(lec-1)   / /

* استخدامات ال JS :

1. الويب Web .
2. Mobile : (react native & flutter, ionic)
3. Games .
4. Server . (client side, Server side) عامل مع الداتا
5. desktop Apps

* أماكن كتابة ال JS :

1. في ال head : مثل الرابط الذي كنا نكتبه لتعريف ال tags على الأجهزة المختلفة
2. في نهاية ال body : كما في ال CSS ملف الرابط نضعه في نهاية ال body

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

لماذا نكتبه في نهاية ال body ؟

لأنه عبارة عن أوامر تتم على ال (tags) لذلك غير منطقي تعريف الحدث على ال tag قبل تعريف ال tag .

```
<body>
  :
<script> ....... </script>
```

3. external file : نضع كود الرابط في نهاية ال body (مكتوب بالرقم 2) .

* طرق كتابة ال JS : الطرق ال (4) تعطيني قيمة بال output للwebsite

1. عندما نريد طباعة شيء :

```
document.write("Sally marouch")
```

document ← اذهب إلى صفحة ال html

write ← اكتب بداخلها

[" "] → String   نطبع كما هي

نُفضّل في ال [illegible] استخدام ال (id).

/ /

٢ أريد استدعاء عنصر والكتابة بداخله :

```
<body>
<div id="div1" > </div>
```

JS):

```
document.getElementById("div1").innerHTMl = "ahmad"
```

اجلب العنصر الذي له id=div1 ← اكتب بداخل صفحة ال html ← كلمة ahmed

٣ إذا أردت أن تطبع في ال Console :

```
Console.log("islam")
```

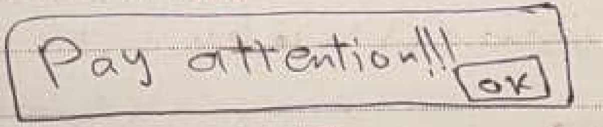
ال Console نستخدمه حتى نعمل check على قيمة معينة وهو يظهر عندما نضغط inspect .

*- مثلاً عندما يحدث error فيجب علينا تجريب اصلاحه واختبار النتائج في ال Console اولاً .

٤ إذا أردنا أن نُخرج مربع من نص الشاشة من فوق :

① alert ، (تنبيه) تتميز بوجود زر ok يعني رسالة عبارة عن تخبير فقط .

```
alert("Pay attention!!!")
```

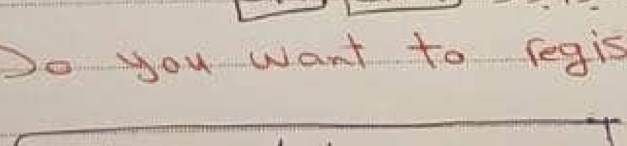

Pay attention!!!

← output

② Confirm : (رسالة تأكيد) , إذا أردنا تخيير ال user بين أكثر من شيء ، تتميز بوجود Cancle ok

```
Confirm("Do you want to register on your web?")
```

← output

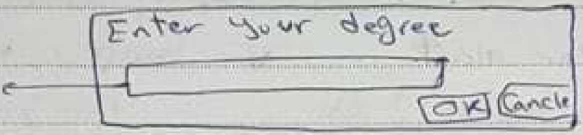

Do you want to ...

عند الضغط على ok يبرز لنا النتيجة بمتغير

٣ Prompt : إذا أردنا أن نطلب من ال user إدخال قيمة (يطلب قيمة)

Prompt ("Enter your degree:")

← output



مربع نص (كتابة إدخال)

*مصطلحات :

① Statement : جملة برمجية

② Syntax : طريقة الكتابة

مثلاً: ما ال Syntax لل Confirm ؟

نكتب الكلمة Confirm ونفتح الأقواس () ونضع الجملة ضمنا "" .

هذا هو ال Syntax.

٣ Comment : // فقط في بداية السطر الذي تريد جعله Comment

← إذا أردت Comment لأكثر من سطر : /* - - - - */

Variables

- يوجد (3) طرق لتعريف المتغيرات : Var, let, Const .

- ال Var :

- فرضنا متغير اسمه (X) وقيمته ⑤ . Var x=5

يتم فتح خانة بال data ويضع فيها القيمة (5)

```
var y = 9 ;
Console.log(x+y) ——————→ # 14
Console.log("x+y") ——————→ # x+y
Var z = x + y
Consol.log(z) ——————→ #14
```

/ /

- قواعد تسمية المتغيرات :

Syntax error

(1) ممنوع تبدأ برقم .

(2) حاول تبدأ باسم يدل على معنى .

(3) ممنوع استخدام كلمات محجوزة : مثل , Var Console, Var Var

Var alert , Var map , Var log

data typs

[1] number : 6,5 , 100 ..

[2] string : a, b, Sally --

[3] boolean : true, false

[4] array, object, null

[5] un defined : غير معرف

```
Var x=100
Console.log(x) ——————→ # 100
Console.log (typeof x) ——————→ # number
```

$\underbrace{\text{typeof}}_{\text{نوع ال data}}$

```
Var x="islam"
Console.log(x) ——————→ # islam
Console.log(typeof x) ——————→ # string

Var x = true
Console.log(x) ——————→ # true
Console.log(typeof x) ——————→ # boolean
```

/ /

ال. array يستخدم عندما تريد ان تكون ال data مرتبة

ال. object عندما يكون الترتيب غير مهم.

```
Var x = [ "ahmed" , "Sally", 100, "siba", true, "shimaa"]
                 Coma
Console.log (x)             # ["ahmed", "sally", ——]
Console.log (typeof x)  # object
Console.log ( x[3]) # siba
Console.log ( x[0]) # ahmed
Console.log (typeof x[1]) # string
Console.log (typeof x[2]) # number
Console.log (typeof x[4]) # boolean
```

ملاحظة: ال array وَ object وَ null نوعهم object.

