# **JavaScript interview Questions - Part 01**

## Basic js interview questions:

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

Ans:- There are 7 different data types in javaScript.

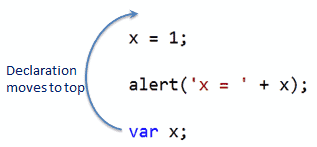
1. Number
2. String
3. Boolean
4. Undefined
5. Null
6. Symbol and
7. Object (Non primitive data type)

Top 6 data types come in “primitive data type” and the last one (Which is object), coms in “Non primitive data type”.

Primitive data types are only available in ES6.

**2. Explain Hoisting in javascript.**

Ans: During the creation phase, the JavaScript engine **moves the variable and function declarations to the top** of the code. And that is called Hoisting in JavaScript. (Hoisting doesn’t move the initialization value, so **if you invoke the value before initialization, It will come “Undefined”**).



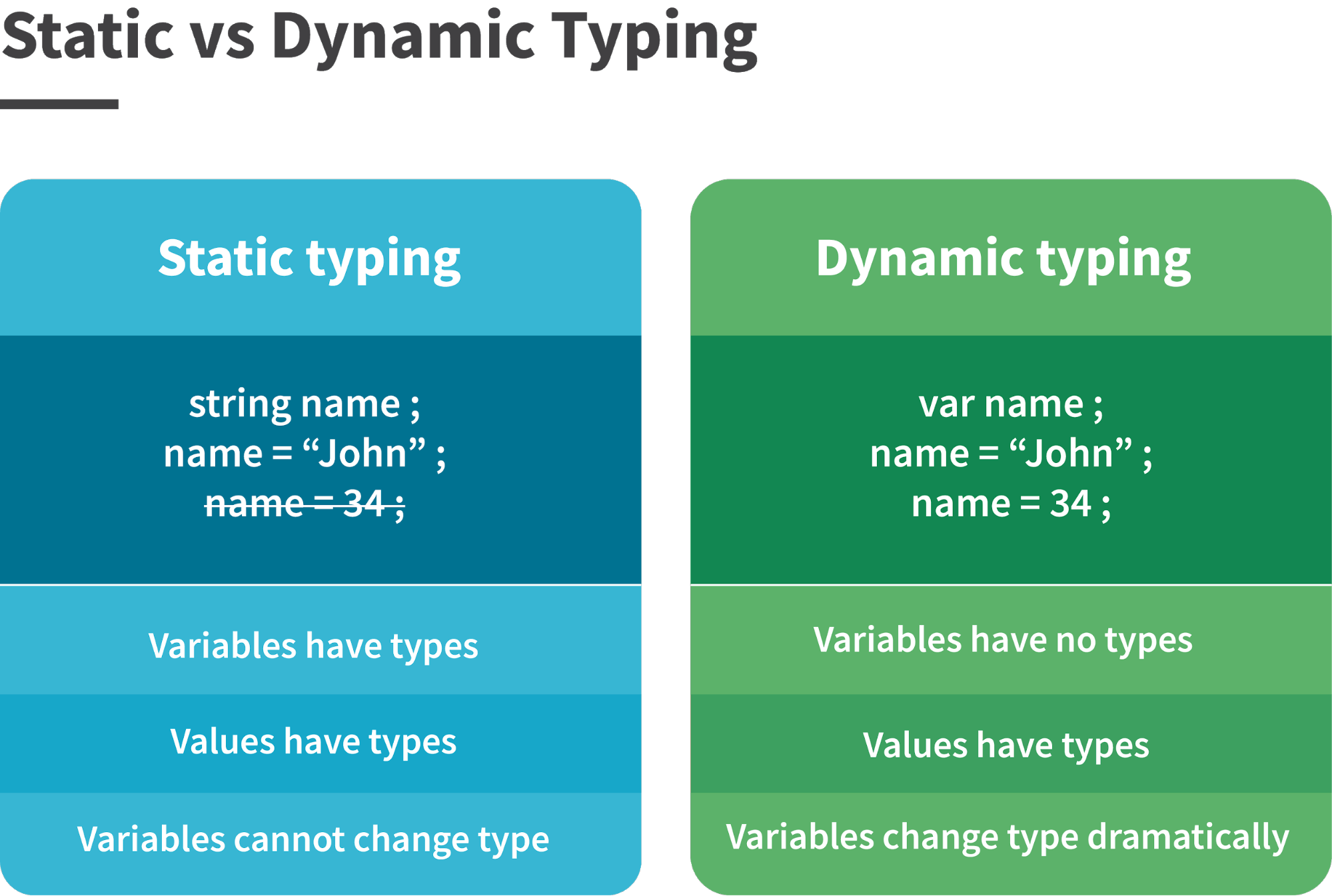
**3. Difference between “==” and “===” operators.**

Ans: Both are comparison operators. The difference between both is “==” compare the value whether “===”(Strict Equal) compare the value and data type both. (Or you can say Strict equal “===” **doesn’t convert the value** before comparison.)

