

javascript

```
طريقة الربط بينهم والافضل تتكتب فى نهايه جمله البودى
<script src="main.js"></script>

ولو حصل العكس ساعتها نعمل فى صفحه الجافه

window.onload=function(){document.querySelector("h1").style.color="powderblue";
};

//this is comment
/* comment*/ or ctrl + /

console.log("read this")
//consolo تعرض فى صفحه ال
console.error("Error");
//عمل رساله خطأ فى صفحه الكونسول
console.table(["hossam","hassan","mohamed"]);
//عمل جدول فى صفحه الكونسول

console.log("hello in %cthis page ","color:red;font-size:20px;");
//%c تغير التنسيق بتاع كلمه معينه بنحط %c

var myname="hossam";
console.log("welcome"+ myname);
//ندى قيمه لمتغير ونطبعه

console.log(typeof("hossam hassan"));
console.log(typeof 1500);
console.log(typeof [15, 16, 17]);
console.log(typeof{name:"hossam",age:20,country:"egy"});
console.log(typeof true);
console.log(typeof undefined);
console.log(typeof null);
//معرفة نوع البيانات

var user="hossam";
console.log(user);
//عمل متغير باسم معين

<div id="hello"> Hello </div>
console.log(hello);
hello.innerHTML="wellcome";
//عمل اى دى فى الاتش تى ام ال واستعدائه هنا وتغير قيمته

var thi="red";
var thi="blue";
//تتاخذ اخر قيمه عملتها ومشي بينهك للخطا
//بتنحط فى الوندز اويجكت وممكن نبحث عنها وممكن نعمل مشكله بعدين

let a="os";
//مش بتغير لو فى قيمتين بنفس الفاريو وتبينك بالخطا

const b= "or";
//مش بتغير لو فى قيمتين بنفس الفار وتبينك بالخطا

console.log("welcome \\\to \\'first\' ");
console.log("welcome \\\to \\'first\' ');
```

```

console.log("welcome \
to \
first ");
//or
console.log("welcome \nto \nfirst ");
// ده علامه تجاهل عشان لو حاب احط اى علامه غريبه \

let a= "love";
let b= "you";
let c= "jav";
document.write( a+" "+ b);
document.write(` ${a}\n ${b} \n ${c} `);
// لو عايز اعمل مسافه بينهم علامه بتعمل بى ال د

let title="TEMPLATE ";
let markup=`<div id="hello"> Hello </div>
<div class="card">
  <div class="child">
    <h2>${title}</h2>
    <P> SOME INFORMATION </P>
  </div>
</div>`;
document.write(markup);
// عمل تامليلت جاهز داخل الجافا نفسها

.....الارقام.....
document.write(10 ** 2); // علامه الاس
console.log(+ "100"); // استرجاع القيمه
console.log(+a + b );// جمع عددين على بعض

console.log(100000);
console.log(1_000_000);
console.log(1e0);
console.log(1**6);
// علامه ال_ و غيرها بطلع الرقم زي ماهو عادى

console.log((100).toString()); // يتحول الرقم لى استرنج
console.log(100.555555.toFixed(2)); // يتاخذ رقمين بعد علامه
console.log(parseInt("100")); // بترجعها لى رقم
console.log(parseFloat("100.55"));
console.log(Number.isInteger("100")); //false
console.log(Number.isInteger(100)); //true

console.log(Math.round(99.5)); //تعمل تقريب للعدد
console.log(Math.ceil(99.1)); // يتقرب الرقم لى اعلى حاجه يعنى ميه
console.log(Math.floor(99.9)); // يتقرب لى اقل حاجه يعنى تسعه وتسعين
console.log(Math.min(10,20,30,40,50)); // اكبر قيمه
console.log(Math.max(10,20,30,40,50)); // اقل قيمه
console.log(Math.sqrt(2,6)); // الاسم
console.log(Math.random(2,5,7,8)); // رقم عشوائى
console.log(Math.trunc(99.6)); // يتجاهل الرقم العشري

let theName = " ahmed";
console.log(theName[4]); // =e
console.log(theName.charAt[3]); // =h
console.log(theName.length); // =6
console.log(theName.trim()); // يحذف المسافات
console.log(theName.toLocaleUpperCase()); // يحول لى حروف كبيره
console.log(theName.toLocaleLowerCase()); // حروف صغيره
console.log(theName.trim().charAt(2).toLocaleUpperCase()); // جعل الحرف الثالث كبير بعد حذف المسافات

.....
let a= "hossam abu hasen";

```

```

console.log(a.indexOf("abu"));
البحث عن العنصر كذا وهبطلع 8=
console.log(a.indexOf("abu",9))
هبطلع 1- عشان هبيدء العد من بعد الرقم
console.log(a.lastIndexOf("hossam"));
معناها هبيدء عد من اخر عنصر
console.log(a.slice(7, 10))
معناها هبيدء من العنصر كذا لحد كذا
console.log(a.slice(-6))
هياخد اخر ست حروف
console.log(a.repeat(4));
معناها تكرار العنصر اربع مرات
console.log(a.split());
معناها قص كل الكلمه
console.log(a.split(""))
معناها قص كل حرف لوحده
console.log(a.split(" ",2))
معناها قص كل جمله لوحده والرقم الاخير معاناها قصها مرتين
console.log(a.length)
console.log(a.substring(1,7))
معناها يقص من العنصر واحد لحد سبعة
console.log(a.substring(-10))
هيقص كل العناصر اللي تحت العنصر العاشر
console.log(a.substring(a.length-1))
هيقص اخر رقم
console.log(a.includes("ali",4))
هل بعد العنصر كذا في كلمه على
console.log(a.startsWith("h",0))
هل الكلمه بتبدء بحرف كذا
console.log(a.endsWith("o"))
هل تنتهى الكلمه بحرف كذا
-----
console.log(10=="10");
هل بتساوى نعم=
console.log(10!="10");
لا تساوى= لا تساوى
console.log(10===10);
بتقارن القيمه والنوع =لانتساوى عشان العشره نوعها استرنج
console.log(10!==10);
نعم=
console.log(!(10=="10"));
بتنفى الكلام not
console.log(10=="10" && 10>8 );
لازم الشرطين يتحققو and
console.log(10=="10" || 10>12 );
شرط واحد يتحقق عشان يدى صح
-----
let price=100;
let disc=false;
let discA=30;
let contry="egypt";
let gender="male"
let student=true
let theName= "Hossam"
if(disc===true){
    price-=discA //price = price - discA
}
else if(contry==="egypt") //لو متحققش الشرط الاول جرب ده
{
    price-=discA+10
if(student===true) //شرط جوه شرط
{
    price-=discA+20
}
}

