51) what are classes

=> classes are template for creating objects. They are encapsulated data with code to work on data. classes consists of function called constructor.

52) Destructuring of objects.

→ It allows us to extract multiple properties or items from an array at a time.

53) Inheritance

→ It enables you to define a class that takes all functionality from a parent class & allows you to add more.

```
class Person {
constructor(name) {
    this.name = name;
}
greet() {
    console.log('Hi ${this.name}');
}
}
class Student extends Person
{
let student1 = new Stud('Tim');
student1.greet();
```

54) what is super in class.

→ The super keyword is used to call the constructor of its parent class to access the parent properties & methods.

ode for event on ...
d constructor.

49) closure.
It is the ability to remember the variable and function that are declared in the outer scope is known as closure.

```
function add()
{ let counter =10
    function increment ()
    {
        count += 1;
    } increment()
    return counter;
} console.log (add())
```

50) what is Armstrong number ? write the function
A positive integer is called an Armstrong number if the sum of a cubes of each digit is equal to the number itself.

```
function Arm. (original-num){
    let sum=0;
    let temporary-num = original-num;
    while (temporary-num >0)
    {
        let remainder = temporary-num%10;
        const const-remainder = remainder.
        for (let i=1 ; i<original-num.tostring().length; i++)
        {
            remainder = const-remainder * remainder
        }
            sum += remainder;
            temporary-num = parseInt (temporary-num/10);
    }
    if (sum == original-num)
        { console.log (original-num);
        } else {
        console.log (original-num);
        }
```

44) . write function . to Reverse a string . (with using built in).

```
function reverse(str)
{
   var a='';
for (var i=0 ;i<= str.length /2;i++)
   {  a=str[i];
      str[i]= str [str.length-i-1];
      str [str.length-i-1]=a;
   } return str;
} var str ='abcde';
      reverse (str);
```

45) . function to calculate the frequency of ocurace of each number . eg:-[1,1,2,3,4]

```
const counts ={};
const sampleArray =[1,2,3,4,4];
sample Array.forEach((x)=> { counts[x]=(counts[x]||0)+1;});
```

46) How do you loop through an object.

→ using for in loop. .

47) Function for palindrome

```
function .pali .(string ){
      const len = .string .length;
for (let i=0;i<len/2;i++)
   { if [string[i]!==string[len-1-i])
      { return 'not palindrom';
      }
   else { return 'is palindr';
   }
const string = "CANNAC".
```

41) write function to merge 2 arrays & removing duplicate without using inbuilt function

```
function mergeArrays ( arr1, arr2)
 {
     let arr = arr1.concat (arr2) ;
     let newArr = [] ;
     for (let i of arr)
     {   if (newArr.indexOf (i) == -1)
     { newArr.push(i) ; }
      } console.log ( newArr);
      }
Let arr1 = [ 2,4,6 ]
Let arr2 = [ 6,4,8 ]
 mergeArrays(arr1, arr2).
```

42) write a function on sorting the number in ascending order.

```
function sorting (nums)
 { return Number (nums.toString().split ('').sort ((a,b) => b-a).join('')
  }.
console.log. ( sorting ( 54321) ) ;
```

43) write function on removing duplicate.

```
function uniqueA(arr){
    let uniqueArr = [];
    for(let i of arr) {
        if (uniqueArr.indexOf(i) == -1)
        { uniqueArr.push(i); }
    } console.log (UniqArr); }
of uniqueA(array)        Const array = [1,2,2,3,3,9]
```

31) ... to Reverse a string (with using built in).

31) What are escape characters in JS

→ characters used to begin an escape command in order to execute some operation. They are characters that can be interpreted in some alternate way that what we intend to. JS uses \ backslash in front as an escape character.

32) Type of null, undefined, array, object.

→ object
→ undefined
→
→ object.

33) Null == undefined → true

Null === undefined → False.

34) Difference b/w == & ===

→ The strict operator takes type of variable in consideration while non-strict operators make type correction based upon values of variable.

35) Explain Ternary operator with example.

→ A ternary operator evaluates a condition and executes a block of code based on the condition.

Ex:- let price = 900;

let checker = (price >= 1000) ?

console.log (checker);

29) Explain spread operator , rest parameter.

→ Spread operator (...) allows us to quickly copy all part of an existing array or object into another array or object. It is also used in combination with destructuring

```
const cabins 1 = [ 'JP' , 'AUS' ]
    cabins 3 = [ ... cabins 1 , ... cabins 2 ]
    console.log (cabins 3)
```

→ Rest parameter (...)