- ال index يبدأ من ال (0). ال index ترتيب المصفوفة.

- ال length طول المصفوفة، وعدد العناصر أي يبدأ من ①.

ال. object : يستخدم في تسجيل البيانات.

طريقة الكتابة: تكتب في هيئة {key: value}

```
              key
Var x = {  name : "ahmed" ;
                   value
           City : "mansoura ;
           age  :  26      ; }

Console.log(x) # {  name:
```

برنامج التشغيل

Console.log(type of x) # object

- إذا أردت أن تصل access لقيمة معينة ← array

① - تكتب اسم الـ array

② تفتح الأقواس

③ تكتب الـ index   ex x[4]

- إذا أردت أن تصل access لقيمة في الـ object :

مثلاً بدنا مدينة الشخص يلي اسمه x

Console.log(x.City) # mansoura

اسم الـ object

key

Console.log(x.name) # ahmed

object ← .key

array ← [index]

الـ null :

```
Var x=60
x=null
Console.log(x) # null
Console.log(typeof x)# object
```

هام

عندما نستدعي قيمة عرفتها سابقاً (Var) لا داعي لإعادة تعريفها.

الـ (0) قيمة

ولكن null فاضي تماماً مافيه حتى (0).

5 undifined : غير معرفة أصلاً. لا null ولا حتى

لأني نسيت ماعرفتها

Console.log(x) $\xrightarrow{?}$ #error

نوع الخطأ :

ReferenceError

خطأ مرجعي

Var x ⟶ ادي عال data واحجز متغير تسمية x

Console.log(x) ⟶ # undifined ↘

يعني لا يعرف... الجهاز لا يعتبر فاضي إلا في حال قلتله x=null .

Console.log(typeof x) ⟶ # undifined

* لو اعمل سؤال بالانترنت فيه عن نوع متغير undifined ولكن في الحقيقة هي `not rated` `unclassified` يعني هي بانتظار ان تأخذ قيمة بـ type و بـ value .

lec-2 JS / /

Operators

1/ arthimatic operators : +, -, *, /, %, **, --, ++

لـ موديولاس (%) تجي لك امتحان

```
Var x = 50;
Var y = 2
Var Z = x+y
Console.log (Z) ⟶ 25
Var x1 = x%y
Console.log (Z) ⟶ # 0
```
باقي قسمة x على y

! عندما يكون ناتج باقي القسمة على ② ← 0 ⇐ عدد زوجي
وعندما يكون ناتج باقي القسمة على ② ← 1 ⇐ عدد فردي

* الزيادة والنقصان

```
Var x = 40
++x
Console.log (x) ⟶ # 41
```
تم اضافة رقم ①

```
Var y = 50
y++
Console.log (y) ⟶ # 50
```
يطبع قيمة (y) ثم يضيف ①

```
Console.log (y) ⟶ # 51
```

زيادة بعد المتغير

x++ : Post increment

زيادة قبل المتغير

++x : Pre increment

/ /

Var x = 6
y = x ** 3
⟹ Console.log(y) ⟶ # 216 (** ⟶ 6^3 → 6×6×6
(لازم اسمع التشغيل مهم)

2 Assignment operators : =, +=, -=, /=, *=, %=
plus equal ↲ mines equal

Var x = 10
x += 50
Console.log(x) ⟶ #60
x+ = 60 ⟶
x الجديدة = $x_{\text{القديمة}}$ + 60
x- = 50 → x = x - 50

var x = 50 (تشغيل جديد)
x = 40 ⟶ re assignment اعادة اعطاء قيمة جديدة
x = = 40 ⟶ Comparison (مقارنة)
يساوي

if Condition

Var x = 80 الشكل العام:
if (x = = 5) { ___ } else { ___ }
لو إذا كانت x=5
(لازم اسمع وتشغيل)
* أول سؤال بالاختبار: سأل الـ user عن درجته
لو درجته 85 وما فوق ← ممتاز ....

```
Var x = prompt ("Enter your degree:")
if( x >= 85) {
Console.log("EXellent") }
elseif (x >= 75) {
Console.log("Very good") }
elseif( x >= 50) {
Console.log ("Acc") }
else {
Console.log("faild") }
```

3] logical operators — معاملات منطقية

&& ← و (تحقق الاثنين)

|| ← أو (تحقق على الأقل واحد)

Not ← != أو ! ← العكس (عكس true ← false)

```
Var x = 20
Var y = 50
if( (x == 20) && (y == 40)) {
Cosole.log("yes") }
else {
Console.log("no") }
```

الجواب: no

لأن الشرط الأول x=20 تحقق ولكن الشرط الثاني y=40 لم يتحقق

```
Var x = 20
Vay y = 50
if ((x == 20) || (y == 20)) {
console.log("yes") }
else { Console.log("no") }
```

الجواب: yes

/ /

إذا كان ((true) || ____)
لا داعي لأن أكمل بعد علامة || لأن لدي شرط واحد على الأقل true
⇐ النتيجة true
أما لو كان (true && ____) يجب أن أكمل.

