

Q1.What is javascript ? Why use javascript?

Ans: Javascript is client side scripting language.

Why use JS

1. To change HTML content.
2. To change CSS.
3. To add or delete HTML content.
4. For ajax call
5. To create api by using json
6. For calculation work, like emi calculator, crm calculation work

Q2. What is ecmaScript? What is the latest version of ecmaScript?

Ans: EcmaScript is a standard which contains the javascript updates, it ensure the interoperability of web pages across different web browsers.

Q3. Javascript Selectors?

Ans: Javascript Selectors are:

1. **document.getElementById():** It selects the element by their id name.

Example:

```
<div id="box"></div>
```

```
document.getElementById("box");
```

2. **document.getElementsByClassName() :** It selects elements by their class name.

Example:

```
<div class="box"></div>
```

```
document.getElementsByClassName("box");
```

3. document.getElementsByTagName(): It selects the elements by their tag name.

Example:

```
<p>hello para</p>
```

```
<p>hello another</p>
```

4. querySelectorAll() : It selects the element that matching css selector.

Example:

```
<h2 class="example">A heading</h2>
```

```
<p class="example">A paragraph.</p>
```

```
document.querySelectorAll(".example");
```

Q4. What is DOM?

Ans: DOM is called Document Object Model, when our browser load our web page then it creates a structure of the code, that structure is called DOM.

DOM is used to change html content, change css, add html content, remove element by using javascript selectors and DOM methods.

Q5. What is BOM?

Ans: BOM stands for Browser Object Model, In this we learn window related topics like

- onload,
- setTimeout,
- history,
- location,
- popup alert , confirm,
- cookies,
- screen size

Q6. Type Conversion in JS?

Ans: Type conversion is used to convert data type of the data. There are two methods available in javascript which are following:

parseInt(): It is used to convert data into integer format.

parseFloat(): It is used to convert data into decimal format.

It is mostly used when we work on calculation related work like emi calculator, when we get data from user through input we get the data in string format then we calculate input data it gives wrong output. So here we need type conversion.

Q7. ES6 most used Features?

Ans: ES6 most used features are following:

1. Backticks(`): It is a modern version to concatenate strings.

Example:

```
const name = 'digital';  
const surname = 'oceans';  
const telephone = '9090887766';  
// "Old syntax"  
const userInfo = 'User info: ' + name + ' ' + surname + ' ' + telephone;  
// "New syntax"  
const userInfo = `User info: ${name} ${surname} ${telephone}`;
```


2. Spread Operator(...): It was introduced in ES6. It looks like a Rest Parameter but it works opposite of that. The spread operator is used to copy,concat or expand an array.

Example: Copying an array

```
let fruits = ['Apple','Orange','Banana'];  
let newFruitArray = [...fruits];  
console.log(newFruitArray); // ['Apple','Orange','Banana']
```

Example: Concatenating array

```
let arr1 = ['A', 'B', 'C'];  
let arr2 = ['X', 'Y', 'Z'];  
let result = [...arr1, ...arr2];  
console.log(result); // ['A', 'B', 'C', 'X', 'Y', 'Z']
```

3. Let & Const

let : Let is an improved version of **var**. Let keyword introduced in ES6, it is used for variable declaration, it resolved the problem of **hoisting** that occurs when we used **var** keyword.

In **var** keyword, there is a problem, in var keyword we can access the value of a variable without declaration, but in let we cant.

const: It is used defined a constant which will never change or update like `pie=3.14`.

4. Map() : map() method is used to modify each element of an array.

Example:

```
const array1 = [1, 4, 9, 16];  
const map1 = array1.map(x => x * 2);  
console.log(map1);
```

5. class: classes are templates for creating objects. The class contains methods, constructors, to access class data we must create an object first.