```

```

else{
    price-=discA+10
}
}
else // لو متحققش اي حاجه نفذ ده
{
    price -=10
}
console.log(price);

gender==="male" ? console.log("mr"):console.log("mrs")
//==if اختصاره

let result =gender==="male" ?("mr"):( "mrs") ;
console.log(result);
console.log( `Hellow ${result} ${theName}` ) ; // بتجمع التلت عناصر
-----
let pric=0;
console.log(`the price is ${pric|| 200}`)
// لو القيمة كانت بتساوى صفر منش بيطلعها وبخيش على البديل
console.log(`the price is ${pric ?? 200}`)
// لو فيه قيمة فولس بيرجعها زي ماهي

-----
let day=1;
switch(day){
    case 0:
        console.log("starday");
        break;
    case 1:
        case2: // لو الامرين بيساوى نفس الحاله نكتبهم تحت بعض
            console.log("monday");
            break;
        default:
            console.log("unknown day ");
            break;
}

-----

//array

let myFriends=["Hossam","Ahmed","Mohamed",["Hasen","Ali"]];
console.log(`hello ${myFriends[0]}`) // بتاخذ اول اسم
console.log(`scond letter : ${myFriends[1][0]}`);
// بتاخذ اول حرف من ثاني
console.log(`hello ${myFriends[3][0]}`);
// بتروح عند الاري الزيادة وتاخذ اول اسم
console.log(`hello ${myFriends[3][1][0]}`);
// بتروح عند الاري الزيادة وتاخذ اول حرف من اول اسم
console.log(myFriends);
myFriends[1]="gehad"; // بيعدل على الاسم الثاني
console.log(myFriends);
myFriends[3]=["jehad","esra"]; // تغير كل الاري الاخير
console.log(Array.isArray(myFriends));
// هل ده تعتبر اري
console.log(myFriends.length); // معرف عدد عناصر الاري
myFriends[6]="unknown";
// لو حطيت تغير اسم عنصره منش موجود فايطلعهم زي ماهم ويخلي اللي فى النص فاضى زي ماهو
console.log(myFriends)
myFriends[length-1]="new name"; // بيغير اسم اخر عنصر
console.log(myFriends)
myFriends.length=2 // بيخلي عدد الاري يبقى اقل

```

```

let sot=[10,"hossam","90",1000,100,20,"10",-20,10];
console.log(sot);
console.log(sot.sort());
// بترتب السالب اولاً ثم الرقم الاسترجاعي ثم الرقم الارقام العادية ثم الاسم
console.log(sot.reverse()); // بتعكس الترتيب

let myFriends=["Hossam","Ahmed","Mohamed","Hasen","Ali"];
myFriends.unshift("gehad","hazem") // تصيف عناصر بدايه الاري
console.log(myFriends.slice());
console.log(myFriends.slice(1,3)); // اطبع من الاندكس كذا لحد كذا

myFriends.splice(1,0,"Eman");
// امر جديد يحدد بدايه العنصر ومن ثم عدد عناصر الحذف ومن ثم اضافته الاسم الجديد

let myFriends=["Hossam","Ahmed","gehad","Hasen"];
let oldFriends=["Eman","hazem"];
let schoolFriends=["baha","maged"]

let Allfriends= myFriends.concat(oldFriends,schoolFriends);
// دمج اكثر من اري

console.log(Allfriends);
console.log(Allfriends.join(" # ").toLocaleUpperCase());
// دمج مع خط علامه بين كل عنصر وجعلهم حروف كابتل
-----

//For تكرار اللوب
let myFriends=[1,2,"Hossam","Ahmed","gehad","hazem","Eman"];
let onlyFriends=[];

for(let i=0 ; i < myFriends.length; i++) // نستخدم i < 5 نستخدم الثابت بتاع الاري
{
    if(typeof myFriends[i] === 'string') // البحث في العناصر وهات الاسترجاعي فقط
    {
        onlyFriends.push(myFriends[i]);
    }
    // console.log(myFriends[i]); // تكرار اللوب ويطبعهم
}
console.log(onlyFriends)

-----
let product =["keybowrd","mouse","montior"];
let color =["red","green","black"];
let models =[2020, 2021];
for (i=0 ; i<product.length;i++){
    console.log("#".repeat(15));
    console.log(" the product: ");
    console.log(`# ${product[i]}`);
    console.log("#".repeat(15));
    console.log(" the color: ");
    for(j=0 ; j<color.length;j++){
        {
            console.log(`- ${color[j]}`);
        }
    }
    console.log(" the models: ");
    for(k=0 ; k<models.length;k++){
        {
            console.log(`- ${models[k]}`);
        }
    }
}
}
-----
// تطبيق على ال For عرض خمس منتجات وكل منتج تحته الالوان بتعته

```

```

let product = ["keybowrd", "mouse", "montior", "pen", "iphone"];
let color = ["red", "green", "black"];
let showCont=5;
document.write(`<h1> show ${showCont} Product </h1>`);
for (i=0 ; i<showCont;i++){
    document.write(`<div>`);
    document.write(`<h3>[${i+1}] ${product[i]} </h3>`);
    for(let j=0 ;j<color.length;j++ )
    {
        document.write(`<p> ${color[j]} </p>`)
    }
    document.write(`</div>`)
}
-----
//while استخدام
let index=0;
while(index<product.length) // طول ما الشرط نرو هيسمر بطابعه
{
    console.log(product[index]);
    index++; // ضروري عشان الجهاز ميفصلش
    if(index===5){
        break;
    }
}
-----
// do while استخدام
let g=1;
do{ // بتنفذ الشرط سواء ترو او فولس
    console.log(g)
    i++;
}
while(false)
{
    console.log(g);
}

```

```

-----fuction-----
function sayHello(user,age) // اسم الفانكشن اللي هنعملها بادلها المتغير
{
    age = age || "unknown"; // فى حاله المستخدم محطش العمر
    if(age<20)
    {
        console.log(`hello ${user} this app not sutiable of you `)
    }
    else{
        console.log(`Hello ${user} your age :${age} `) // الامر اللي هيتنذ
    }
}
sayHello("Hossam",21); // نستدعى الفانكشن ومش بتكرر
sayHello("ahmed");
sayHello("jehad",19);
-----
function gYears(start,end,del) // السنه اللي مش عايزنها تنفذ
{
    for(let i=start ; i<=end; i++)
    {
        if(i===del)
        {
            continue;
        }
    }
}

```

```

    }
    console.log(`the range of year ${[i]}`);
  }
}
gYears(2018,2022 ,2020);

-----
// تطبيق يظهر الاسم والسن والساعات وفي حاله ضغط yes يظهرله يكتب عدد لانهاى من الاسكل ولو نو يقوله تم الاخفاء الاسكل
بندى قيمه افتراضيه فى الحاله المستخدم مكتيش //
function showInfo(us="unknown",ag="unknown",rt=0 ,show="yes",...skills)
{
  document.write(`<div>`);
  document.write(`<h2> welcome ${us} </h2>`);
  document.write(`<p> your age is : ${ag} </p>`);
  document.write(`<p> your hour rate is : ${rt} </p>`);
  if(show==="yes")
  {
    if(skills.length>0){
      document.write(`<p> your Skils is : ${skills.join(" | ")} </p>`);
    }
    else{
      document.write(`<p> Skils : No skills to show </p>`);
    }
  }

  }
  else{
    document.write(`<p> skills is hidden </p>`);
  }

  }
  document.write(`</div>`);
}
showInfo("Hossam",21,3,"yes", "HTML","CSS");
-----
// فانكشن بدون اسم لتنفيذ امر معين عند الضغط على الزر يظهر الاسم
document.getElementById("show").onclick=function() //
{
  document.write("show");
}
-----
// فانكشن داخل فانكشن
function sayMessage(fname, last, )
{
  let message="welcome";
  function concat(){
    message= `${message} ${fname} ${last} `
  }
  concat();
  return message;
}
console.log(sayMessage("Hossam", "Hasen" ));
-----
// arrow تستخدم فى حاله كانت الفانكشن سطر واحد

let print = (num1,num2) => num1+num2;
console.log(print(100,50));
-----
//scope
var a=1;
let b=2;
function showText(){
  // كده هو هياخد ده عشان الفانكشن الاول بنحب على المتغير بنعها داخل القوس ولو مش موجود هتاخذ من بره

