

# Interview Questions – JAVASCRIPT

## (Web Design)

### 1. What are the different data types present in javascript?

To know the type of a JavaScript variable, we can use the **typeof** operator.

#### 1. Primitive types

**String** - It represents a series of characters and is written with quotes. A string can be represented using a single or a double quote.

Example :

```
var str = "Vivek Singh Bisht"; //using double quotes
var str2 = 'John Doe'; //using single quotes
```

- **Number** - It represents a number and can be written with or without decimals.

Example :

```
var x = 3; //without decimal
var y = 3.6; //with decimal
```

- **BigInt** - This data type is used to store numbers which are above the limitation of the Number data type. It can store large integers and is represented by adding “n” to an integer literal.

Example :

```
var bigInteger = 234567890123456789012345678901234567890;
```

- **Boolean** - It represents a logical entity and can have only two values : true or false. Booleans are generally used for conditional testing.

Example :

```
var a = 2;
var b = 3;
var c = 2;
(a == b) // returns false
(a == c) //returns true
```

- **Undefined** - When a variable is declared but not assigned, it has the value of undefined and its type is also undefined.

Example :

```
var x; // value of x is undefined
var y = undefined; // we can also set the value of a variable as undefined
```

- **Null** - It represents a non-existent or a invalid value.

Example :

```
var z = null;
```

- **Symbol** - It is a new data type introduced in the ES6 version of javascript. It is used to store an anonymous and unique value.

Example :

```
var symbol1 = Symbol('symbol');
```

- **typeof of primitive types :**

```
typeof "John Doe" // Returns "string"
```

```
typeof 3.14 // Returns "number" typeof
```

```
true // Returns "boolean"
```

```
typeof 234567890123456789012345678901234567890n // Returns bigint typeof
```

```
undefined // Returns "undefined"
```

```
typeof null // Returns "object" (kind of a bug in JavaScript)
```

```
typeof Symbol('symbol') // Returns Symbol
```

## 2. Non-primitive types

- Primitive data types can store only a single value. To store multiple and complex values, non-primitive data types are used.
- Object - Used to store collection of data.
- Example:

```
// Collection of data in key-value pairs
```

```
var obj1 = {  
  x: 43,  
  y: "Hello world!",  
  z: function(){  
    return this.x;  
  }  
}
```

```
// Collection of data as an ordered list
```

```
var array1 = [5, "Hello", true, 4.1];
```

## 2. Difference between “ == “ and “ === “ operators.

Both are comparison operators. The difference between both the operators is that “==” is used to compare values whereas, “===” is used to compare both values and types.

**Example:**

```
var x = 2;
```

```
var y = "2";
```

```
(x == y) // Returns true since the value of both x and y is the same
```

```
(x === y) // Returns false since the typeof x is "number" and typeof y is "string"
```

### 3. Difference between var and let keyword in javascript.

Some differences are

- From the very beginning, the 'var' keyword was used in JavaScript programming whereas the keyword 'let' was just added in 2015.
- The keyword 'Var' has a function scope. Anywhere in the function, the variable specified using var is accessible but in 'let' the scope of a variable declared with the 'let' keyword is limited to the block in which it is declared. Let's start with a Block Scope.
- In ECMAScript 2015, let and const are hoisted but not initialized. Referencing the variable in the block before the variable declaration results in a ReferenceError because the variable is in a "temporal dead zone" from the start of the block until the declaration is processed.

### 4. What is NaN property in JavaScript?

NaN property represents the “**Not-a-Number**” value. It indicates a value that is not a legal number. **typeof** of NaN will return a **Number**.

