Yatheesh. D.C

30/01/2024.

Experiment No. | Date

Name of the Experiment | Page No.

01 What is the purpose of CSS media queries?

The purpose of media queries is to create responsive web design that adapt to different screen sizes resolutions and device.

02 How do you write a basic media query in CSS?

By using @media and the type and condition.

Ex:-
```
@media screen and (max-width : 992px) {
    body {
        background-color: blue;
    }
}
```

03 Explain the difference b/w max-width and min-width in media queries.

Max-width	Min-width
* The max-width property lets you specify an element's maximum width. This means that an element can increase in width until it reaches a specific unit, at which point it should fix its width to that until	* In contrast to the max-width property specifies the minimum width. It indicates the minimal possible width for an element.

chandra's

* It apply style when it the width is less than (or) equal to specified value. * Once the viewport width is exceeds the specified value media query will no longer apply. * ~~so, simply put, max-width~~ Max-width <= setwidth	* It apply style when the width greater than equal to specified value. * Within the minimum width media query is not apply untill specified width is reached. * Min-width >= set width.

04] What is the purpose of the viewport meta tag in responsive web design?

* A viewport is the user's visible area of a webpage.
* A viewport meta tag is HTML code that tells browsers how to control viewport dimensions and scaling.
* It's a key ingredient of responsive web design and ensures your content is easy to view. So that webpage can adjust its width according to viewport.

05] How can you apply different styles for landscape and portrait orientations using media queries?

=> we can apply style by using CSS media Queries based on orientation of the device.

06] Explain the concept of a mobile first approach in responsive design.

Experiment No.
Name of the Experiment

Date
Page No.

* A mobile-first approach involves designing a desktop site starting with the mobile version, which is then adapted to larger screens.
* Generally speaking, a mobile-first approach means building your website with your mobile users in mind, with the main goal of improving these mobile user experience on your site.

07 What are the common break points used in responsive design?
Mobile devices - 320 px - 480 px
ipads, Tablet - 480 px - 768 px
Small screen, laptops - 769 px - 1024 px
Desktop, large screen - 1025 px - 1200 px
Extra large screen, TV - 1200 px and more.

08 What is the purpose of the rem units in media Queries?
It is the measurement unit that refers to font-size of the root element of a document like 1rem = 16px, 2 rem = 32 px.

09 How you can combine multiple media query in css?
By using logical operator like 'and', 'or' 'not'
AND - Used if both conditions are true.
OR - Used only atleast one condition - is true.

chandra's

NOT- Used to apply if the condition is not true.

10] What is the significance of the 'all' keyword in media Queries?

The 'all' keyword in media Queries is a Universal media type keyword that target all devices regardless of their type (screen, print etc)

11] How do you use media Queries to apply styles only for ~~print~~ print stylesheets?

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

Ex:
```
@media print {
    background color: white;
    color: black;
    margin: 0
}
```

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

12] What is the different b/w screen and print in media Queries?.

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

Experiment No. Date

Name of the Experiment Page No.

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 display: none, we apply this property inside the media Query it comes true when the specified condition is met.

Ex:
```
@media (max-width: 600px){
    .cd{
        display: none;
    }
}
```

14] Explain the ride of orientation property in media Query.

The orientation property in CSS media Queries is used to apply style based on orientation of devices whether it is ~~pro~~ portrait (or) landscape mode.

Orientation property has 2 values.

(i) portrait: Height is greater than (or) equal to width.

(ii) Landscape: Width is greater than (or) equal to height.

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 not true.

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

17] How can you use media Queries to adjust font sizes for different screen sizes?

We adjust font size for different screen size using media Queries in css.

Ex:

```
body {
    font-size : 16px;
    }
@ media (min-width : 600px) and (max-width :
                                    760 px) {
    body {
        font-size : 50 px;
    }
}
```

Experiment No.
Name of the Experiment

Date
Page No.

18] What is the box-sizing property in CSS & what does it control?

Box-sizing property in CSS control how the width & height of element is calculated.

Box-sizing property has two values
* content Box
* Border Box

(i) Content Box:- This is the default sizing.
* width & height properties only apply to elements content box.
* It doesnot overlap the content
Formula:- Total width = width of content.

(ii) Border Box:- The width & height property include the content padding & border.
* It overlaps the content when border size is ~~inner~~ increase.

Formula:- Total width = width of content + width of Border + width of padding.

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

content-box	Border-box
* It consists height & width.	* It consists height, width of border, padding and content.

chandras

* It doesnot overlap the content. * The default behaviour in CSS.	* It overlap when we give the border & padding more than content. * It pretered for modren web development.

