**Printing in javascript**

**console.log("")** = it is command used to printo display on console.it will print new line also like java's println

**defining variable**

**let** a=10 (a is variable name variable naame can be anything)

**let** is used to define any variable

Syntax: **let** *variablename*=*value*

we dont have to specify varibale type in javascript means varibales in javascript are dynamic in nature,we can store any type of valuelike string value,integer etc

**Loop**

We can define loops just same as java

For loop: for(i=1;i<5;i++){

Console.log(i\*5);

}

While loop: let a=1;

While(a<=5){

Console.log(a\*5)

}

**Data types in javascript**

In Javascipt variable can include any value like number,string,boollean etc.

**Number**:it includes integer,float,double etc.

In javascript number can have integer float double etc.

**Boolean**:it contains either true or false

**String**:collection of characters or words is string

**Undefined:**undefined value exist but does not have value

**Symbol**

**Null**:null means it does not exist and does not defined(nothing)

**Functions in javascript**

Functions are dynamic type in nature in javascript means we can return any type of value in function

We can define function by using **function** keyword

**Function mul(**x**){**

**return** x\*x;

**}**

If don’t define return value or return then undefined will be displayed

Ex: **Function mul(**x**){**

**return**;

**}**

**Function mul(**x**){**

**}**

**Default value for function**

We can default value in function if we are not giving value in function by

**Ex:Function mul(**x=10**){**

**Return x\*x**;

**}**

**Console.log(mul()) //**function will take default value which is define at mul() funvtion argument

**Argument mapping for function**

**argument mapping** means giving argument for function is starts from left to right

ex:function(x,t){}

console.log(2,3);

//x will have 2 and y will have 3 value from left to right

**Default argument mapping** means giving default argument for function is starts from right to left

Ex:function mul(x,y=10){

Return x\*y;

}

Console.log(mul(2))// this 2 will go to x

we can not define function like

function mul(x=10,y){

    return x\*2;

}

console.log(mul(3))

as above example this will give undefined as default values for function can be put from right to left andit doestnt define rifght argument value

**Funtion with no name**

**Function expression:** Function with no name are called function expression.

Ex:function(){

Console.log(“hi”)

}

We canstore function expression inside an variable

**Ex: let a=function(){**

**Return 5;**

**}**

//function value is store in a variable

**IIFI(Immediate Invoke function Expression):**

IIFI are Function with no name but its is diffent than function expression

**Arrays in javascript**

Arrays are dynamic in nature in javacript

**Syntax: Let a=[]**

**Ex: let a=[]**

**a[0]=1**

**a[1]=3**

**a[2]=”hello”**

**a[6]=[‘a’,’b’]**

**console.log(a)**

**output**



* We can store any number of values in array we don’t have to define array length
* We can add element at any index even if it does’nt exist

Ex:let a=[]

**a[2]=”hello”**

**a[6]=[‘a’,’b’]**

**Arrays methods in javascript**

**Slice():**The slice() is method of array which returns the selected elements in an array, as a new array object.

It selects the elements starting at the given start argument, and ends at, but does not include, the given end argument.

Ex:**let a=[‘a’,’b’,’c’,’d’]**

**String in javascript**

A JavaScript string is zero or more characters written inside quotes.

Ex:var x = "John Doe";

Template literals/String literals

They are string literals allowing embedded expressions.

String templates is written inside the backtick ` `

Ex:var x =`hello world`

**Template string/literals features**

**1.Multi—line String:** Any newline characters inserted in the source are part of the template literal

Ex:var x =`hello world

Hiieeeknck `;

**2.[Expression interpolation](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Template_literals" \l "expression_interpolation" \o "Permalink to Expression interpolation):** we can add or embed variables(expression) within normal strings in template by using $( )

Ex: var x =20;

var s =`number is $(x) `;

output: number is 20

Any newline characters inserted in the source are part of the template literal.

