**2.1)Syntax**

* var,fun ,class & operator name are case-sensitive.
* Identifiers :- Name of class,var,function, fun parameter
* Include a-z|A-Z|\_|0-9|$ | ext ASCII | Unicode too.
* Semicolon:- To Make code read-able , can help to compress & combine JS code in production environment to remove
* extra space and save bandwith
* Expression :- Piece of code that evaluate to any value.

**2.2) Variables**

They are loosely types can hold any type of datatype.

Named placeholders of values.

Undefined:-Declared but Not initialise. vs Undeclared :- Not declared i.e. not exist

Scope :- All variables exist within a scope that determines the lifetime of the variables and which part of the code can access them.

TYPES OF SCOPE:-

JavaScript mainly has global and function scopes. ES6 introduced a new scope called block scope.

var declare inside fun , added the var to function scope.

var outside the fun added that var in global scope

Scope shadowing :-

If a var is declare and defiend inside a fun which was also decalre outside of fun with same name ,

then global var will shadowed cannot be accessed inside fun till local var

While executing a script engine goes through two phases:-

1.)Parsing/Hoisting:- Move global var to top of file, or top of fun if variables are declared in fun.

2.)Executing

The JavaScript engine moves only the declaration of the variables to the top. However, it keeps the initial assignment of the variable remains intact.

In case of let const , they are not assigned,but var with undefined.

**2.3) DataType**

* Six Type of DataType:-
  + 1.)undefined
  + 2.)null:- empty object pointer
  + 3.)string
  + 4.)number
  + 5.)boolean
  + 6.)symbol
* JS is dynamic language,means var is not asociated with any type of data type , but its values does.
* NaN != NaN
* String:- Sequence of zero or more characters.
  + Begins with and ends with single quote or double quote.

Strings are immutable unlike objects

**2.4)Number**

var temp=new Number(15)

console.log(temp.valueOf());

typeof temp===Object

**3.1)Unary Operators**

Uniary are simplest operator, work on single operand.

+, -, ++, --

let a = 10;

a = +a; // 10

a = -a; // -10

let s = '10';

console.log(+s); // 10

let f = false,

t = true;

console.log(+f); // 0

console.log(+t); // 1

**3.2)Logical Operators**

Three type of Logical operators

NOT (!) 2.)OR ( || ) 3.) AND ( && )

!’’ == true

!NaN == false

Double Negation (!!)

Use twice Negate operator to convert a value to its real boolean value.

**3.3) Assignment Operator**

a=a+b == a+=b

a=a-b==a-=b

a=a\*b === a\*=b

a=a | b === a |=b

a=a & b === a &=b

Strict equal

**7.1)Function:-Intro**

Arguments:- Object can be accesesed inside fun ,which behave like array though not instance of array.

Function Hoisting :- Allow to call function before function declaration.

Moving fun dec at the top of code.

**7.2)Function:– First–class**

Means function be stored as value , an be passed as argument in fun and can be return as function from another function. In other word , can be treated as value.

**7.3)Function:– Anonymous Function**

Function without name .

No name between function keyword and open ‘(‘

let show = function () {

console.log('Anonymous function'); }; show();

**7.4)Function – IIFE(Immediately Invoked Fun Expression)**

A Function defined as expression and executed immediately.

When a var and func are declared in global , both will be added to global object means pollute the global object.

One way to prevent this pollution of global onject is to use IIFE.

IIFE with semicolon (;):- When concatenating js code of two file having two IIFE , into single file may be by using code bundler,without semi-colon it may render syntax error.

const calculator = (function () {

function add(a, b) {

return a + b;

}

function multiply(a, b) {

return a \* b;

}

return {

add: add,

multiply: multiply

}

})();

**7.5)Callback**

The function passed as argument into other function.to call later.