

Important Questions

ITQ defined

HTML & CSS Questions

Q What is HTML and what is its purpose?

→ HTML Hyper Text Mark Up Language

- It is a programming language used to describe the structure of the web page
- By using HTML we can create static pages with text, headings, images, links and tables.

Q What is the difference between HTML & XHTML?

→ HTML → Hypertext Markup Language

XHTML → Extensible Hypertext Markup Language

- XHTML is stricter than HTML
- In XHTML case sensitivity and syntax must be correct

HTML → less expensive

XHTML → more expensive

Q What are the new features introduced in HTML5?

→ Features of HTML5 are:

- It is rich with media elements like adding audio and videos to the website

① HTML5 is simple

↳ we can capable to build our webpage within a day

② Platform independent

↳ can run in any browser if it has browser

③ we can easily read HTML tags.

④ Flexible: Using CSS we can create & design the webpage according to our requirements.

⑤ Links: Multiple webpages can be linked to each other.

⑥ How do you include Comments in HTML?

→ <!-- They will up -->
Commenting the Line.

⑦ What are Semantic Elements in HTML 5 and why are they important?

→ Semantic Elements:
↳ They will clearly describe the meaning of both browser & the developer.

Ex: <form>, <table>, <articles>

They are useful because

- ① They will help to structure the code & we can create a
- ② It will make more readable & easier.
- ③ They make web page more informative.

⑧ Explain the diff b/w <div> & tag?

→ Both <div> and are generic HTML elements.

div tag → Used for block-level organization
Used for styling of page elements.

span tag → used for subtle styling & organization

Ex: #merging-test

<div id="merging-test">, <spn> = " " </div>

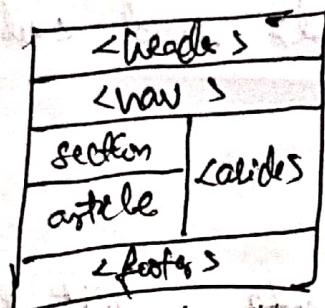
Ex ②: <p> -----

>

> SPAN

Q What is the purpose of <header>, <naw>, <sections> and <footers> tag in HTML?

→



header: It separates headings and sub-headings on a webpage
naw bar: It provides navigation links either within current document or other documents
a) nav tag will be inside the header

sections: It will define the section of document like
① chapter
② header, footers and
③ other sections

Ex: Section 1:

Content of Sec 1

Sub section

footers: used to define the footer of the HTML document

if it has information like

- ① author sup
- ② copyright sup

∴ footer tag is used within the body tag

Q How do you create a hyperlink in HTML?

→ Steps:

<a> ...

using ~~attribute~~ anchor tag and href attribute

Q <a>

Q

Q What is the difference between and elements?

→ → ordered list

 → unordered list.

unordered list: is used to create a list of items in no particular order.

i.e., the order of items is not relevant

ordered list: used to create a list of items in a specific order.

Ex: <table> <table>
 <tr> <tr>
 <td> <td>
 <td> <td>
 </tr> </tr>
 <tr> <tr>
 <td> <td>
 </tr> </tr>

Q How do you embed image in HTML?

→ Using tag.

Ex:

Q1 Explain the difference between the and tag?

→ tag:- Basically we will use to indicate that "this is more important than the surrounding text".

 tag:- To mark text that has stress emphasis.

↳ used to define emphasized text.

→ Content will be displayed in italic.

Q2 How do you create a table in HTML?

→ <table> tags defines an HTML table.

The <table> element will consist of

<tr>

<th> and

<td>

<tr> → table row

<th> → table header

<td> → table cell

Ex:- <table>

<tr> FirstName

<th> SRD <th>

<th> ITEM <th>

<th> Age <th>

<td> Darshan <td>

<td> Harshi <td>

<td> 22 <td>

<tr>

</table>

Q3 What is the purpose of the <form> tag in HTML and how do you create a form?

