

JAVA SCRIPT

```
VAR M;
```

```
M = 40;
```

```
M = 30;
```

هنا المتغير هياخذ اخر قيمة له و هي 30

```
VAR M = 20;
```

```
VAR M = 30;
```

هنا ينفع اكرر الفار بالمتغير و هياخذ اخر قيمة له و هي 30

```
LET X;
```

```
X = 80;
```

```
X= 90;
```

هنا المتغير هياخذ اخر قيمة له و هي 90 زي الفار

```
LET V = 70;
```

```
LET V = 50;  ERRORR
```

دە مینفەش هەنا مش هیدی للمتغیر اخر قیمة له ، لكن ینفع مع
الفار بس ملحوظة الفار و اللیت متشابهین بنسبة 90%

```
CONST Y = 50;
```

دە ثابت لا ینفع یتکرر بالمتغیر و لا ینفع ادبیلە قیمة تانیة غیرها

ملحوظة : ممکن محطش السیمیکولون اخر الامر لكن
لو الاکواد علی سطر واحد لازم طبعا احطها
مثال:-

```
DOCUMENT.WRITE(' HELLO!')
```

```
DOCUMENT.WRITE(10)
```

هی ببتکتب اوتوماتیک

```
DOCUMENT.WRITE('HI')DOCUMENT.WRITE(20)
```

ERRORR

هنا لازم طبعا نخط السیمیکولن کدة

```
DOCUMENT.WRITE('HI');DOCUMENT.WRITE(20);
```

DATA TYPES

1_STRING

TEXT

2_NUMBER 12 10.5 0.0958

3_BOOLEAN TRUE , FALSE

4_UNDEFINED UNDEFINED

5_NULL NO THING , OBJECT

EXAMPLES

1_ VAR X = 'MOHAMMED'

VAR X = '10'

DOCUMENT.WRITE(TYPEOF X)  STRING

2_VAR X = 30;

DOCUMENT.WRITE(TYPEOF X)  NUMBER

3_ DOCUMENT.WRITE(TYPEOF TRUE)
DOCUMENT.WRITE(TYPEOF FALSE) → BOOLEAN

4_ DOCUMENT.WRITE(TYPEOF UNDEFINED)
VAR X; → UNDEFINED

5_ DOCUMENT.WRITE(TYPEOF NULL) → OBJECT

Concatenation(الدمج)

Let name = 'MOHAMMED';

Let age = 25; space

Console.log(name + age);

or

Console.log(name + ' ' + age);

```
Console.log(` my name is ${ name }  
and my age is ${age}`);
```

العمليات الحسابية

```
Let num1 = 4;
```

```
Let num2 = 5;
```

```
Console.log( num1 + num2 );
```

```
Console.log( num1 - num2 );
```

```
Console.log( num1 * num2 );
```

```
Console.log( num1 ** num2 );
```

```
Console.log( num1 / num2 );
```

```
Console.log( num1 % num2 );
```

```
Console.log( 'num1' + num2 );
```

Num14

```
Console.log( 'num1' - num2 );
```

Nan

```
Console.log( 'num1' * num2 );
```

Nan

```
Let num = 10;
```

```
num = num + 1; or num += 1;
```

```
or num++;
```

```
num = num - 1 ;
```

or

num--;

or

num -= 5;

Console.log(++num);


ممکن تکتب بین الاقواس عادى جدا


Console.log(num++);

هنا مش هيزود القيمة فالكونسول بس هتزيد
فالبرنامج يعنى لو عملت بعدها


Console.log(num);

هتظهر القيمة الجديدة فالكونسول

Console.log('5' + '4');  54

Console.log('5' - '4');  1

Console.log('5' * '4');  20

Console.log('5' / 'ahmed');  nan


لو عوزت اجمع ای نص فالعالم یبقی


Console.log(+ '5' + + '4');


هیدینی ناتج جمعهم


Console.log(typeof + '5');  number


Console.log(typeof - '5');  number

Console.log(- '5');  -5

Console.log(- 'ahmed');  nan

Console.log(+true);  1

Console.log(-true);  -1

Console.log(+false);  0

`Console.log(+null);` → 0

`Console.log(+'4 mohammed');` → nan

وفى طريقة ثانية اخليها رقم ... وهى

`Console.log(number'4 M');` → nan

وفى طريقة احدث عشان احول الكلام لرقم

عشان لو فى مع النصوص رقم وهى

`Console.log(parseInt('4 M'));` → 4

طيب لو فى رقم بكسر مثلا ودة العيب الاول

`Console.log(parseInt('4.9 M'));` → 4

هيدينى الرقم الصحيح فقط طيب و الحل! :-

`Console.log(parseFloat('4.9 M'));`

نستخدم ال Float

4.9

وال `parseInt` ليها عيب كمان وهو لو فى

نص قبل الرقم هيديني Nan مثال

```
Console.log(parseInt('M 4 M'));
```

nan

وال parseFloat ليها نفس العيب برضك

```
Console.log(parseFloat('M4.9 M'));
```

Nan

شرح اوبجيكت الماكس فاليو

```
Console.log(Number.MAX_VALUE )
```

```
Console.log(Number.MAX_VALUE + 1566665)
```

هيديني اكبر قيمة برضك

```
Console.log(Number.MAX_SAFE_INTEGER )
```

هيديني اكبر قيمة آمنة

```
Console.log(Number.MIN_SAFE_INTEGER )
```

هيديني اصغر قيمة آمنة

Console.log(Number.isSafeInteger)

بيشوف هو رقم آمن ولا لا

مثال

Console.log(Number.isSafeInteger(56756))

true

Console.log(Number.isInteger(56))

true

Console.log(Number.isInteger('56'))

false

Console.log(Number.isNaN('56'))

false

Console.log(Number.isNaN('ahmed' - 9))

true

Console.log(Number.isNaN('ahmed' / 9))

true

Console.log(Number.isNaN('ahmed' + 9))

تحويل الارقام الى نصوص

```
let x = 100;
```

```
console.log(typeof string(x) )
```

```
console.log(typeof string(10) )
```

كانى كتبته بين علامات تنصيص

```
let x = 100;
```

```
console.log(typeof x.toString() )
```

```
console.log(typeof (20).toString() )
```

```
console.log(typeof 20..toString() )
```

اوبجكت الماث

الثابت ط Math.PI

الابسليوت Console.log(math.abs(4)) 4

Console.log(math.abs(-4)) 4

الجذر التربيعى

Console.log(math.sqrt(9)) 3

الاس

Console.log(math.power(3,3))	27
التقريب راوند	
Console.log(math.round(10.4))	10
Console.log(math.round(10.7))	11
الثيل تقرب لاقرب رقم فوق	
Console.log(math.ceil(10.1))	11
الفلوور تقرب لاقرب رقم تحت	
Console.log(math.floor (10.8))	10
Console.log(math.min(1,-4,6 ,10.4))	-4
Console.log(math.max(100,6,77,99,34))	100

تقطيع النصوص

Repeat

Length

Access

Search

Slice

Split

Let name = 'ali' ;

Console.log(name.repeat(100))

هیکرره 100 مرة

Let name = 'ali' ;

Console.log(name.length)  3

هیدینی 3 لان کلمة علی 3 حروف و لو
سبت سبیس بعد علی هیدینی 4

```
Let name = 100 ;
```

```
Console.log(name.length);
```


Undefined



```
Let name = 'ali'
```


```
Console.log(name[0]);
```

a



```
Console.log(name[1]);
```

L



```
Let name = 'ali' ;
```


```
Console.log(name.charAt(2) );
```



i


```
let name = 'ali' ;
```

```
Console.log(name.charAt(10) );
```



مش هيدي اي حاجة

```
Console.log(name[10]);
```



Undefined

الفرق فى انى لو هعمل برنامج عايز يدينى
Undefined يبقى استخدم [] ولو عايزة يدينى
فاضى يبقى استخدم charAt