```

```

    console.log(`function _ from global ${a}`);
    console.log(`function _ from global ${b}`);

}
console.log(`from global ${a}`); //مش هيقدر ياخذ من القيمة اللي جوه فانكشن مش تبعه
console.log(`from global ${b}`);
showText();

var x=10;
if (x===10){
    let x=50; //10 = كده هو هياخذ القيمة اللي بره
    // var x=50 كده هياخذ القيمة ده ومش هيركز على اللي بره
}
console.log(x);

-----

//Higher : map

let mynum=[1,2,3,4,5,6];
let newarray=[];
for(let i=0 ; i<mynum.length;i++){
    newarray.push(mynum[i]+mynum[i]);
}

//الماب بتقبل ثلاث قيم وتستخدم افضل من الطريقة السابقة
let addself =mynum.map (function(elment,inde,arr)
{
    //console.log(` the elments : ${elment} `)
    // console.log(` the index : ${inde} `)
    //console.log(` the array : ${arr} `)
    //console.log(` this : ${this} `) بتعمل استعداد للفيكشن
    return elment+elment; // بتجمع العنصر + نفسه
}
,5); //عدد العناصر المطلوبه
console.log(addself);

//or
let addarrow =mynum.map((elment) => elment+elment);
console.log(addarrow);

//or
function additon(ele){
    return ele + ele;
}
let ad= mynum.map(additon)
console.log(ad)

-----
EX
// تطبيق تحويل الحروف من كابتل لى سمول والعكس
swapCasee="hoSSAM";
let sw= swapCasee.split(" ").map(function(ele) // اول حاجه بنحول الاسترنج لى ارى
{
    return ele ===ele.toUpperCase() ? ele.toLowerCase:ele.toUpperCase() // اختصاره جمله اف
})
.join(""); // بنرجعها تانى لى استرنج
console.log(sw)
//تم تحويلها لى ارى

let inversNumber= [-10,20,-30,40,-50]; // تحويل الموجب لسالب والعكس
let inv= inversNumber.map(function(v)
{
    return -v; // سالب فى سالب بموجب

```



```

});
console.log(inv);

let ignoreNumber ="H0S123sa4m "; // عايزين نلغي الارقام
let ignor=ignoreNumber.split("").map(function(ig)
//لو العنصر ده مش رقم يبقى رجعه انما لو رقم مترجيش حاجه
{
return isNaN(parseInt(ig))? ig :"";
}).join("");

console.log(ignor);
-----
// Filter

let Friends=["hossam","hazem","hasen","ahmed","eman"];
let filterFriends=Friends.filter(function(name){
return name.startsWith("h"? true: false;
});
console.log(filterFriends)
-----
// بتجمع كل عنصر مع التاني reduce

let adN=[10,20,30,40,50]
let tot=adN.reduce(function(acc,current,index,arr){
return acc+current;
})
console.log(tot);

// يرجع الكلمة الاكبر بين العنصر الاول والتاني لحد مايتصفي
let theBigest=["bla","propgand","other","AAA","Battery","test"]

let cheek =theBigest.reduce(function(ac,curent,index,ar){
console.log(`acc => ${ac}`);
console.log(`curent Elements => ${curent}`);
console.log(ac.length >curent.length ? ac : curent);
console.log("###");
return ac.length >curent.length ? ac : curent;
})

-----
// object

let user ={
// الخاص
theName:"hossam",
age:21,
//العمليات
say:function(){
return `welcome`;
}
}
console.log(user.theName);
console.log(user.age);
console.log(user.say());

//اكثر من اوبجكت
let user ={

theName:"hossam",
age:21,
skil:["Html","css","JS"],
ava :true,
addres:{

```

```

        ksa:"gada",
        Egypt: {
            one: "cairo",
            two : "luxor",

        },

    },
    cheekAv:function(){
        if(user.ava===true)
        {
            return 'is av to work';
        }
        else{
            return 'is not av to work';
        }
    },
};
console.log(user.theName);
console.log(user.age);
console.log(user.skil.join(" | "));
console.log(user.addres.Egypt.two);
console.log(user.cheekAv());

```

// عمل ابدیت للقيمة

```

let use ={
    nam:"moh",
    age:20,
};
console.log(use)
use.nam="hossam";
use["contry"]="luxor"
use.age=21;
use.sa=function(){
    return`welcome`;
};
console.log(use)
console.log(use.sa());

```

//id يقوم باسترجاع العنصر المالك لهذا ال

```

document.getElementById("bu").onclick=function(){
    console.log(this)
}

```

```

let us={
    a:30,
    ageInDay:function(){
        return us.a*365;
        //or this.a*365;
    },
}

```

```

}
console.log(us.a);
console.log(us.ageInDay());

```

--

```

let user ={
    age:20,
    doubleAge:function(){
        return this.age*2;
    }
}

```

```

    }
  }
  console.log(user);
  console.log(user.age);
  console.log(user.doubleAge());

  // إنشاء اوبجكت جديد من اوبجكت اليوزر
  let obj=Object.create(user)
  obj.age=30;
  console.log(obj);
  console.log(obj.age);
  console.log(obj.age);
  console.log(obj.doubleAge());

  let obj1={
    prop1:1,
    Math1:function(){
      return this.prop1;
    }
  }

  let obj2={
    prop2:2,
    Math1:function(){
      return this.prop2;
    }
  }

  let targte={
    prop1:1,
    prop3:3,
  }

  let finialObject=Object.assign(targte); // ياخذ التارجت من اكثر من اوبجكت ويجمعهم
  finialObject.prop1=100;
  finialObject.prop2=200;
  console.log(finialObject)

```

DOM

```

// استدعاء العنصر من الـ id
let myIdElement=document.getElementById("my-div");
console.log(myIdElement);

// استدعاء العنصر بناء على البرجراف
let myTagElement=document.getElementsByTagName("p");
console.log(myTagElement[0]); // نستدعي اول برجراف حسب الاندكس

// استدعاء حسب الكلاس
let myClassName=document.getElementsByClassName("my-span");
console.log(myClassName[0]);

// تغيير العنصر وكتابته في الصفحة
myTagElement[1].innerHTML="test";

// وبتاخذ اول عنصر سواء اي دي او كلاس
let myQuerySlector= document.querySelector(".my-span");
// بتاخذ كل العناصر
let myQuerySlectorAll= document.querySelectorAll(".my-span");
console.log(myQuerySlectorAll)

// اعرض العنوان
console.log(document.title);
// اعرض الهيكل
console.log(document.body);
// اعرض القيمة بتاغت اول فورم
console.log(document.forms[0].one.value);
// استدعاء الرابط بتاغ اللينك
console.log(document.links[0].href);

let elment= document.querySelector(".js");

