**1. CSS Basics**

CSS (Cascading Style Sheets) is used to control the style and layout of web pages. It helps separate the content of a webpage (HTML) from its presentation (styles like colors, fonts, spacing, etc.).

**Selectors:**

Selectors are patterns used to select the elements you want to style.

**Types of Selectors:**

* **Universal Selector (\*)**: Selects all elements in the document.
* \* {
* color: black;
* }

This will set the text color of every element on the page to black.

* **Type Selector (Element Selector)**: Selects all instances of a specific element.
* p {
* font-size: 16px;
* }

This targets all <p> tags and sets the font size to 16px.

* **Class Selector (.)**: Selects elements with a specific class.
* .highlight {
* background-color: yellow;
* }

This applies a yellow background to any element with the class highlight.

* **ID Selector (#)**: Selects an element with a specific ID.
* #header {
* font-weight: bold;
* }

This targets the element with the ID header and makes the text bold.

**CSS Properties:**

Some common CSS properties include:

* **Color**: Sets the color of text.
* p {
* color: blue;
* }
* **Font Size**: Sets the size of the text.
* h1 {
* font-size: 24px;
* }
* **Background**: Sets the background color of an element.
* div {
* background-color: lightgray;
* }
* **Margin and Padding**: Adjusts the space around elements.
* .container {
* margin: 20px;
* padding: 10px;
* }

**Example (CSS Basics):**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>CSS Basics</title>

<style>

body {

font-family: Arial, sans-serif;

}

h1 {

color: darkblue;

font-size: 30px;

}

p {

font-size: 16px;

color: green;

}

.highlight {

background-color: yellow;

}

#footer {

background-color: gray;

padding: 10px;

text-align: center;

color: white;

}

</style>

</head>

<body>

<h1>Welcome to CSS Basics</h1>

<p>This is an example paragraph with <span class="highlight">highlighted</span> text.</p>

<div id="footer">

This is the footer content

</div>

</body>

</html>

**2. CSS3 Features**

CSS3 introduced new features and capabilities, allowing developers to create more visually appealing and responsive web designs. Some of the key CSS3 features include:

**2.1. Borders and Border Radius**

The border-radius property allows you to round the corners of elements.

div {

border: 2px solid black;

border-radius: 10px;

}

This will create a div with rounded corners. The higher the value of border-radius, the more rounded the corners will appear.

**2.2. Box Shadows**

The box-shadow property adds shadow effects around an element’s frame.

div {

width: 200px;

height: 100px;

background-color: lightblue;

box-shadow: 5px 5px 10px gray;

}

This will add a shadow of 5px horizontal and vertical offset, and 10px blur with a gray color around the div.

**2.3. Gradients**

CSS3 allows the use of gradients for backgrounds. A gradient smoothly transitions from one color to another.

div {

background: linear-gradient(to right, red, yellow);

height: 200px;

width: 100%;

}

This will create a linear gradient transitioning from red to yellow from left to right.

**2.4. Transformations**

The transform property allows you to apply transformations like rotation, scaling, or skewing.

div {

width: 100px;

height: 100px;

background-color: teal;

transform: rotate(45deg);

}

This will rotate the div by 45 degrees.

**2.5. Transitions**

CSS transitions allow you to change property values smoothly (over a given duration).

button {

background-color: blue;

color: white;

padding: 10px;

border: none;

cursor: pointer;

transition: background-color 0.3s ease;

}

button:hover {

background-color: green;

}

Here, when the button is hovered, the background color transitions smoothly from blue to green over 0.3 seconds.

**2.6. Flexbox (CSS3 Layout)**

Flexbox is a powerful layout model for creating flexible and responsive layouts.

.container {

display: flex;

justify-content: space-between;

}

.item {

width: 30%;

background-color: lightgreen;

padding: 10px;

text-align: center;

}

<div class="container">

<div class="item">Item 1</div>

<div class="item">Item 2</div>

<div class="item">Item 3</div>

</div>

The container uses Flexbox to distribute the items evenly, with space between them.

**Example (CSS3 Features):**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>CSS3 Features</title>

<style>

.box {

width: 200px;

height: 100px;

background: linear-gradient(to right, blue, lightblue);

border-radius: 15px;

box-shadow: 3px 3px 8px rgba(0, 0, 0, 0.2);

transform: scale(1.1);

transition: transform 0.3s ease;

}

.box:hover {

transform: scale(1.2);

}

</style>

</head>

<body>

<div class="box"></div>

</body>

</html>

**3. CSS Layouts**

CSS layouts define how elements are arranged on a webpage. There are several ways to create layouts in CSS, including Flexbox, Grid, and traditional methods like floats.

**3.1. Flexbox Layout**

Flexbox is ideal for one-dimensional layouts (either row or column). It allows easy distribution of space between items and alignment.

Example of a simple Flexbox layout:

.container {

display: flex;

justify-content: space-around;

align-items: center;

height: 200px;

}

.item {

width: 50px;

height: 50px;

background-color: teal;

}

<div class="container">

<div class="item"></div>

<div class="item"></div>

<div class="item"></div>

</div>

In this example, the .container is a Flexbox container. The child .item elements are spaced around and centered vertically within the container.