Let name = 'I love java script';

Console.log(name.indexOf('o'));

3



Console.log(name.indexOf('a'));

8 هنا هي جيلي حرف آ



Console.log(name.indexOf('a', 9));

10



هنا هي ديني اول حرف آ بعد الحرف التاسع

Console.log(name.indexOf('java'));

7



هنا يجلي اول حرف فالكلمة فقط

والبحث يبدأ من الشمال

ولو عايزة يبدأ البحث من اليمين يبقى استخدم
بس الترتيب من الشمال برضك

```
Console.log(name.LastIndexOf('i'));
```

```
Let name = 'ali gamal ';
```

```
Console.log(name.slice(0,3) );
```

هنا هي قطع الكلام من و هي جيلي اول 3 حقول
لان $3=0-3$ و هكذا يعنى الحقل الرابع و هو
3 مش هيظهر

```
Let name = 'ali gamal hsan karem ';
```

```
Console.log(name.split(' ',3) );
```

هنا هي قطع على السبيس 3 مرات

```
Console.log(name.split('a',2) );
```

هیشیلی حرف ال آ مرتین و مش هیجیب ای
حاجة بعد حرف ال آ التانی

```
Console.log(name.slice(0,3) );
```

هنا هیجیلی اول 3 حروف

```
Console.log(name.substring(0,3) );
```

هنا هیجیلی اول 3 حروف برضك

```
Console.log(name.substr(0,3) );
```

هنا هیجیلی اول 4 حروف

Tutorial

```
Let name = 'I love java script';
```

```
Console.log(name.includes(""));
```

True or false



True لو موجود

لو مش موجود False

Let name = 'I love java script';

Console.log(name.indexOf('z'));

-1 - عشان مش موجود



Let name = 'I love java script';

Console.log(name.includes('0',4));

False



Let name = 'I love java script';

Console.log(name.startsWith('i',2));

False



Console.log(name.endsWith('t'));



True

```
Console.log(name.endsWith('e',6));
```

true ملحوظة هنا ال Lenth مش الستارت

يعنى يعد من ال 1 مش من ال 0 زى ال ستارت

Data type : {ARRAY}

```
LET NAMES = ['ahmed','ali','omar']
```

```
Console.log(names);
```

هيجبلى المتغيرات كلها بس مش كل متغير

```
Console.log(names[1]);    ali
```

```
Console.log(names [0]);   ahmed
```

```
Console.log(names.length); 3
```


```
Let objects = [1,2,3[4,5,6] ]
```

```
Console.log(objects[3]);
```



[4, 5, 6]

```
Console.log(objects[1[ ]3]);
```



5

```
Let objects = [1, 2, 3[4, 5, 6[7, 8, 9] ] ]
```

```
Console.log(objects[1[ ]3] [3]);
```



8

```
Let names = ['ali' 'karem' 'tamer'];
```

```
Console.log(names);
```

هـيـظـهـر لى الـاسـمـاء

طـيـب لـو عـايـز اـعـدـل فـالـاسـمـاء اـعـمـل اـيـة ؟

الحل

Names [0] = 'mohammed'

Names [1]= 'hady'

Names [2] = 'abdo'

Console.log(names);

اضافة عناصر فى الاراى

Let names = ['ali', 'karem', 'tamer'];

Names.push('gamal')

كدة هيضيف العنصر فالآخر

Console.log(names);

Let names = ['ali' , 'karem' , 'tamer'];

```
Names.unshift('gamal','mai');
```

كدة هيضيف العنصر او العناصر فالاول

```
Console.log(names);
```

مسح العناصر

```
Let names = ['ali', 'karem', 'tamer'];
```

```
Names.shift();
```

```
Console.log(names);
```

كدة هيمسحلى اول عنصر من الاول

```
Let names = ['ali', 'karem', 'tamer'];
```

```
Names.shift();
```

```
Names.shift();
```

```
Console.log(names);
```

كدة هيمسحلى اول عنصرين من الاول

```
Let names = ['ali', 'karem', 'tamer'];  
Console.log(names.shift() );
```

كدة هيمسحلى اول عنصر من الاول من الاراى
لكن هيحفظ بيه

```
Let names = ['ali', 'karem', 'tamer'];  
names.pop()  
Console.log(names);
```

كدة هيمسحلى اخر عنصر وهو تامر

```
Let names = ['ali', 'karem', 'tamer'];  
Console.log(names.pop() );
```

كدة هيمسحلى اخر عنصر فى الاراى لكن
هيحفظ بيه

Splice

```
Let names = ['ali', 'karem', 'tamer'];  
Names.splice (0, 1);  
Console.log(names);
```

كدة هيحذفلى العنصر رقم 0 وعنصر واحد بس

```
Let names = ['ali', 'karem', 'tamer'];  
Names.splice (1, 2);  
Console.log(names);
```

طدة هيحذفلى العنصر الثانى و التالت لانى
قولتله يحذف عنصرين من العنصر رقم 1

```
Let names = ['ali', 'karem', 'tamer'];  
Names.splice(0,1,'mona','haidy');  
Console.log(names);
```

كدة هيحذفلى العنصر الاول و يضيف منى و
هايدى من الاول

```
Let names = ['ali', 'karem', 'tamer'];  
Names.splice(0,0,'mona','haidy');  
Console.log(names);
```

كدة مش هيحذف اى حاجة و هيضيف منى و
هايدى من الاول

```
Let names = ['ali', 'karem', 'tamer'];  
Names.splice(1,0,'mona','haidy');  
Console.log(names);
```

كدة هيضيفهم فى لاندكس 1 بعد على دون حذف

Slice

```
Let names = ['ali', 'karem', 'tamer'];
```

```
Names.slice(0,2);
```

```
Console.log(names);
```

كدة مفيش حاجة هتحصل عشان هى بتاخذ
نسخة منها و مش هتأثر فى الاراى

```
Let names = ['ali','karem', 'tamer'];
```

```
Console.log(Names.slice(0,2) );
```

كدة هتظهر فى الكونسول و هيقتطع من الاندكس
0 لحد الاندكس واحد $2=0-2$

يعنى هيجبلى الاندكس 0 و 1

```
Let names = ['ali','karem', 'tamer'];
```

```
Console.log(Names.slice(-3,-1) );
```

كدة هيقتطع اخر عنصرين لان $2=1+3-$

Search

```
Let names = ['ali', 'karem', 'tamer'];  
Console.log(names.indexOf(ali)); 0  
Console.log(names.indexOf(ali, 1));  
Start _____ -1
```

هیدینی 1- عشان بعد الاندکس 1 مفیش علی

```
Let names = ['ali', 'karem', 'ali'];  
Console.log(names.indexOf(ali, 1));  
2
```

```
Let names = ['ali', 'karem', 'ali'];  
Console.log(names.lastIndexOf(ali));  
2
```

هیدا البحث من اليمين

```
Let names = ['ali', 'karem', 'ali'];  
Console.log(Names.lastIndexOf(ali,  
1)); ➡ 0
```

```
Let names = ['ali', 'karem', 'ali'];  
Console.log(Names.lastIndexOf(ali,-  
1)); ➡ 2
```

```
Let names = ['ali', 'karem', 'ali'];  
Console.log(Names.includes('ali'));  
True
```

```
Let names = ['ali', 'karem', 'samy'];  
Console.log(Names.includes('ali',1))  
False
```

هیدینی False عشان قتلته یبدأ بحث من

الاندكس 1

Reverse

```
Let names = ['ali', 'karem', 'samy'];
```

```
Console.log(Names.reverse() );
```

هيعكس الترتيب هيبقى سامى الاول بعد كدة
الباقى

```
Let names = ['ali', 'karem', 'samy'];
```

```
Console.log(Names.sort() );
```

هيرتبهم ابجديا ولو ارقام هيرتبهم تصاعديا

الدمج

```
Let arr1= ['ali','karem','tamer'];
```

```
Let arr2 = ['hady','mona','omar'];
```

```
Arr1 = arr1 + arr2 ;
```

```
Console.log(arr1)
```