Switch Case

Switch: Check Values not range اختبار على قيم وليس رانج
if-else: Check range (من كذا إلى كذا)

```
Var x = prompt("Enter your degree:")
Switch (x) {
Case "a" : Console.log("Car")
 break ;  آخر الحالة إذا تحقق شرط
Case "b" : Cosole.log("bic")
break;
default : Console.log(" ولا شي ") }
الحالة الأخيرة
لو ما تحقق أي من
هدول.
```

* في حال أدخل الـ user: A سينفذ default
لأن switch متعرفة فقط على (a).

Function

* أريد أن أعمل function (لاختصار التكرارات) يجمع عددين:
طريقة تعريف الـ function:

function اسم ال ← Parameter البارامترز

```
function Code1 ( x , y ){
Console.log(x+y) }
```

طريقة استدعاء ال function :

Code1(10 , 20) → arguments

اسم ال function   x=10   y=20   Parameter وليست

لأنه هدول يلي انت بتعطيهن

أما ال (x,y) المعرفة في function ← هي Parameters

- هذه الطريقة من تعريف ال function تدعى

① function declaration أو function tradition

- الطريقة الثانية لتعريف ال function

② function expression:

```
Var Code2= function (a,b){
    Console.log(a+b)}
```

نضع اسم ال function بمتغير

هام سؤال الترشيح + رح يسألنا فيه المرشح آخر ال JS :

SIF = Self Invoked Function = IIFE = Immediately Invoked Function Expressim

= En Eabsulation → التغليف: تغلف ال fun داخل أقواس.

جميع هذه المصطلحات نفس بعضها وتجيب نفس الجواب لو سألنا بأي واحد منهن هدول رح يسألنا لواحدة منهن.

المعنى : function تعمل لوحدها (بدون استدعاء)

/ /

```
function Code3(){
 Console.log("Islam")}
```

Code3() ⟶ هذا السطر هو الاستدعاء

ال calling function أو trigger function

- إذا ما كتبت Code3() ما راح تشتغل لأن ما عملنا trigger

- أمثلة ال function التي تعمل بدون استدعاء :

١

```
( function Code3(){
 Console.log("Islam")}) ()
```

٢

القوس الأول: يمثل ال trigger أو Calling.

والقوس الثاني () : لإكمال ال trigger أو Calling لتزكر لا تعمل بدون أقواس البارامترات.

هذا المثال هو : SIF أو IIFE أو Encabsulation lec-3 JS   / /

Events

mouse الأحداث الخاصة بال

عندما تشيل الماوس

Events: click   dbklick   mousecenter   mousemove   mouseleave

ضغطة ←   ضغطتين ←   hover ←   في غيرهم ولكن هدول الرئيسين.

```
<div class="div1" id="div" >

</div>

<button id="btn1" >...</button
```

عندما أضغط عليه

```
* document.getElementById("btn1").onclick = function(){
      document.getElementById("div1").innerHTML = "islam"}
```

- الآن عندما أضغط على الزر (btn1) ← يكتب بداخل ال (div1) كلام.

```
* document.getElementById ("btn1").dblClick = function(){
  document.getElementById("div1").innerHTML = "islam"}
```

- لما أضغط ضغطتين بالماوس وليس ضغطة واحدة.

```
* document.getElementById("btn1").onmouseenter = function(){
  document.getElementById("div1").inner HTML = "islam"}
```

mousenter: نفس ال hover لما تمرر عالزر بدون ضغط

```
* document.getElementById("btn1").Onmouseleave = function(){
  document.getElementById("div1").innerHTML = "islam"}
```

on mouseleave لما الماوس يغادر الزر.

leave ≠ enter

onmousemove : طول ما انت عم تحرك (مع كل تحريك انت تحرك
علي بيشغل ال function . يعني تلقائياً (for) وعداد .. كل ما حرك
الماوس حركة بسيطة بيزيد العداد .. مع كل حركة بعملها انا .

ملاحظة هامة: أي شي بدك تعمله بال java script أوّل خطوة بتعملها :

الطريقة الصحيحة للكود.
1. تعريف المتغيرات .
2. شو يلي بدك تعمله (يعني ال function)
3. ما الحدث الذي سيشغل ال function .

★ الآن سنختصر الكود الذي في الصفحة السابقة

1] تعريف المتغيرات

```
Var btn = document.getElementById("btn1")
Var div1= document.getElementById("div1")
```

2] دائماً حاطط تشغيل ال fun بطريقة

طريقة + ع

```
function x(){ div1.innerHTMl = "islam"}
```

3] الحدث .

```
btn.onclick = x ;
```

الآن ضغط ع btn بيشغل ال fun بالاسم (x) .

ملاحظة هامة في حال كتبت : btn.onclick = x()

خطأ قاتل لأنه بالحالة رح يشتغل مباشرة ما رح ينظر أي اجل
onclick ⇐ عندما ارتبط ال function بحدث معين لا يجوز
أن أكتب الأقواس باستدعاء ال function .
- إذا عملت الأقواس x() ⇐ صار اسمها trigger أو
Calling ⇐ يعني استدعاء وتشغيل مباشرة ما رح ينتظر الحدث.