Example:

```
class Car {  
  constructor(name, year)  
  {  
    this.name = name;  
    this.year = year;  
  }  
  age() {  
    let date = new Date();  
    return date.getFullYear() - this.year;  
  }  
}
```

```
let myCar = new Car("Ford", 2014); document.getElementById("demo").innerHTML =  
  "My car is " + myCar.age() + " years old.";
```

6. object: objects are like an array that contains many values in a single variable. Objects are a little bit different from an array, an array stores data in the indexed format while objects store data in key-value pairs. The object is more simple as compared to the array.

Example:

```
<p id="demo"></p>
```

```
<script>
```

```
let person = {  firstName : "John",  lastName  :  
    "Doe",  age    : 50,  eyeColor : "blue" };
```

```
document.getElementById("demo").innerHTML =  
    person.firstName + " " + person.lastName;
```

```
</script>
```

7. inheritance: is a concept of object-oriented programming that is used to inherit all methods and properties of the parent class. **extends** keyword is used for inheritance in JavaScript.

Example:

```
<p id="demo"></p>
<script>
class Car {
  constructor(brand) {
    this.carname = brand;
  }
  present() {
    return 'I have a ' + this.carname;
  }
}
class Model extends Car {
  constructor(brand, mod) {
    super(brand);    this.model = mod;
  }
  show()
  {    return this.present() + ', it is a ' + this.model;
  }}

let myCar = new Model("Ford", "Mustang");
    document.getElementById("demo").innerHTML = myCar.show();
</script>
```

8. Array Destructuring: Array Destructuring means breaking down complex arrays into simpler parts. It can be used for assignments and declaration of a variable.

Example:

```
const foo = ['red', 'yellow', 'green'];  
const [a, b, c] = foo;  
console.log(a); // "red"  
console.log(b); // "yellow"  
console.log(c); // "green"
```


9. Arrow function: Arrow function is nothing it is a modern version of function it is written by an arrow that's why it is called an arrow function. It is a modern syntax of the function.

Example:

```
myfunction()=>{ console.log("arrow function"); }
```

Q8. **this** keyword in JS?

Ans: **this** keyword is used when you want to point out the particular element in a list of elements.

Example:

there are 5 to 6 images and you want to pass image source to function on click of image then we use **this** keyword to point out particular clicked image.

this keyword is also used in class to point out the class variable.

Example:

```
<div>  
  
</div>
```

```
<div>  
  
  
  
  
</div>
```

```
<script>  
function imagechange(smallimage)  
{  
    document.getElementById("bigimage").src=smallimage;  
}  
</script>
```

9. Event Bubbling vs Event Capturing?

Ans: There are two problems when we use event on nested elements are following:

Event Bubbling: In the event bubbling the innermost event executes first then its parent , ancestors. It goes form bottom to top.

Example: when we use events on nested element then it executes first where we occurred event then its parents. So it is problem to resolve this problem we use **event.stopPropagation()**.

Event Capturing: In the event capturing the outermost event executes first then its child. It goes form top to bottom.

Example: In the event capturing when we use event on nested element then it executes first where we occurred event then its childs. So it is problem to resolve this we first convert event capturing into bubbling then we use **event.stopPropagation()**.

Q10. Mostly used JS Events?

Ans: There are following mostly used events in javascript:

- **click:** click event is used when the user click a mouse.
- **keyup:** when the user releases a keyboard key.
- **change:** when the user change the value of select,radio,checkbox.
- **mouseover:** when the user mouse over.
- **scroll:** when the user scroll the scrollbar.
- **Load:** When the browser load the content.

Q11. Difference Between === and == operator in JS?

Ans: === operator match content and its data type also.

== operator match content only it does not match data type.

Q12. What is NAN in JS?

Ans: NAN is short for not a number, javascript shows an error when use use number illegally or in a wrong way.

In a javascript there is a function **isNaN()**, it is used to check a number is nan or not.

Example: Whenever we try to parse a string or “undefined” to an int, it returns NaN.

```
console.log(parseInt("The Digital Oceans"));
```

Q13. Timing Events in JS?