هنا فى مكلة هيدمج ما بينهم بس هيعملهم
سترينج

```
Arr1 +=arr2
```

هنا هيعمل نفس المشكلة مش هيدمج اراى
والحل اهو

```
Let arr1= ['ali','karem','tamer'];
```



```
Let arr2 = ['hady', 'mona', 'omar'];  
Console.log(Arr1.concat(arr2));
```

هیدمجهم کارای

وممكن كمان اضيف ارای ثانية وتالته او
سترينج كمان

```
Let arr1= ['ali', 'karem', 'tamer'];  
Let arr2 = ['hady', 'mona', 'omar'];  
Console.log(Arr1.concat(arr2, 'hoda  
'));
```

طیب لو مش عایز ال Square brackets

تتعرض اعمل اية؟! الحل

```
Console.log(Arr1.join() );
```

ممكن كمان اضيف ای حاجة بين العناصر

من خلال ال Join مثال


```
Console.log(Arr1.join('-') );
```

```
Console.log(Arr1.join('#') );
```

المقارنة


```
Let x = 'ahmed';
```

```
Let y = 'ahmed';
```

```
Console.log(x == y);  true
```


```
Let x = 'ahmed';
```

```
Let y = 'ali';
```

```
Console.log(x == y);  false
```

Let x = 'ahmed';

Let y = 'ali';

Console.log(x = y);  ali

Let askemail = prompt('enter your email');

Let email =

'mohammed@gmail.com' ;

Console.log(askemail == email);

دە معناه هل البيانات ال هيكتبها فى

ال Prompt مطابقة للايميل و لا لا

Let x = 'ahmed' ;

Console.log(x.toUpperCase()); 

ahmed

```
Console.log(x.toLowerCase());
```

AHMED



```
Let askemail = prompt('enter your  
email');
```

```
Let email =
```

```
'mohammed@gmail.com' ;
```

```
Console.log(askemail.toLowerCase(  
) == email);
```

كدة اى بيانات هتدخل فى البرومبت هتبقى
حروف سموول بس

طيب لو فى مسافات مثلا اعمل اية ؟!

```
Console.log(askemail.toLowerCase()  
)trim() == email);
```

كدة هيحذفلى المسافات

مثال اخر

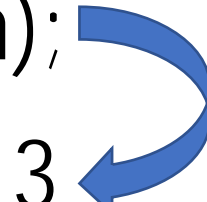
```
Let x = ' ali ' ;
```

```
Console.log(x.length);
```



6

```
Console.log(x.trim().length);
```




3

كدة هيديني 3 لانه حذف المسافات لكن لو في
مسافة وسط الكلام مش هيحذفها لانه بيحذف
المسافات ال في اخر الكلام و اوله فقط
مثال


Let x = 'ali ali' ;

Console.log(x.trim().length);




7


المقارنة جزء 2


Console.log(5 == '5')  true


لية؟ لانها بتقارن قيم مش نوع القيم

Console.log(5 === '5')  false


هنا بتقارن نوع قيم دة number و دة String

Console.log(5 > 6)  false

Console.log(4 < 3)  false

Console.log(4 <= 4)  true

و عندی کمان حاجة تانية ممکن اسال بیها

Console.log(4 != 4)  false

هنا دة معناه هل القيمة دی لا تساوى القيمة دی

مثال عملی


Let product = 'TV-Samsung';

Let size = 50;

Let salary = 10000;

Console.log(salary > 10000); false

Console.log(salary >= 10000); true


Console.log(salary > 12000 && size
== 50);  true

&&يعنىand ||يعنىor

Let product = 'TV-Samsung';

Let size = 50;

Let salary = 10000;

Console.log(salary > 12000 || size
== 40);  true

الجمل الشرطية

اداة الشرط if

الشرط بين الاقواس ()

نتيجة الشرط بين الكيرلي بريسس {}

مثال

```
Let role = prompt (your role?);
```

```
If (role == 'admin')
```

```
{
```

```
Document.write('search,create');
```

```
}
```

```
Else if (role == 'moderator');
```

```
{
```

```
Document.write('update,create');
```

```
}
```

```
Else
```

```
{ Document.write('hello user');
```

```
}
```

مثال اخر

```
Let result = prompt('your result?');
```

```
If (result >= 90)
```

```
{
```

```
Document.write('excellent');
```

```
}
```

```
else If (result >= 80)
```

```
{
```

```
Document.write('very good');
```

```
}
```

```
Else If (result >= 70)
```

```
{  
Document.write('good');  
}  
Else If (result >= 50)  
{  
Document.write('accepatable');  
}  
Else  
{  
Document.write('fail')  
}
```

طريقة ثانية لكتابة الشرط

Age = prompt ('your age ?');

Age > 18 ?

Document.write('hello user')

:age == 18?

Document.write('hello')

: Document.write('hello');

لازم احط السيميكلون فى نهاية الشرط

يدل على انتهاء الشرط

Let age = prompt ('your age ?');

Let result = age > 18? 'hello'

:Age == 18? 'you are 18 now'

: 'no you are very young';

Console.log(result)

السويتش

تقريبا زي ال If

مفيهاش اكبر من ولا اصغر من

لكن لو في جوا الإف عندى إف تانية يبقى

لازم استخدم السويتش هتوفر عليا

Let role = prompt(your age?)

Wswitch (role)

{

Case 'admin' :

Document.write(create,update);

Break;

Case 'moderator':

```
Document.write(update);  
Break;  
Default:  
Document.write(hello user);  
}
```

Loop

```
For (let m = 0 ; m <=10 ; m +=1)  
    Start    condition    steps  
{  
    Console.log(m)  
}
```

مثال اخر

```
For (let m = 10 ; m >=0 ; m -= 2)
```



```
{  
                                end  
Console.log(m)  
}
```

مثال عملی علی ال loop

```
Let names = ['ahmed','mai','ali'];  
For (let m = 0 ; m <= 3 ; m++)  
{  
Console.log(names[m]);  
}
```

مثال آخر لو انا عندی عدد عناصر معروفش
مثلا

```
Let names = ['ahmed','mai','ali'];
```

```
For (let m = 0 ; m < names.length ;  
m++)
```

```
{
```

```
Console.log(names[m]);
```

```
}
```

```
Let x = 'ahmed';
```

```
For(m = 0 ; m < x.length ; m++)
```

```
{
```

```
Console.log(x[m]); a h m e d
```

```
}
```

طيب لو عايز اعمل عرض لمنتجات السيارات

ال عندى

الانواع

الموديل الالوان

مثال

```
Let cars =  
['bmw','mercedse','passat'];  
Let model = ['2019','2020','2023'];  
Let colors = ['red','white','blue'];  
For (let k = 0 ; k < car.length ; k++)  
{
```

```
Console.log('car: ${cars[k]}');  
For (let j = 0 ; j < model.length ; j++)  
{  
  Console.log('model: '${model[j]}');  
  {  
    For (let z = 0 ; z < colors.length ;  
    z++)  
    {  
      Console.log('color: ${colors[z]}')  
    }  
  }  
  Console.log('_____');  
}
```

Continue loop

مثال لو عايز استبعد الارقام بس

Let users =

['ali','ola',1,2,6,7,'hoda'];

```
For(let i = 0 , i<users.length, i++ )  
{  
  If(typeof users[i] == 'number')  
  {  
    Continue;  
  }  
  Console.log(users[i])  
}
```

مثال اخر لو عايز استبعد 'ali'

```
Let users =  
['ali','ola',1,2,6,7,'hoda'];  
For(let i = 0 , i<users.length, i++ )
```

```
{  
  if( users[i] == 'ali')  
  {  
    Continue;  
  }  
  Console.log(users[i])  
}
```

