JAVA SCRIPT BASICS

1. What are JavaScript data type ? What are primitive and non-primitive data types ? List them.

Ans=> in java script there are two type data type

Primitive data types: The predefined data types provided by JavaScript language are known as primitive data types. Primitive data types are also known as in-built data types.

1. **Number**
2. **String**
3. **Boolean**
4. **Undefined**
5. **null**

Non-primitive data types: The data types that are derived from primitive data types of the JavaScript language are known as non-primitive data types. It is also known as derived data types or reference data types.

1. **Object**
2. **Array**

2. What are truthy and falsy values in JS ?

Ans=>

In java script, a truthy value is a value that is considered `true` when encountered in a Boolean context. All values are truthy unless they are defined as falsy(i.e., except for `false`, `0`, `-0`, `0n`, `""`, `null`, `undefined`, and `NaN`).

Java script uses type coersion in Boolean contexts.

Examples of *truthy* values in JavaScript (which will be coerced to `true` in boolean contexts, and thus execute the `if` block):

The following values are always falsy:

- `false`
- `0` (zero)
- `-0` (minus zero)
- `0n` (BigInt zero)
- `''`, `""`, ``` `` ``` (empty string)
- `null`
- `undefined`
- `NaN`

Everything else is truthy. That includes:

- `'0'` (a string containing a single zero)
- `'false'` (a string containing the text “false”)
- `[]` (an empty array)
- `{}` (an empty object)
- `function(){}` (an “empty” function)

3. What are var, let and const ? How are they different and when they should be used ?

Ans=>

Diffrence between var let and const

var	let	const
The scope of a *var* variable is functional scope.	The scope of a *let* variable is block scope.	The scope of a *const* variable is block scope.
It can be updated and re-declared into the scope.	It can be updated but cannot be re-declared into the scope.	It cannot be updated or re-declared into the scope.
It can be declared without initialization.	It can be declared without initialization.	It cannot be declared without initialization.
It can be accessed without initialization as its default value is “undefined”.	It cannot be accessed without initialization, as it returns an error.	It cannot be accessed without initialization, as it cannot be declared without initialization.

4. Difference between '==' operator and '===' operator ?

Ans=>

They both are comperission operator and used to compare the two values

"==" compare only value not type(only equal)
"===" compare both value and the type also(check strikt equall)

5. What is typecasting in JavaScript ? What is implicit type coercion. Give an example.

Ans=>Typecasting or coercion in simple term means to *change the data type of a value to to another data* type like for example, integer to a string or a string to a boolean etc.
There are two types of coercion, implicit and explicit. Implicit coercion is when there is automatic conversion of data type, where as When a developer expresses the intention to convert between types by writing the appropriate code, it’s called explicit type coercion (or type casting).
JavaScript only supports three type of conversion

- to string
- to boolean
- to number

Exp=>

```
val = 5
String(val) //explicit coercion
'10' + val //105 not 15 and o/p is a String, implicit coercion
```

6. What is JSON and its common operations? How can you convert a String to its equivalent JSON object, if valid, and vice versa, JSON object to string ?

Ans=>

JavaScript Object Notation (JSON) is a standard text-based format for representing structured data based on JavaScript object syntax. It is commonly used for transmitting data in web applications (e.g., sending some data from the server to the client, so it can be displayed on a web page, or vice versa). You'll come across it quite often, so in this article we give you all you need to work with JSON using JavaScript, including parsing JSON so you can access data within it, and creating JSON.

If we want copy a json first we use `json.stringify()` to convert a json into string

And the second thing we convert a string into json with the help of `Json.parse()`

7. What do you mean by scope in JavaScript ?

Ans=>

***Scope* in JavaScript refers to the current context of code, which determines the accessibility of variables to JavaScript. The two types of scope are *local* and *global*:**

- **Global variables are those declared outside of a block**
- **Local variables are those declared inside of a block**

8. What is NaN property in JavaScript ?

Ans=>

In JavaScript, NaN stands for Not a Number. It represents a value which is not a valid number. It can be used to check whether a number entered is a valid number or not a number. To assign a variable to NaN value, we can use one of the two following ways.

In JavaScript, NaN is short for "Not-a-Number".
In JavaScript, NaN is a number that is not a legal number.
The Global NaN property is the same as the Number.Nan property

9. How do you add and access a property in a JavaScript object?

Ans=>

To add a new property to a Java script object, define the object name followed by the dot, the name of a new property, an equals sign, and the value for the new property.

Object .newProperty = value

To access a property to a java script object we can acceswith the help of dot syntax and square bracket notation

Object .ExistingProperty

It does not matter if you have to add the property, change the value of the property, or read a value of the property; you have the following choice of syntax.

1. Dot Syntax
2. Square Bracket Notation. (Dot syntax is more straightforward and is generally sufficient).

10. How do you sort numbers in a given array? What is the output if the Array contains strings instead of numbers. Example ['Oct', 'Jan', 'Feb', 'Dec'].

