**13 Tricky CSS3 Interview Questions And Answers to Stand Out on Interview in 2018**

*Based on the latest data from Seek.com.au in 2018 proficient experience with CSS and HTML is listed for 95% of all web development job descriptions. Come along and follow us through the list of the most fundamental and tricky CSS Interview Questions and Answers from [FullStack.Cafe](https://www.fullstack.cafe/).*

**Top of Form**

**Bottom of Form**

**Q1: Describe floats and how they work**

Topic: **CSS**  
Difficulty: ⭐⭐

*Float* is a CSS positioning property. Floated elements remain a part of the flow of the web page. This is distinctly different than page elements that use absolute positioning. Absolutely positioned page elements are removed from the flow of the webpage.

#sidebar {

float: right; // left right none inherit

}

The CSS clear property can be used to be positioned below left/right/both floated elements.

🔗 **Source:** css-tricks.com  
💡 **Don't miss:** [50 More CSS Interview Questions & Answers](https://www.fullstack.cafe/CSS)

Q2: How is responsive design different from adaptive design?

Topic: **CSS**  
Difficulty: ⭐⭐⭐

Both *responsive* and *adaptive* design attempt to optimize the user experience across different devices, adjusting for different viewport sizes, resolutions, usage contexts, control mechanisms, and so on.

**Responsive design** works on the principle of flexibility — a single fluid website that can look good on any device. Responsive websites use *media queries*, *flexible grids*, and *responsive images* to create a user experience that flexes and changes based on a multitude of factors. Like a single ball growing or shrinking to fit through several different hoops.

**Adaptive design** is more like the modern definition of progressive enhancement. Instead of one flexible design, adaptive design detects the device and other features, and then provides the appropriate feature and layout based on a *predefined set of viewport sizes* and other characteristics. The site detects the type of device used, and delivers the pre-set layout for that device. Instead of a single ball going through several different-sized hoops, you’d have several different balls to use depending on the hoop size.

**Q3: How does CSS actually work (under the hood of browser)?**

Topic: **CSS**  
Difficulty: ⭐⭐⭐

When a browser displays a document, it must combine the document's content with its style information. It processes the document in two stages:

* The browser converts *HTML* and *CSS* into the *DOM (Document Object Model)*. The DOM represents the document in the computer's memory. It combines the document's content with its style.
* The browser displays the contents of the DOM.

**Q4: What does Accessibility (a11y) mean?**

Topic: **CSS**  
Difficulty: ⭐⭐⭐

**Accessibility (a11y)** is a measure of a computer system's accessibility is to all people, including those with disabilities or impairments. It concerns both software and hardware and how they are configured in order to enable a disabled or impaired person to use that computer system successfully.

Accessibility is also known as *assistive technology*.

**Q5: Explain the purpose of clearing floats in CSS**

Topic: **CSS**  
Difficulty: ⭐⭐⭐⭐

The clear CSS property specifies whether an element can be next to floating elements that precede it or must be moved down (cleared) below them.

*Clearing floats* (or clearfixing) basically forces the containing element to expand to contain its child elements. It thus forces the subsequent elements to appear below it.

**Q6: How do you optimize your webpages for print?**

Topic: **CSS**  
Difficulty: ⭐⭐⭐⭐

* Create a stylesheet for print or use media queries.

@media print {

...

}

* Add page breaks

<style>

.page-break {

display: none;

page-break-before: always;

}

</style>

Lorem ipsum dolor sit amet, consNullam aliquet. Aliquam ut diam...

<div class="page-break"></div>

Lorem ipsum dolor sit amet, consectetuer adipiscing elit....

**Q7: Can you explain the difference between coding a website to be responsive versus using a mobile-first strategy?**

Topic: **CSS**  
Difficulty: ⭐⭐⭐⭐

Note that these two 2 approaches are not exclusive.

Making a website responsive means the some elements will respond by adapting its size or other functionality according to the device's screen size, typically the viewport width, through CSS media queries, for example, making the font size smaller on smaller devices.

@media (min-width: 601px) {

.my-class {

font-size: 24px;

}

}

@media (max-width: 600px) {

.my-class {

font-size: 12px;

}

}

A mobile-first strategy is also responsive, however it agrees we should default and define all the styles for mobile devices, and only add specific responsive rules to other devices later. Following the previous example:

.my-class {

font-size: 12px;

}

@media (min-width: 600px) {

.my-class {

font-size: 24px;

}

}

A mobile-first strategy has 2 main advantages:

* It's more performant on mobile devices, since all the rules applied for them don't have to be validated against any media queries.
* It forces to write cleaner code in respect to responsive CSS rules.

