

Short Answer:

Answer the following questions with complete sentences in **your own words**. You are encouraged to conduct your own research online or through other methods before answering the questions. If you research online, please consult multiple sources before you write down your answers.

1. What is flow?

A flow refers to the relationship between elements and how this relationship affects the position of these elements

2. How are inline elements and block elements displayed in flow? Out of flow?

Both inline elements and block elements are stacked along each other in a flow. When it is out of flow, elements act independently, have their own layers. And don't take up space on the webpage

3. When is *float* useful?

It is useful when you want to wrap text or inline elements around images or element

4. What would happen if I set *float:left* to a block element?

The element will move to the left side of the webpage and the text will wrap around it from the left side

5. What does *clear* do?

Clear is used on a float element and it ends up appearing below the floating element

6. How is *position* useful?

7. What is the difference between *position:fixed* and *position:absolute*?

Both of them are out of flow. However, *position:fixed* is fixed with respect to the viewport. It stays where it is, even if the document is scrolled. On the other hand, *Position:absolute* is relative to viewpoint and it stays on the same place while scrolling.

8. What does *z-index* do?

Z-index specifies the stack order of an element

9. What is Flexbox?

Flexbox is a one-dimensional layout model. It lays, aligns and distributes items inside a container

10. What is a flex container? Can you explain some flex container properties?

Flex container is a flexbox that defines the properties of the ancestor element by setting its display to either flex or inline-flex.

flex-direction: it sets the direction of elements present in a flexbox,

flex-wrap: it determines whether the elements should be wrapped around the flex line or not

flex-flow: shorthand property for setting flex-direction and flex-wrap.

justify-content: it aligns them across the main-axis.

align-items: it sets the arrangement of elements in a container vertically.

align-content: it changes the behavior of the flex-wrap property. It is similar to the justify content property but it aligns the flex elements vertically

11. What are flex items? Can you explain some flex item properties?

Flex items define the properties of the successor elements. There can be numerous flex items present inside a flex container.

order: it defines the order of a flex element relative to other elements present in a flexbox,

flex-basis: it defines the starting length of a flex item

flex-grow: it defines the extent an element will grow relative to the other elements present inside the container

flex-shrink: it defines the extent an element will shrink relative to the other elements present inside the container

flex: it is the concise way to shorthand to define flex-grow/flex-shrink/flex-basis

align-self: it defines the arrangement of items inside a flex container,

12. What is Grid?

Grid allows its users to arrange elements appearing on a website into various rows and columns.

13. What is a grid container? Can you explain some grid container properties?

Grid container is a parent element that holds grid items within rows, and columns.

grid-template-columns: it defines the number of columns and their widths

grid-template-rows: it defines the number of rows along with their height

justify-content: it aligns the entire grid within a container horizontally

align-content: it aligns the entire grid within a container vertically

column-gap: it determines the distance between columns

row-gap: it determines the distance between rows.

gap: it allows you to define column-gap and row-gap in one spot

14. What are grid items? Can you explain some grid item properties?

Grid items represents the child element that is present inside a grid container

grid-column-start: this determines which column the grid item will start displaying

grid-column-end: this determines which column the grid item will end displaying

grid-column: This is a concise way to define grid-column-start and grid-column-end

grid-row-start: this determines which row the grid item will start displaying

grid-row-end: this determines which row the grid item will end displaying

grid-row: This is a concise way to define row-column-start and row-column-end

15. What are the major differences between CSS transitions and animations?

CSS transitions are generally best for simple from-to movements, while CSS animations are for more complex series of movements.

16. What is responsive web design (RWD)? What are some examples of RWD on a website? How do we achieve this?

RWD is a design approach where you use HTML & CSS to make the website UI look great on all devices and screen sizes, such as a smart phone.

Some examples include Mobile First approach so that the website will be readable on a mobile phone and defining the viewpoint in the meta tag of the html.

RWD can be achieved by editing Flexbox, Grid, Images ,font-sizes, and Media queries

17. What is a CSS preprocessor? What are the advantages and disadvantages, if any, to using them over plain CSS?

CSS preprocessor will extend the functionality of vanilla CSS by adding logical syntax that you may see in other programming language.

The advantages include having features that make CSS structure more readable and easier to maintain through variable, Mixins, function, avoiding repetition, joining multiple files, and Nested Syntax,

The disadvantages include debugging, slow development, large CSS file,

18. How do we declare a SCSS variable? How do we declare CSS variables?

We declare a SCSS variable with a \$ before the variable. we declare CSS variable with the var() function

19. What is the usage of '&' in SCSS?

The & is the parent selector in nested CSS block

20. What is @mixin? What are some use cases?

A mixin allows you to create CSS code that can be reused. Some use cases include Media Query, hiding an element, hiding overflow, creating a circle, and vendor prefixes

21. What is @extend? What are some use cases?

@extend allows you share a set of CSS properties from one selector to another. Some use cases include duplication, moving multiple classes out of HTML, extending complex selectors,

22. What is the usage of !important? What are some use cases?

!important allows you to prioritize a particular style. Some use cases include utility classes and User style sheets

23. Could you explain accessibility, usability, and inclusion? Give some examples of each one in terms of web design.

Accessibility, usability and inclusion are meant to make websites accessible, and appropriate for everyone and all demographic.

Some examples include making it accessible for people with disabilities such as the blind, and color-blind, and making it appropriate for all cultural demographic whether it be sex, gender, ethnicity

