Javascript

- undefined: A variable is declared but not defined. assigned.
- Null : Represent non-existent or Invalid value
- Primitive datatypes can store single value, e.g Number, string, boolean, undefined, null, symbol
- Non-primitive datatype :- Used to store multiple and complex values.
  Eg objects.
- **Difference b/w "==" or "==="**

"==" : Loose comparison, only value is compared. If value is same return true else return false.

"===" : Strict comparison, compare value and datatype both.

- **Difference b/w Var and let.**

1. The keyword var has a functional scope. Anywhere in the function, variable var is accessible	The keyword Let has a block scope. It is limited to the block in which it is declared

Var are hoisted and are initialized to undefined and results in no reference error.	let and const are not hoisted but not initialized Referencing a variable before its declaration results in Reference Error.

- **Implicit type coercion**

It is the automatic Conversion of value from one datatype to another. • It take place when the operands of an expression are of different datatype.

1. String Coercion
- Take place while using '+'

```
var x = 3          (Number type is converted to String)
var y = "Hello"
x + y  // "3Hello"
```

2. Take plaqce while using '-' operator

```
var x = 3
var y = "3"   (String converted to Number)
x - y  // return 0
```

2. Boolean Coercion
Take place while using boolean/Logical operators, ternary operator, if statements, and loop checks.

All values except false, 0, 0n, -0, "", null undefined and NaN are Truthy values.

DATE: __/__/__
PAGE: ____

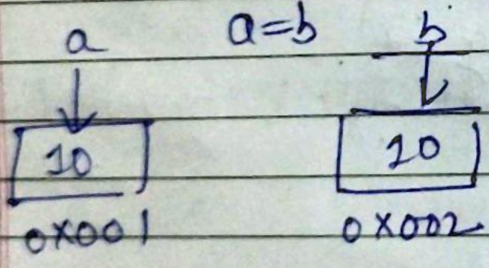
- JavaScript is dynamically typed language. In dynamically typed language the type of variable is checked during runtime incontrast to statically typed language where type of variable is checked during compile time.

- **NaN (Not a Number)**
It indicates the value is not a legal Number.

- **Passed by value**
Primitive Datatypes are passed by values

a=10

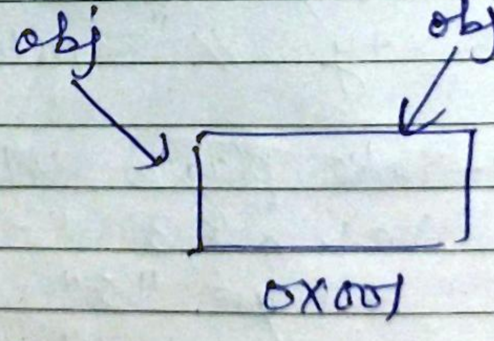
a=b



Passed by Reference
Non-primitive datatypes are passed by reference.

var obj = { Name:
age:
}

var obj1 = obj



- **Immediately invoked function**: — A function that run as soon as its ~~call~~ defined.

```
        No Name
(function() {  });)(); // run

function (){ }; // Compile time error as
                   function is declared without
                   name.
```

DATE : __/__/__
PAGE : _____

- **Strict Mode**

In Strict Mode all types or forms of errors, including silent errors will be thrown. As a result debugging becomes alot simpler.

↳ In strict mode, duplicates arguments are not allowed

→ You won't be able to use javaScript Keyword as parameter or function name.

→ You are not allowed to create global variables in strict mode.

- **Higher Order Function**

Functions that operates on other functions, either by taking them as arguments or returning them. are called higher order function.

Ex setTimeout

- **this Keyword**

The this keyword refers to the object that the function is a property of.

The value of the "this" keyword will always depend on the object that is invoking the function.