→ <Form> :- To Collect user Input

Ex:- <body>

<form>
User ID : <input type = "text" name = "user-ID" />

PassWord : <input type = "password" name = "password" />

</form>

(14) What are some new input types introduced in HTML5?

→ Date, Datetime

time

Week

month

email

tel

URL

search

range

Color

Numbers

To improve the user experience and to make form more interesting.

(15) How to include audio & video content in HTML?

→ Creating the new HTML file in the same directory, called index.

Ex: `<audio controls autoplay>`

`<source src="audio.mp3" type="audio/mpeg">`

`</audio>`

`<video controls>`

`<source src="movie.mp4" type="video/mp4">`

`</video>`

(16) What is the purpose of <iframe> tag and how to use it?

→ <iframe> tag: It will specify the inline frame.

→ used to load another HTML page within the document.

→ Used to display the web page within a webpage.

- (17) How do we add CSS styles to HTML elements?
- We have 3 ways:-
- ① Inline → by using "style" attribute inside HTML element
 - ② Internal → by using `<style>` element in `<head>` section
 - ③ External → by using `<link>` element to link external CSS file.

Inline:

```
<h1 style="color: red;"> SRI </h1>
```

Internal:

```
<head>
```

```
<style>
```

```
body { color: blue; }
```

```
h1 { color: red; }
```

```
p { color: white; }
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<h1> Welcome </h1>
```

```
<p> The site para </p>
```

```
</body>
```

External:

```
<link rel="stylesheet" href="style.css" />
```

- (18) What is the role of alt attribute in `` tag?

Ex:- ``

`alt = "Smiley face"`

→ Used to specify the alternate text for an image. If the image is not displayed.

- Q 19 How do we create a numbered list with custom numbering style in HTML?
- We can create an ordered list using `` tag and define the list items using ``

- Q 20 What is the difference between `<script async>` and `<script defer>`?
- `<script async>`: It used to load script asynchronously
`<script defer>`: It means our script will only execute after the page has finished loading

→ `async` will allow script to run as soon as it is loaded, without blocking other elements on the page

Ex:-
`<script async src="script.js"></script>`
`<script defer src="script.js"></script>`

- Q 21 What is responsive web design, and why is it important?
- Responsive Web Design: It is an approach to web design that makes your web page look good on all devices.
- Responsive Web Design uses only HTML and CSS.
- It is important because to solve a lot of problems for our website.

Q2 How do you make a website responsive using CSS?
→ <meta> tag has to be included

③ By using breakpoint & media query break point.

Q3 What is a media query in CSS, and how is it used for responsive design?

→ Media queries → allow us to apply CSS styles depending on a device's general type

Ex: @media and @support

→ Used at: They allow us to create different layouts depending on the size of the viewport.

Q4 Explain the difference b/w a fixed layout and fluid layout in terms of responsiveness?

→ Fluid layout → set has width 100% and it element arrange via the screen resolution.

Fixed layout → the user will be fixed the size

The user will be giving the size
Ex: in pixels. Ex: 100px 80px

Q5 How do you make image responsive in CSS?

→ we need to give the ~~width~~ value new value to its width property.
Then the height will adjust automatically

Ex) element.style {

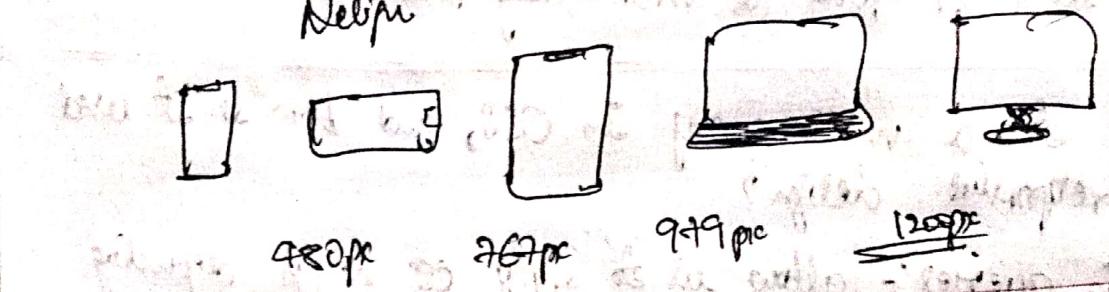
img {

width: 80%;

height: 100px;

Q1 What are breakpoint in responsive design and how are they determined?