```

```

console.log(element.innerHTML); // بترجع جمل الامر الى مكتوب
console.log(element.textContent); // بترجع محتوى التكتست

element.innerHTML="TEXT FROM <span> main.js </span> file "; // بتطبع الاسيان كانها امر
element.textContent="TEXT FROM <span> main.js </span> file "; // بتطبع الجملة كامله

document.images[0].src=" google "; // تغير المصدر بتاع الصورة
document.images[0].alt=" alternate "; // تغير الشرح بتاع الصور
document.images[0].title=" pic "; // تغير عنوان الصورة
document.images[0].id=" pc "; // نعطلها اى دي
document.images[0].className=" go "; // نعطلها كلاس

let myLink= document.querySelector(".link");
console.log(myLink.getAttribute("class")); // ظهور اسم الكلاس
console.log(myLink.getAttribute("href")); // ظهور المصدر
myLink.setAttribute("href","https://google.com") // اللينك بيودي على جوجل

<p class="para" title="parag" data-src="testing" > pargrph </p>
console.log(document.getElementsByTagName("p")[0].attributes) // اظهار جميع الصفات

let myP=getElementsByTagName("p")[0]; // بنشوف هل العنصر ده فى الصفات ده ولا لاء
if(myP.hasAttribute("date-src")){
  if(myP.getAttribute("date-src")===" " ) // بنشوف هل العنصر موجودوفاضى
  {
    myP.removeAttribute("date-src") // لو لقيته يبقى احذفه
  }
  else{
    myP.setAttribute("date-src","new value") // ولو مش فاضى غير قيمته
  }
}

console.log("fond")

}
else{
  console.log(" not fond")
}

----

if (myP.hasAttribute()) // هل العنصر فيه صفات
{
  console.log(`has att Attribute`);
}

-----

let myElements= document.createElement("div"); // انشاء عنصر فى الجافا بدل الانش تى ام ال
let myAtter= document.createAttribute("data-custom"); // عمل صفات داخل لعنصر
let mytext= document.createTextNode("product one"); // كتابه النص داخل الدف اللى عملناه فوق
let myComment=document.createComment("this is div") // انشاء كومنت

myElements.className="prodcut"; //

myElements.setAttributeNode(myAtter); // عشان لو عنصر قيمته فاضيه وجبيت تغيره فى المستقبل

myElements.setAttribute("data-test","testing"); // او نغيرها من هنا

myElements.appendChild(mytext) // عشان نظهر النص اللى عملناه

myElements.appendChild(myComment); // اظهار الكومنت

document.body.appendChild(myElements); // عشان يتفد داخل الصفحه

```

```

-----

let myElements= document.querySelector("div");
console.log(myElements);
console.log(myElements.children); // بيرجع العناصر اللي موجوده داخل الاب ده
console.log(myElements.children[0]); // بيرجع اول عنصر فقط
console.log(myElements.childNodes); // يعرض كل حاجه العنصر والنود والتكست والعدد
console.log(myElements.childNodes[0]);
console.log(myElements.firstChild); // بتجيب اول عنصر سواء تكست او كومنت داخل الاب
console.log(myElements.lastChild); // بتجيب اخر عنصر داخل الاب
console.log(myElements.firstElementChild); // بتجيب اول عنصر فقط داخل الاب

-----

let mybtn=document.getElementById("btu"); // لما نضغط عليه يعمل ايفنت
mybtn.onclick=function(){
    console.log("clicked");
}

mybtn.oncontextmenu=function(){} // لم تضغط كلك يمين يظهر قائمه خاصه بيها
mybtn.onmouseenter=function(){} // اول ماالموس يعمل اتر يظهر الالفنت
mybtn.onmouseleave=function(){} // اول ماالموس يطلق من الزر

window.onscroll=function(){ // لم ننزل فى الصفحه
    console.log("scol");
}

-----

// جعل الفورم يقبل قيم معينه
let userInput =document.getElementById("use") // نستدعى العنصر عن طريق الاي دى
let ageInput =document.getElementById("ag");

document.forms[0].onsubmit=function(e){
    let userValid = false ; // فولس عشان لسه مش اتملت
    let ageValid = false ;

    if (userInput.value!="" && userInput.value.length <=10){
        userValid=true;
    }
    if(ageInput!=""){
        ageValid=true;
    }

    if (userValid===false || ageValid===false ){
        e.preventDefault() // معناها لو اى خانه ممتلئش يبقى ميرسلش البيانات
    }
}

document.links[0].onclick=function(event){
    console.log(event);
    event.preventDefault()
}

-----

<div id="my-id" class="one two show test" > div with many classes</div>

let elemnt=document.getElementById("my-id")
console.log(elemnt.classList) // بنشوف فيها كام كلاس ويتكون نوعها اوبجكت
console.log(elemnt.classList.contains("show")) // هل يحتوى على الكلاس
console.log(elemnt.classList.item(1)) // بيقولك اسم الكلاس
console.log(elemnt.classList.item(1)) // بيقولك اسم الكلاس

```

```

-----

elemnt.onclick=function() //لما اضغط عليه يضيف اثنين كلاس
{
    elemnt.classList.add("add-one","add-two");
}
--
elemnt.onclick=function() //لما اضغط عليه ييحذف الكلاس
{
    elemnt.classList.remove("one","two");
}
---
elemnt.onclick=function() //لو الكلاس موجود هيشلو ولو مش موجود هيحطه
{
    elemnt.classList.toggle("one");
}
-----

let elemnt=document.getElementById("my-id")
elemnt.style.backgroundColor="red"; //تغير الخصائص
elemnt.style.cssText="color:blue;font-weight:bold"; //كتابه التسميات فى سطر واحد
elemnt.style.removeProperty("color"); //ازاله الخاصيه
elemnt.style.setProperty("color","red"); //اضافه خاصيه جديده

document,StyleSheet[0].rules[0].style.removeProperty("line-height"); //الدخول لملف الاستيل وعمل خط عريض عليه وحذفه

-----

let elemnt=document.getElementById("my-id")
let createdP=document.createElement("p");

elemnt.before("hi"); //بتحطه قبل العنصر
elemnt.after(createdP); //بتحطه بعد العنصر

elemnt.append("end");//بيحط الكلمه فى نهايه الكلام
elemnt.prepend("start");//بيحط الكلمه فى بدايه الكلام
elemnt.remove(); //بتحذف العنصر خالص

-----
let span=document.querySelector(".two");

console.log(span.nextSibling) //بتاخذ اول عنصر بعده حتى لو كان كومننت
console.log(span.nextSibling.remove())
console.log(span.nextElementSibling); //بتاخذ العنصر اللى بعده
console.log(span.previousElementSibling); //بتاخذ العنصر اللى قبله

span.onclick=function(){
    span.parentElement.style.opacity="0"; //ابحث عن العنصر الاب وغير خاصيه
}

-----
let myp=document.querySelector("p");
let mydiv=document.querySelector("div")
mydiv.appendChild(myp) //هياخد البرجراف ينقله داخل الدف

let mypp=document.querySelector("p").cloneNode(true); //بياخذ نسخه من كل العناصر ولازم نعط نرو عشان ياخذ المحتوى كله
mypp.id=`${mypp.id}-clone`; //عشان بعد ما هينسخ لازم نغيره الاى دي

-----

let myp=document.querySelector("p");

