**Javascript basis**

Java script က body ထဲမွာ scripte ဖြင့္ပိတ္ထဲမွာ ေရးတယ္

java script Es6 ဆိုတာ version

var က javascript Es5 version မွာ သုံးတာ

**၁။** var name=”kyaw”; (var က keyword/ name က variable )

name; (kyaw ထြက္လာမယ္)

ခုေတာ့ Es6 version ဖစ္တဲ့ let ပဲသုံးမယ္။

**1.** let age=25; (**let** က keyword/ **age** က variable )

age; -> 25

**2.** const mynrc=12/thagaka(N)198781; (**const** keyword က မေျပာင္းလဲတဲ့ တန္ဖိုးေတြ မွာသုံးတယ္/ )

mynrc; -> 12/thagaka(N)198781

const ကို override လုပ္လို႔မရ var တို႔ let တို႔ကို ပဲလုပ္လို႔ရတယ္။

**Data Type 7 မ်ိဳးမွာ (၃)မ်ိဳးပဲသုံူမယ္ [string , number , object]**

String

let firstname=”than”;

let lastname=”shwe”;

let fullname=firstname + “ ”+ lastname (“” ဒါနဲ႕ေရးထားတာက space လိုခ်င္တယ္လို႔ေျပာတာ) ->than shwe

Template literals နဲ႕ 5’ 11’’ ေတြပါ ေသခ်ာေရလို႔ရတယ္။

let height= ` 5’ 11’’ ` ; (`bactip` ထဲမွာ ေရးရင္ siglecode ေရာ dobulbe code ေရာတြဲသုံးလို႔ရတယ္)

**‘** my height is **${ height } `**; ($ {} နဲ႕ေရးတာကို templeate string လို႔ေခၚတယ္)

**Array နဲ႕ object ေတြက key value နဲ႕အလုပ္လုပ္တယ္လို႔မွတ္ထား key ကိုေခၚ value ေတြထြက္လာတာ**

**Array (array သည္ value ေတြအမ်ားႀကီးသိမ္းတာ)**

let box=[“apple”,false,15]; (false က bollean တန္ဖိုး, apple က string, 15 က no. အဲ့တန္ဖိုးေတြက valueေတြဖ္စတယ္)

box[2] -> [2]ခန္းေျမာက္မာရွိမဲ့ 15 ထြက္လာမယ္။ ၂ခႏ္ုးေျမာက္မွာရွိတဲ့key လို႔မွတ္ထားပါ။

override လုပ္နည္း

box[2]=20 -> အခန္း (၂)ထဲက 15 ကို 20 လို႔ေျပာင္းလိုက္တာ။ box[2] က 20 ဖစ္သြားမယ္ ။

let arr=[];

arr[0]=”than”;

arr[1]=”shwe”;

arr လို႔ေခၚလိုက္ than shwe ထြက္လာမယ္

**Object (object က key ေတြ value ေတြ အစုံလိုက္သိမ္းတာ)**

let fruit ={ ( fruit က object)

name=”orange”, (name က key, organe က value )

price=200,

unit=”kb”

};

fruit[“price”] ->(200)ထြက္လာမယ္

furit.price အလိုလဲထုတ္လို႔ရတယ္ ->(200)ထြက္လာမယ္

fruit.test=this is test -> test ဆိုတဲ့ key ကို **this is test** value လို႔ေပးပီး အခန္းထပ္တိုးတာ

override လုပ္ခ်င္ရင္

fruit.test=this is override test အလိုေရးေပးရတယ္

Key နဲ႕ value နဲ့တူရင္ တခုေပးေရးလို႔ရတယ္ eg. fruit=”fruit” ဆို fruit လို႔ တခုပဲေရးလို႔ရတယ္

**typeof** (string လား no ပါလား စစ္တာ)

typeof fruit ဆို object ထြက္လာမယ္

**Operator (w3 school ထဲမွာ ၾကည့္)**

5 \*\* 6 (5 ေျခာက္ထပ္ကိုေျပာတာ expontential)

87%32 (moudules အကြ်င္ကို တြက္ခိုင္းတာ)

a=1

a+=2 -> 3 ထြက္မယ္ (1 နဲ႕ 2 နဲ႕ ေပါင္းပီးထြက္တာ)

**string operator (+)**

let fname=than

let second=shwe

fnam+second

**Comparison Operators နဲ႕ logical Operatore (Boolean တန္ဖိုးေတြပဲထြက္တာ )**

**Ternary operator**

**4>3 ? “” : “”** (4 သည္ 3 ထက္ႀကီးတာမွန္ရင္ ေရွ႕မွာေရးထားတဲ့ double code “” အထဲက အလုပ္လုပ္တယ္။ မွားရင္ ေနာက္က ကုဒ္ထဲကအလုပ္ုလပ္တယ္။ ျပဳံးတဲ့ပုံ မဲ့တဲ့ပုံကို ထည့္လို႔ရတယ္)

