PAGE NO :
DATE :

CSS interview questions

1) what is the purpose of css media queries.?

A) The purpose of the media queries is to create responsive web designs that adapt to different screen sizes, resolutions and devices.

2) How do you write a basic media query in css.

A) we write basic media query by using @media and the type & condition.

3) Explain the different between max-width and min-width in media queries.

Max-width	Min-width
* it apply style when it width is less than or equal to specified value	* it apply style when width greater than equal to specified value.
* style will be applied until the specified width reached	* within the min-width media query is not apply until specied width is reached.

4] what is the purpose of the viewport meta tag in responsive web design?

A: The HTML viewport meta tag is used for creating responsive website. so that web page can adjust its width according to viewport.

5] How can you apply different styles for landscape and portrait orientations using media queries.

A: we can apply style by using css media queries based on orientation of the device.

6) Explain the concept of a mobile-first approach in responsive design.

A 'Mobile-first' approach involves designing a desktop site starting with the mobile version, which is then adapted to larger screens.

7) What are common breakpoints used in responsive design?

The common breakpoints are:

* Extra small (xs): Range 0PX to 575 PX
* Small (sm): Range 576 to 767 PX
* Medium (md): Range 768 to 991 PX
* large (lg): Range 992 to 1199 PX
* Extra large (xl): Range 1200PX and above.

8) What is the purpose of the rem unit in media queries?

The purpose of using rem in media queries. you can create flexible and adaptable designs that respond to changes in font size and provide a better experience for user.

9) How can you combine multiple media queries in css?

In CSS, you can combine multiple media queries using logical operators such as 'and', 'or', and 'not'.

→ and: The 'and' keyword to create a media query that matches all specified conditions.

→ OR: ',' (comma) as a logical OR to create a media query that matches either of the specified conditions.

→ NOT: The 'NOT' keyword to create a media query that matches the inverse of specified condition.

10) What is the significance of the all keyword in media queries?

A) The 'all' keyword in media queries is a universal type that includes all media types. it is the default value if no media type is specified. when you don't specify a media type in a media query. it's assumed to be all.

11) How do you use media queries to apply styles only for print stylesheets?

→ first we use media type as print after we apply whatever the styles want to print.

Ex: @media print {
body {
background-color: yellow;
color: white;
margin: 0;
}
}

@page is used to modify different aspects of printed pages.

12) what is the differences b/w screen and print in media queries?.

screen	print
* This media-type is for on-screen viewing	* This media type is for print style sheets.
* used for computer screen tablets, smartphone etc	* It is effectively work on printable pages
* This is the default media type. if we don't specify any one in media query	* we do print media type as adjustment the font size, hide non-essential elements.

PAGE NO :
DATE :

13) How can you hide an element on a specific screen size using media query?

-> we can hide an element by using display property as none. we Apply this property inside the media query it comes true when the specified condition is met.

14) Explain the role of the orientation property in media query

-> This property is particulary useful for adjusting the layout and styling of a webpage depending on the device's orientation.

15) How do you target specific devices using media queries?

-> we are targetting the specific devices by using condition and apply styles based on screen size, resolution & other features.

16) what is the purpose of the not keyword in media queries?

-> The not keyword in css media queries apply these styles only if the condition is true.

ex: @media not screen and (min-width: 600px) {
color: green;
font size: 30px;
}

17) How can you use media queries to adjust font sizes for different screen sizes?

-> we adjust font size for different screensize using Media queries in css

forex:

Body {
default font size of all screen
}

body {
font-size: 16px;
}

PAGE NO :
DATE :

18) what is the box-sizing property in css and what does it control?

→ The css box-sizing property is used to adjust or control the size of an element.
There are two main values for the 'box-sizing'.

(1) content-Box.
(2) Border-Box.

(i) content-Box (default).
* in content Box it consists only width and height
* it does not includes Border & padding.

(ii) Border-Box
* In Border-Box it includes only width & height.
* And it also include Border & padding.

19) Explain the difference between box-sizing: content-box; and box-sizing: border-box;

→

content-Box	Border-Box
* it includes height & width	* it includes height & width & also Border & Padding.
* content-Box is doesn't overlap.	* Border-Box can overlap.
* This is the default behavior in css	* it preferred for modern web development.