```

```

let mydiv=document.querySelector("div")
mydiv.appendChild(myp) // هياخذ البرجغراف ينقله داخل الدف

let mypp=document.querySelector("p").cloneNode(true); // يباخذ نسخه من كل العناصر ولازم نخط نرو عشان ياخذ المحتوى كله
mypp.id=`${mypp.id}-clone`; // عشان بعد ما هينسخ لازم نغيره الاى دي

// عشان نخط اكثر من ايفنت داخل فانكشن معينه عشان ميعملش اوفر ريدن
myp.onclick=one;
function one(){
    console.log("from onclik 1");
}
myp.addEventListener("click",function(){
    console.log("from event 1");
})
myp.addEventListener("click",one) // عند الضغط تنفذ تانى الفانكشن
-----
// clone استنساخ

let myp=document.querySelector("p") // عند الضغط عليه ينشئ اكثر من برجغراف مستنسخ
myp.onclick= function(){
    let newp=myp.cloneNode(true);
    newp.className="cloned"; // مستنسخ
    document.body.appendChild(newp); // اظهره فى الصفحه
}

// عشان لما ناجى نضغط على سطر مستنسخ يطالع رساله
document.addEventListener("click",function(e){
    if(e.target.className===`cloned`){
        console.log("im cloned");
    }
})
-----
BOM

window.alert("errorr"); // رساله تحذيره
let confirmMessage=confirm("are you sure to delete this itme") // طباعه رساله تاكديه
if(confirmMessage==true){
    console.log("the item deleted")
}
else{
    console.log("the item not deleted")
}
--
let PrompMas=prompt("write the reason","should have 3 char") // طباعه مكان لكتابه السبب
console.log(PrompMas);
-----
setTimeout(function() // اظهر رساله بعد وقت
{
    console.log("masg")
},3000) // = s3 ميلى ثانيه
// or
setTimeout(sayMsg,3000,"Hossam",38)
function sayMsg(user,age){
    console.log(`im massge form${user} his age${age}`)
}
---
// لالغاء الوقت
let cont =setTimeout(sayMsg,3000)
function sayMsg(){
    console.log(`im massge `)
}
console.log(cont)

let btn=document.querySelector("button")

```

```

btn.onclick=function(){
    clearTimeout(cont)
}
-----

// يتخلّى الرقم داخل الدف ينقص تلقائى
let div=document.querySelector("div") ;

function countdown(){
    div.innerHTML -=1;
    if(div.innerHTML ==="0"){
        clearInterval(conter)
    }
}
let conter = setInterval(countdown,1000);

----
console.log(location); //تعرض معلوماتك كامله

//تخليه يفتح لينك فى صفحه ثانيه تعديل طول وعرض ومكان ظهورها
let myWindoe=window.open("https://www.google.com", "_blank", "width=400 ,height=100 ,left=200,top=200")
myWindoe.focus(); //تعمل تركيز على الصفحه

```

```

عمل زر للصعود والنزول
button{
    background-color: royalblue;
    border: none;
    color: white;
    font-weight: bold;
    border-radius: 4px;
    position: fixed;
    bottom: 20px;
    right: 20px;
    display: none;
    cursor: pointer;
}

let btn= document.querySelector("button");
window.onscroll=function(){
    if (window.scrollY>=600){
        btn.style.display="block"
    }
    else{
        btn.style.display="none"
    }
}
btn.onclick=function(){
    window.scrollTo({
        left:0,
        top:0,
        behavior:"smooth"
    });
};

-----
// تخزين قيمه فى حاله كان المستخدم عايز يغير اللون
window.localStorage.setItem("color","red");
window.localStorage.fontWeight="bold";

// استدعاء القيم
window.localStorage.getItem("color");
window.localStorage.fontWeight;

```



```
// اظهارها فى الصفحه
window.body.style.backgroundColor=window.localStorage.getItem("color")
console.log(window.localStorage)
```

```
window.localStorage.removeItem("color"); // حذف العنصر
window.localStorage.clear(); // حذف جميع العناصر
```

```
// destructuring نقل الاري الى المتغيرات
let myFirends=["Hossam","ahmed","hazem"];
let [a,b,c]=myFirends; // كل حرف بيقابل الاسم
// [a,b,c]=myFirends لو كانت متعرفه قبل كده نستخدم ده
// [a, , b ,c] لو مش عايزين اسم فى النص نسيبه فاضى
console.log(a);

-
let myFirends=["Hossam","ahmed","eman",["shady", "moha",["ge", "ys"]]];
let [, , [a, , [ , c]]]=myFirends; // للوصول لشخص معين
console.log(a); //shady
console.log(b); //ge
-
let {thename,theage,contry}=user;
// or ( {thename,theage,contry}=user);

-----
// عمل اوبجكت فانكشن بطيع العناصر
const user={
  thename:hossam,
  theage:21,
  contry:egypt,
  skills:{
    html:90,
    css:67,
  },
};

showDetails(user);
function showDetails({thename:N,theage:A,skills:{css:C}} =user) //عمل تسميه للعناصر
{
  console.log(`your name is ${A}`);
  console.log(`your name is ${B}`);
  console.log(`your name is ${C}`);
}
console.log(user);

-----
let myDate=[1,1,2,2,2,3,"A"]; // بطيع كل الارقام
let uniQueDate=new set([1,1,2,2,2,3]); // بتجيب رقم وحيد بس
// or uniQueDate.add(1).add(2);
// uniQueDate.delete(3) حذف رقم تلاته
console.log(myDate);
console.log(uniQueDate);
console.log(uniQueDate.size); // عددهم
console.log(uniQueDate.has("A")); // هل تحتوى على هذا العنصر

-----
//set & weakset
//size
let mySet=new set([1,1,2,2,2,3,"a"]); // بتجيب رقم وحيد بس
let myweSet=new WeakSet([A:1,B:2]); // بتقبل اوبجكت
console.log(`size of elment inside set ${mySet.size}`);
```

```

let iteratotr =mySet.key();
console.log(iteratotr.next().value); // وفى نواتج القيمه هتكون فيها فولس عشان لسه قصاها عنصر تانى ولما يوصل للاخر يدى نرو

//for each
mySet.forEach((el) => console.log(el)); // تطيع كل عنصر لوحده

-----
// بتجيب البيانات بترتيب وافضل من الاوبجكت
let myEmptyobject=Object.create(null);
myMap=new Map();
let myobj={
  10:"Number",
  "10":"string"
}
console.log([10]);// كده هياخد اخر قيمه
let myMap=new Map(); // مش هتعمل اوفر رايدن وهتعرض الكل
myMap.set(10,"number","10","string")
console.log(myMap.get(10));
console.log(myMap.get("10"));

---
let Mapp=new Map // طريقة الكتابه
([
  [10,"Number"],
  ["10","string"]
  [false, "Boolean"]
])
console.log(Mapp.get(false)) // طريقه الاستدعاء
console.log(Mapp.delete("Boolean")) // للحذف
console.log(Mapp.has(false)) // هل يحتوى

// or let week=new WeakMap ويوجد ايضا
-----
//array.from
console.log(Array.from("hossam")) // هتحولها لى ارى
console.log(Array.from("12345",function(n){
  return +n + +n ; // اجمع كل رقم على نفسه
})))

-----
let myArray=[10,20,30,40,"a","b"];
myArray.copyWithin(3) ; // هياخد من اندكس صفر لى اثنين يطيعهم ويعملهم كوى بست ويحذف اللى بعده
myArray.copyWithin(3,5) ; // هينسخ الاندكس خمس وخطها بدل تلاته ويحذف القديم
console.log(myArray);

-----
// بتشبه اللوب بتقعد تمشى تنفذ لحد ماتلاقى الشرط
let nums=[1,2,3,4,5,6,7,8,9]
let check=nums.some( function(e){
  console.log("test") // هتطيعهم ست مرات عشان لسه ملاقاش الشرط
  return e>5;
});
console.log(check)

let number=5; // نفس المثال السابق بس لو عندنا عنصر بره نشيك عليه داخله
let cheak= nums.some( function(e){
  return e>this;
},number);

-----
// لازم يكون كل الشروط تنفذ عشان يرجع نرو عكس الفانكشن اللى قبلها
const pl={