→ Breakpoint are: They are the building block of responsive design.



Q2 How can we hide elements on specific screen size using CSS?

→ By display property set to st. "none"

Q3

By the CSS height, width & overflow properties may be used to hide elements.

Ex: setting height: 0
width: 0

Overflow: hidden on an element

Q4 What is the purpose of the max-width property in responsive CSS?

→ defines the maximum width of an element.

Ex: Using max-width: 80%;

• Comma only screen and (max-width: 768px) {

Using {

width: 100%;

}

Q9) How do we create a navigation menu using CSS?
→ The `<nav>` tag defines a set of navigation links.

Ex:-
`<nav>`
` Name `
` Contact `

`</nav>`

Q10) Explain the concept of mobile-first design and how it will relate to CSS.

→ Instead of ignoring the CSS styles that need to be added in the end, we have to work on it first, which gives a responsive site from the beginning.

• Designing the desktop site starting with the mobile version, which then adapts to the larger screens.

Q11) What is CSS Flexbox, and what problem does it solve?

→ It is a CSS used layout model.

→ It makes easier to design flexible layout without using float or positioning.

It will solve:-

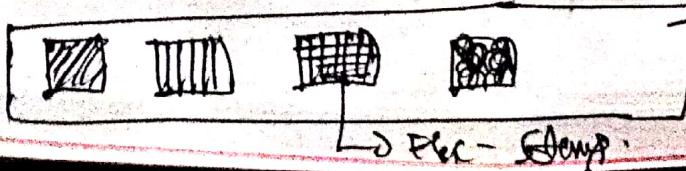
• arranging items in rows or columns.

• We use "flex-grow" property.

Q12) Difference b/w flex Container & flex items?

→ Flex Container: It is on HTML element whose display property will be `flex` or `inline-flex`.

Flex-items: They are the direct children of flex Container



③ How do we create a flex Container in CSS?

→ flex-container

display: flex;
flex-direction: column;
background-color: yellow; } head of flex

↓
Body

<h1> This is first </h1>

<div> </div>

<div> & </div>

</body>

→ we'll first specify on head element using flex-container
& then providing credit will for it in the body using
 tag & class = "flex-container"

④ Properties used to Control the layout in flexbox?

→ Properties for parent (flex-container)

• display → flex

• flex-direction → Row || Column || Column-reverse || Row-reverse

• justify-content → flex-start || flex-end || center || space-between
|| space-around || space-same

• align-items → flex-start || center || stretch || baseline

• align-content → space-around

• align-self → auto || baseline || stretch

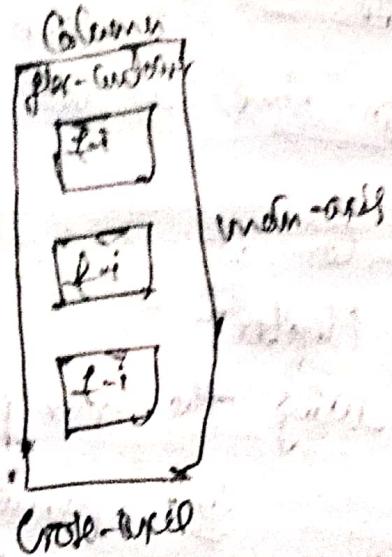
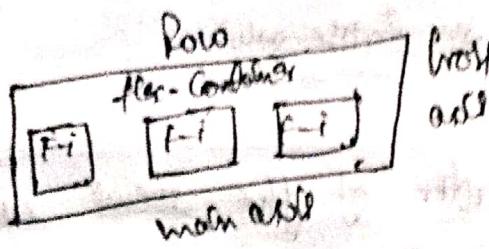
• order → any positive value

• flex-grow → 1 1 1 1

• flex-shrink → 1 1 1 1

• flex-wrap → no-wrap || wrap || wrap-reverse

Q35 How do you specify the direction of flex-grow within a flex-container?
→ By adding flex-direction property to the flex-container



Q36 What is the purpose of the flex-grow, flex-shrink and flex-basis properties?