Ans=>

TO SORT A NUMBERS IN AN GIVEN ARRAY WE USE THE JAVA SCRIPT SORT METHOD
ARRAY.SORT()

TO SORT ASSENDING ORDER
ARRAY.SORT((a,b)=>a-b)

TO SORT DECENDING ORDER
ARRAY.SORT((a,b)=>b-a)

To short a array of string we use sort method to sort in alphabetic order

11. Name the Array functions to add an element, remove the last element, remove the first element, add an element at the beginning, find the index of an element (what happens when there are more than one such elements)
Ans=>

Array.push() => array.push useto add a elemnt at the end of array.

Array.pop() => array.pop userd to remove a element at the end of array.

Array.shift() => array.shift used to remove a first element of array.

Array.unshift() => array.unshift used to add a first element of array.

Array.indexOf() => array.indexOf method used to find the Index of an element of array

12. How do you find maximum and minimum value in an array ?
Ans=>

In java script theare are two approach to find the minimum and maximum value in an array

1. Java script inbuild function

Math.min(...array) => with the help of this function we can find the minimum element of array using spered operator.

Math.max(...array) => with the help of this function we can find the maximum element of array using spered operator

2. For iterative approach
Using for loop we can check the all element in array one by one and check the highest oe lowest value

13. What is the difference between splice and slice string methods?
Ans=>

Array.slice() =>1. This method is used to get a new array by selecting a sub-array of a given array.
2.The changes do not reflect in the original array.

array_name.slice(starting index, ending index)

The parameter 's' indicates the starting index and 'e' indicates the ending index.
They denote the index of the sub-array to be taken. By default, the value for start is '0' and end is 'n'.

Array.splice() => 1. This method is used to add/remove an item from the given array.
2. The changes reflect in the original array

array_name.splice(i, n, item 1, item 2, ...item n)

The parameter 'i' denotes the starting index, 'n' denotes the number of items to be removed from the specified starting index.'item 1, item 2, …..item n' represents the list of new items to be added at the given index. If n=0, no item is removed, the new items are just added to the specified starting index.

14. What do trim(), substr(), toLowerCase() and toUpperCase() and concat() string methods do?
Ans=>

Trim() => The trim() method removes whitespace from both sides of a string.
The trim() method does not change the original string.

SubStr() =>The substr() method extracts a part of a string.
The substr() method begins at a specified position, and returns a specified number of characters.
The substr() method does not change the original string.

ToLowerCase() =>The toLowerCase() method converts a string to lowercase letters.
The toLowerCase() method does not change the original string.

ToUpperCase() =>The ToUpperCase() method converts a string to uppercase letters.
The ToUpperCase() method does not change the original string.

Concat()=> The concat() method joins two or more strings.
The concat() method does not change the existing strings.
The concat() method returns a new string.

15. What is the difference between slice(), splice() and split() methods?
Ans=>

Array.slice() =>1. This method is used to get a new array by selecting a sub-array of a given array.
2.The changes do not reflect in the original array.

array_name.slice(starting index, ending index)

The parameter 's' indicates the starting index and 'e' indicates the ending index.
They denote the index of the sub-array to be taken. By default, the value for start is '0'
and end is 'n'.

Array.splice() => 1. This method is used to add/remove an item from the given array.
2. The changes reflect in the original array
array_name.splice(i, n, item 1, item 2, ...item n)

The parameter 'i' denotes the starting index, 'n' denotes the number of items to

be removed from the specified starting index.‘item 1, item 2, …..item n’ represents the list of new items to be added at the given index. If n=0, no item is removed, the new items are just added to the specified starting index.

Array.split() => The split() method splits a string into an array of substrings.
The split() method returns the new array.
The split() method does not change the original string.
If (" ") is used as separator, the string is split between words.

16. What are functions in JavaScript. How can we declare functions and call them ?
Ans=>
Functions are one of the fundamental building blocks in JavaScript. A function in JavaScript is similar to a procedure—a set of statements that performs a task or calculates a value, but for a procedure to qualify as a function, it should take some input and return an output where there is some obvious relationship between the input and the output. To use a function, you must define it somewhere in the scope from which you wish to call it.

We can declare a function with the function keyword and varaiabel name and urly braces

```
function square(number) { return number * number; }
```

17. What is difference between While and do-while loop. What is for loop and what is its syntax ?
Ans=>

- While loop checks the condition first and then executes the statement(s), whereas do while loop will execute the statement(s) at least once, then the condition is checked.
- While loop is entry controlled loop whereas do while is exit controlled loop.
- In the while loop, we do not need to add a semicolon at the end of a while condition but we need to add a semicolon at the end of the while condition in the do while loop.
- While loop statement(s) is executed zero times if the condition is false whereas do while statement is executed at least once.
- While loop allows initialization of counter variable before starting the body of a loop whereas do while loop allows initialization of counter variable before and after starting the body of a loop.

18. What does break and continue does in loop ?
Ans=>
Break statement break the loop while condition is fullfield
Continue statement contnu ethe loop while the condition is not fullflled

19. What is the function of a return statement?
Ans=>

Return statemnt return the output of function if we cant us th ereturb statemnet at the end of function return already exicuted

20. What is typeof operator ? And for an undeclared variable what will be its typeof ?
Ans=>

A type of oprator use the typeof operator to find the data type of a JavaScript variable.

For an undeclared variable the type of variable is undefined