بدي كلما أضغط عالزر يضيف ① مع كل ضغطة

```
Var x=1
Var btn1=document.getElementById("btn1")
Var div1= document.getElementById("div1")

function Code(){
    div1.innerHTMl = x++;}
btn1.onclick=Code ;
```

بدون "" ـــ "" كما عرفتها

⇐ مع كل ضغطة رح يضيف واحد

← سؤال : أول ضغطة شو رح يطلع ؟ (1) ليش ؟!

لأنه ++ⓧ ← بيطلع ① وبيضيف ①

← ثاني ضغطة ② وثالت ضغطة ③ وهكذا !

نفس الكود ← :

```
btn1.onmouseleave=Code;
```

كل ما ضغطت ماشي

حتى .. ← بتحرك بيشتغل

```
btn1.onmousemove= Code;
```

طول ما هو بتحرك عال (btn) كله بيضيف ① .

* الـ Events الخاصة بالـ Keyboard (لما اضغط عالـ Keyboard) .

KeyEvents: keypress keydown keyup

← الأشهر استخداماً

- عبارة عن key معناه رح اكتب مو اضغط ← وطبعا كتابة مكانها input .

/ /

```
<input type="text" id="input1">

Var input1 = document.getElementById("input1")
```

أول ما اضغط (قبل ما ارفع ايدي) ((طول ما انا ضاغط))

```
input1.onkeypress = function(){
  Console.log("islam") }
```

(كل شي وأنا ضاغط

* keypress : مثلا بالألعاب طول ما انت ضاغط على (w) بيطلع لفوق.

```
input1.onkeydown = function(){
  Console.log("islam") }
```

* keydown : ما بيستناك ترفع ايدك (أول ما تضغط ما صدق وطبع إيدك وكبست فوراً بينفذ ولكن لو ضليت ضاغط مارح يضل ينفذ طول بينفذ مرة وبس تشيل ايدك وترجع تكبس برجع بينفذ مرة وهكذا .

```
input1.onkeyup = function(){
Console.log("islam") }
```

* Keyup : أول ما تشيل ايدك .. يعني إذا ضغطت على حرف وضليت ضاغط وكتب حتى ألف حرف ورا بعض ما بيكتب بالConsole islam لحتا تشيل ايدي بيكتبها مرة.

← بستخدم keyup مثلاً إذا ضل ضاغط ومطول (5) ثواني وما شال إيده .. معناها بجوز ضغط بالغلط => ما بينفذها. أما إذا ضغط عادي وبعدها شالها بينفذ .

هدول أصلاً الـ keypress رح يلغوها ويحل محلها keydown

يعني اول ما صفحة ال window تحمل عندي --- window.onload = function

```
→ window.onload = function(){
alert("هل تريد التسجيل في الموقع ؟") }
```

- طرق اضافة الأحداث :

1- اسم الحدث + on : onclick, onkeypress, ondblClick

2- addEventListener : اضيف حدث

قيمتين

```
btn.addEventListener("Click", Code )
```

اسم الـ function يلي بدك ياه يشتغل ، لـ اسم الحدث يلي بدك ياه يشتغل لما يصير
بس من دون on

الطريقة 1

```
btn.onclick = code;
```

الطريقة الأفضل في الاستخدام Best Practices هي ال
addEventListener

loop
حلقات

1 For — الأكثر استخداماً.. أكثر من 95% من استخداماتك.

- طريقة الكتابة :

```
for (initialization ; Condition ; update) { ___ }
```

لـ القيمة الابتدائية — لـ الشرط — Semi Colon

```
for(Var x=0 ; x<=100 ; x++){
Console.log(x) }
```

يطبع الارقام من 0 ← 100

- طريقة عمل ال for : هل ال (0) أصغر من 100 ؟ إذا تحقق الشرط ينزل الى عند Console.log وينفذها ← وبعدها ليصعد الى x++ وينفذها ليختبر الشرط على الرقم ① ثم ينزل الى Console.log لينفذها وبعدها تضيف ① وتكبر وهكذا..

* - في حال Console.log(X++) هون بيطبع القيمة القديمة وبعدين بيضيف ولكن ما تكون ضمن ال (for) فلما هون بيضيف واحد

2 while

طالما

```
Var x=0
while( x<=100)
 Console.log (x)
 x++;
 }
```

طالما ال (x) أصغر أو تساوي 100

اطبع (x)

وبعد ما تطبع ضيف ①

3 do-while :

```
Var x=0
do {
Console.log(x)
 x++; } while(x<=100)
```

اعمل (نفذ)

اطبع وضيف ① طالما ال x أصغر من 100

Bad For:

```
for(Var x=10 ; x>2 ; x++){
 Console.log(x) }
```

طولما ال (x) أكبر من (2) يطبع ⇐ لا ينتهي شغال إلى ما لانهاية يسمى infinite Code ⇐ بيخرب الجهاز، إذا شغلتو

- مطلوب من الجهاز في حوالي (72) مليون ثانية (بأقل من ثانية) لينفذ ال for كلها.

* مثال هام جداً جداً: رح يجي بالاختبار، ورح نكتب له كود JS.