→ flex-grow: It will show how much it will grow in proportion to sibling elements.

flex-shrink: It will determine how much it will shrink.

flex-basis: Sets the initial main size of the flex-item

∴ If flex-grow = undefined

↳ nothing happens

flex-shrink enabled = item will shrink below loop to prevent an overflow of the container

Q37 How to align flex items horizontally and vertically within a container?

→ We use align-items property to align our item.

↳ main-axis = justify-content ; cross-axis = align-items

③ Difference b/w justify-content & align-items
proposed in Flexbox

→ Justify-content: Controls alignment of all items on the main axis
align-items: it will control in cross-axis

④ How can we control the order of flex items using CSS Flexbox?
→ By using the order property

Ex:- parent
display: flex;

child L
height: 20px;
background: green;
padding: 5px;

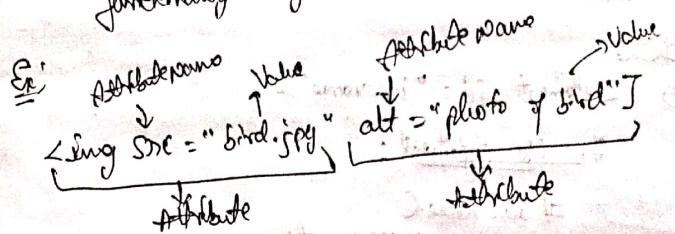
⑤ What are the flexbox breakpoints, & how can they be used for responsive design?
→ It is a size where the design adjusts for a specific screen width
→ The breakpoint can operate up & down

default breakpoint are:-

- 1) desktop (1200px - 1399px)
- 2) wide desktop (1400px & up)
- 3) tablet (768px - 1024px)
- 4) mobile (767px & below)
- 5) mobile landscape (488px - 767px)

⑥ What are HTML attributes?

→ It is a piece of markup language used to adjust the behaviour of display of an HTML element.
Attributes can be used to change the color, size & functionality of HTML elements.



⑦ Difference b/w global attributes & element-specific attributes in HTML.

→ Global attributes; Common to all HTML elements.
They can be used on all elements

Ex:-
accesskey
class
contenteditable
data-*
dir
draggable
hidden
id
long
spellcheck

→ Element-specific attributes: used to adjust the behaviour of display of an HTML element

→ Used to change color, size & functionality.

⑧ How do you add attributes to HTML elements?

→ element.setAttribute("Name", "Value");
↓
name value

- Q4. What are diff b/w class & id attributes?
- class: A class name can be used by multiple HTML elements.
- ID: It is used only by one HTML element within the page.

Syntax:

① `<element id="id-name">`

' In CSS styleSheet:

id-name {

 /* CSS property */

② `<element class="class-name">`

' In CSS styleSheet:

. class {

 /* CSS property */

- Q5. What is the use of id attribute in HTML and how is it unique?

→ Specifies the unique id for an HTML element.

Unique:

point to a specific style definition in a style sheet (#)

Q6. Explain the use of href attribute in HTML, particularly in the context of links and anchors.

→ Specifies the URL of the page the link goes to.

` DBASS ` → answer text

- Q7. How do you add alternative text to an image using the alt attribute?
- We use "alt attribute"

Ex: ``

- Q8. What is the purpose of the target attribute? In HTML links and what are its possible values?
- It specifies where the linked document will open when the link is clicked.

Ex: ``

``

- Q9. How do you use the src attribute to embed an external resource such as an image or video in HTML?