**Funciton**

1. function ဆိုတာ က block of code (code ေတြကို စုပီး blog လုပ္ထားတာ) function name ကိုလွမ္းေခၚရင္ အလုပ္လုပ္တယ္။

task ဆိုတဲ့ လုပ္ငန္းေဆာင္တာေတြကိုလည္း အလုပ္ုလပ္ေပးတယ.္။

function ရဲ႕ code ေတြကို{} ထဲမာေရးတယ္။

   let currentage=23;

            function birthyear(){ -> function name

                let result=2021- currentage; -> let က keyword , result က variable,

                let end= `သင့်အသက်က ${result} မှာ မွေးဖွားတာ`;

                return end; -> function name ကိုေခၚ၇င္ return ရဲ႕ end ဆိုတဲ့ var ကို ထုတ္ေပးပါ

            }

**Parameter in Function** (function parathesis ထဲမွာ parameter ေတြ ၁ခုထက္ပိုပီးေ၇းလို႔ရတယ္၊ တကယ္ဆို variable ေတြေရးတာလို႔လဲမွတ္ထား)

    function birthyear(currentage=20){ -> cur=20 လို႔ေပးထာတာက default ေပးထားတာ console မွာေခၚရင္ birthyear ()

အလိုေခၚတာနဲ႕ 20 နဲ႕တြက္ေပးလိမ့္မယ္ currentage ပဲေပးထားမယ္ဆိုရင္္ console မွာ

ေခၚရင္ birthyear(ၾကဳိက္တဲ့ ႏွစ္ကိုထည့္လို႔ရတယ္) ေမြးဖြားတဲံေန႔ တြကေပး လိမ့္မယ္

                let result=2021- currentage;

                let end= `သင့်အသက်က ${result} မှာ မွေးဖွားတာ`;

                return end;

            }

parameter ၂ ခုေရးနည္း

function birthyear(name,currentage){ ->console မွာ birthyear(“kyaw”,20) kyaw နဲ႕ 20 ကိုႀကိဳက္တာေရးထားလို႔

argument လို႔ေခၚတယ္ enter ရင္ အေျဖထြက္လာမယ္

default လဲေပးလို႔ရတယ္ (name=”kyaw”,currentage=20) default ကို

console မွာေခၚရင္ birthyear() အလိုပဲေခၚ ႐ုံပဲ။

                let result=2021- currentage;

                let end= `${name}အသက်က ${result} မှာ မွေးဖွားတာ`;

                return end;

            }

console.log

let a=15;

            console.log(a); ->browser ထဲက console မွာ a လို႔ေရးစရာမလိုပဲ တႏ္ူးျပမယ္ 15 လို႔

function expression

let test= function(){

                return "this is function expression test";

            }

console မွာ test() လို႔ေခၚေပးရတယ္။

IIFE (Immediately Invoked Function Expression)

(function(){

                console.log("thisis IIFE");

            })()

**Scope (3)မိ်ဳး**

scope ေတြကို console.log နဲ႕ လွမ္းေခၚတယ္ ။function ထဲမွာပဲ လွမ္းေခၚလို႔ရတယ္။ return အစား console.log ကိုသုံးသလိုမ်ိဳးမွတ္ထား။

Block Scope

{

let a = this a

var b= this b

}

console.log(b); var နဲ႕ ေျပာထားတဲ့ ေကာင္ပဲထုတ္လို႔ရတယ္ Carli bracket ထဲကဟာကို console.log(a)ဆို error

တက္ယမ

**local scope**

function ထဲက ဟာကို console.log နဲ႕ ေခၚမယ္ဆိုရင္ carli bracket အတွငးက ပဲ ေခၚရတယ္။

            function test(){

                let a="this a";

                let b="this b";

                console.log(a);

                console.log(b);

            }

**global scope fun ရဲ႕ အျပင္မာ ေ၇းထားတဲ့ variable ကို function ထဲကေန console.log နဲ႕ေခၚတာ**

 let a="gloabl scope";

            function test(){

                console.log(a);

            }

ကိုတြက္ထုတ္လိုက္တဲ့ တန္ဖိုးေတြကို array[] နဲ႕သိမ္းခ်င္ရင္

let record=[]; -> ဒီ record ကို ေခၚရင္ resultရဲ႕ ၇လဒ္ထည့္ခ်င္လြတ္ အခန္းလြတ္ေဆာက္ထားတာ။

          let index=0; -> index ကို ++ လုပ္လို႔ရေအာင္ေပးထားတာ

           function calculate(width,height)

           {

                let result=width\*height;

                record[index]= result; -> multiple ရဲ႕ ရလဒ္ေတြ ကို record ထဲက 0 အခန္းထဲကို သိမ္းမာ တခုပဲသိမ္းလို႔ရမယ္။

index တန္ဖိုးကက zero တခန္းပဲဖစ္ေနေတာ့ ။

               index ++; -> တခန္းပီးတခန္းသိမ္းခ်င္လို႔ index တန္ဖိုးကို ++ေပးတာ

                return result;

           }

အေပၚက record ေနရာမာ ဒီပုံစံနဲံ ေရးလို႔ရတယ္ width height result ေတြကို array တခန္းခ်င္းစီမွာသိမ္းခ်င္ရင္