20) ~~Also define that box-sizing: border-box; as the default box model for your page~~

PAGE NO :
DATE :

20) How does the box-sizing property affect the calculation of an elements width & height in CSS?

-> In the box sizing property has two values in content box calculate : Total width = width of content in ~~border~~.

Border box calculate : Total width = width of content & width of border + width of padding.

21) why might you choose box-sizing: border-box: as the default box modern for your project.

-> Because the box-sizing property allow us to include the padding & border include elements total width & height.

22) Different between normalizing and resetting?

-> Normalizing

* The goal of normalizing css is to apply built-in-browser styling across all browsers uniformly

resetting

* Rest are intented to remove all default browser styles.

23) what is a CSS combinator, & how is it used in a selector?

-> The CSS 'combinator are something realationship b/w Multiple selector.

* Descendent combinator : the descendent combinator selects all elements that are descendants of a specified element.

* child combinator : The child combinator select all elements that are a direct child of a specified element.

PAGE NO :
DATE :

③ Adjacent sibling: The adjacent sibling combinator selects an element that is immediately preceded by specified element.

24) Differentiate b/w descendant and child combinatory in CSS selectors

-> Descendant	child
* The descendant combinator matches a child, grandchild and so on	* The child combinator selects elements that are direct children of another element.
* selects direct children of a specified element	* select all descendant of a specified element.

25] Explain the purpose of the adjacent sibling combinator. (+) In CSS provide a use case:

-> The adjacent sibling selector is used to select an element that is directly after another specific element.

-> Sibling elements must have the same parent element and 'adjacent' means "immediately following".

Ex: div +p {
background-color: 'yellow';
}

26) How does the general sibling combinator (~) differ from the adjacent sibling combinator (+)?

-> general sibling

* The general sibling selector in CSS is used to select all general siblings of an element that follows that element
* it represented by using a tildle (~) b/w two selectors.

adjacent sibling

* it is used to select all the immediate siblings of an element in a webpage.
* It is represented by using plus(+) b/w two selectors.

27) what is the significance of the child combinator (>) in CSS selectors?

* The child selectors in CSS is used to select elements in a webpage that are the direct children of a specified element
* in other words, it selects only those elements that are nested directly inside a specified element and not those which are nested inside a child or grand child of an element.

28) How can you select all paragraphs that are direct descendants of a div using a CSS selectors?

-> TO select all paragraphs that direct descendats of a div using the child selector (>) along with the 'p' selector for paragraphs.

Ex:
```
div > p {
    /* style */
}
```

29) Provide an example of using the descendant combinator to style nested elements.

Ex:
HTML:
```
<div class = container>
```

<P> this is a paragraph </P>

<P> this is a paragraph </P>
 this is a span inside the nested div

CSS part

```
.container p {
    /* your styles here */
    color: blue;
}
```

10) Explain how the space b/w two selectors represents a descendant combinator.

→ * In CSS, the space b/w two selectors is known as the descendant combinator.
* it is used to select all elements that are descendants of the first specified element.

31) How would you select an element that is the immediate next sibling of another element in CSS?

→ In CSS, you can use the adjacent sibling combinator (+) to select an element that is the immediate next sibling of another element.

32) In what scenarios would you choose one combinator over another, and what are the considerations when using combinators for efficient CSS selectors

→ choose CSS combinations based on the specific relationship & proximity b/w elements, balancing performance, specifity & maintainability.

33) Explain the concept of CSS pseudo-selectors Provide example of commonly used pseudo-selectors and their purpose.

* CSS pseudo-selectors are special keywords that are used to select and style a specific portion of an elements content or state of an element.

(i) :hover: Selects and styles an element when the user hover over it.

Ex: a:hover {
color: red;
}

ii) :active: Selects and styles an elements while it is being activated.

Ex: button:active {
background-color: green;
}

iii) :focus: Selects and styles an element that has focus (mainly keyboard)

v) :nth-child(): Selects elements based on their position within a parent element (even/odd)

Ex: :nth-child(odd) {
background-color: red;
}

34) Differentiate b/w pseudo-classes and pseudo-elements in CSS. Give examples of each.

