JavaScript Samurai from Zero To advanced Level Summery

Mebo

######################################################################

بسم الله الرحمن الرحيم نبتدى بقه بان التلخيص ده هيبقى فيه اغلب شرح الجافا سكريبت الى مش هتحتاجوا بعده حاجه ان شاء الله

Mebo

#######################################################################  
let , var , const

----------------------------

الفرق بين التلاته دول ايه بقه   
بص يا سيدى ال var دى ملهاش سكوب جوه IF لانها بتتعترف فقط بالاسكوب بتاع الفانكشن بس و معتبره ان البلوك سكوب ملوش لازمه اصلا فعشان كده معتبره اقواس ال if كانها هواء هههههه و تناديها بره ال IF و تلاقيها شغاله عادى خالص & تانى حاجه ملهاش نوع معين يعنى بتعرف الحاجه و تعملها انشياليز ب سترينج او نامبر او اراى او اى حاجه و ممكن بعدها تحط فيها داتا مختلفه خالص فى التايب

/\*eslint-env browser\*/

var Myname = "Mahmoud";

var Myage = 27;

var Myfamily = ["mahmoud", "Mohamed", "Mostafa", "karim", 25,

(function () {

'use strict';

return 5;

}())

];

window.console.log(Myfamily);

window.console.log(Myage);

window.console.log(Myname);

او

var MyVar = "Mahmoud";

console.log(MyVar); // Mahmoud

MyVar=5;

console.log(MyVar); // 5

if(true)

{

var x = 5;

}

console.log(x);

ال let ليها سكوب ذيها ذى اى فاريبول فى اى لغه تانيه شايف عامل ايرور اذاى :D لانها معترفه بالفانكشن و البلوك سكوب

if(true)

{

let MyNum = 5;

}

console.log(MyNum);

الفرق بين ال function Scope و البلوك سكوب ؟؟

البلوك سكوب انى اجى اعمل قوسين كده غير بتوع الفانكشن سواء بقه اقوس لوحدها كده او اقواس الايف او السويتش المهم انها مش اقواس فانكشن

ولان ال let معترفه بالبلوك سكوب فهتضرب هنا عشان بناديها خارج البلوك سكوب

{

let x = 6

}

console.log(x); // error undefined as it is out of Block Scope

بينما نفس الكود بالفاريبول مش هيضرب لان ال var مش معترفه بالبلوك سكوب اصلا

{

var x = 6

}

console.log(x); // 6

اما الاتنين داخل الفانكشن سكوب الى هو يعنى لو اى واحده فيهم داخل فانكشن انسى انها تتشاف بره لان الاتنين معترفين بالفانكشن سكوب

function Myfunc() {

var x = 6

}

console.log(x); // error undefined as it is out of Function Scope

ذى ما انت شايف كده ال var مفيهاش حاجه اسمها تحديد نوع على عكس let

دى بقه بتتعامل مع كله عادى و كذلك اقدر برضوا احددلها نوع معين اجبره يتعامل معاه   
فهتلاقىه بيتعامل مع الفاريبول كسترينج و كانمبر و كبوليان و كأراى و ادى مثال اهه و ممكن اقوله any لو مش حابب احدد برضوا :D

var x = [5,6,4] ;

var z = "Mostafa"

let Name: string="Mahmoud"; // accept String Only

let Num: number = 5;

let IsGood: boolean = true;

let MyArr1: string[]; // first Method to write Array of string

let MyArr2: Array<string>; // second Method (Generic) to write Array of string

let MyArr3: number[]; // first Method to write Array of number

let MyArr4: Array<number>; // second Method to write Array of number

let MyArrAny: Array<any>; // this array like list because now it accept any data type

عندك بقه حاجه اسمها Const دى بتثبت رقم بشكل جلوبال بحيث محدش يعرف يغيره و ده بيبقى معمول مثلا للكونيكشن سترينج

const Myname:string = "mahmoud";

Mebo

#######################################################################  
IIFE (Immediately Invoke Function Expression)

-----------------------------------------------------------

بص يا سيدى دى فانكشن بترن مره واحده بس و بترن من تلقاء نفسها اهميتا انى بعملها كانها سكوب بالظبط للكود الى انا بكتبه عارفين ليه عشان لو انا عامل مثلا فاريبول جلوبال فى فايل ما و جيت فى فايل تانى عملت نفس الفاريبول و انا مش واخد بالى و خليته برضوا جلوبال ساعتها هيعملوا اوفرايد على بعض انما لو انا بكتب اى كود جافا سكريب داخل السيلف انفوك فانكشن فكده انا خليت لكل فايل سكوب خاص بيه داخل ال Self Invoke func