**3.2. Grid Layout**

CSS Grid Layout is a powerful two-dimensional layout system that allows you to design both rows and columns.

.container {

display: grid;

grid-template-columns: repeat(3, 1fr);

grid-gap: 10px;

}

.item {

background-color: coral;

padding: 20px;

text-align: center;

}

<div class="container">

<div class="item">Item 1</div>

<div class="item">Item 2</div>

<div class="item">Item 3</div>

</div>

In this layout, the .container creates a grid with three equal-width columns (1fr means one fraction of the available space).

**3.3. Float-based Layouts (Old Method)**

Floats were traditionally used to create multi-column layouts. Though they are less commonly used now due to Flexbox and Grid, they are still useful for certain types of layouts.

.container {

width: 100%;

}

.left {

float: left;

width: 50%;

}

.right {

float: right;

width: 50%;

}

<div class="container">

<div class="left">Left Column</div>

<div class="right">Right Column</div>

</div>

Here, the .left and .right divs are floated to create a two-column layout. The container div wraps both of them.

**Conclusion**

In this detailed overview, we covered:

1. **CSS Basics**: The foundational concepts of CSS, such as selectors, properties, and basic styling techniques.
2. **CSS3 Features**: Advanced features introduced in CSS3, such as border-radius, box shadows, gradients, transforms, transitions, and Flexbox.
3. **CSS Layouts**: Layout techniques for structuring content on a webpage, including Flexbox, Grid, and traditional float-based layouts.

Each of these concepts plays an important role in designing modern, responsive, and visually appealing webpages. Using CSS3 features effectively allows you to create interactive, engaging, and professional web designs.

Example

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>CSS Basics and Layouts</title>

<style>

/\* General Page Styling \*/

body {

font-family: Arial, sans-serif;

margin: 0;

padding: 0;

background-color: #f4f4f4;

}

/\* Header Section \*/

header {

background-color: #4CAF50;

color: white;

padding: 20px;

text-align: center;

box-shadow: 0 4px 8px rgba(0, 0, 0, 0.2);

}

/\* Navigation Bar \*/

nav {

background-color: #333;

color: white;

padding: 10px 0;

text-align: center;

}

nav ul {

list-style: none;

padding: 0;

}

nav ul li {

display: inline;

margin: 0 15px;

}

nav ul li a {

color: white;

text-decoration: none;

font-size: 18px;

}

nav ul li a:hover {

text-decoration: underline;

}

/\* Main Content Layout (Flexbox) \*/

.main-content {

display: flex;

justify-content: space-between;

padding: 20px;

}

.content {

width: 70%;

background-color: #fff;

padding: 20px;

margin-right: 20px;

box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);

}

.sidebar {

width: 25%;

background-color: #fff;

padding: 20px;

box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);

}

/\* Footer Section \*/

footer {

background-color: #333;

color: white;

text-align: center;

padding: 15px 0;

position: fixed;

width: 100%;

bottom: 0;

}

/\* Box Styling \*/

.box {

width: 100%;

height: 150px;

background-color: #FF6347;

margin: 20px 0;

color: white;

text-align: center;

padding-top: 50px;

font-size: 24px;

border-radius: 8px;

box-shadow: 5px 5px 10px rgba(0, 0, 0, 0.2);

transition: transform 0.3s ease;

}

.box:hover {

transform: scale(1.1);

}

/\* Grid Layout Example \*/

.grid-container {

display: grid;

grid-template-columns: repeat(3, 1fr);

gap: 20px;

padding: 20px;

}

.grid-item {

background-color: #FF9800;

color: white;

padding: 30px;

text-align: center;

border-radius: 8px;

box-shadow: 2px 2px 5px rgba(0, 0, 0, 0.2);

}

/\* Responsive Design \*/

@media (max-width: 768px) {

.main-content {

flex-direction: column;

align-items: center;

}

.content, .sidebar {

width: 100%;

margin-right: 0;

}

.grid-container {

grid-template-columns: 1fr;

}

}

</style>

</head>

<body>

<!-- Header -->

<header>

<h1>Welcome to My Website</h1>

<p>Learn CSS Basics and Layouts</p>

</header>

<!-- Navigation Bar -->

<nav>

<ul>

<li><a href="#home">Home</a></li>

<li><a href="#services">Services</a></li>

<li><a href="#about">About</a></li>

<li><a href="#contact">Contact</a></li>

</ul>

</nav>

<!-- Main Content (Flexbox Layout) -->

<div class="main-content">

<!-- Content Area -->

<div class="content">

<h2>Main Content Area</h2>

<p>This area contains the main content of the webpage. Here you can add text, images, or other media.</p>

<div class="box">Hover over me!</div>

</div>

<!-- Sidebar -->

<div class="sidebar">

<h3>Sidebar</h3>

<p>This is a sidebar with related content, links, or advertisements.</p>

</div>

</div>

<!-- Grid Layout Example -->

<div class="grid-container">

<div class="grid-item">Grid Item 1</div>

<div class="grid-item">Grid Item 2</div>

<div class="grid-item">Grid Item 3</div>

</div>

<!-- Footer -->

<footer>

<p>&copy; 2024 My Website. All rights reserved.</p>

</footer>

</body>

</html>