**Q8: Explain the basic rules of CSS Specificity**

Topic: **CSS**  
Difficulty: ⭐⭐⭐⭐

* ID selectors have a higher specificity than attribute selectors.
* /\*wins\*/
* a#a-02 { background-image : url(n.gif); }
* a[id="a-02"] { background-image : url(n.png); }
* Contextual selectors are more specific than a single element selector.
* The embedded style sheet is closer to the element to be styled.
* The last rule defined overrides any previous, conflicting rules.
* p { color: red; background: yellow }
* p { color: green } // wins
* A class selector beats any number of element selectors.
* .introduction {} //wins
* html body div div h2 p {}
* The universal selector has a specificity of 0, 0, 0, 0.

**Q9: Is there any reason you'd want to use translate() instead of absolute positioning, or vice-versa? And why?**

Topic: **CSS**  
Difficulty: ⭐⭐⭐⭐

translate() is a value of CSS transform. Changing transform or opacity does not trigger browser reflow or repaint but does trigger compositions; whereas changing the absolute positioning triggers reflow. transform causes the browser to create a GPU layer for the element but changing absolute positioning properties uses the CPU. Hence translate() is more efficient and will result in shorter paint times for smoother animations.

When using translate(), the element still occupies its original space (sort of like position: relative), unlike in changing the absolute positioning.

**Q10: What the code fragment has the greater CSS specificity?**

Topic: **CSS**  
Difficulty: ⭐⭐⭐⭐

Consider the three code fragments:

// A

h1

// B

#content h1

// C

<div id="content">

<h1 style="color: #fff">Headline</h1>

</div>

What the code fragment has the greater specificy?

**Answer:**  
The CSS specificity:

* A is 0,0,0,1 (one element),
* B is 0,1,0,1 (one ID reference point and one element),
* C is 1,0,0,0 (since it is an inline styling).

Since 0001 = 1 < 0101 = 101 < 1000,

the third rule has a greater level of specificity, and therefore will be applied. If the third rule didn’t exist, the second rule would have been applied.

**Q11: What clearfix methods do you know? Provide some examples.**

Topic: **CSS**  
Difficulty: ⭐⭐⭐⭐⭐

**Method 1. The Old School Way**

Insert an empty element that has the clear property declared on it at the bottom of the container of floated elements.

.clear {

clear: both;

}

<div class="container">

<div class="el-1">I'm floated...</div>

<div class="el-2">I'm also floated...</div>

<br class="clear">

</div>

<div class="main">

Bravo, sirs and madams. I'm in the clear.

</div>

**Method 2: The Overflow Way**

Using the overflow property on our .container, we can actually force the container to expand to the height of the floated elements.

.container {

overflow: hidden; /\* can also be "auto" \*/

}

<div class="container">

<div class="el-1">I'm floated...</div>

<div class="el-2">I'm also floated...</div>

</div>

<div class="main">

Bravo, sirs and madams. I'm in the clear.

</div>

**Method 3: The “clearfix” Class**

The “clearfix” (which means fixing the clearing of floats) defines a .clearfix class in our stylesheet that we can apply to any float-containing element. This will force the container element to expand, pushing subsequent elements beneath it. It uses the CSS pseudo-elements ::before and ::after.

.clearfix:before,

.clearfix:after {

content: "";

display: table;

}

.clearfix:after {

clear: both;

}

.clearfix {

zoom: 1; /\* ie 6/7 \*/

}

<div class="container clearfix">

<div class="el-1">I'm floated...</div>

<div class="el-2">I'm also floated...</div>

</div>

<div class="main">

Bravo, sirs and madams. I'm in the clear.

</div>

**Method 4: The Future contain-floats Value**

The W3C specification has added a new value to the min-height property (and to other min/max properties), in order to help solve this problem. It looks like this:

.container {

min-height: contain-floats;

}

This basically will do the same thing as the clearfix or the overflow method, but with a single line of code.

**Q12: How to style every element which has an adjacent item right before it?**

Topic: **CSS**  
Difficulty: ⭐⭐⭐⭐⭐

Use **The Lobotomized Owl Selector**:

\* + \* {

margin-top: 1.5em;

}

The idea is that only elements that have a previous sibling inside a container get margin on top.

Another example:

<div class="container">

<div>Box 1</div>

<div>Box 2

<div>Sub-box 1</div>

<div>Sub-box 2</div>

</div>

<div>Box 3</div>

</div>

Css:

\* + \* {

margin-top: 0.5em;

}

html, body {

height: 100%;

}

body {

font-family: "Roboto", san-serif;

margin: 0;

overflow: hidden;

}

.container {

background: #ccc;

min-height: 100%;

padding: 2em;

}

.container div {

background: #fff;

border: 2px dashed #666;

padding: 2em;

width: auto;

}

.container div:nth-child(2) > div {

background: #ccc;

}

**Q13: Write down a selector that will match any links end in .zip, .ZIP, .Zip etc...**

Topic: **CSS**  
Difficulty: ⭐⭐⭐⭐⭐

Use *case-insensitive attribute matching*, i. This is a new feature in CSS Selectors Level 4.

Consider:

a[href$=".zip" i]:after {

content: '↓'

}

🔗 **Source:** css-tricks.com