و دى عينه من الكلام الى بقوله

var x = 7;

console.log(x); // 7

(function Myfunc() {

console.log(x); // undefined as it does not take from outer Scope

var x = 6

console.log(x); // 6

})()

هتلاحظ هنا انه مأخدش من الجلوبال سكوب عشان انا حطه جوه سيلف انفوك فانكشن و اخد فقط من الاسكوب بتاعه يبقى احسن حاجه ان كل اما افتح فايل جديد للجافا سكريبت اروح كاتب كل الكود بتعها داخل فانكشن IIFE

Mebo

######################################################################  
Data Type  
------------------------------

بصوا هو من الاول كده الجافا سكريبت بتتعرف على نوع الفاريبول بشكل اوتوماتيك من غير ما انت تحدده هى بتشوفه تعرفه على طول و فيها بينقسم نوع الداتا ل   
Number : و ده بيبقى رقم صحيح او رقم عشرى او اى رقم عمتا   
String : ده بقه بيبقى مجرد كلام و رغى ههههههههه نص يعنى   
Boolean : ده بقه ذى يا ابيض يا اسود يعنى يا صح يا غلط الى هى يا اما true او false   
Null : ده معناها انها اتعرفت بس انا اديتها قيمه فاضيه ذى قيمه النول بتاعت الداتا بيز دى قيمه على فكره بس اسمها قيمه الولا حاجه هههههه  
undefined : دى معناها انها اتعرفت بس انا مدتهاش اى قيمه هى بس حجزت مكان فى الميمورى انما مخدتش اى قيمه ولا حته قيمه الولا حاجه الفاضيه ذى ال Null ههههههه   
  
- اما بقه لو معرفتش الفاريبول اصلا و حبيت استدعيه فطبعا هيدينى ايرور لانه اصلا مش متخزن فى الميمورى و بالتالى هيقولى ده مش موجود اصلا   
 و دى امثله على الداتا فاليو

**var *firstName*** = **'John'**; *// string****console***.log(***firstName***); *// John***var *lastName*** = **'Smith'**; *// string***var *age*** = 28; *// Number***var *fullAge*** = **true**; *// Boolean****console***.log(***fullAge***); *// true***var *MyNull*** = **null**;  
***console***.log(***MyNull***); *// null***var *job***;  
***console***.log(***job***); *// Undefined****console***.log(Myname); *// Error Uncaught ReferenceError: Myname is not defined*

Mebo

######################################################################  
concatenation  
------------------------------

دى ياعم انى اجمع شويه داتا جنب بعض كده فى سترينج واحد و ادى مثال

**var *firstName*** = **'John'**;  
**var *age*** = 28;  
**var *job***, ***isMarried***;  
***job*** = **'teacher'**;  
***isMarried*** = **false**;  
*// Type coercion or concatenation****console***.log(***firstName*** + **' is a '** + ***age*** + **' year old '** + ***job*** + **'. Is he married? '** + ***isMarried***);

طيب احنا عشان تبقى عارف ممكن نبقى مخزنين فى فاريبول قيمه و نغيرها عادى و دى هيظهر فى المثال الى جاى

**var *firstName*** = **'John'**;  
**var *age*** = 28;  
**var *job***, ***isMarried***;  
***job*** = **'teacher'**;  
***isMarried*** = **false**;  
*// Type coercion****console***.log(***firstName*** + **' is a '** + ***age*** + **' year old '** + ***job*** + **'. Is he married? '** + ***isMarried***);  
*// Variable mutation****firstName*** = **'Mahmoud'  
*age*** = **'twenty eight'**;  
***job*** = **'driver'**;  
***isMarried*** = **true**;  
***console***.log(***firstName*** + **' is a '** + ***age*** + **' year old '** + ***job*** + **'. Is he married? '** + ***isMarried***);  
  
*// prompt***var *lastName*** = *prompt*(**'What is his last Name?'**);  
***console***.log(***firstName*** + **' '** + ***lastName***);

جرب بقه و شوف النتايج و كمان يا سيدى فايده البرومبيت بقه انها ذى الاليرت كده بس الفرق بينهم ان البرومبيت بتاخد انبوت و بتخزنه فى فاريبول ذى ما انت شايف

Mebo

######################################################################  
coergion  
------------------------------

ده معناه تحويل الداتا تايب للفاريبول بشكل اوتوماتيكى يناسب العمليه الى بتحصل عليه دلوقتى فهنلاحظ ان اضافه رقم 5 على رقم 5 يساوى 10 بينما اضافه رقم 5 على اى سترينج ساعتها هيعتبر ال 5 استرينج هى كمان و يعملها كونكاتيونيشن ده فى حاله الجمع اما فى حاله الضرب بقه هتلاقى ان لو الارقام مكتوبه فى استرينج هيحولها لارقام و يضرب عادى اما لو حاصل ضرب استرينج كلامى مع رقم هيطلعلك نتيجه NAN