Syntax:

① `<video src="video-url" controls>`
`</video>`

② ``
``
``

Ex:

`<video width="500" height="300" controls>`
`<source src="javascript.mp4" type="video/mp4" />`
`</video>`

- Q) What is the purpose of the disabled attribute, and how is it used in HTML form elements?
- disabled attribute: It is a Boolean attribute
- ① Disabled element is invisible
 - It makes the element not clickable, focusable
 - ② even disabled
 - ③ The user can neither edit nor focus on the control.

Syntax:

```

<head>
  <title> Disabled </title>
</head>
<body>
  <button type="button" disabled> Click </button>
</body>

```

JAVA Script Questions:

- Q) Is there any relation b/w Java & Java Script?
- JS has no direct relation to Java
- ① they are very different
 - ② Both languages are written and executed differently and are used for a variety of different things.
- No → not related.
- Q) Is JavaScript a compiled or interpreted language?
- It is a interpreted language.
- ③ The languages such as C++ or Java needs to be

Compiled before it is executed.

- Q) Is JavaScript a Case-sensitive language?
- Yes, it is a case-sensitive language.
- That means that language keywords, variable, function names, and any other identifier must always be typed with consistent capitalization of letters.
- For Ex: must be typed "while", not "While".
or "WHILE".
- Q) What is node.js?
- It is a cross-platform, open-source JS runtime environment that can run on windows, Linux, Unix, macOS and more.
- It runs on V8 JS engine, and executes the JS code outside the web browser.
- Used for server-side programming.

- Q) What is the difference b/w let, var & const?
- let → used to store value
 Ex: let x=10
- var → declare the variable → place where we want to store value
 Ex: var y=10; var name="John"
- const → the value remains constant
- Ex: const x=10
const pi=3.142
- ③ let & const are block-scoped

Q) var declarations are either globally scoped or function scoped.

Let → can be updated but not re-declared

Const → can't be updated & re-declared

Var → both be updated & declared

Q) What are the difference between undeclared and undefined variables?

→ Undeclared: It occurs when we try to access any variable that is not initialized & declared.

Undefined: It occurs when a variable has been declared but has not been assigned any value.

Q) What is hoisting? → `const log(x)` is undefined

→ It is a default behaviour of moving all the declarations at the top of the scope before the code executes.

Q) No matter where function & variable are declared.

Q) What is scope in JavaScript?

→ It refers to the current context of code, which determines the accessibility of variables to JavaScript.

The two types of scope are local & global

→ Global → are declared outside the block

→ Local → -u-u- scope of a block

Global Scope:

`let a = "Hello"`

`function A () {`

`console.log(a);`

`}`

`greet()`

`of;` Hello

Local Scope:

`var dogSound = "woof";`

`const animalSound = () => {`

`const dogSound = "woof";`

`};`

`animalSound();`

`console.log(dogSound);` woof

Q) What are reserved words? (Can I use reserved word as identifier?)

→ Reserved Keywords: the words which are given within class, constructor statement.

↳ var delete for let break super void
case do static function new switch while
interface catch else if package finally else

→ They can't be used as identifier for variables, functions and classes etc.

Q) Why do you need strict mode? How do you declare strict mode?

→ It enforces stricter parsing and error handling on the code at runtime

→ It will help us to write clean code and

(Catch error and flag that our code undefined

if you)

Add the "use strict" at the first statement in your file.

④ What are global variables?

→ Variables that are declared in the global scope

• They can be accessed from anywhere in JS program

⑤ What are the problems with global variables?

→ If many variable are declared in global then they may remain in memory till the program execution is completed

• Memory Consumption

• Risk of Overwriting Variable

⑥ What is NaN property?

→ NaN is a number that is not a legal number.

NaN → Not-a-Number

Ex: isNaN('Not') = /True.

Number. isNaN('String'); // false.

NaN (undefined) /True

NaN (null) //false

NaN (' ') /True

NaN ("MSD") /True

NaN ("f") //false

⑦ What is the purpose of delete operator?