Provide improved way of handling parameter in function any number of arguments will be converted into an array using rest. it also helps in extracting some part of array.

```
function total (...args) {
  let sum = 0;
  for ( let i of args) {
      sum += i;
    }
  return sum;
}  console.log (fun(1,2));
```

30) Difference between null, undefined & empty.

→ Null . is an assignment value it primitive value that represents the null , empty.
It indicates the absence of a value for a variable
The type of null is object .

→ undefined . Type of undefined is undefined , indicates a not assigned a variable to it. converted to NaN while performing Primitive operations.

24) Difference between arrow function and normal function

→ Arrow function is a shorter syntax for a function expression and doesn't have its own this, arguments. These functions are best suited for non-method funcion.

```
const hardly = () => console.log('This is arrow fuction');
```

→ Functions are a set of statements that perform tasks. in this fuction has to be declare using function key word, it could not be used as an expression.

```
function name (number)
{ return number;
}
```

25) what is the output of 5+4+'4'-9-4-2.

→ 94942.

27) what is destructuring in JS give an example.

→ It allows us to extract multiple properties or items from an array at a time its a java script expression to extract data from arrays, objects, maps & sets.

```
Var [one, two, three] = ['Jan', 'Feb', 'Mar']

console.log (one);
console.log (two);
```

28) what is typeof operator

→ We can use the javascript typeof operator to find the type of a javascript value, it returns the type of a variable or an expression.

```
type of "Arjunce";   (String)
```

Q4) Difference betw....

20) what are objects methods.

21) Explain SetTimeout.

→ The setTimeout () method is used to call a function or evaluate an expression after a specified number of milliseconds.

```
SetTimeout (function() {
console.log ("Hi");
}, '2000');
```

22) what is callback function?

→ A callback function is a function passed into another function as an argument. This function is invoked inside the other function to complete an action.

```
function callback (name) {
    console.log (name);
}
function outterfunction (call).
    {
    let name = 'sam';
      call (name);
    }
    outterfunction (callback());
```

23) Difference between declaration and definition.

→ Declaration in java script is done by var, let, and const are the ways to declare in java script variable These are just the containers for storing data

```
Let x *
Var y *
```

→ Definition is where the system allocates memory by seeing the above function definition

```
let x = 'Anu';
```

18) Explain call and apply methods.

→ The call() method invokes a function with a given this value and argument provided one by one.

```
var employee = { first_name : 'Tim',
                 last_name : 'cook' ;
               }
function invite ( greeting ) {
    console.log ( greeting + this.first_name + this.last_name );
}
invite.call ( employee , "Hello" ) ;
```

→ The apply() method invokes a function with a given this value and allows you to pass in arguments as an array

```
var employee = { first_name : 'Joe' ;
                 last_name : 'Tucker' ;
               }
function invite (greeting ) {
    console.log ( greeting + " + this.first_name + " + this.last_name ) ;
}
invite.apply( employee , [ 'Hello' ] ) ;
```

19) What is primitive & non primitive data types in JS.

→ Primitive data types are the type of data which are immutable

- → String
- → Boolean
- → number
- → null
- → undefined

→ Non Primitive data types are mutable

- → objects
- → Functions
- → Array.

14) what are all position property

→ static : default value . the element is positioned according to normal flow of element

→ absolute : Here the element is positioned relative to its parent.

→ element : the position is determined by values of left, right, top, bottom

→ fixed : this is similar to absolute except here the elements are positioned relative to <html> element.

→ relative : the element is positioned according to normal flow of the document & positioned relative to its original . normal position.

→ initial : This reset the property to its default value.

→ inherit : Here the element inherits to takes the property of its parents.

15) what . class name will you give if we want to display a . div

ϕ flex . align . items - centre = to place vertically centre

d flex . justify - content - center to place horizontally centre

17) Explain hoisting

Hoisting is a default behaviour of javascript where all the variable and function declaration are moved to top of the scope.

```
function dosomthing() {
    x = 33
    console.log(x);
    var x ;
}
```

→ row - reverse & stack items horizontally from right to left in ascending
→ row column reverse : stack item vertically from bottom to top of ascending
→ flex-wrap : This property specifies of flex items should be wrapped or not
→ wrap : The flex items would be wrapped if needed.
→ wrap ÷ reverse : This specifies that the items would be wrapped in reverse order
→ flex-flow : is used for setting both flex-direction & flexwrap properties in one statement.
→ justify-content : used for aligning the flex item in possible ways are : center , flex start , space around , space-between, align-items , align-content.

