

# UpSkills Report

## JavaScript Session 1:

Topic :

1. History of JavaScript
2. Pareto Principle
3. Variables and classification of variables

On this day as it was the first lecture, it began with the introduction of JavaScript. Samal Sir told about history and journey of JavaScript.

- The first browser are introduced in 1995 names as NETSCAPE and it is developed in Mocha.
- The first and original name of JavaScript is ECHMAScript.
- ECMAScript is a subset of JavaScript. JavaScript is basically ECMAScript at its core but builds upon it. Languages such as ActionScript, JavaScript, JScript all use ECMAScript as its core.
- JavaScript is used for frontend programing used to developed web pages using html, css, etc
- JavaScript enables interactive web pages  
But as we know java and JavaScript is named as almost same but there is no similarity between java and JavaScript. JavaScript is used for front end development and java is used for backend development.
- In JavaScript object contains key value pair, key contains only string value.
- JavaScript is dynamic type language, we do not need to specify type of variable because it is dynamically assigned by the JavaScript engine. JavaScript is a loosely coupled language and there Is no limitation in JS because Everything in JS is an object and object is container and Container holds anything.

## 2: Parento Principle :

**The**  **Rule**

The Pareto Principle, also known as the 80/20 Rule, The Law of the Vital Few and The Principle of Factor Sparsity, illustrates that 80% of effects arise from 20% of the causes – or in lamens terms – 20% of your actions/activities will account for 80% of your results/outcomes.

### 3: Variables and Classification:

**Var** keyword to assigned variable In JavaScript (ES5).

JavaScript provides 5 data types that is

- String
- number
- Boolean
- undefined
- null;

And JavaScript classified into 2 types that is **primitive and Non-primitive**

**Primitive** data Type are

- String
- Number
- Boolean
- Undefined
- Null

Properties of primitive type is pass by value or copy by value.

**Non-primitive** dataTypes are

- Object and Array

Properties of non primitive type is pass by reference or copy by reference

Array and object are container it means it can contain and hold anything, but actually javascript array is object in nature.

Object contains key value pair, key contains only string

undefined is assigned by javascript engine.

Undefined means variable has been declared but no assigned value to variable.

e.g. `var a;`

`var b;`

Null is assigned by programmer or developer.

Null is datatype or keyword which is assigned to the variable by developer.

## JavaScript Session 2:

Topic :

1. Copy by value and copy by reference
2. Concatenation with String in JavaScript
3. Operators in JavaScript
4. Difference between == and === operator
5. Different Tech Debt in JavaScript

1:

### Call by value:

- When a variable is passed as a parameter to a function, if any changes are made to parameter, the original variable will not get affected. This is known as **call by value**.
- In JavaScript all primitive data type are call by value i.e string, number, Boolean, null, undefined.

### Call by reference:

- When a variable is *reference* and not its value is passed to a function's parameter, it will get affected to the original variable and update the original variable reference. This is known as **call by reference**.
- In javascript all the non-primitive datatypes are call by reference i.e object and array.

## 2: String Concatination:

The condition where plus operator in JS acts as string concatenation

- Number + String = String
- String + Number = String
- String + String = String
- Array + Number = String
- Number + Array = String
- Number + Object = String
- Array + Array = String
- String + object = String
- String + Null = String
- String + Undefined = String
- Object + Object = String
- Number + undefined = NaN
- Null + null = 0
- Boolean + null = Number

### 3: Operators in JavaScript

- Arithmetic Operator.
- Logical Operator.

Arithmetic Operators : -

+	: Addition.
-	: Subtraction.
*	: Multiplication.
/	: Division.
%	: Modulus (Division Reminder)

Logical Operator : -

&&	: Logical and.
	: Logical or.
!	: Logical not.

Comparison operator : -

==	: Equal to.
===	: Equal value and equal type.
!=	: Not equal.
>	: Greater than
<	: Less than.
>=	: Greater than or equal to.
<=	: Less than or equal to.

### 4: Difference between == and === operator

- == and === is a conditional operator which compare one value with other and check the equality and returns true or false.
- In Js == operator is tech debt, generally == operator is used to check equality but in JavaScript it is not properly work.
- To overcome this problem === operator is introduced in Js.
- === operator first checks type and then checks content(value) if both condition is true it return true otherwise returns false.

So in javascript developer always use === operator for correct Answer.