→ It will remove a given property from an object

• Delete both the value of property & property itself

Ex: var animal = {

name: 'Tom',

age: 22

delete animal.age;

⑧ Diff b/w null and undefined?

→ Null: empty value & is also a primitive type undefined: variable is declared and value not assigned

undefined

• undefined

• Invalid

• VariableName === undefined

• Non error

• Cannot be assigned

Null

→ It has no

object

Value

• valid

• VariableName === Null

• treated as 0 Value

• Can be assigned

Q6. What are bitwise operators available in JS?

| | | | |
|----|------------------------|---------|-----|
| & | Bitwise AND | 6 & 3 | 2 |
| | OR | 10 10 | 10 |
| ^ | XOR | 2 ^ 2 | 0 |
| ~ | Bitwise 1's Complement | ~9 | -10 |
| << | left shift | 10 << 2 | 40 |
| >> | right shift | 10 >> 2 | 2 |

Q7. Can I declare let and const variables?

→ They can't be redeclared.

Q8. Does const variable make the value immutable?

→ Yes, the const declaration creates an immutable reference to a value.

- The value it holds is not immutable.
→ The variable itself can be reassigned.

Q9. What is ES6? Get down its features.

→ ES6 stands for "ECMAScript".

ES6 was created by standards JS, and ES6 is the 6th version of ECMAScript.

Published in 2015 also known as ECMAScript 15.

Features:

| | |
|-------------------|--------------------------|
| let keyword | Module |
| const keyword | rest parameter |
| for... of loop | Spread operator |
| arrow function | Destructuring Assignment |
| multi-line string | Default parameters |
| class | Ternary If-Else |

Q10. What are the possible ways to Create objects in JS?

→ Using

- ① Object literals
- ② constructor functions
- ③ ES6 class and object
- ④ Create method

Ex- ① Using object literals

const obj = {
 name: "John"
};

② Using the new keyword

function Obj(prop){
 this.prop = prop;
}
const obj = new Obj("value");

③ Using object.create()

function newObject(prop){
 this.prop = prop;
}
const obj = object.create(newObject);

Cri: #1

```
Var pizza = {  
    name: 'margherita',  
    size: 'medium',  
    isVegetarian: true}
```

#2

```
Var pizza = new Object();  
pizza.name = 'margherita';  
pizza.size = 'medium';  
pizza.isVegetarian = true;
```

#3

```
Var anotherPizza = Object.Create(pizza)
```

Q7: What is the difference b/w slice and splice?
→ slice: It returns the piece of array, but it doesn't affect the original array.

splice: Changes the original array by removing, replacing or adding values and returning the affected value.

Cri:
Cont numbers = [1, 2, 3, 4, 5]
out = numbers.slice(1, 4) → [2, 3, 4]
console.log(out);

```
Cont array = [1, 2, 3, 4, 5]  
array.splice(1, 1)  
array.slice(1, 1)
```

Cri: var array = [1, 2, 3, 4, 5]

console.log(array.splice(3))

Q8: Difference b/w

a. == & === operator

b. != & !==

c. != & !==

Ans: ==

① equal to

② perform loose equality comparison

③ equal value

④ equal type

⑤ perform strict equality comparison

⑥ == & ===

True

→ First 5 is converted to a string & then compared with 5 → Number & string are different

⑦ 5 is number

⑧ 5 is string

(b) == & ===

① used for assigning the value to variable

② used to compare two variables.

100 == 100 } Both are equal
100 === 100 }

⑥ $a =$ $a = \underline{3}$
 $\rightarrow a = \rightarrow$ It is a read/write assignment

Ex: let a=3

Console.log((a += 2)); op= 1

Console.log((a += 0)); op= NaN

Console.log((a += 'Hello')) op= NaN

⑦ What is higher order function?
→ The function that accept functions as parameters and return a function.

