How Many Ways To Insert JS

JavaScript, also known as JS, is one of the scripting (client-side scripting) languages, that is usually used in web development to create modern and interactive web-pages. The term "script" is used to refer to the languages that are not standalone in nature and here it refers to JavaScript which run on the client machine.

1. **internal JS:**

   By using script tag at the bottom of the document

   Syntax:

```
<body>
    <script>
    Console.log(“hello world”);
    </script>
</body>
```

2. **inline JS:**

   we can apply inline js within the element.

   Syntax:

```
<body>
    <button onclick=”alert(‘hello world’)”>click</button>
</body>
```

3. **external JS:**

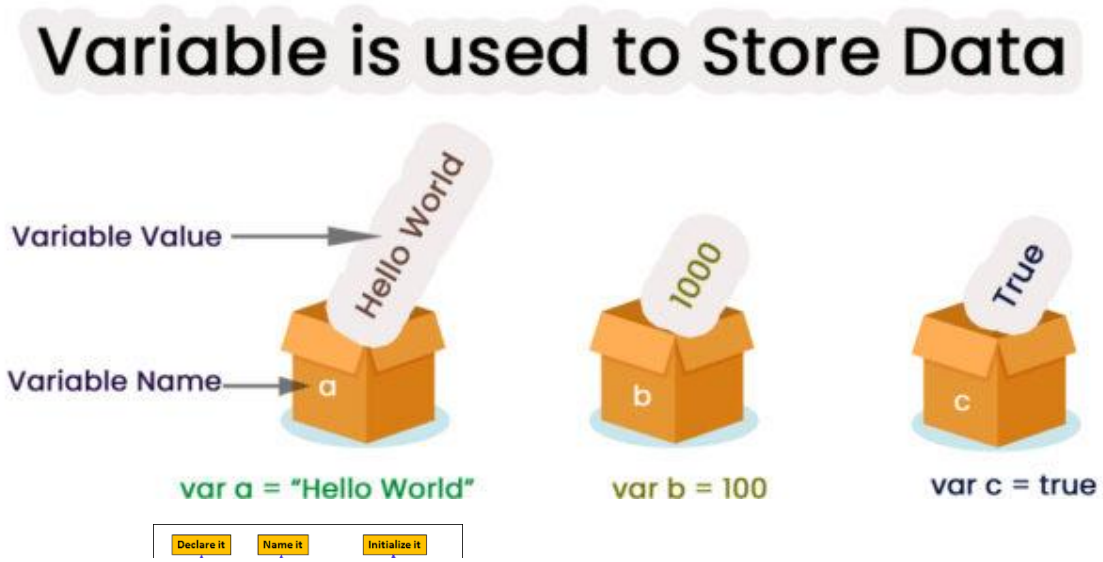
   By creating a js file with .js extention we can insert js file into the html document. That js file is linked in the script tag.

   Syntax:

```
<body>
    <script src=”js file with .js extention”></script>
</body>
```

Variables:

Variables are used to store data in JavaScript. Variables are used to store reusable values. The values of the variables are allocated using the assignment operator(“=”).



JavaScript Variables can be declared in 4 ways:

- **Automatically**
- **Using var**
- **Using let**
- **Using const**

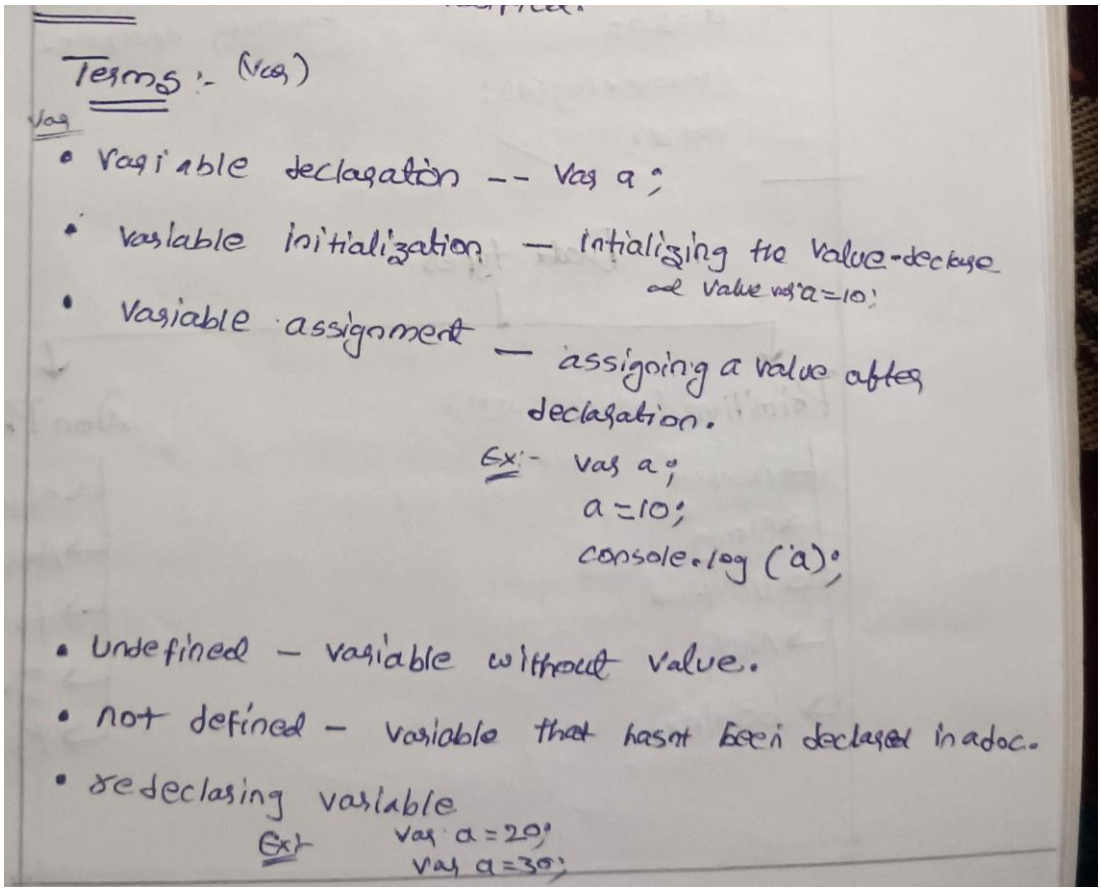
Example:

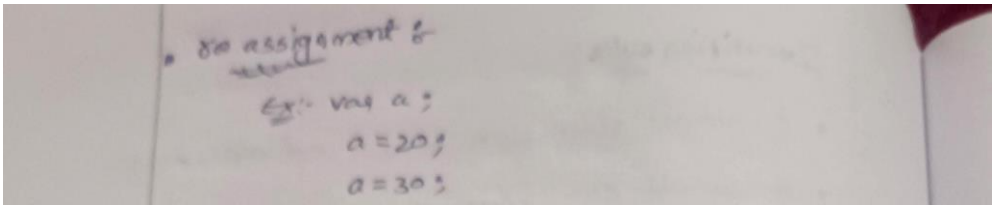
```
a=10;
Var b=20;
let c=5;
Const d=15;
console.log(a);     //10
console.log(a);    //20
console.log(a);   //5
console.log(a);   //15
```

Rules for Identifiers:

- Names can contain letters, digits, underscores, and dollar signs
- Identifier should not start with number
- Names must begin with a letter or _ or $
- Names are case-sensitive
- Reserved words cannot be used as Identifier.

Terms:





Dynamic typing:

Js is a dynamically typed, meaning you do not have to specify the datatype of the variable when declared. The Data type of the variable is determined automatically in a runtime.

Hoisting

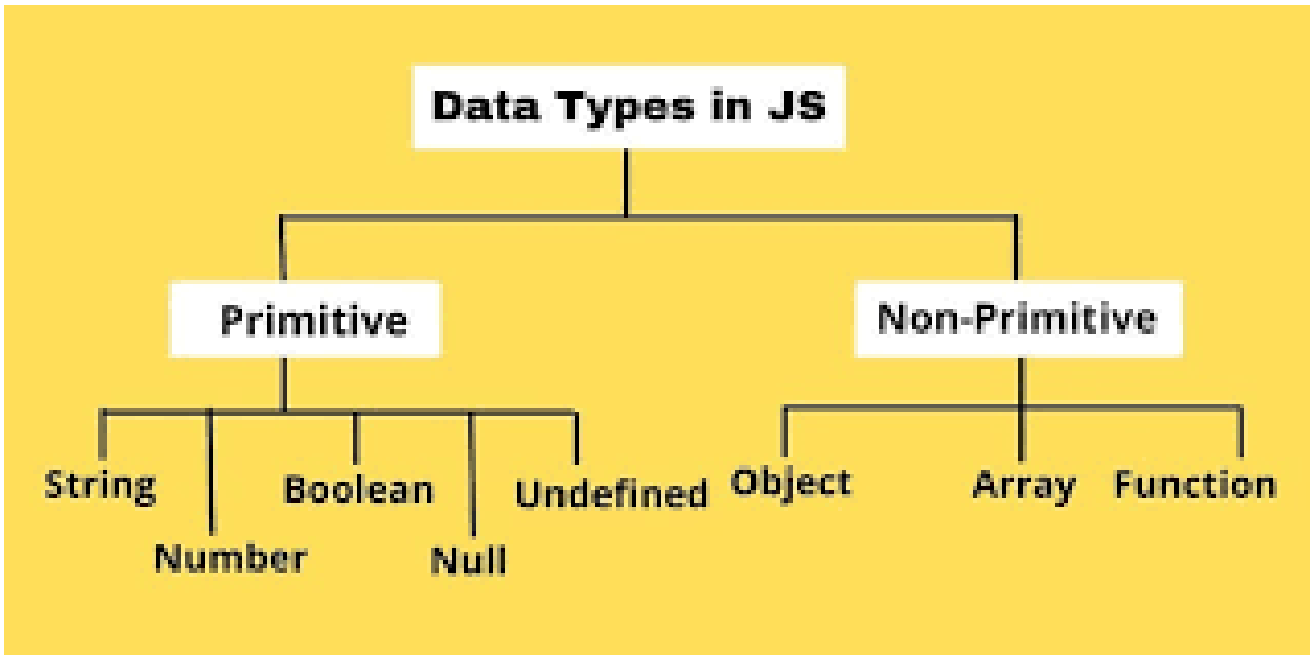
- It is a behaviour where the declaration of the variable and functions are moved to the top even before the execution.
- Only Declaration is hoisted not the Initialization
- Only works in **var** remaining **let** and **const** goes to temporary deadzone