مثال آخر لو عایز استبعد ای عنصر
لا یساوی 'ali'

```
Let users =  
['ali','ola',1,2,6,7,'hoda'];  
For(let i = 0 , i<users.length, i++ )  
{  
  If( users[i] != 'ali')  
    {  
      Continue;  
    }  
  Console.log(users[i])  
}
```

Break loop

مثال لو عایز افرمل عند ال قبل هدی

```
Let users =  
['ali','ola',1,2,6,7,'hoda'];  
For(let i = 0 , i<users.length, i++ )  
{  
  If( users[i] == 'hoda')  
  {  
    break;  
  }  
  Console.log(users[i])  
}
```

مثال لو عايز افرمل عن هدى بالضبط

```
Let users =  
['ali','ola',1,2,6,7,'hoda'];  
For(let i = 0 , i<users.length, i++ )  
{  
Console.log(users[i])
```

```
    If( users[i] == 'hoda')  
        {  
            break;  
        }  
}
```


While loop

Let $z = 0$;

While ($z < 3$)

```
{  
Document.write('hello world')  
Z++  
}
```

كدة هينفذلى الامر طالما الشرط متحقق
فهيعرض كلمة 'hello world' 3مرات

Dowhile loop

```
Let z = 0 ;
```

```
do
```

```
{
```

```
Document.write('hello world');
```

```
z++;
```

```
}while(z < 3);
```

هنا الشرط متحقق فسينفذ 3 مرات

مثال اخر

```
Let z = 0 ;
```

```
do
```

```
{
```

```
Document.write('hello world');
```

```
z++;
```

```
}while(false);
```

هنا الشرط مش متحقق فهينفذه مرة واحدة بس

Function

Function pro (price , tax , ads)

{

Let product = price + tax ;

Let result = product + ads ;

Console.log(result);

} pro(200 , 2.5 , 10)

مثال لو عايز احسب سعر منتج بالضريبة
و الاعلانات

```
function pro(price)
{
    let product = price ;
    let tax = price * 0.07
    let ads = price + 10
    let result = product + tax + ads ;

    console.log(result);
}
```

```
}  
pro(1000);
```

return

```
function days (age)  
{  
  let result = age * 365 ;  
  return result ;  
}  
let day = days (25);  
console.log(day);  
  
function calcAgeByHours (day)  
{  
  let result = day *24 ;  
  return result;  
}  
let x = calcAgeByHours(day);  
console.log(x);
```

HOISTING

لو انا كاتب كود بالشكل دة

```
console.log('hello');  
console.log('hello');  
console.log('hello');  
var x = 10 ;  
var z = 20 ;  
var m = 40 ;
```

الجافا سكريبت هتقراه بالشكل دة

```
var x ;  
var z ;  
var m ;  
console.log('hello');  
console.log('hello');  
console.log('hello');  
x = 10 ;  
z = 20 ;  
m = 40 ;
```

باختصار الجافا سكريبت بتعمله

بتعمله Hoisting

طيب لو انا مش عايز كدة الحل اية

الحل هو ال Let

مش بيتعملها Hoisting

مثال اخر للتوضيح الفرق بينهم

```
console.log(x)  
let x = 10 ;
```

كدة هيديك ايرور فوشك علطول

```
console.log(x)  
var x = 20 ;
```

كدة هيديك undefined

ونيجي للفانكشن دلوقت الديكلريشن

لو انا كاتبه بالشكل دة

```
hello()

function hello()
{
  console.log('hello')
}
```

کده انا ناديت عليها قبل ما اعملها فانكشن
بس هو هيقراها كده

```
function hello()
{
  console.log('hello')
}
hello()
```

اما ال expression

فلأ مش بيتعملها HOISTING

ملحوظة :

اي فانكشن فالدنيا بيبدأ ب Function

تبقى Declaration

ای فانکشن پیدا بای حاجة غیر Function

تبقى Expression

ملحوظة لو الفانکشن باداً باقواس (يبقى

Expression

Scope and self invoked

المتغيرات و الفانکشن عندنا بتكون Global

یا اما Local

مثال على المتغيرات و الفانکشن ال Local

```
function hello(){  
    local ای متغيرات او فانکشن بتکتب جوا هنا تبقى  
}  
hello();
```

اما المتغيرات و الفانكشن ال Global
بنتكتب في اى مكان تانى برا الكيرلى بريس
مثال للتوضيح

```
function hello(){  
  
    var x = 10 ;  
}  
hello ()  
console.log(x);
```

كدة هيديك ايرور لانه ما شافش ال 10
لكن لو كتبتها بالشكل دة

```
function hello(){  
  
    var x = 10 ;  
    console.log(x);  
}  
hello ()
```

كدة هو قراها (ال 10)

دى كلها بلوك اوف كود

```
function hello()  
if () {}  
for () {}  
while (condition) {  
  
}
```

فى الفنكشن فقط ممكن استخدم ال

Local وال global

مثال للتوضيح

```
var x = 10 ;  
function hello ()  
{  
    console.log(x)  
}  
hello()
```

كدة هيطبعها عادى يا معلم طب لية ؟

عشان ال Var جلوبال (متشافة فكل حته)

مثال اخر بالليت

```
let x = 20 ;  
function hello ()  
{  
  
    console.log(x)  
}  
hello ()
```

هيطبعتها يا معلم

مثال اخر عاللويب

```
for (var i = 0 ; i < 10 ; i++)  
{  
  
}  
console.log(i)
```

هيطبعلك عادى

عشان الفار جلوبال متشاف فكل حنة

طيب لو استخدمت ال Let

مثال

```
for ( let i = 0 ; i < 10 ; i++ )  
{  
  
}  
console.log(i)
```

كدة هيديلك ايرور فوشك علطول طيب لية ؟

لان ال Let لوكال يا برنس

فالخلاصة : فى الفنكشن ممكن استخدم

اللوكال و الجلوبال variables

لكن فى الباقي لازم استخدم الليت كلوكال

او الفار كجلوبال

القيم الافتراضية Arguments
دلوقت لو عملت فنكشن بالشكل دة
و ناديت عليه

```
function hello ( name = 'ahmed' )  
{  
  console.log(`hello ${name}`);  
}  
hello()
```

كدة هيطبعلى "hello ahmed"
لكن لو دخلت قيمة تانية فعلية

Argument لل

مثال

```
function hello ( name = 'ahmed' )  
{  
  console.log(`hello ${name}`);  
}  
hello('mohammed')
```

كدة هيطبعلى

'hello mohammed'

مثال اخر لو عايز اعمل ارأى

يبقى اعمل ...numbers

```
function calc ( ...numbers)
{
    let result = 0 ;
    for( let i = 0 ; i < numbers.length
; i++ )
    {
        result += numbers[i];
    }
    console.log(result);
}
calc(9, 333, 44, 6777, 7654, 44);
```

Arrow


```
let x = () => 1 ;
```

دی بتساوی

```
let x = function ()  
{  
  return 1;  
}  
console.log(x())
```

ملحوظة ای حاجة بعد السهم هی ال return

بشرط يكون عندی line واحد بس

ولو معندیش بارامیتر

يبقى اكتبها بالشكل دة

```
let x = _ => 1;
```

ولو عندی بارامیتر يبقى اكتبها بالشكل دة

```
let x = (num) => num*5;  
console.log(x(7))
```

ولو عندی اكثر من بارامیتر يبقى اكتبها كدة

```
let x = (num, m) => num*m;  
console.log(x(5, 6))
```

object

دە مثال علاوبجكت

```
let car =  
{  
  title : 'BMW',  
  price : 600000,  
  color : ['white', 'red', 'black'],  
  model : '2023',  
  hello: function()  
  {  
    return 'hello world';  
  }  
}  
console.log(car);
```

كدة هيجبلى كل بيانات الاوبجكت

طيب لو عايز اجيب بيانات معينة جوا
الاوبجكت

Console.log (car.color);