13) what are different display properties in html
→ inline : this can display any block level element as an inline element.
→ block : we can display any inline element as a block level element.
→ inline-block : its similar to inline except by using display as inline block we can format the element using height, width
→ flex : displays elements as an flexible structure using flex property.
→ inline-flex : displays flex containes as inline element in content follows.
→ grid : it : displays html elements as grid containes
→ none : this property we can hide the html elements

8) what are merits and demerits of external CSS

-> merits : the style of a few documents can controlled from the side by utilizing the
: multiple html elements can have numerous documents where class can be made
: To assemble styles in complex circumstances, selector & grouping

-> Demerits : The additional download is expected to import documents having style information
: To render the documents the outer template ought to be stacked
: Not practical for small style definitions.

9) How can you create a nested web page in html?
Using <iframes>

10) Explain CSS box model?
-> A rectangular box is wrapped around every HTML element. The box model is used to determine the height & width of the rectangular box. The CSS box consists of widths & height, padding, margin box.

11) what is Doctype? explain
->. The html document type declaration, also known as doctype is the first line of code required in every html document. The doctype declaration instructs the web browser what version of html page is written in. There are three kinds of doctypes available. strict doctype transitional doctype, Frameset doctype.

12) Explain. flexbox property.
-> Flex-direction : This property helps defining direction
-> Row : stacks items horizontally from left to right in flex
-> column : stacks items vertically from top to bottom

Attribute Selector : The attribute selector targets elements based on the presence & / or value of html attributes, it is declared using square

```
input [type='text'] {
    bg-color : #FFF;
    width : 100px;
},
< input type='text' >
```

4) If we are applying two classes for same div with different style with apply either class 1 or class 2.

→ class 2.

5) what is CSS and why we use it ?

CSS stands for cascading style sheet. It is a popular styling lan which is used in Html to design websits.

6) How CSS works ? CSS are styling sheet which could be applied document in three diffent ways which are inline, internal and ext methods.

Inline method - is used to insert the style sheets in html docu

Internal method - is used to add a unique style to sigle docu

External method - is used when you want to make change multiple pages.

7) what are the new features in HTML 5

→ It provides support for local storage

→ New form controls. like calender, date, time, email, url

→ <canvas> element is provided for facilitate 2D drawing.

→ The <video> & <audio> elements are provided for me playback.

→ Child combinator :- It is similar to descendant combinator, except it only targets immediate child elements.

#container.box { float: left ;
Padding-top : 10Px }

// selector will match elements that have class of box & that are elements of children of # container elements.

<div id= "container">
<div class= "box"></div>
<div> <div class = 'box' > </div>
</div>
</div>

→ General sibling combinator :- selector uses a general sibling combinator to match elements based on sibling relationship . the selected elements are beside each other in HTML

h1 ~ p { margin-bottom 20PX ; }
<div>
<h1> Title </h1>
<P> Paragraph </P>
<P> — x — </P>
<P> — x — </P>
</div>

→ Adjacent sibling combinator :- A selector that uses the adjacent sibling combinator uses the plus symbol(+) , and is almost the same as the general sibling selector . difference is the targeted element must be immediate sibling not just a general sibling .

P+P { text-indent :1.5em ; margin-bottom : 0; }
<h2> Title </h2>
<P> Paragraph .ex </P>
<P> — x — </P>
<div class='box' >
<P> Paragraph </P>
</div>

Full Stack Assignment

1) Explain Semantic tags -> Tags which describe the particular meaning to the browser and the developer <form> <table> <img>

2) Explain types of Selectors with order of precedence

-> Universal Selector: The selecting all elements on a page, the provided style will be applied to all the elements on the page.

```
* {
    background-color: #FFFFF
}
```

-> Element type selector: This selector matches one or more elements of the same name.

```
ul {
    border: solid black 1px;
}
```

-> ID selector: This selector matches an HTML element that has ID attribute with the same value as that of the selector

```
<div id="cont">...</div>

#cont {
    width: 100Px;
}
```

-> class selector: class selector matches all elements on the page that have their class attributes set to the same value in the whole

```
<div class="classes">...</div>

.classes {
    margin: 10PX
    color: white;
}
```

3) Combinator selector

-> Descendant combinator: lets you combine two or more selectors such that you can be more specific in your selection method

```
div p {
    bg-color: grey;
}
```