CERTIFIED QUIZ

Introduction to CSS

Cascading Style Sheets (CSS) is a stylesheet language used to describe the presentation of a document written in HTML or XML. CSS controls the layout of multiple web pages all at once. It allows developers to apply styles, such as colors, fonts, and spacing, to web pages, making them visually appealing. It was first introduced in 1996 by the W3C (World Wide Web Consortium) and has become an integral part of modern web design.

CSS Syntax

The basic syntax of CSS consists of a selector and a declaration block. A selector specifies which HTML element(s) the style applies to, while a declaration block contains one or more declarations separated by semicolons.

Each declaration includes a CSS property and a value, separated by a colon. Properties are aspects of the element you want to change (e.g., color, font-size), and values specify the new appearance.

Example:
css
p {
color: red;
font-size: 16px;
}

Selectors

Selectors are used to select the HTML elements that you want to style. There are various types of selectors:

- Element Selector: Targets all elements of a specified type (e.g., `p`, `h1`).
- ID Selector: Targets a specific element with a unique ID, prefixed with a `#` (e.g., `#header`).
- Class Selector: Targets all elements with a specified class, prefixed with a `.` (e.g., `.menu`).
- Attribute Selector: Targets elements with a specific attribute (e.g., `input[type="text"]`).

Example:
css
#header {
background-color: blue;
}
.menu {
font-size: 18px;
}

Box Model

In CSS, the box model describes the rectangular boxes that are generated for elements in the document tree and governs how elements are laid out.

It consists of four parts:
- Content: The actual content of the box, such as text or an image.
- Padding: Clears an area around the content inside the element.
- Border: A border that surrounds the padding (if any) and the content.
- Margin: Clears an area outside the border, creating space between the element and others.

Example:
css
```
div {
padding: 10px;
border: 2px solid black;
margin: 20px;
}
```

CSS Positioning

CSS provides several positioning schemes to define how elements are arranged in the document:

- Static: The default positioning. Elements are positioned according to the normal flow of the document.
- Relative: Positioned relative to its normal position.
- Absolute: Positioned relative to its nearest positioned ancestor or the initial containing block.
- Fixed: Positioned relative to the browser window, and it does not move when scrolling.
- Sticky: Toggles between relative and fixed, depending on scroll position.

Example:
css
```
div {
position: absolute;
top: 50px;
left: 100px;
}
```

Flexbox

CSS Flexbox is a layout model that provides an efficient way to align and distribute space among items in a container, even when their size is unknown. It is particularly useful for creating responsive layouts.

Key properties include:
- display: flex: Defines a flex container.
- flex-direction: Specifies the direction of the flex items (row, column, etc.).
- justify-content: Aligns items horizontally.
- align-items: Aligns items vertically.

Example:
css
```
.container {
    display: flex;
    flex-direction: row;
    justify-content: space-between;
    align-items: center;
}
```

CSS Grid

CSS Grid is a two-dimensional layout system that allows you to create grid-based layouts, which can arrange items both in rows and columns.

Key properties include:
- display: grid: Defines a grid container.
- grid-template-columns: Specifies the number and width of columns.
- grid-template-rows: Specifies the number and height of rows.

- gap: Defines the space between grid items.

Example:

```
.container {
    display: grid;
   grid-template-columns: repeat(3, 1fr);
    gap: 20px;
}
```

Media Queries

CSS Media Queries are used to apply different styles to different screen sizes or devices. This makes websites responsive, adjusting their layout based on the device's characteristics (e.g., width, height, orientation).

Example:

```
@media only screen and (max-width: 600px) {
    body {
        background-color: lightblue;
     }
}
```

CSS Animations

CSS animations allow elements to transition from one style to another over a certain period of time. Animations are defined using keyframes, which specify the styles at different points in the animation.

Example:
css
```
@keyframes slide {
    0% { transform: translateX(0); }
            100% { transform: translateX(100px); }
}

.box {
            animation: slide 2s ease-in-out infinite;
}
```