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

~~How does the box-sizing~~

In the box-sizing property has two values in content box. Total width = width of content. In border box,

Total width = width of content + width of border + width of padding.

21] Why might you choose box-sizing: border-box; as the default box model for your project?

=> Because the box-sizing property ~~for your proj~~ allow us to include the padding & border include elements total width & height.

Experiment No.
Name of the Experiment

Date
Page No.

22 Difference between normalizing and resetting

Normalizing	Resetting
* The goal of normalizing css is to apply built-in browser styling across all browser uniformly.	* Resets all of the user's agent-bundled styles for the browser.
* Resetting makes a lot of pointless overrides in the styling and has a long chain of selectors.	
* Since it makes use of the User Agent's styles. there aren't many long chains of css selectors to be noticed while normalizing.	* Resetting makes a lot of pointless overrides in the styling and has a long chain of selectors.
* It is simple to debug while normalizing	* As it is practically impossible to find bugs, debugging is challenging.

23 What is a css combinator, and how is it used in a selector?

A combinator is something that explains the relationship between the selectors.

chandra's

(i) Descendant selector:-

The descendant selector matches all elements that are descendant of a specified element.

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

(ii) Child Selector:-

The child selector selects all elements that are the children of a specified element.

Ex:- div > p {
    background-color: yellow;
}

(iii) Adjacent sibling selector:-

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

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

(iv) General sibling selector:-

The general sibling selector selects all elements that are next sibling of a specified element.

Ex:- div ~ p {
    background-color: yellow;
}

Experiment No.
Name of the Experiment

Date
Page No.

24 Differentiate between descendant and child combinators in CSS selectors. Provide examples for each.

Descendant	child.
* It selects all descendants of a specified element, regardless of how deeply nested they are.	* It selects direct children of a specified element.
* It uses a space b/w the selectors.	* It uses the greater-than symbol (>)

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;
}
```

chandra's

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

General sibling combinator (~):-

* The general sibling selector in CSS is used to select all general siblings of an element that follows that element.
* In the other words, it selects those elements in a document that are nested under the same parent element as a specified element and is also present after that element.
* The general sibling selector is especially useful as it can select any sibling element that comes after a specified element, irrespective of the number of elements blw them.
* It is represented by using a tildle (~) blw two selectors.

Adjacent sibling combinator (+):-

* It is used to select all the immediate siblings of an element in a webpage.
* In the other words, it selects those elements that are nested under the same parent as a specified element and also appear immediately after that element.
* It selects only the element that immediately follows an element.
* It is represented by using plus (+) blw two selectors.

Experiment No.
Name of the Experiment

Date
Page No.

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

* The child selector 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) grandchild of an element.

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

To select all paragraphs that are direct descendants of a 'div' using the child selector (>) along with the 'p' selector for paragraphs

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

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

HTML:
```
<div class = container>
<p> This is a paragraph </p>
<div>
    <p> This is a paragraph </p>
```

chandra's

```
<span> This is a span inside the nested div </span>
</div>
</div>
```

CSS part:-

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

30] Explain how the space between 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.
* The descendant combinator is represented by a whitespace character.

Ex:-

```
selector1 selector2 {
        /* styles */
}
```

* Selector-1:- This is the ancestor element. It represents the element whose descendants you want to style.
* Selector-2:- This is the descendant element. It represents the elements that are descendants of the elements selected by 'selector 1'.

Experiment No. Date

Name of the Experiment Page No.

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.

Syntax:

```
element1 + element2 {
    /* style */
}
```

Here, element 1 and element 2 are the two elements involved and the '+' symbol is the adjacent sibling combinator. This selector will match 'element 2' only if it is an immediate sibling that directly follows 'element 1'.

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

When choosing combinators in CSS selectors, it's important to consider the structure and complexity of your HTML document.

(i) Descendant selectors:

* Use case: When you want to select nested elements, regardless of their level of nesting.

chandra's

* Considerations: This is useful for styling elements within a specific container or context. However, it can also lead to less efficient selector if not used carefully, as it targets all descendants.

(ii) Child Combinator (>):

* Use case: When you want to select immediate children of a parent element.
* Considerations: This is more specific than the descendant combinator and can lead to more efficient selectors since it only targets direct children.

(iii) Adjacent sibling Combinator (+):

* Use case: When you want to select an element that is the immediate next sibling of another element.
* Considerations: Useful for styling elements that follow each other directly. Can be efficient for selecting specific elements in a sequence.

(iv) General sibling Combinator (~):

* Use case: When you want to select elements that share the same parent and appear after a specified element.
* Considerations: Similar to the adjacent sibling combinator but less strict, as it selects all matching siblings, not just the immediate next one.

Experiment No.
Name of the Experiment

Date
Page No.

33] Explain the concept of css pseudo-selectors. provide examples of commonly used pseudo-selectors and their purposes.

* CSS pseudo-selectors are special keywords that are used to select and style a specific portion of an element's content or state of an element.
* They allow you to target elements based on criteria that cannot be expressed using regular element selectors alone.

(i) :hover :-

purpose:- Selects and styles an element when the user hover over it.

Ex:
```
a:hover {
    color: red;
}
```

(ii) :active :

purpose: selects and styles an element while it is being activated.

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

(iii) :focus :

purpose: selects and styles an element that has focus (mainly keyboard).

chandra's

(v) :nth-child():

Purpose: Selects elements based on their position within a parent element (even/odd).

Ex:

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

34] Differentiate blw pseudo-classes and pseudo-elements in CSS. Give examples of each.

Pseudo-classes	Pseudo-elements

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

35] How can you use the :nth-child pseudo-class to select specific elements in a list (or) container? Prove an example.

Experiment No.
Name of the Experiment

Date
Page No.

The ':nth-child' pseudo class in CSS allows you to select and style elements based on their position within a parent container.

Ex:- Selecting odd and Even Elements

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

JAVASCRIPT :-

01] What are the primitive Datatypes in Javascript?

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

* Numbers
* strings
* Boolean
* Undefined
* Symbole

02] Explain the difference between null and undefined in Javascript.

* Null :- * It is a value can explicitly set to indicate the absence of any meaningfull value.
* It explicitly set by programmers.

* Undefined :- * Means a variable has been declared but has not been assigned value.
* It is built-in-primitive value in JS.

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

We can check the datatype of variable using the 'type of' operator.

04] Explain the concept of truthy & falsy values in Javascript. Provide examples.

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

05] What is the difference between == and === operators in Javascript and How do they relates to data types?

"==" :- This operator compares values for equality.

"===" :- This operator (03) comparing the both values and type of operands.

Examples:-

console.log (6==6) Comparing value.

=> It return true.

Console.log (10===10) comparing the types.

=> It returns false.

07] Explain the difference blw 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.

08] How does Javascript handle NaN (not a number) values, & how can you check if value is NaN?

In Javascript, 'NaN' is a special value representing the result of an operation that cannot produce a meaningfull numeric result.

Exi- let x=5
console.log (x * "abc")

Q9] Explain the concept of type coercion in Javascript. Provide examples.

Type coercion in Javascript refers to the automatic conversion of one data type to another during the execution of a program.

There are two types:

(i) Implicit type coercion:

It happens automatically when values of different types are used together in an operation.

Exi let x = 5;
let y = "10";
let result = x+y // Number + string.
console.log(result).

(ii) Explicit Type coercion:

If also known as type casting, occurs when a developer explicity converts a value from one type to another using built-in functions & methods.

Exi let a = "5".
let b = 10
let result = Number(a) + b // Explicit coercion
console.log(result)

Experiment No. | Date

Name of the Experiment | Page No.

10 What is the purpose of the undefined data type, and when might it be explicitly assigned to a variable?

In Javascript, undefined is a primitive data type that represents the absence of a value. It is often used to indicate that a variable has not been assigned a value.

Developers might explicitly assign the value 'undefined' to a variable to indicate that it currently lacks a meaningful value or that it should be reset to an uninitialized state. Ex: let x;

console.log(x).

11 How do you create and use template literals (string operation) in Javascript?

We use template literals enclosed within the backticks (` `) instead of single or double quotes & we use in ${} symbol & we use curly braces.

Ex: let x = 10

y = 20

console.log(`The sum of ${x} and ${y} is ${x+y}`)

chandra's

12] What is hoisting?

The process of moving all the elements declaration to the top of the scope.

13] What is IIFE?

* IIFE - Immediately Invoked Function Expression
* The function can auto call it when the page is ready for regular execution.

14] What is meant by Default parameter passing?

Default parameter means we are allocw 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 into(name, age=16){
    console.log('myself ${name} and
        my age is ${age}')
}
into('virat')
```

15] What is the default return value?

The default return value is undefined.

Ex:
```
function a(x){
    return
}
a(2)
console.log(a(2))
```

Experiment No. | Date

Name of the Experiment | Page No.

16] How to pass unlimited number of parameters to a function?

We can pass unlimited number of parameter by using arguments keyword and in the arrow function, use rest parameter (...)

chandra's