Console.log (car.model);
Nested object

مثال

```
let x = 3;  
let user = {  
  firstname: 'mohammed',  
  lastname : 'barakat',  
  email : 'test@gmail.com',  
  age : x ,  
  
  getage: function() {  
    if (user.age >=18 )  
    {  
      return 'available';  
    }  
    else {  
      return 'unavailable';  
    }  
  },  
}
```

```
};  
console.log(user.getage());  
console.log(user['getage']());
```

Create object

لو عايز اضيف اوبجكت من غير داتا

واضيف الداتا من برا هل

ينفع ؟ اة ينفع

مثال بال Dot notation

```
let user = {} ;  
user.name = 'ahmed' ;
```

دى هي هياها الكود دة

```
let user = {  
  name : 'ali'  
} ;
```

وممكن كمان اضيفها

بال Bracket notation

```
let user = {} ;
```

```
user['age'] = 25  
console.log(user.age)
```

ودة مثال لاضافة method

```
let user = {} ;  
user.hello = function()  
{  
  
    return 'hello';  
}  
console.log(user.hello())
```

this Keyword

بتشير للمكان ال تنتمى اليه
فقيمتها حسب مكان استخدامها
مثال

```
let user = {  
  
age : 25 ,
```

```
getage : function ()  
{  
    return this.age  
}  
  
console.log(user.getage())
```

كدة هى بتشير للاوبجكت التتنمى اليه

هى هى كانى كتبت user.age

طيب لو مش كاتبها جوا اوبجت

مثال

```
console.log(this)
```

هتطبعك ال window

ولو عايز تتأكد اكرر

```
console.log(this == window)
```

هيطبعك true يا معلم

وصلت ؟

طيب تعالى نكتبها في فنكشن عادية جداً

```
function hello()  
{  
    return this ;  
}  
console.log( hello () );
```

هل في هنا اوبجكت ؟ لا

فهترجلك Window علطول

استخدام ال this مع ال Use strict

```
'use strict';  
function hello()  
{  
    return this ;  
}  
console.log(hello())
```

هيطبعلك Undefined

Object-create

دلوقت لو انا عندي اوبجكت و عايز

انسخ البيانات ال فيه ل اوبجت تاني

يبقى استخدم ال Object.create

مثال

```
let user1 = {  
  name : 'mohammed',  
  getname : function()  
  {  
    return `hello ${user1.name}`;  
  }  
}  
  
let user2 = Object.create(user1);  
console.log(user2.getname())
```

كدة هيطبعلى hello mohammed

طيب لو انا عايز اعدل عاليانات

عايز اغير الاسم مثلا فى
User2 فى اللابجكت
مثال

```
let user1 = {  
  name : 'mohammed',  
  getName : function()  
  {  
    return `hello  
    ${user1.name}`;  
  }  
}  
  
let user2 = Object.create(user1);  
user2.name = 'ali'  
console.log(user2.name)
```

كدة هيطبعلى ali

لكن اللابجكت الاول هيظل محتفظ
بقيمه mohammed

واقدر كمان اضيف Property جديد

مثال

```
let user1 = {  
  name : 'mohammed',  
  getname : function()  
  {  
    return `hello  
    ${user1.name}`;  
  }  
}  
  
let user2 = Object.create(user1);  
user2.age = 25;  
console.log(user2.age)
```

هيطبعلك ال age بكل بساطة

طيب كدة انا بضيف و بعدل من برا

ال Create

لكن لو عايز اضيف من جوا

مثال

```
let user1 = {  
  name : 'mohammed',  
  getname : function()  
  {  
    return `hello  
    ${user1.name}`;  
  }  
}  
  
let user2 = Object.create(user1, {  
  age: {value: 25}  
});  
console.log(user2.age)
```

هيطبعك ال age عادى جدا

طيب لو انا دلوقت عايزه يطبعلى

ال Getname للاوبجكت الثانى

بعد ما اعدل اسمه

مثال

```
let user1 = {  
  name : 'mohammed',  
  getname : function()  
  {  
    return `hello  
    ${user1.name}`;  
  }  
}  
  
let user2 = Object.create(user1);  
user2.name = 'ali';  
console.log(user2.getname())
```

كدة هيطبعلى

Hello mohammed

طيب انا عايزه يطبعلى Hello ali

الحل بمثال

```
let user1 = {  
  name : 'mohammed',  
  getName : function()  
  {  
    return `hello ${this.name}`;  
  }  
}  
  
let user2 = Object.create(user1);  
user2.name = 'ali';  
console.log(user2.getName())
```

هنا انا استخدمت ال this

لان وانا بنادي عالابجكت الاول

هتعني User1

وانا بنادي عالابجكت الثاني

هتعني User2

يعنى بتخليك الموضوع ديناميك

Assign object

دمج الاوبجكت

لو انا عندي كذا اوبجكت و عايز ادمجهم في

اوبجكت واحد

مثال

```
let a1 =  
{  
  num1: 1  
}  
let a2 =  
{  
  num2: 2  
}  
let a3 =  
{  
  num3: 3  
}
```

```
}  
let a4 = Object.assign(a1, a2, a3)  
console.log(a4)
```

ولو عاوز اضيف اوبجكت جديد

من جوال Assign

مثال

```
let a1 =  
{  
  num1: 1  
}  
let a2 =  
{  
  num2: 2  
}  
let a3 =  
{  
  num3: 3  
}  
let a4 =  
Object.assign(a1, a2, a3, {num: 4})
```

```
console.log(a4)
```

ممکن کمان اعدل بیانات جوا الاوبجکت

الجديد

مثال

```
let a1 =  
{  
  num1: 1  
}  
let a2 =  
{  
  num2: 2  
}  
let a3 =  
{  
  num3: 3  
}  
let a4 = Object.assign(a1, a2, a3, {num: 4})  
a4.num1 = 10;  
console.log(a4)
```

قيمتها كانت 1

ودلوقت بقت 10

اول درس فى ال

Dom

Document object model

اول ما تسمع كلمة Dom

تعرف انك هتتعامل مع عناصر

ال HTML

عشان تجيب داتا او تغيرها او تعمل فيها اى
حاجة

اول حاجة الوصول للعنصر من خلال ال Id
مثال

```
<body>
```

```
<h1 id="head">Hello</h1>
```

```
<script src="main.js"></script>
```

```
</body>  
</html>
```

```
let head = document.getElementById('head');  
console.log(head)
```

الوصول للعنصر من خلال class مثال

```
<body>  
  
<h1 class="head">Hello</h1>  
<h1 class="head">Hello</h1>  
<h1 class="head">Hello</h1>  
  
<script src="main.js"></script>  
</body>  
</html>
```

```
let head =  
document.getElementsByClassName('head');  
console.log(head)
```

كدة هيجبلى كل العناصر

طيب لو عايز امسك عنصر معين وليكن الاول مثال

```
<body>  
  
<h1 class="head" >Hello</h1>  
<h1 class="head" >Hello</h1>  
<h1 class="head" >Hello</h1>  
  
<script src="main.js"></script>  
</body>  
</html>
```

```
let head =  
document.getElementsByClassName('head')[0];  
console.log(head)
```

الوصول لعنصر من خلال ال Tag name

مثال

```
<body>

<h1 class="head" >Hello</h1>
<h1 class="head" >Hello</h1>
<h1 class="head" >Hello</h1>

<script src="main.js"></script>
</body>
</html>
```

```
let head =
document.getElementsByTagName('h1')
console.log(head)
```

كدة هيجبلى كل عناصر

الH1

طيب لو عايز امسك العنصر الاول ؟

مش دة Array يا معلم !؟

يبقى استخدم الاندكس

مثال

```
<body>
```

```
<h1 class="head" >Hello</h1>  
<h1 class="head" >Hello</h1>  
<h1 class="head" >Hello</h1>  
  
<script src="main.js"></script>  
</body>  
</html>
```

```
let head =  
document.getElementsByTagName('h1')[0]  
  
console.log(head)
```

كدة انا جبت العنصر الاول
من خلال الاندكس وهكذا يا برنس

وفى طريقة كمان عشان اوصل للعنصر

سواء من خلال ال Id

او ال Class

Tag name ال

مثال

```
<body>

<h1 class="head" >Hello</h1>
<h1 id="head" >Hello</h1>
<h1 class="head" >Hello</h1>

<script src="main.js"></script>
</body>
</html>
```

عشان اجيبه من خلال ال Class

يبقى اكتب . واسم الكلاس بعدها

```
let head = document.querySelector('.head')
console.log(head)
```

وعشان اجيبه من خلال ال id

يبقى اكتب # واسم

ال Id

```
let head = document.querySelector('#head')  
console.log(head)
```

وعشان اجيبه من خلال ال tag name

يبقى اكتب ال Tag name

```
let head = document.querySelector('h1')  
console.log(head)
```

ولكن ليها عيب انها هتجلى اول عنصر بس

سواء ناديت عليه من خلال

ال Class

او Id

او Tag name

ودة طبعا لو كلهم واخدين نفس الاسم

طيب والحل ؟!