- بدنا نطبق كل منتج حسب السعر تبعو.

```
<ul>            class="list"
                attribute
<li> iphon14 promax </li>           price="50 000"
<li> iphone12 pro max </li>         price=" 30 000"
<li> iphon reno 8  </li>            price="15 000"
<li> samsung   </li>
                                    Price="8000"

<div  id="Content">
        بدي لما اضغط عليه يظهر هون
</div>
<button  id="btn1"> show price</button>
```

.querySelector(   )

لـ نكتب بداخلها نفس
بال css

/ /

JS

انا بدي ال btn يظهر إلا لما اضغط على منتجات

تعريف المتغيرات

```
<button id="btn1" style="display:none"> ⟸
```

```
1 Var allproducts = document.~~get~~ querySelector("
```

* ④ يوجد طرق كثيرة لاستدعاء العناصر:

```
getElement By Id
getElement By Class Name
getAnimation
getElement By Name
getElement By Tag Name
getSelection
querySelector
querySelector All
```

getElementByName ← اسم العنصر

querySelector ← نكتب بداخلها مثل css مثلا ("#div1")

querySelectorAll ← عندما نريد استدعاء أكثر من واحد

- الآن نحن نريد استدعاء كل ال (Li) لكي نضعهم في Var

```
1 Var allproducts = document.querySelectorAll(".list li")
  Var Content    = document.querySelector("#content")
  var btn        = document.querySelector("#btn1")
```

- ماذا لي المنتجات ① لما أضغط ع أي منتج ② ميلي يظهر بال div

اسم المنتج يلي داخل (Li)

```
allproducts.forEach( function(item) {  ⟶
```

forEach ← يعني لما عدي ع كل منتج بتعامل معه لوحده ← كل عنصر رح يعدي على من ال (Li)

function ← هيدي معناها عند كل ضغطة ع (Li) اعملي function

منحط بارامتر function رح يكون (Li) يلي بدو بارامتر.

نتيجة ، اقرأوا هديك هدي بتعني.

أي item(li)

1- allproducts.forEach(function(item){
2- item.onclick = function(){
3- Content.innerHTML = item.textContent
4- }
5- })

عنصر الي ضغط عليه (li)

روح عال div Content وكتب جواها محتوى ال li item.

المحتوى النصي لل item

صحيح بهافيتي ok + " " لمشان يعملي فراغ بينا بينهم.

الآن: مثلاً بدي اشتري أكثر من منتج ← ولكن كلما ضغطت عم يجيلي الاسم، ولكن إذا رجعت ضغطت ع حد تاني كان يحذف القديم ويحط المنتج الجديد ← أنا ما بدي هيك.
أنا بدي يضيف المنتجات كلهن
← نفس الكود السابق ← منغير بس جزء واحد السطر ③.

Content.innerHTML += item.textContent .

⟸ صار الجديد = القديم + item.textContent . بهالحالة ما عم يحذف
بيصير يضيفهم لبعض.

* بدي قلو، لما يصير في محتوى بال div Content أظهرلي btn السعر. يعني ال btn نظهر لما يكون في محتوى بال div.
لذلك منكتب من بعد السطر ال ③ يعني تبع تعبئة الأغراض.

if (Content.innerHTML != "") {

لما ال Content div لـ تعال ما فاضية   [illegible] فاضية

btn.style.display = "block"
btn.style.Color = "red"
btn.style.backgroundColor = "blue"

أي خاصية css بدنا نكتبها نكتبها بهالشكل

backgroundColor (ذي خاصية بال css) بدنا نكتبها بال js من كلمتين بيكون أول حرف صغير و الكلمة الثانية تبدأ بحرف كبير

هيك بيظهر لما يصير ال div مو فاضي

display: none ≠ display: block

/ /

* الآن : بدي اكتب السعر بال Console
منروح نعرف متغير جديد فوق عند ال Var

```
Var totalPrice = 0
```

طالما ضغط عال ال item تبع بدي عرّفلي ال (Price)
بروح عالكود لكتب تحت السطر ②

```
totalprice += item.getAttribute ("price")
```

فقط بنخزن متغير

وعند السطر ⑤ يعني بال functions ← منخليه يعرضه

```
btn.onclick = function () {
Console.log( totalprice) }
```

هلق لما بضغط عال Showprice صار يعرضلي السعر
ولكن اذا ضغطت ع ② ← بيكتبن تنين

15000 30000

كيف بدي خليه يجمعهن؟!
الفكرة انو item.getAttribute ("price") ما عم يتعرف
عليه كا رقم

⟸ نعدل عالسطر :

```
totalprice += +( item.getAttribute("price"))
```

ضيفنا هي الطريقة لانك بس هيك حولت نوع ال data
يلي بين اقواس ل integer

لتكون...

كود المثال كامل:

```
Var  allproducts = document.querySelectorAll(".list li")
Var  Content = document.querySelector("#Content")
Var  btn = document.querySelector("#btn1")
Var  totalPrice = 0

allproducts.forEach (function(item){
  item.onclick = function(){
        totalprice  +=  +(item.getAttribute("price"))
   Content.innerHTML += item.textContent + " "

   if( Content.innerHTML != "" ){
     btn.style.display = "block"
      btn.style.backgroundColor = "yellow" }

     }
       })
 btn.onclick = function(){
 Console.log( totalPrice)}
```

lec-4 JS كل lec-3 لاحقة لل

/ /

innerHTML: أمر كتابة

```
totalPrice+= ~~item~~ parseInt(item.getAttribute("price"))
```

← تعطيني مابين قوسين كلمة واحدة صحيحة (رقم)

هام + لو جي بالاختبار:

← اكتب كود تسأل فيه الشخص مثلاً (user) أول ما يفتح المتصفح : كم لاعب عندك ؟! كم طالب عندك ؟

← الرقم يلي بجاوبك فيه مثلاً (4) ← تظهرله prompt دخل اسم أول لاعب .. دخل اسم ثاني لاعب .. وهكذا ، بشرط أن تظهر هذا فقط بعدد مرات الجواب

(4) لاعبين ← 4.prompt

(5) طلاب ← 5.prompt

← بعدين تعرض أسماء اللاعبين كلهم يلي كتبتهم بال prompt بال console كلهم سوا.

