

## HTML & CSS Assignments

Q) The CSS box model is a fundamental concept in web design that describes the layout and structure of element on a webpage. The box model consists of four main parts: Content, padding, border and margin.

1) Content:- It refers to the actual content of the element such as text, images or other media.

2) Padding:- It is the space between the content & the border. Padding can be used to create space within the element & around its content.

3) Border:- It is the line that surround the padding and content of the elements.

4) Margin:- It is the space outside the border, which separates the element from other elements on the webpage.

\* These parts affect the overall dimensions and spacing of an element on the webpage.

Exercise:- Simple HTML structure.

<!doctype html>

<html>

<head>

②  
<body>

<div class = "Container">

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

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

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

</div>

</body>

</html>

### Example of using CSS Grid:

<!DOCTYPE html>

<html>

<head>

<style>

• Container {

display: grid;

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

gap : 10px;

}

• item {

text-align : Center;

padding : 10px;

Border : 1px solid #000;

}

</style>

</head>

<body>

<div class = "Container">

layouts such as arranging row or column.

⊕ Flexbox is ideal for navigation menu & card layouts and centering vertically and horizontally.

## 2) CSS Grid

⊕ Grid is used for two dimensional layouts, making it ideal for creating complex grid-based concepts.

⊗ Its suitable for designing grid layouts, image galleries, & even entire webpage layouts.

## Exercise. Example of using flexbox

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
  .container {
```

```
    display: flex;
```

```
    justify-content: space-between;
```

```
    align-items: center;
```

```
  }
```

```
  .item {
```

```
    flex: 1;
```

```
    text-align: center;
```

```
    padding: 10px;
```

```
    border: 1px solid #000;
```

```
  }
```

```
</style>
```

```
</head>
```



```
<style> </style>
```

```
</head>
```

```
<body>
```

```
<div class="box">
```

```
<h2> Sample box </h2>
```

```
<p> This is Sample paragraph inside the box </p>
```

```
</div>
```

```
</body>
```

```
</html>
```

### ① Setting margin

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

### ② Adding padding

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

### ③ Applying a border

```
.box {  
  border: 20px solid black;  
}
```

Q2) The specificity of a CSS selection that determine which CSS styles are applied to an element when multiple competing styles exist. It is important in styling webpages because it helps developers & designers understand how to control the priority of CSS rules & ensure that the intended styles are

applied correctly. Specificity is determined by the combination of selectors in a CSS rule. The more specific a selector is, the higher its specificity value.

Order of importance: 1) Inline styles 2) selectors (ID) 3) Class selectors and attribute selectors 4) Type & Pseudo selectors 5) Universal & descendant selectors.

### Exercise:

```
<!doctype html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
#special-paragraph {
```

```
  colour: red; }
```

```
p {color: blue; }
```

```
!important {color: green; }
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<p id="special paragraph" class=
```

```
"important"> This is a paragraph </p>
```

```
</body>
```

```
</html>
```

Q3) CSS Flexbox & CSS grid are two layout models in CSS that allow web developers to create responsive & flexible page layouts.

### 1) CSS Flexbox:

\* Flexbox is primarily used for one dimensional

```

<div class="item"> item1 </div>
<div class="item"> item2 </div>
<div class="item"> item3 </div>
</div>
</body>
</html>

```

- ②
- Position: relative:- adjusts an elements position relative to its normal flow, useful for minor positioning tweaks.
  - Position: absolute positions:- an element relative to the nearest positioned ancestor or the viewport, often used for overlays or precise positioning.
  - Position: fixed:- Positions an element relative to the viewport, making it stay fixed even when the page is scrolled, commonly used for fixed headers on navigation bars. The choice depends on specific layout and design needs.

#### \* Exercise:-

```

<!DOCTYPE html>
<html>
<head>
<link rel="stylesheet" type="text/css"
href="styles.css">

```