Ans: There are 2 timing methods in JavaScript which are following:

- **setTimeout():** It executes code after a specified duration. It executes code one time.
- **setInterval():** It executes code after a specified duration. It repeats code after a specified duration continuously.

Q14. preventDefault() in JS?

Ans: preventDefault() in javascript is used to stop the default behavior of the element.

Example: the <a> tag is use to link pages but if we use prevent default onclick of a tag then it will not redirect to the page, because preventDefault stops the behavior of the <a> tag.

It is also used in submit button to prevent the form submit on click of submit.

Example:

```
<a id="myAnchor" href="http://the-digital-oceans.com/">Go to the-digital-oceans.com</a>
```

```
<p>The preventDefault() method will prevent the link  
above from following the URL.</p>
```

```
<script>  
document.getElementById("myAnchor").addEventListener(  
    "click", function(event){  
        event.preventDefault()  
    });  
</script>
```

Q15. Difference between let and var in JS?

Ans: var: is used to hold data, it works like container to hold the data, but there is a lack in **var keyword** it can be used first and we can declare at last.

let: we can say that it is an improved version of **var** keyword. It was introduced in es6 , let keyword cannot be used without declaration.

Q16. What do you mean by NULL in JS?

Ans: Null is primitive data type, it is used to declare a variable with no value. It is like a blank value “ ” of the variable.

Example: `var num=null;`

.

Q17. What is an undefined value in JS?

Ans: undefined value is an error, javascript throw an error when a variable has been declared and no value has been given to it.

Example:

```
var a=5;
```

```
var b;
```

```
var c=a+b;
```

Q18. HTML API ?

Ans: HTML API are following:

- 1. web storage:** It is used to store data in a browser.
There are two types:
 - **local storage:** it save data permanently in browser.
 - **Session storage:** It save data temporary in browser , it removes data when the browser closed.
- 2. drag & drop :** it is used to drag and drop element.
- 3. Geolocation :** It is used to get current location.

Q19. Array methods in JS?

Ans: Array methods are following:

To insert data in an array

1. push() : push method is used inserts data at the end of an array.

Example:

```
const fruits = ["Banana", "Orange", "Apple", "Mango"];
fruits.push("Kiwi");
```

2. unshift() : this method inserts data at the beginning of an array.

Example:

```
const fruits = ["Banana", "Orange", "Apple", "Mango"];
fruits.unshift("Kiwi");
```

3. splice(): This method inserts data at specified position wherever we want to insert data.

Example:

```
const fruits = ["Banana", "Orange", "Apple", "Mango"];
fruits.splice(2, 0, "Lemon", "Kiwi");
```

To delete data from array

1.pop(): this method is used to delete data from last of an array.

Example:

```
const fruits = ["Banana", "Orange", "Apple", "Mango"];  
fruits.pop();
```

2.shift(): this method is used to delete data from starting of an array.

Example:

```
const fruits = ["Banana", "Orange", "Apple", "Mango"];  
fruits.shift();
```

3.splice(): this method is used to delete data from the specified position wherever we want to remove.

Example:

```
const fruits = ["Banana", "Orange", "Apple", "Mango"];  
fruits.splice(0, 1);
```


Q20. String methods in JS?

Ans: String methods are following:

There are 3 main methods to extract string parts:

1. slice(): slice() extracts a part of a string and returns the extracted part in a new string.

Example:

```
let str = "Apple, Banana, Kiwi";  
let part = str.slice(7, 13);
```

output:

Banana

Example2:

```
let str = "Apple, Banana, Kiwi";  
let part = str.slice(-12, -6);
```

2. substring(): substring() is similar to slice() method except that it does not accept negative value.

Example:

```
let str = "Apple, Banana, Kiwi";  
let part = str.substring(7, 13);
```

3. substr(): substr() is similar to slice().The difference is that the second parameter specifies the length of the extracted part.

Example:

```
let str = "Apple, Banana, Kiwi";  
let part = str.substr(7, 6);
```