- e.g.     1==1                 // True
  - 1=="1"                 // In js it return true but it is wrong
  - 1==="1"                // False
  - 1===1                  // True
  - 1==(+1)                // True
- In JavaScript, there is a one bug with + operator while between Addition and concatenation the String. Developer need to know this bug.

- To avoid this problem use + operator in front of variable like +val to convert string to number. or there is one built in function in JavaScript i.e. Number().

## 5: Tech debt in JavaScript.

Tech debt is a bug present in JavaScript which will not be solved by anyone. This is an inbuilt bug in logic which developers need to take care of or know when they implement some logic and calculation.

Tech debt in JavaScript is as follows:

- NULL
- == Operator
- + Operator

## JavaScript Session 3:

Topic:

1. Truthy and falsy values
2. Hoisting in JavaScript
3. Local scope and global scope

### 1: Truthy and falsy values

All the non-primitive values are always truthy values because they contain objects & objects are containers.

the number except 0 is truthy, true  
var v1 = "string", [] this are all the truthy values.

Primitive values are falsy values means which contain null, NaN, undefined are falsy.

e.g. var v1 = "" //blank string is a falsy  
var digit = 0 //falsy  
var any = null || undefined // falsy

Use of ! operator also for developers' safer side, it is known as a negation operator; it reverses the Boolean result of the operand.

e.g. var a = null // null  
var a = !null // true  
var a = !!null //false

depends on logical operator Truth table

**Logical Operator : -**

		&&	
0	0 =>	0	0
0	1 =>	0	1
1	0 =>	0	1
1	1 =>	1	1

It is as same as these 0 is true, ans 1 is false.

**2: Hoisting in javascript :**

In JavaScript, variables and functions are declare to the top of the scope before execution is called hoisting So we can use variable and function before declaring them.

e.g     a=100;  
           var b=20;  
           console.log(a/b);  
           var a;

**3: Local and Global Scope:**

The variable declared within a function is a local variable for that function and the variable is declared outside the function scope it is known as global variable.

E.g:  
 Var a=100; //global variable  
 Function check()  
 {         Var a=10; //local variable  
           Console.log(a);  
 }

**JavaScript Session 4:**

Topic:

- 1: Difference between ES5 and ES6 variables
- 2: Difference between let and const keyword which differes ES5 to ES6.

**1: Some points of ES5 and ES6**

- In ES5 Daclare with var keyword where ES6 Let and Const keyword used.
- In ES5 hoisting is possible where in ES6 hoisting is not possible
- In ES5 redeclaration and reassigning is possible but in ES6 with Let keyword we can reassigning multiple times but we cannot redeclare again and with respect to const keyword once we can declare any value with variable we cannot reassign or redeclare the value.

- In ES5 var has a functional scope while in ES6 let and const has lexical or block scope

```
e.g. functional scope
a=10;
function scope()
{
    a=20;
    console.log(a);
}
Var a;
```

- Lexical scope or block scope:

```
Let scope{
    Let scope
        {
            Let scope
                {
            }
        }
    }
}
```

## 2: Some points of Let and Const

- Let and const keyword use in ES6.
- Functionality of both let and const are almost same only the difference is In let we can assign multiple times but in const we cannot reassign.
- Both does not support redacларation and hoisting
- With const it is compulsory to assigning value while declaring variable but with respect to let it is not compulsory.

## JavaScript Session 5:

- We solve some problems related to pass by value and pass by reference variables.

### Const keyword:

We get some more knowledge about const keyword, we know that we cannot redeclare or reassign a const value but we can point to the object created by const variable it will get changed.

We can change and add value if we declare const value as a object.

```
e.g.    const value = {
        Name : "ABC";
```

```
};
To overwrite the value of name, we can overwrite like that
value.name="XYZ";
console.log(value); // XYZ
```

or we can also append some value in const variable like,

```
value.job="developer";
console.log(value); // Name: "XYZ" , Job: "developer".
```

## JavaScript Session 6:

1<sup>st</sup> half:

- Non – Technical Topic (Personality development)
  1. SMART Goal.
  2. Building Confidence
  3. Mindset, Toolset and Skillset

2<sup>nd</sup> half:

### Technical topic :

1. IP and Port
2. HTTP
3. REST API
4. License software's
5. Cloud providers
6. Web Servers

### 1. IP and Port and there ranges

- IP stands for internet protocol is an address is used to identify computer on internet.
- IP address is UNIQUE id which are uniquely defined to the all the computers.
- It is logical & software address and combination of network and host address.
- It helps in connecting your computer to other devices on your network.
- IP addressing are classified into two types
- IPv4 and IPv6 stand for version 4 and version 6
- Ipv4 is 32 bit addressing and ipv6 is 128 bit addressing.
- IPv4 is separated by .(dot) and IPv6 is separated by : (colon).
- IPv4 is representing decimal numbers while IPv6 seperating hexadecimal numbers ( 0 to 9 & A to F )



**Port** is a used Run the application it is endpoint of system to communicate with other system with help of port numbers.

0-1024 ports reserved and 1023-65535 port are available for user

**There are 5 classes of IP address:**

**Class A :**

**Actual range of class A**

255	0	0	0
N	H	H	H
0	0	0	0
127	255	255	255

Where the N is stand for network and H is stands for host

Total network in class A is 128.

Total host in class a is  $256*256*256=1,67,77,216$  host addresses are available in class A.

**Valid range of class A (Register Global Public Static):**

255	0	0	0
N	H	H	H
1	0	0	1
126	255	255	254

**Class B :**

**Actual range of class B**

255	255	0	0
N	N	H	H
128	0	0	0
191	255	255	255

Total network in class B is  $64*256=16384$

Total host network in Class B  $=256*256=65,536$

**Actual range of class B (Register Global Public Static)**

255	255	0	0
N	N	H	H
128	0	0	1
191	255	255	254

**Class C :**

**Actual range of class C**

255	255	255	0
N	N	N	H
192	0	0	0
223	255	255	255

There is total 3 network addressing in class C  $=32*256*256=2097152$ .

The host in class C single network  $=256$

**Valid range of class C (register Global public Static)**

255	255	255	0
N	N	N	H
192	0	0	1

223	255	255	254
-----	-----	-----	-----

**Class D :**

It is used for multicasting purpose and used by router.

224	0	0	0
239	255	255	255

224	0	0	1
239	255	255	254

**Class E**

It is used for research & scientific purpose.

240	0	0	0
255	255	255	255

240	0	0	1
255	255	255	254

**2. HTTP**

- HTTP is a hypertext transfer protocol which establishing connection between client and server.
- It is a protocol which helps to communicate a client with servers by sending datapackets means making request to the server and server will respond to the client.
- HTTP does the request response process between client and server.
- Unsecure (http = 80)
- Secure (https = 443)

**HTTP Verbs are:**

- GET - GET verb is used to getting information or data from the server requesting by the client.
- POST - POST verb is used to adding or inserting new data or record or submitting the essential data of client to the server.
- PUT - PUT verb is simply used to modify the record or credentials on the server by client.
- DELETE - DELETE verb is used to delete or remove the data from the server.
- Option
- Patch

**Status codes (1xx,2xx,3xx,4xx,5xx) :**

1xx = represented for informational

100 – continue

101 – switching protocol

2xx = represented for success

200 – ok      201 – created    202 – accepted

204 – no content

3xx = represented for redirection

304 – not modified

4xx = represented for Client errors

400 – bad request

401 – unauthorized

403 – forbidden

404 – not found

409 - conflict

5xx = represented for server error

500 – internal server error

502 – bad gateway

### 3. REST API

- API (Application program interface) endpoint is the code allows two software programs to communicate with each other.
- It is done by sending request to the server by the client and receiving response send by the server to the client
- It is an application program to request GET,PUT,POST,DELETE data.
- REST suggests to the server for creating an object and send data to the requested client as a response.
- And API is a application program interface which allows communication between client and server.
- REST API performing CRUD (CREATE,READ,UPDATE,DELETE) operation that is nothing but your GET,POST,PUT,DELETE

### 4. Licenses in Software Development

The following are the licences if software development:

- MIT
- Apache 2.0
- GPL
- Mozilla
- Creative commas
- BSD

### 5. Cloud Providers

- AWS
- Google Cloud
- Azure by Microsoft
- Bluemix by IBM
- Digital Otion
- Vultr. Etc...

## 5. Web Servers

- A web server is a computer that runs websites. It's a computer program that distributes web
  - pages as they are requisitioned.
- There is different types of webservers
- NGINX
- Apache
- Kong
- Light HTTP
- IIS
- HTTP Proxy