الحل:

① عدد اللاعبين

```
var PlayersNumber = Prompt("Enter the Players Number")
```

②

```
Var array1 = [
for (var x = 1 ; x <= playersNumber ; x++){
```

③ بدي تطلع كل مرة شكل prompt

```
Var Player Name = prompt("Enter The Name of the player")
```

← ليش ؟! حفظت الجواب بمتغير مشان اقدر / ارجع الو اذا بديه

1 → playerNumber

```
④: array1 [x] =playerName; }
```

```
⑤: Console.log("array1")
```

/ /

- ليش حطينا array ؟ لأنو أكثر من قيمة
- ليش حطينا array ؟ فيني حط array أو object وممكن
هنا لازم يهمني الترتيب لذلك حطينا array.
- ملاحظة: في الكود السابق سيطبع النتيجة empty
ليش ؟!
لأن ال index سيأخذ ال (0) وليس (1) وأنا عاطيها بال for

(X=1)

الكود الصحيح:

```
Var PlayersNumber = prompt("Enter The plyers Number:")
Var array1 = [ ]
for(var x = 0 ; x < playerNumber ; x++) {
Var playerName = prompt ("Enter The Name of The player:")
array1 [x] = playerName ;   }
بتسكر
Console.log("array1")
```

التطبيق (الخرج) لو دخلت (4) Sally SiSi Sara manar

```
playerName = Sally
array1 [0] = Sally
playerName = SiSi
array1 [1] = SiSi
array1 [2] = Sara
array1 [3] = manar

array1 = [0 , 1, 2, 3]
array1 = ["Sally", "SiSi" , "Sara" , "manar"]
```

string methods

1. split قسّم

2. slice شريحة - يأخذ شريحة من index ل index

3. toLowerCase , toUpperCase

4. trim - يلغي المسافات التي قبل وبعد الـ text

5. indexOf - check ع أي قيمة string لو index لشيء معين يبحث

* split

```
Var Code = "Welcome islam"
```

```
Var Code2 = Code.split(" ")
```

قسّم

أي تقسيم من عند الفراغ.

```
Console.log(code2)
```

الـ output:

```
['welcome', 'islam']
```

- مثلاً:

```
Var Code3 = Code.split("e")
```

```
['w', 'lcom', 'islam']
```

⟸ يلغي الـ e ويقسم من عندها w[e]lcom[e]

★: أين نستخدم الـ split ؟: في الشبكات

إذا بدك تقسم .. ت check ع قيمة معينة

بتقسم ال URL هدا

```
Var Code = "file://C:/users/DELL/Desktop/lec27-5
```

```
Var Code2 = Code.split("/")
Console.log(Code2)
```

["file:", "", 'c', 'users', ---] الشي يلي بتقسم من عنده بلغيه (/)

*slice:

```
Var Code = "hello islam"
Var Code2 = Code.slice(2,7)
Console.log(Code2)
```

خذ جزئية من ال index-2 الى index-7

ماعدا خد ال ⑦

output: llo_i
1 2 3 4 5

7 - 2 = 5

```
 hello  islam
0 1 ②③④⑤ ⑥ 7 8 9 10
```

يلي بيوقف عنده

ما بياخده ⟸

ما بياخد ال ⑦

المسافة في ال string تعتبر index.

ملاحظة: يأخذ ال index الذي يبدأ به ولكن لا يأخذ ال index الذي ينتهي به

*toLowerCase

```
Var Code = "HELLO ISLAM MOHAMED"
Var Code2 = Code.toLowerCase( )
~~Var~~ Console.log(Code2)
```

output: hello islam mohamed

/ /

*** toUpperCase**

```
Var Code= "hello islam"
Var Code2= Code. toUpperCase()
Console.log(Code2)
```

- output : Hello ISlAM

أهم (2)

*** trim**

- إذا عندي input ← type="password" ← لو مسيت وقبلها مسافة ← بيعتبرها غلط .

```
Var Code= "          hello islam          "
Var Code2= Code .trim()
Console.log(Code2)
```

⟸ يعتبر المسافة محتوى

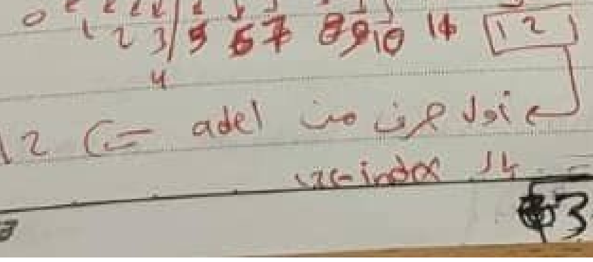
.يلغي المسافات التي قبل أو بعد ال text

***indexOf**

```
Var Code= "hello asmaa adel"
Var Code2= Code.indexOf(" adel")
Console.log(code2)
```

بيعمل Check ع حاجة إذا موجود

output: 12



أول حرف من adel ⟸ 12

ال index=12

$\boxed{!= -1}$ → يعني عم Check ع أي قيمة إذا فاضية أو لا.

