1. **What is CSS and how does it work with HTML?**

CSS stands for Cascading Style Sheets with an emphasis placed on “Style.” While HTML is used to structure a web document (defining things like headlines and paragraphs, and allowing you to embed images, video, and other media), CSS comes through and specifies your document’s style—page layouts, colors, and fonts are all determined with CSS.

1. **What are selectors and what are their different types?**

* CSS selectors are used to "find" (or select) the HTML elements you want to style.
* We can divide CSS selectors into five categories
* Simple selectors (select elements based on name, id, class)
* Combinator selectors (select elements based on a specific relationship between them)
* Pseudo-class selectors (select elements based on a certain state)
* Pseudo-elements selectors (select and style a part of an element)
* Attribute selectors (select elements based on an attribute or attribute value)

**3. What is CSS selector specificity and how does it work?**

If there are two or more conflicting CSS rules that point to the same element, the browser follows some rules to determine which one is most specific and therefore wins out. Think of specificity as a score/rank that determines which style declarations are ultimately applied to an element.

**4. Describe z-index and how stacking context is formed.**

A stacking context is created when an element is positioned and assigned a z-index value other than auto, or when an element has an opacity value less than 1. In WebKit and Chrome 22+ an element with a fixed position always creates a stacking context, even when the z-index value is auto.

**5. Describe BFC (Block Formatting Context) and how it works.**

A block formatting context is a part of a visual CSS rendering of a web page. It's the region in which the layout of block boxes occurs and in which floats interact with other elements. A block formatting context is created by at least one of the following: The root element of the document html.

**6.Have you ever used a grid system, and if so, what do you prefer?**

Before Flex became popular (around 2014), the float-based grid system was the most reliable because it still has the most browser support among the alternative existing systems (flex, grid). Bootstrap was using the float approach until Bootstrap 4 which switched to the flex-based approach. As of writing (2020), flex is the recommended approach for building grid systems and has decent browser support.

**7. Have you used or implemented media queries or mobile specific layouts/CSS?**

Yes. An example would be transforming stacked pill navigation into fixed-bottom tab navigation beyond a certain breakpoint**.**

**8. Explain how a browser determines what elements match a CSS selector.**

A CSS pseudo-element is a keyword added to a selector that lets you style a specific part of the selected element(s). They can be used for decoration (:first-line, :first-letter) or adding elements to the markup (combined with content: ...) without having to modify the markup (: before, after).

**9. Describe pseudo-elements and discuss what they are used for.**

This part is related to the above about writing efficient CSS. Browsers match selectors from rightmost (key selector) to left. Browsers filter out elements in the DOM according to the key selector and traverse up its parent elements to determine matches. The shorter the length of the selector chain, the faster the browser can determine if that element matches the selector.

**10. Explain your understanding of the box mode.**

The CSS box model describes the rectangular boxes that are generated for elements in the document tree and laid out according to the visual formatting model. Each box has a content area (e.g. text, an image, etc.) and optional surrounding padding, border, and margin areas.

**11. What does \* { box-sizing: border-box; } do? What are its advantages?**

By default, elements have box-sizing: content-box applied, and only the content size is being accounted for. box-sizing: border-box changes how the width and height of elements are being calculated, border and padding are also being included in the calculation.

**12. What is the CSS display property and can you give a few examples of its use?**

CSS Display properties are none, block, inline, inline-block, flex, grid, table, table-row, table-cell, list-item. For example, none:- Does not display an element (the element any longer affects the layout of the document). All child elements are also no longer displayed. The document is rendered as if the element did not exist in the document tree

block:- The element consumes the whole line in the block direction (which is usually horizontal)

inline elements can be laid out beside each other.

**13. What's the difference between inline and inline-block?**

Compared to display: inline, the major difference is that display: inline-block allows to set a width and height on the element. Also, with display: inline-block, the top, and bottom margins/paddings are respected, but with display: inline they are not.

**14.What's the difference between the "nth-of-type()" and "nth-child()" selectors?**

 As a general rule, if you want to select an interval of a selector regardless of the type of element it is, use nth-child. However, if you want to select a specific type only and apply an interval selection from there, use nth-of-type.

**15. What's the difference between a relative, fixed, absolute and statically positioned element?**

A positioned element is an element whose computed position property is either relative, absolute, fixed or sticky.

static - The default position; the element will flow into the page as it normally would. The top, right, bottom, left and z-index properties do not apply.  
  
relative - The element's position is adjusted relative to itself, without changing the layout and thus leaving a gap for the element where it would have been had it not been positioned.  
  
absolute - The element is removed from the flow of the page and positioned at a specified position relative to its closest positioned ancestor if any, or otherwise relative to the initial containing block. Absolutely positioned boxes can have margins, and they do not collapse with any other margins. These elements do not affect the position of other elements.  
  
fixed - The element is removed from the flow of the page and positioned at a specified position relative to the viewport and doesn't move when scrolled.

**16. What existing CSS frameworks have you used locally, or in production? How would you change/improve them?**

Bootstrap - Slow release cycle. Bootstrap 4 has been in alpha for almost 2 years. Add a spinner button component, as it is widely used.  
Semantic UI - Source code structure makes theme customization extremely hard to understand. Its unconventional theming system is a pain to customize. Hardcoded config path within the vendor library. Not well-designed for overriding variables unlike in Bootstrap.  
Bulma - A lot of non-semantic and superfluous classes and markup are required. Not backward compatible. Upgrading versions break the app in subtle manners.

**17.Have you used CSS Grid?**

 Yes, we should absolutely use the CSS Grid layout!