4. Explain Implicit Type Coercion in javascript.

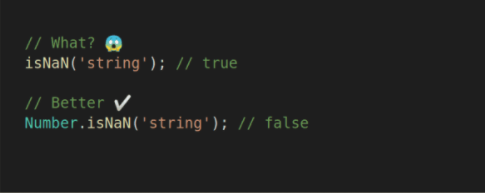
**5. Is javascript a statically typed or a dynamically typed language?**

Ans: JavaScript is a **dynamically typed** language. where the **interpreter assigns variables a type at runtime based** on the variable's value at the time.



**6. What is NaN property in JavaScript?**

Ans: Nan Stands for **“Not A Number”.** But **typeof NaN is a Number**. Also, **Any operation with NaN returns NaN**. Even The NaN does not equal any value, including itself.

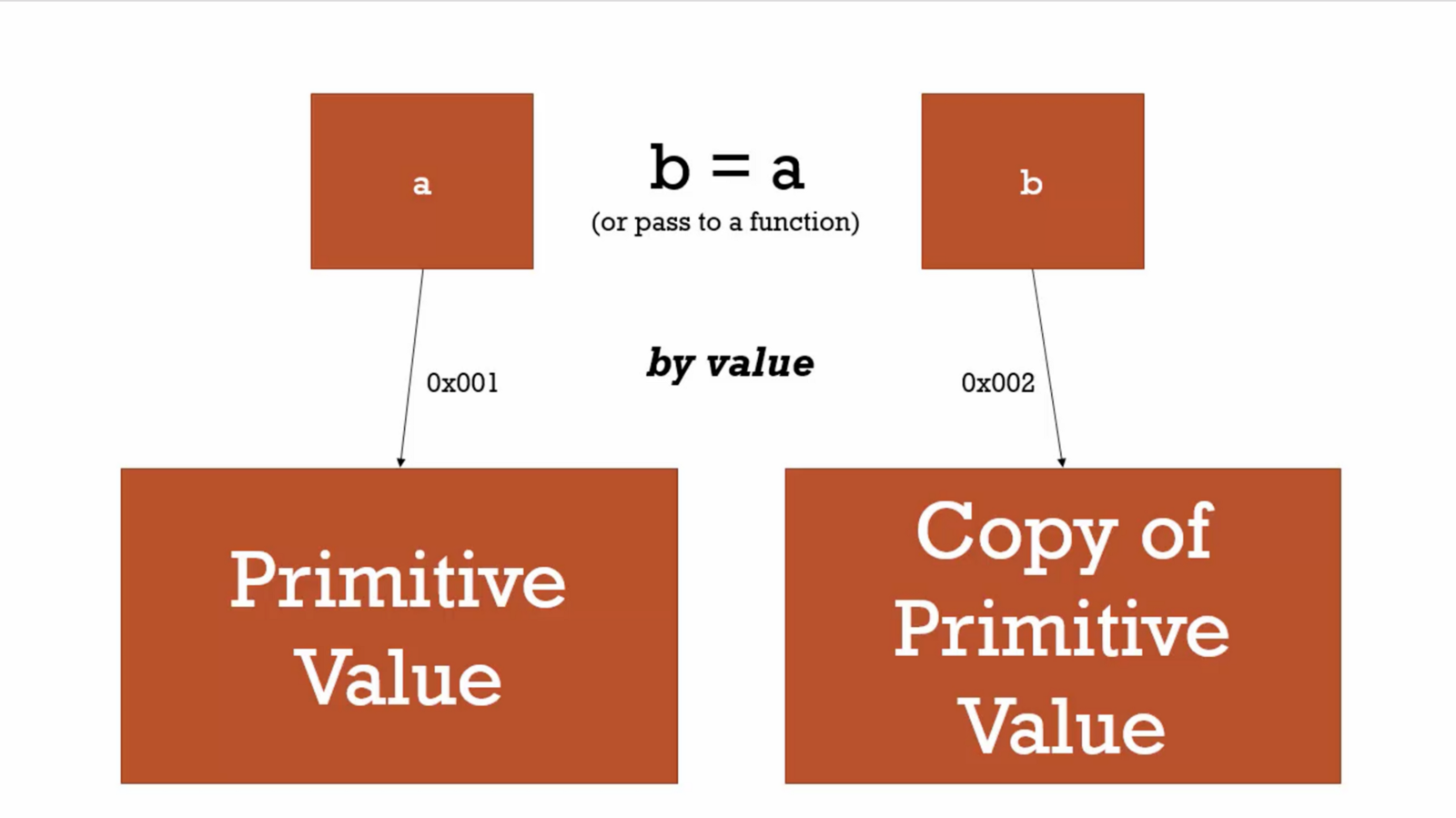


**7. Explain passed by value and passed by reference.**

Ans: In JavaScript, **primitive data types are passed by value** and **non-primitive data types are passed by reference**.