علامة فاضية

/ /

← إذا بدك تتأكد أو تختبر Check أي حرف عندك بالموقع بأي index موجود تستخدم indexOf.

١. استخدامات indexOf : إذا بدك ت Check ع أي شي إذا موجود ولا لأ..

عندك ٥٠٠ كود عمل وبدك تشوف إذا العمل كذا موجود ولا لأ ←

تختبرها بس بال indexOf.

إذا ال indexOf ← رجع لقيمة ← true

إذا ما رجع لقيمة ← false

ما رح يرجع ب $\boxed{6}$

$\boxed{6 != -1}$? true دة بينفذ أول واحدة

```
Var Code="hello islam mohamed"
if (Code.indexOf("islam") != -1){
 Console.log("islam is existed")}
else {
 Console.log("islam is not existed")}
```

يعني إذا أي (رقم) سوى كان بالرقم تبع ال index.

```
Var Code= "hello mohamed"
if ( Code. indexOf("islam") != -1) {

else
        }
```

-1

$\boxed{-1 != -1}$

false → بينفذ (else)

في الرسمة لأنه عم Check ع شي مو موجود بيرجعلك $\boxed{-1}$ دائماً

Array methods أسئلة انترفيو

join , PoP , Push, Shift , unshift, lenght.

ConCat , slice

***join:** → يحول ل string

- يحول ال array ل string بالشي لي تحطه بين قوسين

```
Var array1 =[ "ahmad". "mohamed", "islam", "esraa". "aya", "asmaa"]
Var array2. array1.join(" / ")    ("+") (",") ("+/+")
Console.log (array2)
output → ahmed/mohamed/ islam/esraa/ aya/asmaa
```

***Pop:**

Pop - يحذف آخر عنصر في ال array .

هام: يحذف اخر عنصر (asmaa) ويضيفه في arr2

```
Var arr1 =[ "ahmed", "mohamed", "islam", "esraa", "aya", "asmaa"]
Var arr2 = arr1.PoP( )
Console.log (arr2) output → asmaq
Console.log (arr1) output →[ "ahmed" , "mohamed","islam", "esraa","aya"]
```

***push :**

Push - اذا بدك تضيف عنصر بآخر ال array .

```
Var arr1 . [ "ahmed", "mohamed", "islam", "Sally"]
Var arr2 = arr1. push ("ibraheem")
Console. log(arr1) → ["ahmed", mohammed", "islam", "Sally", "ibraheem"]
```

***shift:**

Shift: يحذف اول عنصر في ال array

```
Var arr1.[ "ahmed", "islam", "sally"]
Var arr2 = arr1.shift()
Console.log(arr1) → ["islam", "Sally"]
Console.log (arr2) →  "ahmed"
```

/ /

*unshift:

unshift : يضيف عنصر إلى بداية ال array

```
Var arr1 = ["ahmed" , "islam" , "sally"]
Var arr2 = arr1.unshift("ola")
Console.log      ("arr1") -> ["ola", "ahmed", "islam", "sally"]
```

في نهاية ال array	في بداية ال array
pop يحذف	= يحذف shift
push يضيف	= يضيف unshift

*length:

length : يحسب عدد العناصر مثال ① ←

```
Var arr1 = ["ahmad", Sally", "islam"]
Var arr2 = arr1.length
Console log(arr2) --output--> [3]
```

سؤال امتحان ✱ ما العلاقة بين آخر عنصر في ال array وال length ؟

يعني لو عاوز اجيب آخر عنصر ال array عن طريق ال length كيف ؟

هو نفسه ال Length ③

وال index لآخر عنصر length-1 ②

*Concat:

Concat : بيلزق تضيف عناصر الأرايز (من مصفوفة لمصفوفة) ليصبح ↑ مصفوفة كبيرة

```
Var arr1 = ["ahmed", "islam", Sally"]
Var arr2 = [ "asmaa", rima" , "sisi"]
Var x =  arr1. Concat        (arr2)✓     x("arr2") X
Console.log(x)
   -> ["ahmed", "islam", "sally", "asmaa", "rima", ]
```

ليست string

← اضافة arr2 الى arr1

لتكون...

/ /

*slice

slice: تأخذ جزئية من المصفوفة

```
Var arr1= ["ahmed", "islam", "sally", "rima", "reem"]
Var arr2= arr1. slice(1, 3)
Console.log(arr2) → ["islam", "sally"           ]
```

ahmed   islam   sally   rima   reem

0   [1]   [2]   [3]   4

X

يبدأ من inda ← [1] و الذي ينتهي عند [index-3] لا يأخذه

Math methods

round   pow   sqrt   random   ciel   floor

* round   * Math.round

round : يقرب لأقرب عدد صحيح

```
Var num1= Math.round (6.4) →
Console.log(num1) → [6]
Var num1= Math.round (6,5)
Console.log(num1) → [7]
```

6 و(4) من عشرة

4 أو أقل ← يحافظ على الرقم الأصغر

5 أو أكثر ← يزيد الرقم

6,49999 ⟶ أقرب لـ(0) من 100

6,51111 ⟶ أقرب لـ(100) من (0)

*pow *Math.pow html: : الأُس *Math.pow

```
Var num1 = Math.pow(2, 7)
Console.log(num1) → 2^7 = 128
```