။ widths ဆိုကိုဘာသာထားတဲ့နာမည္ ၊ width က ကိုေရးတဲ့ width ဥပမ 20 height 30

console.table (record ) လို႔ေခၚ၇င္ arry ကို table နဲ႕ျပေပးတယ္

 record[index]={

                    widths:width,

                    heights:height,

                    res: result

                };

**USD, EUR SGD ကေန MMk ကိုေျပာင္းနည္း**

let rates={

                    usd:1500,

                    eur:2000,

                    sgd:1100

                }

                function exchangerate(amount,currency){

                    //console.log(amount \* rates[currency]); -> console နဲ႕လဲထုတ္လု႔ိရတယ္။

                    let calc=amount \* rates[currency]; -> rates ထဲ က curryc ပဲဖစ္ရမယ္လို႔ေရးထားတာ

                    let result=calc + "MMK";

                    return result; }

**MMK မွ USD, EUR SGD ကိုေျပာင္းနည္း**

let rates={

                    usd:1500,

                    eur:2000,

                    sgd:1100

                }

                function exchangerate(amount,currency){

                    //console.log(amount \* rates[currency]); -> console နဲ႕လဲထုတ္လု႔ိရတယ္။

                    let calc=amount / rates[currency]; -> rates ထဲ က curryc ပဲဖစ္ရမယ္လို႔ေရးထားတာ

                   let result=calc + "" + currency;

                    return result;

**အသီး၀ယ္ အခြန္တြက္ စုစုေပါင္းထုတ္**

let fruits={

                    orange:1500,

                    apple:2000,

                    mango:1100

                }

                function caltax(amount,parcentage=5){

                    return (amount\*parcentage)/100;

                }

                function calcost(qty,fruit){

                    let calc=qty \* fruits[fruit];

                    let result=calc + "" + "mmk" ;

                    return {

                        name : fruit,

                        unitprice : fruits[fruit],

                        quantity: qty,

                        cost: result,

                        tax : caltax(result),

                        total: result + caltax(result) }; }

**Method ဆိုတာ obj ထဲမွာ key နဲ႕ function နဲ႕ တြဲေဆာက္တာကို ေခၚတယ္။**

object ထဲမွာ method ေတြ ေဆာက္လို႔ရတယ္။

let myself = {

                    name : "TS",

                    age : 24,

                    skill : ["html","php","css"],

                    //method

                    test : function(){

                            return "this is test method";

                    },

                    //method ES6

                    run(para){ -> function ေဖ်ာက္ပီးေရးထားရင္ ES6 style နဲ႕ေရးထားတာ

                        return "this is method es6"+" "+para;

                    },

                    showall(){

                        //console.log(this); -> this ဆိုတာ myself ဆိုတဲ့ object ကိုေျပာတာ

                   return `my name is ${this.name} and age is ${this.age} year old and ${this.test()}`; -> obj ထဲက တန္ဖိုးေတြ ျပန္ေခၚသုံးခ်င္လို႔ backtip ထဲေရးတယ္ ${this.name} ဆိုတာ myself objet ထဲက name ကိုလွမ္းေခၚတာ

                    }

                }

**Popular Array Method**

 let fruits=["apple","banana","orange","mango"];

                fruits.isArray(fruits);

1. isArray()

Array.isArray(**fruits**) - Array ဖစ္ရဲ ့လားလို႔ စစ္တာ true လို႔ထြက္မယ္။

2. includes()

fruits.includes(“apple”) - fruits ထဲမာ apple ပါာလးစစ္တာ။

3. indexOf()

fruits.indexOf(“Apple”) - apple ပါတဲ့အခန္းနံပါတ္ကို လွမး္ယူတာ။

4. splice() ဖ်က္္တာ

fruits.splice(2,3) - 2 ခန္းေျမာက္ကေနစဖ်က္မယ္ 3 ခုထိ ျဖက္မယ္။

fruits.splice(2,3, “**fruitone**”, “**fruittwo**”); ဖ်က္သြားမယ့္ေနရာမာ အစားထည့္မယ္။

5. concat (array ရဲ႕ အခန္းေတြ တခုနဲ႕တခုေပင္းတာ)

fruits.concat(marks) - fruits နဲ႕mark ရဲ႕ aray အခန္းေတြေပါငး္တာ ေပါင္းပိးသားကို save ထားမွရမယ္ နဂို fruit နဲ႕ mask array ထဲမာ သြာေပါင္းမာမဟုတ္ဘူး ။

**Array Iteration**

loop တဲ့အခါမွာ ii နဲ႕ထည့္သု႔းတာကိုေျပာတာ

 const fruits=["apple","banana","orange","mango"];

                // for(i=0;i<fruits.length;i++){

                //     console.log(i,fruits[i]);

                // }

                // for(fruit of fruits){

                //     console.log(fruit);

                // }

                // fruits.forEach(function(i){

                //     console.log(i);

                // })

                let ul =document.createElement("ul");

                fruits.forEach(function(i){

                    console.log(i);

                    let li= document.createElement("li");

                    let text=document.createTextNode("i");

                    li.append(text);

                    ul.append(li);

                });

                document.body.prepen(ul);