عندى ال Queryselectorall

وطبعاً اقدر اجيب اى عنصر من خلال الاندكس

مثال

```
<body>

<h1 class="head" >Hello</h1>
<h1 class="head" >Hello</h1>
<h1 class="head" >Hello</h1>

<script src="main.js"></script>
</body>
</html>
```

```
let head =
document.querySelectorAll('h1')[1]
console.log(head)
```

كدة انا جيت العنصر التانى واقدر اعدل فيه
براحتى خالص

ممکن کمان اغير العنوان

مثال


```
let title = document.title ;  
title = 'hello!';  
console.log( title )
```

كدة غيرلى العنوان خلهولى hello!

ولول عندى صور يا معلم اقدر اخليه يجبهالى
مثال

```
let images = document.images;  
console.log(images)
```

ولو عايز امسك اى صورة واعدل فيها او
اضبطها يبقى من خلال الاندكس
مثال

```
let images = document.images[0];  
console.log(images)
```

كدة انا جيت اول صورة واقدر اعمل لنا عايزه
واقدر كمان اوصل لكل لينكات الصفحة

مثال

```
<body>

<a href="https://youtube.com/">youtube </a>
<a
href="https://www.facebook.com/">facebook</
a>
<a
href="https://www.google.com/">google</a>

<script src="main.js"></script>
</body>
</html>
```

```
let links = document.links;
console.log( links )
```

كدة هيوصلنى لكل اللينكات
طيب لو عايز اوصل للينك معين؟!
يبقى من خلال الاندكس برضك

مثال

```
let links = document.links[1];  
console.log( links )
```

كدة هيجباك تانى عنصر يا معلم

ولو عاوز اوصل لل Href للعنصر التانى مثلا

مثال

```
let links = document.links[0];  
console.log( links.href )
```

كدة هيجبلى ال href للعنصر الاول

طيب لو انا عايز اعدل

ال Href

مثال

```
let links = document.links[0];  
links.href = 'http://www.google.com/';  
console.log( links.href )
```

كدة هو خلى اللينك الاول يجبلى جوجل

تانی دروس ال

DOM

Attributes

```

```

دلوقت انا عندي 3 اتریبوتس

اقدر اوصل للعنصر من خلالهم

مثال

```
let img = document.getElementById('img');  
console.log(img.alt)
```

كدة هيجبلى ال alt بتاعه

ال هو Alt1

وممكن كمان اقولع اعرضلى ال Src

مثال

وممكن كمان اقلوله اعرضلى ال Id

مثال

```
let img = document.getElementById('img');  
console.log(img.id)
```

كدة هيعر ضلك اسم

ال Id

ولو فى كلاس

مثال

```

```

```
let img = document.getElementById('img');  
console.log(img.className)
```

كدة هيجبلك اسم

ال Class

ال هو Test

كل الفات دة عبارة عن عرض البيانات

ال Attributes

بالطريقة العادية

اما لو عاوز تعدل البيانات دى

يبقى برا ال Console

مثال

```
let img = document.getElementById('img');  
img.src = '0.jpg'
```

```
img.className = 'test2'  
console.log(img.className)
```

كدة هو غير ال

Src

وغير ال Class name كمان

ولو عايز ازود ال attribute مثلا

مثال

```
let img = document.getElementById('img');  
img.title = 'image';  
console.log(img.title)
```

كدة انا زودت العنوان

وحصلت عليه كمان

دلوقت لو عايز اتأكد العنصر فيه

Attributes

ولا لا؟ احنا عندنا حل

مثال

```
<body>  
  
  
  
<script src="main.js"></script>  
</body>
```



```
let img = document.getElementById('img');  
console.log(img.hasAttributes())
```

هيطبعك true

يا معلم

ولو عايز اتأكد من Attribute

معين ؟

مثال

```
<body>  
  
  
  
<script src="main.js"></script>  
</body>
```

```
let img = document.getElementById('img');  
console.log(img.hasAttribute('src'))
```

هيقولك موجود (true)

واقدر كمان اعرض كل

Attributes ال

الموجودة فى العنصر

مثال

```
<body>
```

```

```

```
<script src="main.js"></script>
```

```
let img = document.getElementById('img');
```

```
console.log(img.attributes)
```

كدة هيطبعلك كل ال

Attributes

الموجودة

وممكن اوصل لاول Attribute

مثال

```
<body>
```

```

```

```
<script src="main.js"></script>
```

```
let img = document.getElementById('img');
```

```
console.log(img.attributes[0])
```

واقدر كمان اعدل attribute

مثال

```
<body>
```

```

```

```
<script src="main.js"></script>
```

```
let img = document.getElementById('img');  
img.setAttribute('alt', 'mohsen')  
console.log(img.alt)
```

كدة هو غيرلى ال attribute (alt)

من alt1

Mohsen

واقدر كمان اضيف

Attribute

جديدة

مثال

```
let img = document.getElementById('img');  
img.setAttribute('title', 'ahmed')  
console.log(img.title)
```

كدة انا ضفت ال

Title

Ahmed وخليته

كدة اتعلمنا ال get و ال Set

كدة فاضل ال Remove

Remove

مثال

```
<body>
```

```

```

```
<script src="main.js"></script>
```

```
let img = document.getElementById('img');
```

```
img.removeAttribute('alt')
```

```
img.removeAttribute('class')
```

```
console.log(img)
```

كدة هو حذفلى

Alt ال

Class و ال

Inner and outer html

Outer

بيجيب العنصر كله بمحتواه

مثال

```
<div id="container">
```

```
<h2>hello! </h2>
```

```
</div>
```

```
let container =
```

```
document.getElementById('container')
```

```
console.log(container.outerHTML)
```

كدة هيطبعلى العنصرين الموجودين

ال H2 و ال Div

Inner

هيجبلى المحتوى الداخلى العنصر

مثال

```
<div id="container">
```

```
<h2>hello! </h2>
```

```
</div>
```

```
let container =
```

```
document.getElementById('container')
```

```
console.log(container.innerHTML)
```

كدة هو طبعلى المحتوى

الموجود داخل العنصر

ال هو ال H2

Outer

مثال

```
<body>

<div id="container">

<h2>hello! </h2>

</div>

<script src="main.js"></script>
</body>
```

```
let container =
document.getElementById('container')

container.outerHTML = '<p>hello world
</p>'
```

كدة بدلت العنصر بالكامل بالبراجراف

ال برا و ال جوا

Inner

مثال

```
<body>  
  
<div id="container">  
  
<h2>hello! </h2>  
  
</div>  
  
<script src="main.js"></script>  
</body>
```

```
let container =  
document.getElementById('container');  
  
container.innerHTML = '<p> hello world  
</p>';
```

