

91 CSS Interview Question

Sr.No	Topic	Number of Questions
1	Basic CSS	10
2	Selectors	4
3	CSS Box Model	7
4	Layout	8
5	Typography	7
6	Colors and Backgrounds	9
7	CSS Transition	5
8	CSS Animation	5
9	CSS Transform	10
10	CSS Flex	8
11	Grid Layout	10
12	Responsive Design	8
Total		91

Here are 99 CSS questions that cover a wide range of topics, including theory, concepts, and practical questions. These questions will help you prepare effectively for an interview.

Basic CSS

Q.1] What does CSS stand for?

- CSS stands for Cascading Style Sheets.

Q.2] What is the purpose of CSS?

- CSS is used to style and format the appearance of web pages written in HTML and XML.
- It allows web developers to control aspects like **layout**, **colors**, **fonts**, and **spacing** of HTML elements.

Q.3] How do you link a CSS file to an HTML document?

- You can link a CSS file to an HTML document using the `<link>` element within the `<head>` section of the HTML document.

```
<link rel="stylesheet" type="text/css" href="styles.css">
```

- **rel** – It is used to indicate different relationships between the current document and the linked resource. Some common values are
 - stylesheet
 - icon
 - preload
 - alternate
- **type** – The `type` attribute specifies the MIME type of the linked resource.

Q.4] What is the syntax of a CSS rule?

- A CSS rule consists of a **selector** and a **declaration block**.



Q.5] What are the different ways to apply CSS to a web page?

CSS can be applied in three ways:

1. **Inline styles:** Directly within an HTML element.
2. **Internal stylesheets:** Within a `<style>` element in the `<head>` section of an HTML document.
3. **External stylesheets:** linked to an HTML document using the `<link>` element.

Q.6] What is an inline style?

- Inline styles are **CSS** styles applied directly to an individual HTML element using the `style` attribute

```
<p style="color: red; font-size: 16px;">This is a paragraph with inline styles.</p>
```

Q.7] What is an internal stylesheet?

- An internal stylesheet is CSS code written within a `<style>` element in the `<head>` section of an HTML document.
- It applies styles to that particular HTML document only.

Q.8] What is an external stylesheet?

- An external stylesheet is a separate CSS file linked to an HTML document using the `<link>` element.
- It allows styles to be applied across multiple HTML documents.

Q.9] How do you apply multiple styles to a single element?

- Multiple styles can be applied to a single element by separating each style declaration with a semicolon within the style attribute (for inline styles) or within the CSS rule:



Q.10] What is the difference between a class selector and an ID selector?

Class selector

- Defined using a dot (.) followed by a class name.
- Classes can be applied to multiple elements, and one element can have multiple classes.

```
.classname {  
    property: value;  
}
```

ID selector

- Defined using a hash (#) followed by an ID name.
- IDs are unique within a document and should only be applied to one element.

```
#idname {  
    property: value;  
}
```

Selectors

Q.11] What is a CSS selector?

- A CSS selector is a pattern that is used to select and style HTML elements based on various criteria such as **element types**, **IDs**, **classes**, **attributes**, and **relationships with other elements**

Q.12] What are the types of CSS Selectors

1. Universal Selector (*)

- ✓ Selects all elements on the page

- ✓ Selects all elements of a specific type

```
p {  
    property: value;  
}
```

3. Class Selector (.)

- ✓ Selects elements with a specific class attribute

```
.classname {  
    property: value;  
}
```

4. ID Selector (#)

- ✓ Selects a single element with a specific ID attribute

```
#idname {  
    property: value;  
}
```

5. Attribute Selector

- ✓ Selects elements based on the presence or value of an attribute.

```
input[type="text"] {  
    property: value;  
}
```

6. Pseudo-classes

- ✓ Selects elements based on their special state

```
a:hover {  
    property: value;  
}
```

8. Descendant Selector (space)

- ✓ Selects elements that are descendants of another element.

```
div p {  
  property: value;  
}
```

9. Child Selector (>)

- ✓ Selects elements that are direct children of another element.

```
div > p {  
  property: value;  
}
```