passed by value passed by reference

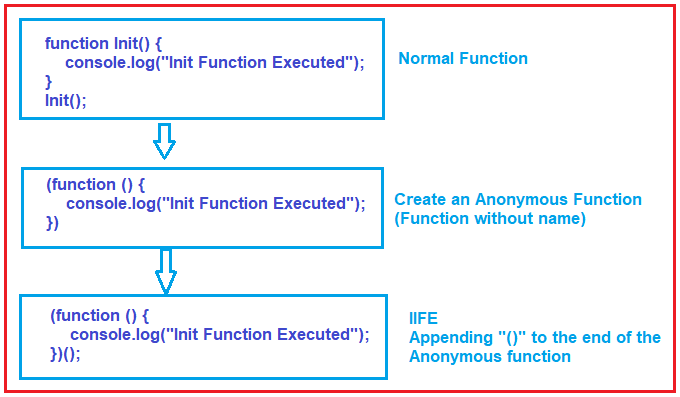
Var a=10; b=a;



**8. What is an Immediately Invoked Function in javascript?**

Ans: An Immediately Invoked Function ( known as IIFE and pronounced as IIFY) is a function that runs as soon as it is defined.

There is three type for immediately invoked Function:



**9. Explain Higher Order Functions in javascript.**

Ans: Functions that operate on other functions, either by taking them as arguments or by returning them, are called higher-order functions.

Higher order functions are a result of functions being first-class citizens in javascript.

**function** higherOrder2() {

**return** **function**() {

**return** "Do something";

}

}

**var** x = higherOrder2();

x() *// Returns "Do something"*

**10. Explain “this” keyword.**

Ans: The “this” keyword refers to the object that the function is a property of. The value of “this” keyword will always depend on the object that is invoking the function.

In other words you can say it points to a particular Object.

For ex:

const person = {

firstName: "John",

lastName : "Doe",

id : 5566,

fullName : function() {

return this.firstName + " " + this.lastName;

}

};

So if you invoke this, the answer will come “**John Doe**”. because we have pointed **“This”** keyword to **firstName and this.lastName.**

11. Explain call(), apply() and, bind() methods.

12. What is Currying in javascript?

**13. Explain Scope and Scope Chain in javascript.**

Ans: **Scope** determines the **visibility and accessibility** of a variable.

JavaScript has three Scope:

1. Global Scope/Global Variable
2. Local Scope/local variables and
3. Block Scope/ block scoped variable

**Now let's talk about Scope chain**: JavaScript resolves a variable by looking it in its current scope, if it cannot find the variable, it goes up to the outer scope, is called the scope chain.

**var** y = 24;

**function** favFunction(){

**var** x = 667;

**var** anotherFavFunction = **function**(){

console.log(x); *// Does not find x inside anotherFavFunction, so looks for variable inside favFunction, outputs 667*

}

**var** yetAnotherFavFunction = **function**(){

console.log(y); *// Does not find y inside yetAnotherFavFunction, so looks for variable inside favFunction and does not find it, so looks for variable in global scope, finds it and outputs 24*

}

anotherFavFunction();

yetAnotherFavFunction();

}

favFunction();

14. Explain Closures in JavaScript.

15. What are object prototypes?

**16. What are callbacks?**

Ans: A callback is a function passed as an argument to another function. This technique allows a function to call another function. A callback function can run after another function has finished.

function myDisplayer(some) {

document.getElementById("demo").innerHTML = some;

}

function myFirst() {

myDisplayer("Hello");

}

function mySecond() {

myDisplayer("Goodbye");

}

myFirst();

mySecond();

17. What is memoization?

18. What is recursion in a programming language?

19. What is the use of a constructor function in javascript?

**20. What is DOM?**

Ans: The Document Object Model ( DOM) is an application programming interface (API) for manipulating HTML and XML documents

The DOM represents a document as a tree of nodes. It provides an API that allows you to add, remove, and modify parts of the documents effectively.