- **Self Invoking**

var x = xoyz() { })();

DATE : __/__/__
PAGE :

- **Call**

i) This method invokes method (function) by specifying the owner object.

```
function SayHello(){
    return "Hello" + this.name;
}
var obj = {name: "Sandy"};
SayHello(obj); // Hello Sandy
```

ii) Call() method allows an object to use the method of different objects.

```
var person {
    age: 23
    getAge: function()
    {
        return this.age;
    }
}
var person2 = {age: 54}
person.getAge.call(person2) // 54.
```

iii) call Accept Arguments.

- **Apply Method.**

The apply method is similar to call Method. The only difference is that in call Method takes argument seprately, whereas apply method takes argument as array.

DATE : __/__/__
PAGE : _____

- **Bind() Method**

This method returns a new function, where the value of "this" Keyword will be bound to the owner object, which is provided as a parameter.

- **Exec() vs test()**

test() and Exec() are RegExp expression methods used in javaScript

Exec() method is use to search a string for a specific pattern, and if it finds it, it will directly return the pattern directly. else it will return an empty result	test() will find a string for a specific pattern. It will return true on finding the given text otherwise it will return false.

- **Currying**

Currying is an advanced technique to transform a function of arguments n, to n functions of one or fewer arguments.

```
Ex function 'add(a) {
        return function(b) {
            return a+b;
        }
    }
3.
add(3)(4)
```

If we have function f(a,b) then after currying, it will be transformed to f(a)(b).

By using Currying technique, we do not change the functionality of a function, we first change the way it is invoked.

Ex:

```
function multiply(a,b)
{
  return a*b;
}

function Currying(fn)
{
  return function(a){
    return function(b)
    {
      return fn(a,b);
    }
  }
}

var curriedMultiplied = currying(multiply);
multiply(4,3)          // 12
curriedMultiply(4)(3)  // 12
```

- **Advantages of using External JavaScript** (code of javascript written in seperate file)
  - → Allow web designers and developers to collaborate on HTML and JavaScript files.
  - → we can reuse the code.
  - → Code readability is simple.

DATE: __/__/__
PAGE:

• Scope and Scope Chain

scope in JS determines the accessibility of variables and functions at various parts of one's code.

Types
- Global
- Local
- Block

Global: Functions declared in global namespace have global scope which means all variables and functions having global Scope can be accessed from anywhere inside the code.

Function Scope: Any variable or function declared inside a function have local/function scope means variables declared inside function can only be accessed inside the function & not outside.

Block Scope:- Block scope is related to the variables declared using let and const. Variables declared with using var do not have block scope.

Scope Chain

When javaScript engine does not find variable in local Scope, it tries to check for the variable in the outer scope. If the variable does not exist. In the outer scope, it tries to find the variable in global scope like a chain

DATE : __/__/__
PAGE :

• Advantages of JavaScript

1. JavaScript can be executed on client side as well as server side.
2. Simple to learn.
3. To the end user, JavaScript is quite quick.

• **Callbacks**: A callback is a function that will be executed after another function gets executed. The functions that are passed as arguments and can be returned from other functions are called first order or callback functions.

• Types of Error in JavaScript

1) Syntax Error :- Syntax Error are mistakes or spelling problems in the code the cause program to not execute at all.

2) Logical Error :- It occur when Logic of program is incorrect.

• **Memoization** :- It is a form of caching where the return value of function is cached based on the its parameters.

It is used for expensive function calls.

DATE :__/__/__
PAGE :_____

- **Constructor functions** in JavaScript are used to create objects in JavaScript.
If we want to create multiple objects having similar properties and methods then we use constructors.

Ex
```
var Person(name, age, gender)
{
    this.name = name;
    this.age = age;
    this.gender = gender;
}
var Person1 = new Person("Vinet", 76, "male");
```

- **DOM** (Document Object Model)
DOM is a programming interface for HTML and XML documents

Whenever the browser tries to render an HTML document. It creates an object based on the HTML document called DOM. Using this Dom we can manipulate or change various elements inside HTML document.

- **BOM** (Browser Object Model)
It allow user to interact with browser.
A browser's initial object is window