كدة هو غير محتوى ال ديف بالبراجراف

يعنى غير ال جوا

وتقدر كمان تضيف البراجراف و تسيب
ال H2 ودة من خلال ال(+=)
مثال

```
<body>  
  
<div id="container">  
  
<h2>hello! </h2>  
  
</div>  
  
<script src="main.js"></script>  
</body>
```

```
let container =  
document.getElementById('container');  
  
container.innerHTML += ' <p> hello world  
</p>';
```

وبكدة هو اضافلى البراجراف جوا الديف

Sibling and parent

Sibling

بيوصلنى للعنصر التالى او السابق

عشان احذفه او اضيف له او اعدله

مثال

```
<body>

<div id="container">

<p>first</p>
<p id="second">second</p>
<p>third</p>

</div>
<script src="main.js"></script>
</body>
```

```
let element =
document.getElementById('second')
```

```
console.log(element.previousElementSibling)
```

كدة هو جابلى العنصر ال قبله (البراجراف)

مثال اخر عشان اوصل للعنصر التالى

```
<body>

<div id="container">

<p>first</p>
<p id="second">second</p>
<p>third</p>

</div>
<script src="main.js"></script>
</body>
```

```
let element =
document.getElementById('second')

console.log(element.nextElementSibling)
```

كدة هو جابلى العنصر ال بعده

البراجراف

<p> third </p>

واقدر كمان اعدله

مثال

```
<body>

<div id="container">

<p>first</p>
<p id="second">second</p>
<p >third</p>

</div>
<script src="main.js"></script>
</body>
```

```
let element =
document.getElementById('second')
element.nextElementSibling.innerHTML =
'test'
```

كدة هو عدلى العنصر التالى خلاه

"test"

ممکن کمان احذف العنصر

Remove

مثال

```
<body>

<div id="container">

<p>first</p>
<p id="second">second</p>
<p >third</p>

</div>
<script src="main.js"></script>
</body>
```

```
let element =
document.getElementById('second')
element.nextElementSibling.remove()
```

دکة هو حذفلى العنصر ال بعده

ونفس الكلام ينطبق عال
الفرق بين

previousElementSibling

و

previousSibling

ال previousElementSibling

بتجيب العنصر ال قبله

اما ال previousSibling

بتجيب ال قبلها اياً كانت

سواء عنصر او كلمة

نفس الكلام ينطبق على ال Next

دلوقت لو عايزين نوصل للعنصر الأب
ال هو الديف
مثال

```
<body>

<div id="container">

<p>first</p>
<p id="second">second</p>
<p >third</p>

</div>
<script src="main.js"></script>
</body>
```

```
let element =
document.getElementById('second')
console.log(element.parentElement)
```

كدة هو الديف علطول يا معلم

وصلت؟

واقدر احذف الديف كمان

مثال

```
<body>

<div id="container">

<p>first</p>
<p id="second">second</p>
<p>third</p>

</div>
<script src="main.js"></script>
</body>
```

```
let element =
document.getElementById('second')
element.parentElement.remove()
```

كدة هو حذف الديف بكل حاجة

Css style

طرق التعديل عال CSS

مثال

```
<body>
<div id="container">
<p>hello world</p>
</div>
<script src="main.js"></script>
</body>
```

```
let container =
document.getElementById('container')
// element.style.property = value
container.style.backgroundColor = '#444';
container.style.color = '#fff';
container.style.padding = '10px';
container.style.borderLeft = '7px solid
#fa0';
```

و فى طريقة ثانية للتعديل مثال

```
container.style.cssText = `
color: red;
margin: 20px;
background: blue;
padding: 22px;
`
```

حذف ال Property مثال

```
container.style.removeProperty('color')
container.style.removeProperty('background')
)
```

كتابة ال html داخل الجافا سكريبت

مثال

```
// اول خطوة انشاء العنصر وامسكه بمتغير
let container =
document.createElement('div');
let head = document.createElement('h1');
let img = document.createElement('img');
// تانى خطوة اضافة المحتوى
let content =
document.createTextNode('hello world');
head.appendChild(content);
img.src = '1.png';
// ثالث خطوة اضافة العنصر فى المكان الصحيح
container.appendChild(head);
container.appendChild(img);
document.body.appendChild(container)
console.log(container)
```

المحاضرة 54

مشروع ال Cards

```
let names = ['ahmed', 'ali', 'mo', 'barakat']
let ages = ['15 years old', '17 years old', '29 years old', '22 years old']

let container = document.createElement('div')

document.body.appendChild(container);
container.style.textAlign = 'center';
function element(names, ages)
{

    let card = document.createElement('div')
    let title = document.createElement('h2')
    let age = document.createElement('p')
    let img = document.createElement('img')

    let head = document.createTextNode(names)
    let agecontent = document.createTextNode(ages)
    img.src = '1.png'
    title.appendChild(head);
    age.appendChild(agecontent);
// style of card
    card.style.width = '200px';
    img.style.width = '100%';
    card.style.background = '#444'
    card.style.padding = '15px'
    card.style.margin = '5px'
    card.style.color = 'fff'
    card.style.display = 'inline-block'

    card.appendChild(title)
    card.appendChild(age)
```

```

card.appendChild(img)

container.appendChild(card);

}
for ( let i = 0 ; i <= 3 ; i++)
{

    element(names[i], ages[i]);

}

```

Events

بنكتب الحدث فى العنصر نفسه
مثال

```

<body>

    <button
oncl i ck="console.log(' Hi i i i i i i i i ' )" >cl i ck</
button>

<scri pt src="mai n. j s" ></scri pt>
</body>

```

ممکن تكتب اى امر برمجى
كنت بتكتبه فى الجافا سكريبت
طيب الامر ده بيحصل امتى ؟!

بيحصل عند الضغط على العنصر او الزر

تقدر كمان تنادي على اى فانكشن
مثال

```
function hello()  
{  
  
    console.log('hello')  
  
}
```

```
<body>  
  
<button onclick="hello()">click</button>  
  
<script src="main.js"></script>  
</body>
```


كدة انا عملتلها Invoke
عند الضغط على الزر
دى اول طريقة لكتابة الحدث
للعنصر نفسه

تانى طريقة
مثال

```
<body>  
  
  <button id="btn">click</button>  
  
<script src="main.js"></script>  
</body>
```

```
let btn = document.getElementById('btn');  
btn.onclick = function ()  
{
```

```
console.log('hello')  
}
```

ثالث طريقة مثال

```
<body>  
  <button id="btn">click</button>  
  <script src="main.js"></script>  
</body>
```

```
let btn = document.getElementById('btn');  
function hello()  
{
```

```
console.log('hello')  
}  
btn.onclick = hello;
```

لو عايز اغير لون الخلفية عند الضغط عالزر
مثال

```
<body>  
  <button id="btn">click</button>  
  <script src="main.js"></script>  
</body>
```

```
let btn = document.getElementById('btn')  
function hello()
```

```
{  
    document.body.style.background = 'gold'  
}  
btn.onclick = hello;
```

يوجد طريقة اخرى لاضافة الحدث
مثال

```
let btn = document.getElementById('btn')  
btn.addEventListener('click', function(){  
    console.log('clicked')  
})
```

طيب ايه الفرق بين

On event And addEventListener تعالی نشوف بمثال

```
let btn = document.getElementById(' btn' )  
btn.onclick = function(){  
    console.log(' first' )  
}  
  
btn.onclick = function(){  
    console.log(' second' )  
}  
  
btn.onclick = function(){  
    console.log(' thi rd' )  
}
```

كدة لما نضغط عالزر هيطبعلى اخر امر
برمجى فقط

طيب هل دة منطقى؟! اكيد لأ
لكن فى طريقة تانية

بالaddEventListener

مثال

```
let btn = document.getElementById('btn')  
btn.addEventListener('click', function(){  
    document.body.style.background =  
    'gold';  
})  
  
btn.addEventListener('click', function(){  
    btn.style.background = 'red';  
})  
  
btn.addEventListener('click', function(){
```

```
btn.style.col or = ' green' ;  
})
```