To check if a value is NaN, we use the **isNaN()** function,

Note- isNaN() function converts the given value to a Number type, and then equates to NaN.

isNaN("Hello") // Returns true

isNaN(345) // Returns false

isNaN('1') // Returns false, since '1' is converted to Number type which results in 0 ( a number) isNaN(true) // Returns false, since true converted to Number type results in 1 ( a number) isNaN(false) // Returns false isNaN(undefined) // Returns true

### 5. What are the features of JavaScript?

These are the features of JavaScript:

- Lightweight, interpreted programming language
- Cross-platform compatible
- Open-source
- Object-oriented
- Integration with other backend and frontend technologies
- Used especially for the development of network-based applications

### 6. What are the advantages of JavaScript over other web technologies?

These are the advantages of JavaScript:

### *Enhanced Interaction*

JavaScript adds interaction to otherwise static web pages and makes them react to users' inputs.

### *Quick Feedback*

There is no need for a web page to reload when running JavaScript. For example, form input validation.

### *Rich User Interface*

JavaScript helps in making the UI of web applications look and feel much better.

### *Frameworks*

JavaScript has countless frameworks and libraries that are extensively used for developing web applications and games of all kinds.

## **7. How do you create an object in JavaScript?**

Since JavaScript is essentially an object-oriented scripting language, it supports and encourages the usage of objects while developing web applications.

```
const student = {  
  name: 'John', age:  
  17  
}
```



We believe in quality  
Since 2014

## **8. How do you create an array in JavaScript?**

Here is a very simple way of creating arrays in JavaScript using the array literal:

```
var a = []; var b = ['a', 'b',  
'c', 'd', 'e'];
```

## **9. What are the scopes of a variable in JavaScript?**

The scope of a variable implies where the variable has been declared or defined in a JavaScript program. There are two scopes of a variable:

Global Scope

Global variables, having global scope are available everywhere in a JavaScript code.

Local Scope

Local variables are accessible only within a function in which they are defined.

## 10. What are the conventions of naming a variable in JavaScript?

Following are the naming conventions for a variable in JavaScript:

Variable names cannot be similar to that of reserved keywords. For example, var, let, const, etc.

Variable names cannot begin with a numeric value. They must only begin with a letter or an underscore character.

Variable names are case-sensitive.

## 11. What's the difference between let and var?

Both let and var are used for variable and method declarations in JavaScript. So there isn't much of a difference between these two besides that while var keyword is scoped by function, the let keyword is scoped by a block.

## 12. What are the different ways an HTML element can be accessed in a JavaScript code?

Here are the ways an HTML element can be accessed in a JavaScript code:

`getElementByClass('classname')`: Gets all the HTML elements that have the specified classname. `getElementById('idname')`: Gets an HTML element by its ID name.

`getElementbyTagName('tagname')`: Gets all the HTML elements that have the specified tagname.

`querySelector()`: Takes CSS style selector and returns the first selected HTML element

## 13. What is the difference between Undefined and Undeclared in JavaScript?

Undefined	Undeclared
Undefined means a variable has been declared but a value has not yet been assigned to that variable.	Variables that are not declared or that do not exist in a program or application.

## 14. What is the difference between Undefined and Null in JavaScript?

Undefined	Null

Undefined	Null
Undefined means a variable has been declared but a value has not yet been assigned to that variable.	Null is an assignment value that we can assign to any variable that is meant to contain no value.

### **15. What is the 'this' keyword in JavaScript?**

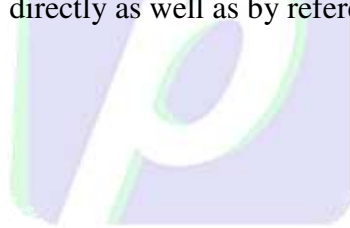
The Keyword 'this' in JavaScript is used to call the current object as a constructor to assign values to object properties.

### **16. Is javascript a statically typed or a dynamically typed language?**

Yes, JavaScript is a dynamically typed language and not statically.

### **17. What is BOM?**

BOM is the Browser Object Model where users can interact with browsers that is a window, an initial object of the browser. The window object consists of a document, history, screen, navigator, location, and other attributes. Nevertheless, the window's function can be called directly as well as by referencing the window.



We believe in quality

Since 2014