```

```

    20:"eg",
    10:"sd",
    30:"js",
  }
  let minlocation=15;
  let locObject=Object.keys(pl);
  let locArrayNumber=locObject.map((n)=>+n); // عشان يرجعهم كالأرقام مش استرنج
  let cheek= locArrayNumber.every(function(e){
    return e>this
  },minlocation) // هترجع فولس
  -----
  console.log(..."hossam") // هتعمل توسيع للحروف

  let allFriend=["hossam","ahmed"];
  let friend=["js","ht"]
  allFriend.push(...friend); // هتعملهم دمج اري واحده

  let myNum=[10,20,30,40,50,60]
  console.log(Math.max(...myNum)) // عشان لو محطناش التلت نقط هنضطر نكتب كل الارقام من تاني جو القوس

  obj1={
    na:hossam,
  }
  obj2={
    na:ys,
  }
  console.log({...obj1,...obj2 , h:4}) // هينشى ابوجكت جديد مدموج بينهم وخط ابوجكت زياده

```

```

// regular لو عابزين نطاق كلمه هيدخلها اليوزر بي كلمه او نهج ثابت بدل مانستخدم جمل الشرط
// i = بتخلي الحروف مش حساسه يعنى ياخذ الكابتل او الاسمور
// g= هيعرض كل النتائج
// m= يبحث فى الاسطر كلها
--
let tld="com org info to";
let tldRe=/ (org |com|info)/i; // رجع اى قيمه موجوده منهم لو كتبت حرف جى هيجلبى كل النتائج
console.log(tld.match(tldRe)) ; // طريقه الطباعة
----
let nums="12345678910";
let numsRng=/[0-9]/g; // هياخد رنج الارقام
console.log(nums.match(numsRng)) ;
----
let notnums="12345678910";
let notnRng=/[^0-2]/g; // هياخد كل الارقام ماعدا الرينج ده
console.log(notnums.match(notnRng)) ;
---
let specialnums="1!2#3%4^7*8@910";
let specialRng=/[^0-9]/g; // هياخد كل العلامات ماعدا الارقام
console.log(specialnums.match(specialRng)) ;
-----
let practic="os1 os3os os4os os3 os10os";
let pract=/os[2-4]os/g; // هياخد الاسم الى اوله حروف مابين رنج معين واخره حروف
console.log(practic.match(practic)) ;
-----
let allLetters=/[a-zA-Z] /g; // جميع الحروف كابتل واسمول

let dot=/./g; // بتجيب كل الحروف والارقام والسافات
let word=/ \w /g; // بتجيب الحروف الكابتل والاسمول
let words=/ \W /g; // بتجيب الروموز
//let words=/ \d /g; // بتجيب الارقام من صفر لتسعه
//let words=/ \D /g; // عكس الى فوق بتجيب يعنى الحروف

```

```
//let words=/ \s /g; بتجيب بالمسافات
//let words=/ \S /g; بتجيب من غير المسافات
let valid =/\w@\w.(com|org)/ // مثال الايميل
let word=/ \bspa, /g; // شرط الكلمة تكون فى بدايه الكلام
//let word=/ spam\b /g; // شرط الكلمة تكون فى نهايه الكلام
//let word=/(\bspam|spam\b) /g; // اول او نهايه الكلام
//let word=/(\bspam|spam\b) /g.test("spamming"); // هل يحتوى
//let word=/ \B /g; // عكس اللى فوق لا بدايه ولا نهايه
//n+ one or more
//n* zero or more
//n? zero or one مثلا https? ممكن اخر حرف يبقى موجود او لاء
// // معنى ممكن تكون فيها الجملة ده او علامه السلاش (www.)? (https?:\\\/) عشان نعمل العلامتين دول نكتبها

//ing $.test() هل بيتنتهى
//^ing .test() هل بيبدء
//.match (/^d{5}(?=z) /) followed by ويكون بخمس حروف
//.match (/^d{5}(?!z) /) not followed by

let r=/@/ig;
console.log(re,"jave") // بتعمل تغير لكلمه بيقم تانيه

-----
// لو مكنش مطابق ميعتش البيانات
document.getElementById("registir").onsubmit=function(){
let phoneInput=document.getElementById("phone").value;
let phone = /^(\\d{4})\\s\\d{3}-\\d{4} /; // (1234) 567-9432
let valdresult=phone.test(phoneInput)

if (valdresult===false){
return false;
}
else{
return true;
}
}
```

```
// بدل ماننشئ كل اوبجكت لوحده ونعدل عليه لاء ننشئ ده oop constructor

class User {
constructor(id, username, salary) {
this.i = id;
this.u = username || "unknown"; // لو مكنش اسم يبقى رجع غير معروف
this.s = salary > 6000 ?salary+ 1000 :salary; // لو اقل من كذا يبقى زود المرتب

//1
this.msg=function(){
return`welcome your salary is : ${this.s} your id is :${this.id} `
}

}
//2
writemsg(){ // عمل ميثود بدل الفانكشن
return`welcome your salary is : ${this.s} your id is :${this.id} `
}

//3 update للبيانات
updateName(newName){
this.u=newName;
}
}
```

```

}

let userone=new User(100,"hossam",5000);
let usertwo=new User(101,"mohamed",6000);
let userthre=new User(102,"ahmed",7000);

console.log(userone.i);
console.log(userone instanceof User); // هل اليوزر ده مشتق من الكلاس ده
console.log(userone.constructor === User);
console.log(userone.msg());
console.log(userone.writemsg());
userone.updateName("khaled");
console.log(userone.u);

let strone="Hossam"; // نوعها استرنج
let strtwo=new String("Hossam"); // تستخدم ده افضل عشان تبقى مشتقه من الاوبجكت نوعها اوبجكت

----
// static
class User {
  constructor(id, username, salary) {
    this.i = id;
    this.u = username ;
    this.s = salary ;
    this.cont=0 ;
    User.cont++; // زود واحد كل ما حد يدخل
  }

  static contMember(){
    return`${this.cont} member created`
  }
}

let userone=new User(100,"hossam",5000);

console.log(userone.contMember()); // عشان استاتك مش هيقدر يستدعيها من اوبجكت فرعى ومش هيطلع ناتج
console.log(User.contMember()); // هيثم استدعها عادى

--
// inheritance
class User {
  constructor(id, username, salary) {
    this.i = id;
    this.u = username ;
    this.s = salary ;
  }
}

class admen extends User { // عشان نورث حاجه لازم نكتب اكستنت
  constructor(id, username, permission) {
    super(id,username); // سوبر عشان نستدعى القيم من الكلاس الثانى
    this.p=permission;
  }
}

};

let userone=new User(100,"hossam",5000);
let admenone=new admen(110,"jaso",10);
console.log(admenone.i);
-----
// encapsulatio = private