```

</head>
<body>
<div class="relative">
<p>Relative position </p>
</div>
<div class="absolute-container">
<div class="absolute">
<p>Absolute position </p>
</div>
<div class="Fixed">
<p>fixed position </p>
</div>
</body>
</html>

```

→ CSS (style.css)

- relative {  
position: relative;  
top: 10px;  
left: 20px;  
}
- absolute - container {  
position: relative;  
height: 200px;  
}
- absolute {  
position: absolute;  
top: 50%;  
}



① ②  
css(style.css)

blockquote::before {

content: "120100"; /\* unicode character  
for right double quote \*/

font-size: 24px;

margin-left: 5px;

}

Responsive web design is an approach to designing and developing websites that ensures they adapt and display properly on various devices and screen sizes, such as desktop computers, tablets, and mobile phones. This is achieved by using flexible grids and layouts, along with CSS media queries.

Media queries are CSS rules that apply styles based on the characteristics of the device's display, like the width, height and resolution. They allow you to define different styles for different screen sizes or device types.

\* Exercise:-

<!doctype html>

<html>

<head>

<title>Responsive web design </title>

③

```
left: 50%;
```

```
transform: translate(-50%, -50%);
```

```
}
```

```
• fixed {
```

```
    position: fixed;
```

```
    top: 0;
```

```
    right: 0;
```

```
}
```

Q5) CSS pseudo elements such as `::before` and `::after` allow you to insert content before or after the content of an element, without the need to add extra HTML markup. They are often used for decorative elements, enhancing design, or for adding additional content.

Ex. `<!DOCTYPE html>`

```
<html>
```

```
<head>
```

```
<link rel="stylesheet" type="text/css"
```

```
href="style.css">
```

```
</head>
```

```
<body>
```

```
<article>
```

```
<blockquote>
```

```
"my name is poorna".
```

```
</blockquote>
```

```
</article>
```

```
</body>
```

```
</html>
```

```

<style>
body {
  font-family: Arial, sans-serif;
  text-align: center;
  margin: 0;
  padding: 50px;
}

```

```

- Container {
  max-width: 800px;
  margin: 0 auto;
}

```

```

- box {
  border: 1px solid #ccc;
  padding: 20px;
  margin-bottom: 20px;
}

```

```

</style>

```

```

</head>

```

```

<body>

```

```

<div class="container">

```

```

  <h1> Responsive design </h1>

```

```

  <div class="box">

```

```

    <h2> Box 1 </h2>

```

```

    <p> Content for Box 1 </p>

```

```

  </div>

```

```

  <div class="box">

```

```

    <h2> Box 2 </h2>

```

```

    <p> Content for Box 2 </p>

```

```

  </div>

```

```

  </div class="box">

```

```

<h2> Box 3 </h2>

```

```

<p> Content for Box 3 </p>

```

```

</div>

```

```

</div>

```

```

</body>

```

```

</html>

```

Accessibility in web development is crucial as it ensures that websites are usable and understandable by people with disabilities, including those with visual, auditory, cognitive, and motor impairments. It aims to provide equal access and opportunities for all users, regardless of their abilities or disabilities.

ARIA (Accessible Rich Internet Applications) is a set of attributes that can be added to HTML elements to provide additional information to assistive technologies, such as screen readers. ARIA roles and attributes.

- 1) Role = "banner": Indicates main header or banner of a page.
- 2) Role = "navigation": Identifies the navigation section of a page.
- 3) Role = "main": Denotes the main content area of a page.
- 4) Role = "form": Marks up a form element.
- 5) Role = "link": Identifies an interactive link.



The <doctype> (or declaration in HTML (Hyper Text Markup Language) serves as a signal to the web browser about which version of HTML or XHTML the document is written. It stands for "Document Type Definition".

The purpose of the <doctype> is declaration to the browser that the browser renders the webpage correctly by interpreting the HTML markup according to the specified version, rules and standards.

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

</meta>

<title>Doctype experiment </title>

</head>

<body>

<h1>Hello, world! </h1>

</body>

</html>

## Exercise

```
<!doctype html>
```

```
<head>
```

```
<title> sample webpage </title>
```

```
</head>
```

```
<body>
```

```
<header>
```

```
<h2> Sample website </h2>
```

```
</header>
```

```
<ul>
```

```
<li><a href="home.html">Home</a></li>
```

```
<li><a href="about.html">about Us</a></li>
```

```
<li><a href="services.html">Services</a>
```

```
<li><a href="contact.html">Contact</a></li>
```

```
</ul>
```

```
</nav>
```

```
<main>
```

```
<h2> Welcome to our website </h2>
```

```
<p> This is some sample content </p>
```

```
</main>
```

```
<footer>
```

```
<p> ©copy; 2013 samplewebsite </p>
```

```
</footer>
```

```
</body>
```

```
</html>
```