PAGE NO :
DATE :

Pseudo-classes	Pseudo-elements
* it uses a single colon(:) before the name	* it uses a double colon (::) before the name.
* it select elements based on their state & position	* it select parts of an element's content or create additional content
* They are applied to entire elements	* They are applied to specific part of an element 'content'.
* Ex: a:hover { color: blue; }	* Ex: p::before { content: ">>"; }

35] How can you use the 'nth-child Pseudo-class to select specific elements in a list or containers? Prove an example.

→ The ':nth-child' Pseudo class in CSS allows you to select & style elements based on their position within a part container

Ex: Selecting Odd and Even Elements

```
li:nth-child(odd) {
    background-color: red;
}
```

PAGE NO :
DATE :

Javascript

① What are the primitive data types in Javascript ?

The primitive data is used to store the single data or one data.

The primitive data types are:

* Numbers
* Strings
* Boolean
* undefined.
* Symbol.

② Explain the difference b/w null and undefined in Javascript.

→ null : is a value can explicitly set to indicate the absence of any meaningfull value.

* it explicitly set by programmer

→ undefined : means a variable has been declared but has not been assigned value.

* It is built-in-primitive value in Javascript

3] How do you check the data type of a variable in Javascript.

→ We can check the datatype of variable using the 'type of' Operator.

4] Explain the concept of truthy & falsy values in Javascript. Provide Examples.

→ In Javascript categorized truthy & falsy value in Boolean data type.

Ex:

```
let x = true
console.log(x).
```

PAGE NO :
DATE :

5] what is the difference between == and === operators in Javascript, and how do they relate to data types?

A "==" : This operator compares values for equality.
"===" : This operator comparing the both values & type of operands.

Example;

console.log(6=="6") comparing values.
it returns true.
console.log(10==="10") comparing the types
=> it returns false..

7] Explain the difference b/w the ++x and x++ increment operators in Javascript.

-> ++x and x++ = Both are Increment operator.
++x -> This is pre-increment operator
x++ -> This is Post increment operator.

8] How does Javascript handle NAN(Not a Number) values, and how can you check if a value is nan?

=> In Javascript 'NaN' (Not a number) is a special value representing the result of an operation that cannot produce a meaningful numeric result.

9] Explain the concept of type coercion in javascript. Provide examples.

=> TYPE coercion in javascript is the automatic conversion of values from one data type to another during runtime of a program.

PAGE NO :
DATE :

There are two types
(i) Explicit Type conversion
(ii) Implicit TYPE conversion

Ex: Implicit coercion:
This occurs automatically during operation, and the Javascript tries to convert one or both operand to a common data type.

```
let y = 50;
    z = "20"
console.log(y+z)
```

Explicit coercion
Developers can also explicitly coerce value from one type to other using function.

```
let x = 489
    y = String(x)
console.log(typeof(x))
console.log(typeof(y))
```

10) what is the purpose of the undefined datatype & when might it be explicitly assigned to a variable?
→ undefined is a primitive datatype, it occurs when we declare a variable but we don't assign the value.

Ex:

```
let z;
console.log(z)
```

11) How do you create and use template literals (String Interpolation) in Javascript.
→ we use template literals enclosed within the backticks (` `) instead of single or double quotes & we use in ${variable} Symbol & we use curly braces.

Ex: let x = 10
y = 20
console.log ('The sum of ${x} & ${y} is ${x+y}')

12) what is hoisting?
A: The process of moving all the element declaration to the top of the scope.

13) what is IIFE?
-> IIFE is stands for "Immediately invoking function Expression" also called as self invoking function and function call itself function.

14) what is meant by default parameter passing.
-> Default parameter means we are allow to specify default values for function parameter when the function is called. if value is not provided for parameter. Then default value is used.

Ex: function info (name, age = 16) {
console.log ('my self ${name} and my age is ${age}')
} info ('Deepak')

15) what is default return value?
-> The default return value is undefined
Ex: function a (x) {
return
} a (2)
console.log (a (2))

PAGE NO :
DATE :

16) How to pass unlimited number of parameters to a function

-> We pass unlimited number of parameters by using Arguments keyword. and in the Arrow function we use rest Parameter (...).