

Assignment Code: FSD-AG-003

HTML & CSS | Assignment

Question 1 : Create a layout with three boxes side by side using Flexbox. The first box should have twice the width of the other two. Provide the HTML and CSS code snippets.

Answer:

HTML Code:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Flexbox Layout</title>
  <link rel="stylesheet" href="styles.css">
</head>
<body>
  <div class="container">
    <div class="box box1">Box 1</div>
    <div class="box box2">Box 2</div>
    <div class="box box3">Box 3</div>
  </div>
</body>
</html>
```

CSS Code:

```
body {
  margin: 0;
  padding: 0;
  font-family: sans-serif;
}
```

```
.container {
  display: flex;
  width: 100%;
  height: 200px;
}
```

```
.box {
  display: flex;
  align-items: center;
  justify-content: center;
  color: white;
  font-size: 1.5rem;
}
```

```

}

.box1 {
  flex: 2;
  background-color: #3498db;
}

.box2 {
  flex: 1;
  background-color: #e74c3c;
}

.box3 {
  flex: 1;
  background-color: #2ecc71;
}

```

Question 2: Using CSS Grid, create a 2x2 grid layout where the first row spans two columns, and the second row has two equal columns. Provide the HTML and CSS code snippets.

Answer:

HTML Code:

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>CSS Grid Layout</title>
  <link rel="stylesheet" href="styles.css">
</head>
<body>
  <div class="grid-container">
    <div class="item item1">Header (spans 2 columns)</div>
    <div class="item item2">Left Column</div>
    <div class="item item3">Right Column</div>
  </div>
</body>
</html>

```

CSS Code:

```

body {
  margin: 0;
  padding: 0;
}

```

```

    font-family: sans-serif;
}

.grid-container {
    display: grid;
    grid-template-columns: 1fr 1fr;
    grid-template-rows: auto auto;
    gap: 10px;
    padding: 20px;
}

.item {
    background-color: #3498db;
    color: white;
    padding: 20px;
    font-size: 1.2rem;
    text-align: center;
    border-radius: 8px;
}

.item1 {
    grid-column: 1 / span 2;
    background-color: #2ecc71;
}

.item2 {
    background-color: #e74c3c;
}

.item3 {
    background-color: #9b59b6;
}

```

Question 3: Write a media query that changes the background color of the body to light grey when the screen width is less than 600px.

Answer:
CSS Code:

```

@media (max-width: 600px) {
    body {
        background-color: lightgrey;
    }
}

```

Question 4: Create a button that changes its background color to blue and scales up slightly when hovered over, using CSS transitions. Provide the HTML and CSS code snippets.

Answer:

HTML Code:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Hover Button</title>
  <link rel="stylesheet" href="styles.css">
</head>
<body>
  <button class="hover-button">Hover Me</button>
</body>
</html>
```

CSS Code:

```
.hover-button {
  padding: 12px 24px;
  font-size: 1rem;
  background-color: #ccc;
  color: #000;
  border: none;
  border-radius: 6px;
  cursor: pointer;
  transition: background-color 0.3s ease, transform 0.3s ease;
}

.hover-button:hover {
  background-color: blue;
  color: white;
  transform: scale(1.05);
}
```

Question 5: Use the ::before pseudo-element to add a decorative icon before each list item in an unordered list with the class name 'decorated-list'. Provide the HTML and CSS code snippets.

Answer:

HTML Code:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Decorated List</title>
  <link rel="stylesheet" href="styles.css">
</head>
<body>
  <ul class="decorated-list">
    <li>Item One</li>
    <li>Item Two</li>
    <li>Item Three</li>
  </ul>
</body>
</html>
```

CSS Code:

```
.decorated-list {
  list-style: none;
  padding: 0;
}

.decorated-list li {
  position: relative;
  padding-left: 24px;
  margin-bottom: 10px;
}

.decorated-list li::before {
  content: "★"; /* You can replace this with any Unicode icon or emoji */
  position: absolute;
  left: 0;
  color: #3498db;
  font-size: 1.2rem;
}
```

Question 6: Create a simple animation that moves a square box from left to right infinitely. Provide the HTML and CSS code snippets.

Answer:

HTML Code:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Box Animation</title>
  <link rel="stylesheet" href="styles.css">
</head>
<body>
  <div class="box"></div>
</body>
</html>
```

CSS Code:

```
body {
  margin: 0;
  padding: 0;
  overflow-x: hidden;
  height: 100vh;
  display: flex;
  align-items: center;
}

.box {
  width: 100px;
  height: 100px;
  background-color: #3498db;
  position: relative;
  animation: moveRight 3s linear infinite;
}

@keyframes moveRight {
  0% {
    left: 0;
  }
  100% {
    left: 100%;
  }
}
```

Question 7: Explain the difference between position: absolute; and position: fixed; in CSS.

Answer:

position: absolute;

- **Positioning Context:** Positioned relative to the nearest ancestor that has a position value other than static (e.g., relative, absolute, or fixed). If no such ancestor exists, it's positioned relative to the <html> element.
- **Scroll Behavior:** Moves with the page when you scroll.
- **Use Case:** Ideal for placing elements inside containers, like tooltips, badges, or overlays within a section.

Ex:

```
.container {  
  position: relative;  
}
```

```
.box {  
  position: absolute;  
  top: 10px;  
  left: 20px;  
}
```

position: fixed;

- **Positioning Context:** Always positioned relative to the **viewport**, not any ancestor.
- **Scroll Behavior:** Stays in the same place even when the page is scrolled.
- **Use Case:** Great for sticky headers, floating buttons, or persistent navigation elements.

Ex:

```
.fixed-box {  
  position: fixed;  
  top: 10px;  
  right: 10px;  
}
```

Question 8: Create a layered layout where a semi-transparent overlay covers an image using z-index. Provide the HTML and CSS code snippets.

Answer:

HTML Code:

```
<!DOCTYPE html>  
<html lang="en">  
<head>  
  <meta charset="UTF-8">  
  <title>Overlay Example</title>  
  <link rel="stylesheet" href="styles.css">
```

```
</head>
<body>
  <div class="image-container">
    
    <div class="overlay"></div>
    <div class="content">Overlay Text</div>
  </div>
</body>
</html>
```

CSS Code:

```
body {
  margin: 0;
  padding: 0;
}
```

```
.image-container {
  position: relative;
  width: 100%;
  max-width: 600px;
  height: 400px;
  margin: 0 auto;
  overflow: hidden;
}
```

```
.background-image {
  width: 100%;
  height: 100%;
  object-fit: cover;
  display: block;
}
```

```
.overlay {
  position: absolute;
  top: 0;
  left: 0;
  width: 100%;
  height: 100%;
  background-color: rgba(0, 0, 0, 0.5); /* semi-transparent black */
  z-index: 1;
}
```

```
.content {
  position: absolute;
  top: 50%;
  left: 50%;
  transform: translate(-50%, -50%);
}
```



```
color: white;  
font-size: 2rem;  
z-index: 2;  
}
```

-.Finished:-