10. Adjacent Sibling Selector (+)

- ✓ Selects the first **element** that immediately follows a specified element, and both elements must have the same parent.

```
h2 + p {  
  property: value;  
}
```

11. General Sibling Selector (~)

- ✓ Selects **elements** that are siblings of a specified element.

```
h2 ~ p {  
  property: value;  
}
```

```
/* Make the first paragraph bold */  
p:first-child {  
  font-weight: bold;  
}  
  
/* Change the color of the last paragraph */  
p:last-child {  
  color: gray;  
}  
  
/* Change the background color of all input elements except those with type="text" */  
input:not([type="text"]) {  
  background-color: yellow;  
}
```

```
/* Change the color of a link when the user hovers over it */  
a:hover {  
  color: red;  
}  
  
/* Add a blue border to an input element when it has focus */  
input:focus {  
  border: 2px solid blue;  
}  
  
/* Change the color of the second child of a list item */  
li:nth-child(2) {  
  color: green;  
}
```

- It is a keyword added to the selectors which will allow to style the specific parts of an element's content.
- Pseudo classes - targets the entire element
- Pseudo elements – targets the specific part of an element
- Pseudo-elements are written with a double colon (::)

Here are some common pseudo-elements

```
/* Insert content before the content of a paragraph */  
p::before {  
  content: "Note: ";  
  font-weight: bold;  
}  
  
/* Insert content after the content of a paragraph */  
p::after {  
  content: " (end)";  
  font-style: italic;  
}  
  
/* Style the first line of a paragraph */  
p::first-line {  
  color: blue;  
  font-weight: bold;  
}  
  
/* Style the first letter of a paragraph */  
p::first-letter {  
  font-size: 200%;  
  color: red;  
}
```

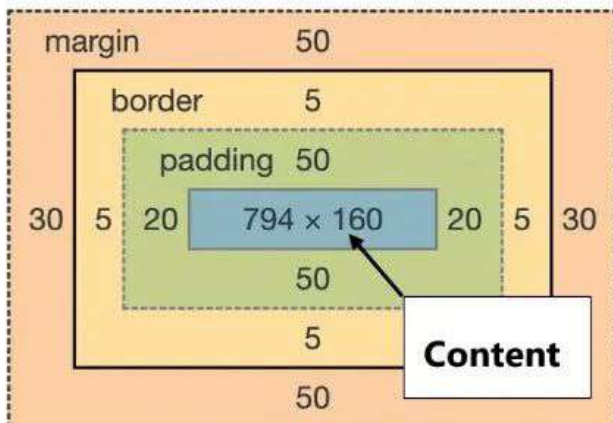
Q.13] What is a pseudo-class?

- A pseudo-class is a way to style an element in CSS **based on its state or position** without needing extra HTML code.

CSS BOX MODEL

Q.15] What is the CSS box model?

- The **CSS** box model is a fundamental concept that describes the rectangular boxes generated for elements in a web page, which includes the element's content, padding, border, and margin



Q.16] What are the components of the box model?

- Content:** The actual content of the box, where text and images appear.
- Padding:** The space between the content and the border.
- Border:** The border surrounding the padding (if any) and content.
- Margin:** The space outside the border, separating the element from other elements.

Q.17] What is the difference between margin and padding?

- Margin:** Space outside the element's border, creating space between elements.
- Padding:** Space inside the element's border, creating space between the content and the border.

Q.19] How do you center an element vertically?

- To center an element vertically, you can use Flexbox

```
container {  
  display: flex;  
  align-items: center;  
}
```

Q.20] What is the box-sizing property?

- The **box-sizing** property defines how the total width and height of an element are calculated.
- With **box-sizing: border-box;** the padding and border are included in the element's total width and height.

```
element {  
  box-sizing: border-box;  
}
```

Q.20] How do you create a rounded border?