Ex: add = (x) \Rightarrow (y) \Rightarrow $x+y$;

Ex: function doSomething()

$\lambda x() \Rightarrow$ Console.log("Hello")

return

doSomething()

⑧ What is the currying function?
Ans: To represent the short form, we can form the callable
 $f(a, b, c) \rightarrow f(a)(b)(c)$.

⑨ Transform a function with multiple arguments into a nested level of function, taking each taking a single argument.

Ex: function multiply(a, b)
returns a * b

Console.log(multiply(3, 2))

160

⑩ What are arrow functions?

→ New way to write anonymous function expression.

Format:

$() \Rightarrow L$

Ex:

const add = (a, b) \Rightarrow a+b;

const square = $a \Rightarrow a \times a$;

⑪ What is a spread operator?

Ans: Spread operator:- (...) allow us to quickly copy all the part of existing array or object into another array or object.

Ex: const numbers = [1, 2, 3, 4, 5, 6];

const [one, two, ..., rest] = numbers;

Ex: const name = [name: "SRD",
age: 22,
Branch: ECE]

const newName = [
name: "Hem",
age: 21,
Branch: ECE]

const newName = [...name, ...newName];

Q7: What is rest parameter?

→ allow a function to accept an indefinite number of arguments or an array.

Ex: function sum(...args) {

let total = 0;

for (const arg of args) {

total += arg;

return total;

Consider: log(sum(1, 2, 3)) → Output: 6

Consider: log(sum(1, 2, 3, 4)) → Output: 10

Q8: What happens if we do not use rest parameter for a last argument?

→ Syntax Error is thrown as it has to be

last parameter & Code is not Executed

Q9: What are regular Expression patterns?

→ patterns used to match character combination in string.

Pattern used with

| | | |
|---------|------------|--------------|
| exec() | match() | search() |
| test() | matchAll() | split() |
| Reg Exp | replace() | replaceAll() |

Ex:

let regex1 = /for/;

let regex2 = Reg Exp('for');

Q10: What is a Regular Expression?

→ Some of Question Q9

Q11: How do we search a string for a pattern?

→ By "search()" method and "replace()" method

Ex: prototype. search()

↳ Syntax

Q12: What is the purpose of switch-case?

Ans: Executes a block of code depending on condition.

Ex: `let a = 3 + 3;`

`switch(a) {`

`case 3:`

`alert('Value is below');`

`case 4:` → `break;`

`alert('Eqd on');`

`case 5:` → `break;`

`alert('Value is too large');`

`default:` → `break;`

`alert("Unknown Value");`

}

Q: What are the conventions to be followed for the usage of switch Case?

→ The default clause is the last clause, but it does not need to be so.

One default clause is OK

multiple default clause = Switch Error

Q: What are primitive Data types?

→ There are 7 primitive types.

① String

② Number → 3

③ Boolean

④ Boolean

⑤ Symbol

⑥ Null

⑦ undefined

⑧ Object

④ Array

⑤ Function

Q: What are the different ways to access the object properties?

→ 3 ways are these:

① Dot property operator

Object.property

② Square bracket

[property]

③ Object destructuring

But {property} = Object.

Ex:

① Obj.prop ② obj[property] ③ {property = obj}

Q: What are the function parameter rules?

Ans: i) The function only gets to know the value, not the argument's location

ii) If a function changes an argument's value, it does not change the parameter's original value

Q: Different ways which create infinite loops?

→ By "omitting all parts of the head of a for() block"

i) Use the goto statement to define infinite loop

Ex: "while(true)" loop & "for(;;)" loop

-? Infinite loop can crash the program browser & freeze our PC

Q2 What are template literals?

Ans: Template literals will allow us to use string expressions in the form of `a${str}`.

Ex: Const name = 'Jack';

Console.log('Hello \${name} !')

Output: Hello Jack

For `str1`)

Const str1 = 'The \${a} string'; ✓

Const str2 = `A "quote" inside a string`; ✓

Const str3 = `A 'quote' inside a string`; ✗

Q3 What are the default values for destructuring assignments?

Ans: The default value can be "any expression".

It will only evaluate when necessary.