Example:

```
a=20;
Console.lob(a);     //20
var a;
```

Data types

JavaScript provides different **data types** to hold different types of values. There are two types of data types in JavaScript.



Primitive Data Types :-

Primitive data types are the fundamental building blocks used to represent single values.

- Primitive data types which is stored in stack (call by value and pass by value)
- which are immutable. we can access the data but cannot change.

Non Primitive Data types :- (or) Composite data type

- used to represent multiple values
- non primitive data types are stored in heap. (call by reference and pass by reference)
- which are mutable (we can change data).

Primitive

- Numbers :- represents numeric values

  Eg :- var a = 100;

- String :- represents Group of characters

  eg :- var b = "Hello"

- Boolean :- represents boolean value either true - 1 or false - 0

- null :- represents null. ie. no value at all (intentionally Empty value)
- Undefined :- variable with out value (declared but not assigned value)

Non primitive

- array :- represents group of similar Elements
- Objects :- represents instance through which we can access members.
- functions :- it is a block of code to perform Particular task & it is reusable.
- date
- reg exp :- represents regular expressions.

Examples :-

primitive

```
var a = 20;  //number
var b = 'hello'; //string
var c = true; //boolean
var d = false
var e = null
var f = undefined

console.log(c+f);  //NaN
```

non primitive

arrays (number index)

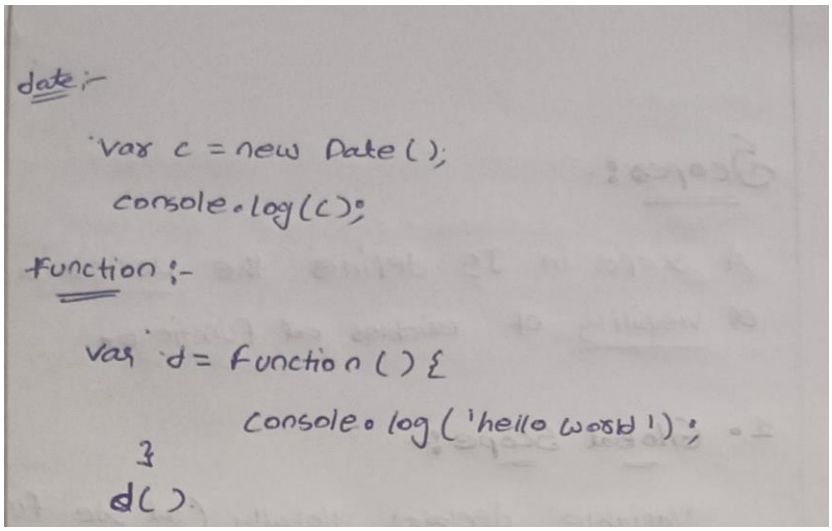
```
var a=[20, 'hello', true];

a[0] = 30;
console.log(a);  //[30, 'hello', true]

Console.log(a[-1]);
```

objects (named index)

```
var b={
        id:1201,
        name: 'abhi',
        age: 22
    }
Consde.log(b.age);  //22
```



Typeof:

The typeof operator returns the data type of a variable.

The JavaScript typeof operator returns the data type of a variable or expression. It's a unary operator placed before its operand and returns a string indicating the data type, such as "number", "string", "boolean", "object", "undefined", "function", or "symbol".