- Use the **border-radius** property to create rounded borders

```
element {  
  border-radius: 10px;  
}
```

Q.22] How do you create a border around an

- Use the **border-radius** property to create rounded borders

```
element {
  border: 2px solid black;
}
```

LAYOUT

Q.23] What is the position property in CSS?

- The **position** property in CSS determines how an element is positioned in a document.
- It can take several values: **static**, **relative**, **absolute**, **fixed**, and **sticky**.

Q.24] Different Values of the position Property

static

[Refer a codepen](#)

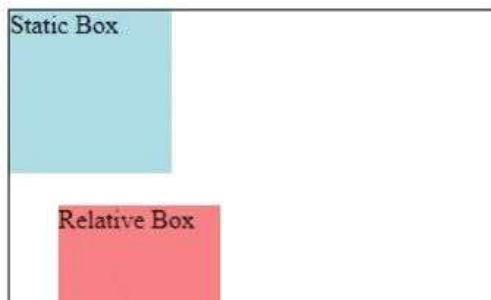
- Default positioning
- Elements are positioned according to the normal flow of the document.
- We can't use offsets like **top**, **right**, **bottom**, and **left**

relative

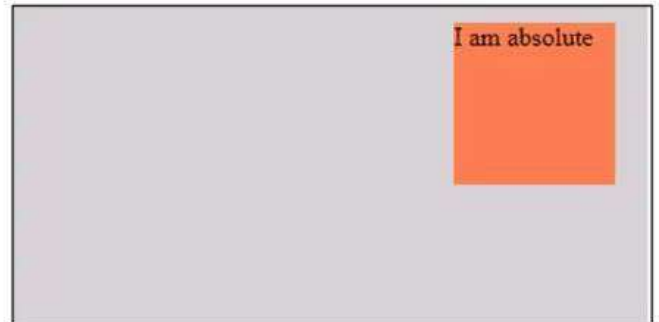
[Refer a codepen](#)

- Positioned relative to its normal position
- We can use offsets like **top**, **right**, **bottom**, and **left**

Here is the diagram which illustrates the **static** and **relative** position



- If there is no positioned ancestor, the element is positioned relative to the initial containing block (usually the **<html>** element).
- The ancestor can have **relative**, **absolute**, **fixed**, or **sticky** positioning
- It does not work with position **static**



fixed

[Refer a codepen](#)

- Positioned relative to the viewport, meaning it stays in the same place even when the page is scrolled



Sticky

[Refer a codepen](#)

- Acts like relative until the element reaches a specified scroll point, then it "**sticks**" in place.

Q.25] What is the float property?

- Used to position an element to the left or right within its container, allowing text and inline elements to wrap around it.

[Refer a codepen](#)

Q.26] How do you clear floats in CSS?

- To clear floats and ensure elements below a

- **Block** - Elements take up the full width available and start on a new line
- **Inline** - Elements only take up as much width as necessary and do not start on a new line.
- **inline-block** - Behaves like inline elements but can have a width and height.
- **Flex** - Turns the element into a flex container
- **grid** - Turns the element into a grid container

Q.28] How do you create a flexbox layout?

- To create a flexbox layout, you set `display: flex;` on the parent container and use various properties like `flex-direction`, `justify-content`, `align-items`, etc., on the child elements to control their layout.

Q.29] What is the flex property?

- The `flex` property is a shorthand for `flex-grow`, `flex-shrink`, and `flex-basis` combined, defining how a flex item will grow or shrink to fit the available space.

Q.30] How do you create a grid layout in CSS?

- To create a grid layout, you set `display: grid;` on the container and define its columns and rows using `grid-template-columns`, `grid-template-rows`, and place items within the grid using `grid-column` and `grid-row` properties.
- [Refer codepen](#)

Grid Item 1	Grid Item 2	Grid Item 3
Grid Item 4	Grid Item 5	Grid Item 6

- Using the `font-family` property in CSS.

```
selector {
  font-family: Arial, sans-serif;
}
```

Q.32] What is the font-family property?

- The **font-family** property specifies the font family for text.
- It allows you to define a prioritized list of fonts to use, **separated by commas**.
- If the first font is not available on the user's system, it falls back to the next one in the list.