*sqrt *Math.sqrt Math.sqrt : الجذر التربيعي

```
Var num1 = Math.sqrt(81)
Console.log(num1) → √81 = 9
```

*Math.Ceil Math.Ceil : تقريب إلى الرقم الصحيح التالي
تقرب للرقم الصحيح الأعلى

```
Var num1 = Math.Ceil(7,1) (7,9) (7,2)
Console.log(num1) → 8
```

يأخذ الرقم الصحيح الأعلى دوماً

floor *Math.floor Math.floor: تقرب للرقم الصحيح الأقل.

```
Var num1 = Math.floor (7,9) (7,5) (7,1)
Console.log(num1) → 7
```

floor ≠ Ceil

*random *Math.random

```
Var num1 = Math.random()
Console.log(num1) → بيعطي أرقام عشوائية بين 0 ← 1 غير
```

- مثلاً بدي رقم عشوائي بين 0 و 100

```
Var num1 = Math.random()*100
Console.log(num1) → 22,981 —   23.2899 —   94,8 —
```

* - إذا بدي أشيل الكسر كيف ؟! floor وليس Ceil

إذا بدي أرفع الكسر كيف ؟! Ceil وليس floor

```
Var num1 = Math.floor( Math.random()*100)
Console.log(num1)
```

هيك بيعتبرهم رقم وعليه بيعمل floor

/ /

Json

Json: Java Script Object Notation طريقة كتابة

ال Json هو طريقة كتابة في ال JS . بتشبه طريقة ال object.

Json: طريقة الكتابة في JS بتشبه ال object ولكن مش object

Json: وسيلة التواصل بين ال frontend وال backend

لـ لغة وسيطة بينهم

أشهر موقع تعليمي ل Json ← Json placeholder

*ـ ال object مع ال key ماكنتش مطروحة بين (" ") فقط ال Value

[name:"islam"]

كنت حطها بين (" ").

*ـ json : ال key وال value بحطهم بين [" "]

*كيف نكتب ال Json ؟!

ضمن مجلد ال (JS) منعمل script2.json

```
[    {  Key         Value
        "name" :   "islam" ,  ——→  Coma
        "age" :    26 ,
        "city" :   "mansoura"   } ,        ,
        {"name" : "islam",
         "age" : 26 ,
         "City" : "mansoura",} ,     ]
```

ـ يعني هي مجرد زي حطها [ومنعطي key (قيمة)

بين كل سطر و سطر Coma

وبين كل object و object

علامة Coma

ال Datatypes المسموح كتابتها في Json : مسموح 6-Values قيم

(true,false)

number , string , boolean , null, [] , {}

array object

```
{ "city": ["islam", "ahmed"]
  { "name": { name: "—"
              age: "—" }}
```

الـ Datatypes الممنوع كتابتها في Json ممنوع 3-Values

Date   undefined   function → ~~"d": function(){ }~~

~~26/2/2020~~

- ولكن إذا بدي اكتب تاريخ كيف ؟! ⟸ نحطه بين ("") ⟸

بهالحالة صار string "26/2/2020"

- إذا بدك تكتب undefined ؟ ⟵ ("") حولها لـ string ⟸

"undefined";

JS

عملت array بـ object بـ JS ;

فيه array مخزن فيه مجموعة objects

```
Var Code = [
  {
   name: "ahmed",
   age: 20;
   city: "mansoura" },

  {  —  esraa
        22
        cario   },

  {  —  sally
        25
        Damascus  } ]

Console.log(Code)
```

/ /

output:

```
{name: "ahmed" , age: 20 , City: "mansoura" }
{ name: "esraa" , age: 22 , city: "Cario" }
{ name: "Sally", age: 25 , City: "Damas" }
length: 3
```

Console.log(typeof Code) → object

لأنو هي array منوع ال array ← object

HTML

```
<div ~~class~~ id="div1" >
```

JS

- بدي اعرض البيانات بالصفحة مو بال Console

```
Var div1 = document.getElementById("div1")

div1.innerHTMl =Code
```

X → [object Object][object object][-

الصفحة ما بتعرف تقرأ ال object

(الحل). تحويل الكلام لقيم string هي ال data تعتبر data json

array ب object وبكون هي شكل object وبسمي json

- بدي اعرض هي ال data بال div كيف ؟؟ هاد شكل ع data

object وصعب الصفحة تقرأو ـ بحولو string

```
Var x = ~~JSON~~ JSON.stringify(Code)
```

لـ يحول ال object لـ string

هي هيه object

```
div1.innerHTML = x
```

لتكون... وبسمي Code لأنو (x) هي ال code بـ string

إذا عندي data ← string وبدي حولها لـ object لكي أقدر
اعمل عليها access وعدل عليها كيف ؟

Var y = JSON.parse (x) → تحول الـ string لـ object
JSON . Parse وتخزنه في (y)

← هيدي string أصلاً

div1.innerHTML = y → [object,object ,object ——]

(y.City) إذا بدك تعدل عالمدينة

انت كـ Developer بتعرف تتعامل مع object بس ما بتعرف تتعامل مع string .

X = JSON.stringify (y) ; تحول الـ object اللي هو (y) إلى string وتخزنه في (x)

y = JSON.Parse(x) : تحول الـ string يلي هو (x) إلى object وتخزنه في (y)
← الـ JSON