If we do not pass any argument in the function, the function automatically assigns the value "undefined".

Q4 How do we swap variables for destructuring assignments?

Ans: 4 ways to swap:

① $[a, b] = [b, a]$

② temp = a

$a = b$

$b = \text{temp}$

(This approach is efficient because it does not push extra values.)

③ $a = a + b$

$b = a - b$

$a = a - b$

Ex: let x = 1

let y = 5

$[x, y] = [y, x]$

④ $a = a * b$

$b = a / b$

$a = a ^ b$

Console.log(*)

Console.log(y)

O/p: 5

O/p: 1

Q5 Is that possible to use expressions in Switch-Case?

Ans: If we use the case matches, the compiler then the code will execute.

Q6 What are the differences b/w for...of and for...in statements?

Ans: for...of: It iterates over property values

for...in: It iterates over each key.

Ex: Const str = "Hello World";

for (element of str) {

Console.log(element);

}

Output: H

o

l

l

o

W

o

r

l

d

o

!

Q7 Const student = {

reg no: "12345"

name: "Sandeep"

g

y

for (key in student) {

Console.log(key, student[key]);

}

O/p:

reg no: "12345"

name: "sandeep"

Q3 Def. the argument object & rest parameters?

A1 Argument object: It represents all the arguments passed to the function.

Rest parameters: It represents the unknown number of arguments above the function

Argument object

Rest Parameter

- If it is not a real array
 - They are array instances
 - like sort(), map(), forEach(), pop(), can be applied on it directly
 - It is a plain array

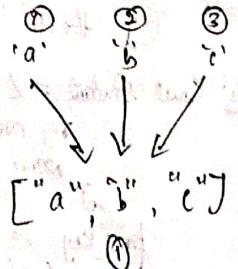
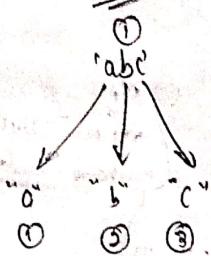
Q4 Def the spread operator & rest parameters?

A1 Spread Rest

- Separates one into individual elements
 - function A(...el1, el2, el3) {
 console.log(el1, el2, el3);
}

A('abc') // "a" "b" "c"
 - function B(...elments) {
 console.log(elments);
}

B('a', 'b', 'c') // "a", "b", "c"



Q5 Explain all the array methods, which are the output and either the method works on the original array or not?

A1 Array methods for JS:

① Push and pop

Ex: let color = ['red', 'gray', 'yellow', 'black'];
out = color.push('orange') } It pushes orange
Console.log(out)

out = color.pop() } pop the last element
Console.log(out) } L black

② "Concat"

const arr1 = [1, 2, 3];
const arr2 = [4, 5];
out = arr1.concat(arr2);
Console.log(out)

Out: [1, 2, 3, 4, 5]

③ "Slice"

const numbers = [1, 2, 3, 4, 5];
out = numbers.slice(1, 4);
Console.log(out)

Out: [2, 3, 4]

④ "Filter"

const age = [25, 18, 30, 40, 15, 16, 17];
out = age.filter(age => age >= 18);
Console.log(out)

Out: [25, 18, 30, 40]

⑤ "for Each"
 const months = ["Jan", "Feb", "Mar"]
 months.forEach(function(month) {
 console.log(month)
 })

o/p: Jan

Feb

Mar

Apr

Some methods are:

- ① every()
- ② filter()
- ③ indexOf()
- ④ lastIndexOf()
- ⑤ map()
- ⑥ reduce()
- ⑦ reverse()
- ⑧ some()

⑨ sort()

⑩ splice()

① "map"

const b = [1, 2, 3, 4, 5]
 const numbers = b.map(function(b) {
 return b + 2
 })

o/p: 3 4 5 6 7

console.log(numbers)

⑥ "for each"

const a = [1, 2, 3, 4, 5]
 a.forEach(function(a) {
 console.log(a + 2)
 })

o/p: 3

4

5

6