ال Event نفذت كل الاوامر

ودة احد الفروق

المحاضرة 57

مشروع لتحويل الجنيه الى دولار و العكس

```
let dollar =  
document.getElementById(' dollar' )  
let pound =  
document.getElementById(' pound' )  
  
dollar.onkeyup = function()  
{  
  
pound.value = dollar.value * 33;  
  
}  
pound.onkeyup = function()  
{  
  
dollar.value = pound.value / 33;
```

```
}
```

محاضرة 58

After , before

```
<body>

<button id="after">after</button>
<button id="before">before</button>
<button id="append">append</button>

<p id="content">content</p>
<div id="container">

</div>

<script src="main.js"></script>
</body>
```



```
let after =
document.getElementById('after')
let before =
document.getElementById('before')
let append =
document.getElementById('append')
let content =
document.getElementById('content')
let container =
document.getElementById('container')

container.style.background = "red";
container.style.height = "55px";
after.onclick = function ()
{
    container.after(content)
}

before.onclick = function ()
{
    container.before(content)
}

append.onclick = function ()
```

```
{  
  
    container.append(content)  
    content.style.textAlign = 'center'  
    content.style.padding = '20px'  
  
}
```

محاضرة 59

Add ,remove and toggle class

مثال

```
<style>  
  
    .name{color: antiquewhite;  
           background: black;  
    }  
</style>  
</head>  
<body>
```

```
<p id="hello">hello! </p>
```

```
<script src="main.js"></script>  
</body>
```

```
let hello =  
document.getElementById('hello');  
  
hello.onclick = function()  
{  
  
    hello.classList.add('name')  
  
}  
  
hello.oncontextmenu = function()  
{  
  
    hello.classList.remove('name')  
  
}
```

الـ Oncontextmenu

يعنى لما اضغط عليك يمين
لكن انا عندي حل افضل من كدة الا وهو

ال Toggle

مثال

```
let hello =  
document.getElementById('hello');  
  
hello.onclick = function()  
{  
  
    hello.classList.toggle('name')  
  
}
```

Toggle يعنى ال

Add , remove

مع بعض

محاضرة 60 Navbar

كود Html

```
<body>

<button class="hide"
id="open">open</button>
<button id="close">close</button>
<div class="container">

<p>HOME</p>
<p>COURSES</p>
<p>NEWS</p>
<p>PROBLEM SOLVING</p>
<p>QUIZS</p>
<p>BLOGS</p>
<p>CHATS</p>

</div>
```

```
<scri pt  src="mai n. j s"></scri pt>  
</body>
```

کود CSS

```
.contai ner  
{  
  
    background-col or:  fi rebri ck;  
    margi n:  2PX;  
    paddi ng:  4PX;  
    wi dth:  150px;  
hei ght:  100vh;  
col or: whi te;  
paddi ng:  2px;  
text-al i gn:  center;  
font-si ze:  20px;  
border-radi us:  25px;  
border-l eft-styl e:  doubl e;  
border-l eft-col or:  whi te;  
border-l eft-wi dth: 10px  ;  
  
}  
#open
```

```

{
    border-radius: 28px;
border: none;
font-size: 27px;
background-color: firebrick ;
color: white;
}
#close
{
    border-radius: 28px;
margin-left: 120px;
border: none;
font-size: 27px;

background-color: firebrick ;
color: white;

}
p: hover
{

    background-color: darkslategrey;
}
p{

    padding: 5px;
}
#open: hover
{

    background-color: darkslategray;

```

```

}
#close: hover
{
    background-color: darkslategray;
}
.hide
{
    display: none;
}

```

کود ال JS

```

let btnOpen =
document.getElementById('open')
let btnClose =
document.getElementById('close')
let container =
document.querySelector('.container')

btnClose.onclick = function()
{
    container.classList.add('hide')
    this.classList.add('hide')
    btnOpen.classList.remove('hide')
}
btnOpen.onclick = function()
{

```



```
this.classList.add('hide')  
btnClose.classList.remove('hide')  
container.classList.remove('hide')  
}
```

محاضرة 61

Auto event

```
<body>  
  
  <input type="text" id="txt">  
  <button id="btn">search</button>  
<script src="main.js"></script>  
</body>
```

```
let txt = document.getElementById('txt')  
let btn = document.getElementById('btn')
```

```
window.onload = function()  
{  
  txt.focus();  
}
```

Focus يعني اول ما يحمل الصفحة السهم

يقف عند ال

Input

محاضرة 62

عمل Gallary

دلة الكود

Html

```
<body>  
  
<div class="gallery">  
  
  <div class="photos">
```

```

```

```

</div>
<div class="main">
    
</div>
<div class="photos">
    
    
    
    
</div>

```

```
</div>  
<script src="main.js"></script>  
</body>
```

Css

```
.photos img{  
width: 200px;  
height: 200px;  
}  
  
.main img  
{  
    height: 450px;  
    width: 450px;  
}  
  
.photos img:hover{  
cursor: pointer;  
    width: 200px;  
    height: 200px;  
}  
  
.gallery  
{  
  
    text-align: center;
```

```
}
```

Scroll

کود Html

```
<body>  
  
    <button id="btn"> &uArr; </button>  
    <script src="main.js"></script>  
</body>
```

کود CSS

```
body{  
  
    width: 4000px;  
    height: 4000px;  
}  
#btn  
{  
bottom: 10px;  
right: 10px;  
width: 30px;
```

```
height: 30px;
position: fixed;
border: none;
font-size: 33px;
display: none;
}
```

کود Java script

```
let btn = document.getElementById('btn');

window.onscroll = function()
{
    if(scrollY >= 400)
    {
        btn.style.display = 'block';
    } else
    {
        btn.style.display = 'none';
    }
}

btn.onclick = function()
{
    scroll({
        top: 0,
```

```
left: 0,  
behavior: "smooth"}  
)  
}
```

محاضرة 66

Location object

```
//  
https://www.youtube.com/watch?v=C4tBiMmrFbc&list=PLknwEmKsW80uTqUDaFRBiAViDZ5uI3VcE&index=66  
== href  
// https == protocol  
// www.youtube.com == hostname  
// watch?v=C4tBiMmrFbc&list=PLknwEmKsW8  
==pathname
```

ای حاجة اقدر اعملها

Get

اقدر اعملها Get

مثال

```
console.log(location.href)
```

عمل set

مثال

```
location.href = 'https://www.google.com/'
```

واقدر اعمل

Get و set

للبروتوكول والهاوس نيم

والباث نيم كمان

الباث نيم بييجى بعد الهاوس

Location.reload

بيخلى الصفحة تحمل تانى

Location.assign

Location.replace

والفرق بينهم

فى ال Assign

بيحتفظ ببيانات الصفحة السابقة

اما ال Replace

هيو جهك بس متقدرش ترجع

محاضرة 67

Settimeout

لو انا عايز فانكشن تشتغل بعد فترة زمنية
معينة يبقى هستخدم

ال Settimeout

مثال

```
setTimeout(function(){  
  console.log('hello!')  
}, 3000)
```

كدة الفانكشن هتشتغل بعد 3 ثوانى

طيب لو حببت اوقف

ال Settimeout

لازم احطها فى متغير الاول

مثال

```
let hello = setTimeout(function(){  
  console.log('hello!')  
}, 3000)  
  
clearTimeout(hello)
```

setinterval

بتكرر الفانكشن بعد كل فترة زمنية

الى ما لا نهاية

مثال

```
let i = 0;  
let hello = setInterval(function(){  
  console.log(i++)  
, 1500)
```

هيفضل يعدلى الى ما لا نهاية

طيب لو عايز اوقف تكرار الفانكشن

عند

i=4

مثال

```
let i = 0;  
let hello = setInterval(function(){
```

```
console.log(i++);  
if(i == 4){  
    clearInterval(hello);  
}  
  
}, 1000)
```