# CSS Course Outline

## 1. Introduction to CSS

* What is CSS?  
  CSS (Cascading Style Sheets) is used to style and layout web pages. It controls the appearance of HTML elements, including colors, fonts, spacing, and layout.
* CSS Syntax  
  CSS rules consist of selectors and declarations. A declaration is made up of a property and a value, like color: red;.
* Including CSS in HTML  
  Discuss different ways to include CSS in an HTML document: Inline, Internal (via <style>), and External (via <link>).

## 2. CSS Selectors

* Basic Selectors  
  Learn how to target HTML elements using simple selectors: element (p), class (.className), and ID (#idName).
* Universal Selector  
  The \* selector applies styles to all elements on the page.
* Group Selectors  
  Use a comma to apply the same styles to multiple selectors, like h1, h2, h3.

## 3. Inline, Block, and Inline-block Elements

* Inline Elements  
  Inline elements, like <span> and <a>, take up only as much space as their content and do not start on a new line.
* Block Elements  
  Block-level elements, like <div> and <p>, take up the full width of their parent container and always start on a new line.
* Inline-block Elements  
  The inline-block display value allows an element to be inline but behave like a block with specific width and height.

## 4. CSS Colors

* Color Values  
  CSS supports different ways to define colors: predefined color names (red), hexadecimal (#ff0000), RGB (rgb(255,0,0)), and HSL (hsl(0, 100%, 50%)).
* Background Colors  
  Apply background colors using the background-color property.

## 5. CSS Units

* Absolute Units  
  Absolute units include px (pixels), cm (centimeters), mm (millimeters), and in (inches).
* Relative Units  
  Relative units like em, rem, %, and vw adapt to the size of the parent or viewport.

## 6. CSS Fonts and Typography

* Font Family  
  Apply different fonts to text using the font-family property. Example: font-family: Arial, sans-serif;.
* Font Size  
  Control the size of text using font-size, which can be defined in px, em, rem, etc.
* Font Weight  
  Adjust the thickness of text using the font-weight property (e.g., bold, normal, or numeric values like 400, 700).
* Line Height  
  Set the height of text lines using line-height for better readability.

## 7. CSS Box Model

* What is the Box Model?  
  The CSS box model defines the rectangular boxes that are generated for elements in the document tree. It consists of content, padding, border, and margin.
* Padding and Margin  
  padding defines the space between the content and the border, while margin defines the space outside the border.
* Border Properties  
  Use the border property to control the width, style, and color of an element's border.

## 8. CSS Display Property

* Display Types  
  The display property specifies how an element is displayed on the page. Common values include block, inline, inline-block, and none.
* Hiding Elements  
  Use display: none; to completely remove an element from the page.

## 9. CSS Positioning

* Static Positioning  
  Elements are positioned according to the normal flow of the document (position: static;).
* Relative Positioning  
  Position an element relative to its normal position using position: relative;.
* Absolute Positioning  
  Position an element relative to its closest positioned ancestor using position: absolute;.
* Fixed Positioning  
  An element with position: fixed; is positioned relative to the browser window.
* Sticky Positioning  
  position: sticky; keeps an element fixed when you scroll past it, but it moves with the scroll until a certain point.

## 10. CSS Flexbox

* Introduction to Flexbox  
  Flexbox is a layout model that provides an efficient way to align and distribute space among items in a container.
* Flex Container and Flex Items  
  Define a flex container using display: flex;, and its children automatically become flex items.
* Main Axis and Cross Axis  
  The flexbox layout uses a main axis (horizontal or vertical) and a cross axis to align items.
* Flex Properties  
  Key properties include justify-content, align-items, align-content, flex-direction, flex-wrap, and flex-grow.

## 11. CSS Grid

* Introduction to Grid Layout  
  CSS Grid is a two-dimensional layout system used to create grid-based layouts.
* Grid Container and Grid Items  
  Use display: grid; to create a grid container, and its children automatically become grid items.
* Grid Template Rows and Columns  
  Define the structure of the grid using grid-template-rows and grid-template-columns to set row and column sizes.
* Grid Gap  
  Use the gap property to create space between grid items.

## 12. CSS Pseudo-classes and Pseudo-elements

* Pseudo-classes  
  Pseudo-classes target elements based on their state. Common examples include :hover, :focus, :nth-child(), and :first-child.
* Pseudo-elements  
  Pseudo-elements create and style specific parts of elements. Common examples include ::before, ::after, ::first-letter, and ::first-line.

## 13. CSS Combinators

* Descendant Combinator  
  Targets elements that are descendants of a specific element, written as div p (targets p inside any div).
* Child Combinator  
  Targets direct children of an element, written as div > p.
* Adjacent Sibling Combinator  
  Targets an element that is the adjacent sibling of another, written as h1 + p.
* General Sibling Combinator  
  Targets all siblings of an element, written as h1 ~ p.

## 14. CSS Transitions and Animations

* CSS Transitions  
  Transitions allow property changes to occur smoothly over a specified duration using transition: property duration;.
* CSS Animations  
  Animations define more complex sequences of property changes using @keyframes and the animation property.
* Keyframes  
  Use the @keyframes rule to define the steps in an animation.

## 15. CSS Media Queries and Responsive Design

* What are Media Queries?  
  Media queries allow you to apply CSS rules based on the characteristics of the user's device, such as screen size or orientation.
* Creating Responsive Designs  
  Learn to create responsive layouts that adapt to different screen sizes using @media rules.
* Viewport Meta Tag  
  Discuss the importance of the viewport meta tag in creating responsive designs for mobile devices.

## 16. CSS Variables (Custom Properties)

* Defining Variables  
  Use custom properties (CSS variables) to define reusable values, like --main-color: #ff0000;.
* Using Variables  
  Apply variables throughout your stylesheet using var(--main-color).

## 17. CSS Z-index

* Understanding Z-index  
  The z-index property controls the vertical stacking order of elements. Higher values appear in front of lower values.
* Stacking Context  
  Explain stacking context and how it is formed in CSS.

## 18. CSS Backgrounds

* Background Color and Image  
  Set background colors with background-color and images with background-image.
* Background Position and Size  
  Control the position and size of background images using background-position and background-size.

## 19. CSS Shadows

* Box Shadow  
  Apply shadows to elements using the box-shadow property.
* Text Shadow  
  Apply shadows to text using the text-shadow property.

## 20. CSS Transforms

* 2D Transforms  
  Use the transform property to rotate, scale, skew, or translate elements in 2D space.
* 3D Transforms  
  Create 3D effects using rotateX, rotateY, and perspective transforms.

## 21. CSS Best Practices

* Organizing CSS Code  
  Tips for writing clean, maintainable, and scalable CSS, such as using comments, naming conventions (BEM), and modularity.