```

```

class User {
    #e;        // اولاً نعملها تعريف
    constructor(id, username, salary) {
        this.i = id;
        this.u = username ;

        this.#e = salary ;
    }
    getSalary(){        // لازم نعمل ميثود عشان نعرف نستدعي قيمتها بره
        return parseInt(this.#e)
    }
}

let userone=new User(100,"hossam",5000);
console.log(userone.getSalary()*0,3);    // نرجع القيمة ونضربها في رقم 3

-----
// prototype المكونات الاوبيجكت
class User {
    constructor(id, username, salary) {
        this.i = id;
        this.u = username ;

        this.s= salary ;
    }
    sayHi(){        // لازم نعمل ميثود عشان نعرف نستدعي قيمتها بره
        return ` hi `;
    }
}

let userone=new User(100,"hossam",5000);
User.prototype.saywelcome=function(){    // انشاء فانكشن جديد تضاف لى البروتو تايب
    return ` welcome `;
}
Object.prototype.love="love y"; // هتضيفها جو البروتو الخاص بالاوبيجكت
String.prototype.addDot=function(val){    // عمل فانكشن من النوع استرنج بتخط قبل الاسم وبعده دوت
    return `.${this}`;
}

-----
// prototype المكونات الاوبيجكت
class User {
    constructor(id, username, salary) {
        this.i = id;
        this.u = username ;

        this.s= salary ;
    }
}

const myObj={
    a:1,
    b:2,
}
Object.defineProperty(myObj,"c",{
    writable:true, // لو فولس يعنى محدش يقدر يعدل على القيمة ده بره الفانكشن
    enumerable:true, // لو فولس يبقى مش هقدر احطها في اللوب
    configurable:true, // لو كانت فولس مش هقدر احذفها بره الفانكشن
    value:4,
})
console.log(delete myObj.c)
//or اكتر من عنصر

```

```

Object.defineProperty(myObj,{
  c: {
    writable:true,
    enumerable:true,
    configurable:true,
    value:4,
  },
  g: {
    writable:true,
    enumerable:true,
    configurable:true,
    value:4,
  }
})
console.log(Object.getOwnPropertyDescriptor(myObj,'d')) // اللى عايزين نجيب بياناته

-----
// date
let dateNow=new Date();
let birthday= new Date("jun 1 , 2001")
let dateDefr=dateNow-birthday
console.log(dateDefr/ 1000 / 60 / 60 /24 /365 ) // نوانى دقايق ساعات ايام سنه
console.log(dateNow)
console.log(dateNow.getTime())
console.log(dateNow.getDate()) // هتجيب كام فى الشهر
console.log(dateNow.getFullYear()) //2022
console.log(dateNow.getMonth()) // هتجيب الاندكس بتاع الشهر يعنى رقم حداثر هتساوى شهر اتناشر
console.log(dateNow.getDay()) // الاحد بيساوى صفر
console.log(dateNow.getHours())
-----
// date set
let dateNow=new Date();
console.log(dateNow)

dateNow.setTime(0); // هيجب التاريخ من بدايه 1970 اللى هو بدايه توقيت اليونكس
console.log(dateNow)

dateNow.setDate(1); // هيجب الاول من الشهر اللى احنا فيه والسنة ولو كتبت صفر هيجب اخر يوم فى الشهر قبله
console.log(dateNow)

dateNow.setFullYear(2021,11); // شهر ديسمبر
console.log(dateNow)
---
// generte
function* genertNumber(){
  yield 1;
  yield 2;
  yield 3;
}
let genertor= genertNumber();
console.log(genertor.next()) // هيطبع اول واحده ويهدا يوقف وهتبقى على شكل ارى

for(let value of genertNumber()){
  console.log(value); // هيطبع واحد اتنين تلاته
}
-----
// generte
function* genertNumber(){
  yield 1;
  yield 2;
  yield 3;
}

```

```
function* genertLetter(){
  yield "a";
  yield "b";
  yield "c";
}
function* genertAll(){
  yield*genertNumber();
  yield*genertLetter();
  yield*[5,6,7]
}
let genertor= genertAll();
console.log(genertor.next()); // هبطيع اول واحده وبهدا يوقف وهتبقى على شكل اري
console.log(genertor.return()); // هتخليها تقف ومش هتطيع حاجه بعدها
-----
function* genertAll(){
  let index=0;
  while(true){ // هيولد عدد لانهاى من الارقام كل مانطلب منه
    yield index++;
  }
}
let genertor= genertAll();
console.log(genertor.next());
```

```
// moduls export import
export let a= 10;
export let arr=[10,20,30];
export function sayh(){
  return `welcom`
}
// بعدها نروح للصفحة التانيه اللى انشائها
//<script src="java.js" type="moduls"> </script>

import {a,arr,sayh as hallo} from "./java.js";
console.log(hallo()); // بعد ماغيرنا اسمها

-----
export let a= 10;
export let arr=[10,20,30];
export function sayh(){
  return `welcom`
}
export {a as myNum ,arr,sayh };
export default function saWe(){
  return `hello`
}
import saWe,{myNum,arr,sayh} from "./java.js"; // ضيفنا الفانكشن الجديد بتكون بره القوس
console.log(myNum);
console.log(sayh());
console.log(saWe());
```

```
// java script object notion =json
وهي بديل لغه ال xml وهي مشاركة البيانات من
الباك اند للفروند
طريقه الكتابه بتكون داخل اقواس مجموعه
وكل عنصر يتكتب بين دابل كوتس
{
  "string":"Hossam",
  "number":3,
  "object":{
```



```

"egy": "luxor",
"mi": "minia"
},
"array": ["html", "css", "json"],
"boolean": true,
"null": null
}
}
}

-----
const myJsonFromServer = { "username": "hossam", "age": 21 }; // البيانات التي جابه من السيرفر الجسون
console.log(typeof myJsonFromServer); // نوعه استرنج
console.log(myJsonFromServer);

const myJSObject = JSON.parse(myJsonFromServer) // بعدها نحول البيانات لي اوبجكت بلغه الجافا
console.log(typeof myJSObject); // نوعه اوبجكت
console.log(myJSObject);

myJSObject["username"] = "ahmed"; // لو عايزين نغير البيانات

const myJOSNObjectToSer = JSON.parse(myJsonFromServer) // بعدها نحول البيانات لي الجسون تاني بعد ماعملنا التعديل
console.log(typeof myJSObject); // نوعه اوبجكت
console.log(myJSObject);

//synchronous المزامنه ولازم العمليات تتم بترتيب
console.log("1");
// window.alert("operation") الكود هيقف هنا
console.log("2");

//Asynchronous كل عمليه هتم ملهاش علاقه بتانيه
console.log("1");
setTimeout(() => console.log("opreation"), 4000);
console.log("2");

console.log("#####");

setTimeout(() => {
    console.log("web api") // application program ineterface هتنتفذ بعد الفانكشن اللي تحت
}, 0);

function one(){
    console.log("one");
}

function two(){
    one();
    console.log("two");
}

function three(){
    two();
    console.log("three");
}

three(); // اخر واحد يدخل هو اول واحد يخرج last int first out زي الاستاك
// هتساوي الترتيب ده انما لو كتينا الفانكشن واحد مش هتطيع الا غيره
console.log("###");
console.log("one");
console.log("two");
console.log("three");
-----
console.log("one");
setTimeout(() => { // queue تنبيه الكيو اللي زي الطابور اللي يا جى الاول يخرج الاول
    console.log("three");
}, 0);
setTimeout(() => {