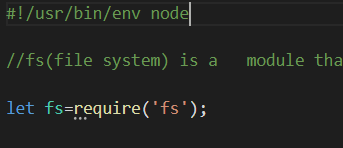
**Steps to create js comman global**

**1)create npm package: we can create npm package using**

**“npm init”.**

**2)put shalang file inside js file:**

**#!/usr/bin/env node inise js file**



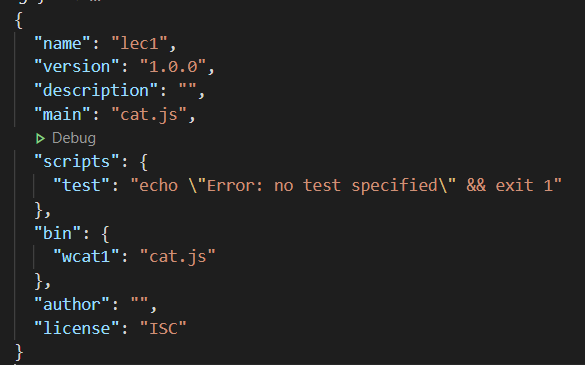
**3)add a "bin" to your package.json file**

add bin to your json package and write the command name you want to give

**syntax:”bin”:{**

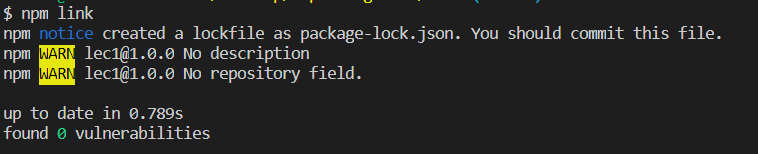
**“commandName”:”your file directory”**

**}**



**4)Write npm link command to make your file global**

**it will make your command global**



**Objects**

Objects in javascript are dynamic in natures

An object can have number,string,function,even object

We can set create object key/proeprties outside object even afte object is created.

**Object creation**

An object can be created with figure brackets {…} with an optional list of *properties*.

A property is a “key: value” pair, where key is a string (also called a “property name”), and value can be anything.

**Syntax: Let variableName={proeprty1,property2,… }**

let user = new Object(); // "object constructor" syntax

let user = {}; // "object literal" syntax

let user = { // an object

name: "John", // by key "name" store value "John"

age: 30 // by key "age" store value 30

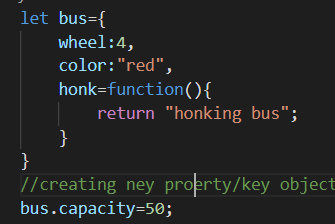
};

User,name,age are keys/properties of object

**Setting property values**

We can set property value of object by

**.Outside object:**we can create and set key/property outside of object

ex: 

we can update object key/property valaue

**accessing object values**

pwe can access object value in three was

**1.accessing object using square bracket**

We can access object value using square bracket bby typing its attribute inside []

**Ex:**bus[“capacity”]

Capacity is key name of object

It search that attribute or try to resolve if there is no attribute

**2.accessing object using square bracket with variable**

We can access object using square bracket by typing variable name inside[]

It will search varibale name and put its value inside []

E**x:let index**=”capacity”;

**bus[**index**];**

index is variable which has capacity value and this value is put inside[];

**3.accessing object using dot(.) operator**

We can access object value using **.**

EX: **bus.capacity=10;**

**Delete property in object**

The delete keyword deletes a property from an object:

Ex:

var person = {firstName:"John", lastName:"Doe", age:50};

delete person.age;   // or delete person["age"];

The delete keyword deletes both the value of the property and the property itself.

fter deletion, the property cannot be used before it is added back again.

The delete operator is designed to be used on object properties. It has no effect on variables or functions.

The delete operator should not be used on predefined JavaScript object properties. It can crash your application.

**Git**

Git is a version control

Git is repository where program modules is saved

If we want to save program in in different module like version1,ver2… we save in different folders but it is bad practices so git solve this problem ,it will save program in differ modules which will save in single repository

Repositary basically where git save our code ,repositort

Git wil can track file which Is tracked in staging area of git

**Git Init:** iit will create git repository

**Git init**

**Git add**:it will add all files inside folder into git rrepositary

**git add <file> =to add a single file:**

**git add . =To add everything at once we use**

**Git commit :**git commit save file in repository from staging area

**Git commit –m “id name”**

**Ex:git commit –m “first”**

