

# CSS INTERVIEW QUESTIONS & SOLUTIONS

## 1. Explain the CSS Box Model.

**Answer:** The CSS Box Model describes how elements are structured with content, padding, border, and margin.

### Components:

- **Content** - The actual content (text, images)
- **Padding** - Space between content and border
- **Border** - Line around the padding
- **Margin** - Space outside the border

### Example:

CSS

```
.box {  
  width: 200px;  
  height: 100px;  
  padding: 20px;  
  border: 5px solid blue;  
  margin: 10px;  
  box-sizing: border-box; /* Include padding/border in width */  
}
```

## 2. What are CSS selectors and their specificity?

**Answer:** CSS selectors target HTML elements for styling. Specificity determines which styles apply when multiple rules target the same element.

### Specificity Hierarchy (highest to lowest):

1. Inline styles (1000)
2. IDs (100)
3. Classes, attributes, pseudo-classes (10)
4. Elements and pseudo-elements (1)

### Example:

CSS

```
/* Specificity: 1 */  
p { color: blue; }  
  
/* Specificity: 10 */  
.highlight { color: red; }
```

```

/* Specificity: 100 */
#main { color: green; }

/* Specificity: 111 */
#main .highlight p { color: purple; }

/* Specificity: 1000 */
<p style="color: orange;">Text</p>

```

### 3. Explain Flexbox and its properties.

**Answer:** Flexbox is a CSS layout method for arranging items in rows or columns with flexible sizing.

#### Container Properties:

- **display: flex** - Creates flex container
- **flex-direction** - Row or column arrangement
- **justify-content** - Main axis alignment
- **align-items** - Cross axis alignment
- **flex-wrap** - Wrapping behavior

#### Item Properties:

- **flex-grow** - Growth factor
- **flex-shrink** - Shrink factor
- **flex-basis** - Initial size
- **align-self** - Individual alignment

#### Example:

CSS

```

.container {
  display: flex;
  justify-content: space-between;
  align-items: center;
  flex-wrap: wrap;
}

.item {
  flex: 1 1 200px; /* grow shrink basis */
}

.special-item {
  flex: 2; /* Takes twice the space */
  align-self: flex-start;
}

```

## 4. What is CSS Grid and how does it differ from Flexbox?

Answer:

Feature	Flexbox	Grid
Dimension	1D (row OR column)	2D (rows AND columns)
Use Case	Component layout	Page layout
Content Flow	Content-first	Layout-first

Grid Example:

CSS

```
.grid-container {
  display: grid;
  grid-template-columns: 1fr 2fr 1fr;
  grid-template-rows: auto 1fr auto;
  grid-gap: 20px;
  height: 100vh;
}

.header {
  grid-column: 1 / 4; /* Span all columns */
}

.sidebar {
  grid-column: 1;
  grid-row: 2;
}

.main-content {
  grid-column: 2;
  grid-row: 2;
}
```

## 5. Explain CSS positioning (static, relative, absolute, fixed, sticky).

Answer:

Position Types:

- **static** - Default, normal document flow
- **relative** - Positioned relative to its normal position
- **absolute** - Positioned relative to nearest positioned ancestor
- **fixed** - Positioned relative to viewport
- **sticky** - Toggles between relative and fixed

### Example:

#### CSS

```
.relative {
  position: relative;
  top: 10px; /* Moves 10px down from normal position */
  left: 20px; /* Moves 20px right from normal position */
}

.absolute {
  position: absolute;
  top: 0;
  right: 0; /* Positioned at top-right of nearest positioned parent */
}

.fixed {
  position: fixed;
  bottom: 20px;
  right: 20px; /* Always 20px from bottom-right of viewport */
}

.sticky {
  position: sticky;
  top: 0; /* Sticks to top when scrolling */
}
```

## 6. How do you create responsive design with media queries?

**Answer:** Media queries apply CSS based on device characteristics like screen size.

### Example:

#### CSS

```
/* Mobile First Approach */
.container {
  width: 100%;
  padding: 10px;
}

/* Tablet */
@media (min-width: 768px) {
  .container {
    width: 750px;
    margin: 0 auto;
    padding: 20px;
  }
}
```

```
/* Desktop */  
@media (min-width: 1024px) {  
  .container {  
    width: 1000px;  
    padding: 30px;  
  }  
}
```

```
/* Print Styles */  
@media print {  
  .no-print {  
    display: none;  
  }  
}
```