```

```

        console.log("four");
    }, 0);
    console.log("two");
    // هتطبع واحد اتنين تلاته اربعه عشان فانكشن الوقت بتطبع فى الاخرس فى
    ----
    let myreq=new XMLHttpRequest();
    myreq.open("GET","https://api.github.com/users/elzerowebsschool/repos");
    myreq.send()
    console.log(myreq);
    //readyState 4 معناه الطلب خلص مع الاتصال مع السيرفر سواء فتح الصفحه او لاء 4
    //status:200 معناه تمت العمليه

    myreq.onreadystatechange=function(){ // متابعه عمليه الاتصال والطلب مع السيرفر
        // console.log(myreq.readyState);
        //console.log(myreq.status);

        if(this.readyState===4 && this.status === 200)
        {
            // console.log(this.responseText) // عباره عن جيثون ابوجكت
            let JsData=JSON.parse(this.responseText) // نحوله لجافا اسكربت
            // console.log(JsData) // نتعرض البيانات الخاصه بالسيرفر وكل عنوان

            for(let i=0; i<JsData.length;i++){ // انشاء كل اسم فى دف لوحده
                let div=document.createElement("div");
                let repName=document.createTextNode(JsData[i].full_name);
                div.appendChild(repName);
                document.body.appendChild(div);
            }
        }
    }

    -----
    // call back function
    function makeItRed(e){ // عند الضغط على الكلمه يتحول للون الاحمر

        e.target.style.color="red"
    }
    let p =document.querySelector(".test");
    p.addEventListener("click", makeItRed);

    setTimeout(() => {
        console.log("download photo from url")
        setTimeout(() => {
            console.log("rezie photo")
            setTimeout(() => {
                console.log("add your logo")
                setTimeout(() => {
                    console.log("show the photo in web site")
                }, 1000);
            }, 1000);
        }, 1000);
    }, 1000);

    ---

    /* promises
    هتطبع المعلومات بدون الحاله
    const myPromies=new Promise((resolveFunction ,rejecFunction) =>{
        let connect=true;
        if(connect){
            resolveFunction("connection Establish");
        }
        else{

```

```

        rejecFunction("connection faild");
    }
}).then(
    (resolveFunction)=>console.log(`Good : ${resolveFunction} `),
    (rejecFunction)=>console.log(`Bad : ${rejecFunction} `),
); */

const myPromies=new Promise((resolveFunction , rejecFunction) =>{
    let connect=true;
    if(connect){
        resolveFunction("connection Establish");
    }
    else{
        rejecFunction("connection faild");
    }
});

myPromies.then( (resolveFunction)=>console.log(`Good : ${resolveFunction} `), // الافضل عشان تطيع الحاله معاها
(rejecFunction)=>console.log(`Bad : ${rejecFunction} `),
);
console.log(myPromies); // هتطيع الاول

/* ممكن نطبق ده ونكررها بدل المثال اللى قبله بدل مانعمل كذا فانكشن فيها وقت
يعنى بعد مالموعده يتحقق نفذ اللى بعده
*/
myPromies.then( (resolveFunction)=>console.log(`Good : ${resolveFunction} `),
(rejecFunction)=>console.log(`Bad : ${rejecFunction} `),
);
console.log(myPromies); */

-----

const myPromies=new Promise((resolveFunction , rejecFunction) =>{
    let employee=["hossam","gehad","eman"]
    if(employee.length===3){
        resolveFunction(employee) // لو كل الموظفين حصرو اطبع اسمائهم
    }
    else{
        rejecFunction(Error("the number not full")) // الايرور ده عشان لو حاجه مكملتش يقولك الخطا فين
    }
});

myPromies.then( (resolveValue)=>{
    resolveValue.length=2; // خلينا عدد الموظفين اتين
    return resolveValue;
})

.then( (resolveValue)=>{
    resolveValue.length=1; // خلينا عدد الموظفين واحد
    return resolveValue;
})
.then( (resolveValue)=>{
    console.log(`we choisen is:${resolveValue}`);
})
.catch((rejectReason) => console.log(rejectReason)).finally(console.log("happy"))
// هتطلع السبب واخر فانكشن هتتغذ فى كل الحالات

-----
// promise and XHR with xml request
const getData= (apiLink)=>{
    return new Promise((resolve , reject)=>{
        let myReq=new XMLHttpRequest();
    });
};

```

```

myReq.onload=function(){
    if(this.readyState===4 && this.status === 200)
    {
        resolve(JSON.parse(this.responseText) )
    }
    else{
        reject(Error("No data fond"))
    }
};
myReq.open("GET",apiLink);
myReq.send();

});
};

getData("https://api.github.com/users/elzerowebsschool/repos").then
((result)=> {
    result.length=10 // هنقلهم بدل ماتلاين عشره
    return result
}).then((result)=> console.log(result[0].name))
.catch((rej)=>console.log(rej)); // لو متفدش

-----
// fetch طريقه للحصول على الريبوسيت
fetch("https://api.github.com/users/elzerowebsschool/repos").then((result)=>{
    console.log(result);
    let myDate=result.json();
    console.log(myDate);
    return myDate; // عشان نستخدمها فى الزن اللى بعدها
})
.then((myDate)=>{ // ممكن نغير البارامتر مى داتا ونستخدمها ونغير فى كل خطوه
    myDate.length=10;
    return myDate;
})
.then((myDate)=>{
    console.log(result[0].name)
})

-----
// Promise.all
const myFirstPromise=new Promise((reslove ,reject)=>{
    setTimeout(() => {
        reslove("im first promies")
    }, 5000);
});

const mySecondPromise=new Promise((reslove ,reject)=>{
    setTimeout(() => {
        reslove("im second promies")
    }, 2000);
});

const myThredPromise=new Promise((reslove ,reject)=>{
    setTimeout(() => {
        reslove("im thred promies")
    }, 1000);
});

// شرط يكون اتحقق كل اللى قبله وبعد كده هيرجعهم فى ارى
Promise.all([myFirstPromise,mySecondPromise,myThredPromise]).then(

```

```

        (resolveValue)=> console.log(resolveValue),
        (rejectValue)=> console(`rejected:${rejectValue} `)
        // لو حاجه كانت ريجكت هطبعها هي لوحدها
    )
    // Promise.allSettled() وتقولك الخطا في اي واحد
    // Promise.race() هترجع القيمه اللي هتطلع قصاده الاول سواء في الحالتين
    -----
    // async Promise وميش هنتحتاج نكتب جوه اي برومز جو الشروط
    async function getData(){

        let user=["HOSSAM"];
        if(user.length>0){
            return ("user is found")
        }
        else{
            throw new Error("no user fond")
        }
    }

    console.log(getData()) // هترجع البرومز

    getData().then(
        (resolveValue)=>console.log(resolveValue),
        (rejValue)=>console.log(" rejected "+ rejValue)
    )

    -----
    // async await نعملها افضل من الزن عشان تنفذ بعد مالاامر اللي قبله يخلص
    const myPromies= new Promise((rs,rj)=>{
        setTimeout(() => {
            rs("im good promies")
            // rj(Error("bad promies"))
        }, 3000);
    });

    async function readDate(){ // لازم نخط الامر ده قبل الفانكشن عشان لو هنعمل اللي جواها
        console.log("befor promise");
        console.log(await myPromies); // المفروض يتنفذ الاول لكن عشان حاطين الامر ده هسييتي يتعمل بعده
        // console.log(await myPromies.catch((err)=> err)); في حاله الريجكت بينطلب منه يرجع الخطا
        console.log("after promise");

    }
    readDate();
    -----
    // async await try معناها هجرب هل البرومز هيتحقق ولا لا واذا كده نفذ الفانكشن اللي بعدها

    const myPromies= new Promise((rs,rj)=>{
        setTimeout(() => {
            rs("im good promies")
            // rj(Error("bad promies"))
        }, 3000);
    });

    async function fetchDate(){ // لازم نخط الامر ده قبل الفانكشن عشان لو هنعمل اللي جواها
        console.log("befor promise");

        try{
            let myData=await fetch("https://api.github.com/users/elzerowebsschool/repos")
            console.log(await myData.json());

        }catch (reason){
            console.log(`reson : ${reason}`);
        }
    }

```

```
    }finally{  
        console.log("after promise");  
    }  
  
    }  
    fetchDate();
```