Q.33] What is the font-family property?

- You specify a fallback font by listing multiple fonts in the `font-family` property, separated by commas.

```
selector {
  font-family: "Helvetica Neue",
  Helvetica, Arial, sans-serif;
}
```

Q.34] What is the font-size property?

- The **font-size** property in CSS specifies the size of the font.
- It can be set to various units like pixels (**px**), **ems** (**em**), or **percentages** (%).

Q.35] How do you change the color of the text?

- Text color is changed using the `color` property.

```
selector {
```


- Text alignment can be controlled using the `text-align` property.

```
.cover-background {
  background-image:
url('image1.jpg');
  background-size: cover;
}
```

```
.contain-background {
  background-image:
url('image2.jpg');
  background-size: contain;
}
```

Colors & Backgrounds

Q.37] How do you change the color of the text?

- You can set the background color of an element using the `background-color` property.

```
selector {
  background-color: #ffcc00;
}
```

Q.38] How do you set a background image?

- You can set a background image using the `background-image` property.

```
selector {
  background-image: url('image.jpg');
}
```

Q.39] How do you set the size of a background image?

```
.specific-size-background {
  background-image:
url('image3.jpg');
  background-size: 100px 100px; /*
width height */
}
```

- Contain:** Scales the background image to be fully visible within the element, maintaining its aspect ratio.
- Specific Size:** You can set specific dimensions for the background image using units like pixels, percentages, or any other CSS length units.
- Auto:** Maintains the original size of the background image.

```
.auto-size-background {
  background-image:
url('image5.jpg');
  background-size: auto;
}
```

- Multiple Background Sizes:** If you have multiple background images, you can specify sizes for each one.

```
.multiple-backgrounds {
  background-image:
url('image6.jpg'), url('image7.jpg');
  background-size: 50% 50%, cover; /*
First image is 50% by 50%, second image
covers the element */
}
```

Q.40] How do you repeat a background image?

- You can control the repetition of a background

- Other values include **repeat**, **repeat-x**, and **repeat-y**.

```
selector {
  background-repeat: no-repeat;
}
```

Q.41] What is the background-position property?

- Is used to specify the initial position of a background image within its container
- You can set the position using keywords, length values, or percentages
- This property helps in aligning the background image as desired within the element.

Syntax

```
background-position: x y;
```

- **x** represents the horizontal position.
- **y** represents the vertical position.

Common Values

```
/* Example using keywords */
.keyword-background {
  background-image: url('image.jpg');
  background-position: top right;
}

/* Example using length values */
.length-background {
  background-image: url('image.jpg');
  background-position: 20px 30px; /* 20px from the left, 30px from the top */
}

/* Example using percentages */
.percent-background {
  background-image: url('image.jpg');
  background-position: 75% 25%; /* 75% from the left, 25% from the top */
}

/* Example using multiple backgrounds */
.multiple-backgrounds {
  background-image: url('image1.jpg'), url('image2.jpg');
  background-position: left top, right bottom;
}
```

REMEMBER

- You can use multiple background images in an

Q.42] How do you create a gradient background?

- You can create a gradient background using the **background** property with a gradient function.

```
selector {
  background: linear-gradient(to right, red, yellow);
}
```

Q.43] How do you create a semi-transparent background?

- You can create a semi-transparent background using the **rgba()** function in the background-color property.

```
selector {
  background-color: rgba(0, 0, 0, 0.5);
}
```

Q.44] How do you create a shadow effect?

- You can create a shadow effect using the **box-shadow** property for element shadows or **text-shadow** for text shadows

```
box-shadow: h-offset v-offset blur spread color;
```

- **h-offset** : The horizontal offset of the shadow
- **v-offset**: The vertical offset of the shadow
- **blur** (optional): The blur radius
- **spread** (optional): The spread radius
- **Color** (optional): The color of the shadow

```
selector {
  box-shadow: 2px 2px 5px rgba(0, 0, 0, 0.3);
}
```

Q.45] How do you change the opacity of an element?