between elements, making layouts look clean and organized.
- **Example:**
  Adding margins can prevent text or images from crowding each other, while padding can ensure that text does not touch the edges of a container.

b. Responsive Design

- **Flexibility:**
  Understanding the box model is crucial for responsive design, where elements must adjust gracefully to different screen sizes.

Tips and Tricks:

1. HTML Learning: https://developer.mozilla.org/en-US/docs/Web/HTML
2. HTML Practice: https://www.naukri.com/code360/web-development

CSS resources:

Responsively: Responsively is a modern, open-source design tool that helps web developers and designers test their websites across multiple devices simultaneously.
CSS Tricks: CSS-Tricks is a highly popular and comprehensive resource for learning and mastering CSS, web design, and front-end development techniques.
Google Fonts: Google Fonts is a free, open-source library that offers a wide selection of web fonts optimized for performance.

Follow tech blogs like Smashing Magazine, and Dev.to to stay current with industry trends