/\*eslint-env browser\*/

console.log(5 +5); // 10 Number

console.log(5+'M'); // 5M string

console.log('M' + 5); // M5 string

console.log('6' + 5); // 65 string

console.log('6' \* 5); // 30 Number

console.log('6' \* '5'); // 30 Number

console.log('M' \* 5); // NAN not a Numner

Mebo

######################################################################  
Basic operators   
------------------------------

ده من الاخر الضرب و القسمه و الجمع و الطرح و باقى قسمه و كل دول انت عارف استخدمهم فبلح يعنى مش لازم اشرحهم هديك امثله و خلاص جك خابط

**var *yearJohn***, ***yearMark***;  
*// variables with out var :D***year** = 2018;  
**ageJohn** = 28;  
**ageMark** = 33;  
*// Math operators****yearJohn*** = **year** - **ageJohn**;  
**yeahMark** = **year** - **ageMark**;  
***console***.log(***yearJohn***); *// 1990****console***.log(**yeahMark**); *// 1985****console***.log(**year**); *// 2018****console***.log(**year** + 2); *// 2020****console***.log(**year** \* 2); *// 4036****console***.log(**year** / 10); *// 201.8****console***.log(**year** % 10); *// 8*

Mebo

######################################################################  
logical operators   
------------------------------

دى ياعم اكبر و اصغر من الاخر   
> أكبر   
< أصغر  
<= اصغر من او بيساوى   
>= اكبر من او بيساوى   
== دى تساوى فى القيمه بس   
=== تساوى فى القيمه و النوع   
!= لا تساوى   
  
الحاجات دى فى الغالب بتستخدم فى ال If condition يعنى بتبقى شرط لو حصل يترتب عليه حدوث شئ اخر   
و على فكره ممكن اختصر كتابه الاف بالطريقه الى مكتوب بيها الكود ده

*// Logical operators***var *johnOlder*** = 15 < 19; *// true****console***.log(***johnOlder***); *// true  
// typeof operator****console***.log(**typeof *johnOlder***); *// boolean****console***.log(**typeof** ageJohn); *// undefined****console***.log(**typeof 'Mark is older tha John'**); *// string***var *x***;  
***console***.log(**typeof *x***); *// undefined*

console.log(0 && 9); // 0 (false and True)

console.log(9 && 0); // 0 (True and false)

console.log(0 || 9); // 9 (false or True)

console.log(9 || 0); // 9 (True or false)

console.log(1 || 9); // 1 (True or True) so take First true

console.log(9 || 1); // 9 (True or True) so take First true

Mebo

######################################################################  
Hoisting  
------------------------------

دى يا سيدى معناها ان الفانكشنز العاديه الكومبايلر بيقراها الاول خالص اول ما بيخش الفايل و بالتالى انا بقدر اعملها كول من قبلها او من بعدها عادى على عكس الفاريبولز الى بتتقرا بعد الفانكشنز و لذلك لا يمكن انى انادى فاريبول قبل انشاءه فى الكود و لذلك مينفعش تيجى تقولى جيبلى قيمه فاريبول قبل ما احطها و فكره الهيوستنيج هنا انى بقه لما بخلى الفانكشن العاديه انونيموس فانكشن و احط قيمتها فى فاريبول فهنا بقه قوانين الفاريبول بتنتطبق عليه ولازم لما اجى اناديها اناديها من بعد الفانكشن وليس قبلها و ادى مثال

console.log(Myfunc()); // it makes Error as it is now like A variable

let Myfunc = function(){

return 5 ;

}

console.log(Myfunc()); // 5

Mebo

######################################################################  
Operator precedence  
------------------------------

هنا بيعرفك بقه ايه العمليات الى بتحصل قبل التانيه   
<https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Operators/Operator_Precedence>

بص بقه على الجدول و انت هتفهم ايه بيجى قبل ايه و كمان يا سيدى مثال صغير اهه

***console***.log(3 + 4 \* 5); *// 3 + 20  
// expected output: 23****console***.log(4 \* 3 \*\* 2); *// 4 \* 9  
// expected output: 36***var *a*** , ***b***;  
***console***.log(***a*** = ***b*** = 5); *// 5***var *now*** = 2018;  
**var *yearJohn*** = 1989;  
**var *fullAge*** = 18;  
*// Multiple operators***var *isFullAge*** = ***now*** - ***yearJohn*** >= ***fullAge***; *// true****console***.log(***isFullAge***);  
*// Grouping***var *ageJohn*** = ***now*** - ***yearJohn***;  
**var *ageMark*** = 35;  
**var *average*** = (***ageJohn*** + ***ageMark***) / 2;  
***console***.log(***average***);  
*// Multiple assignments***var *x***, ***y***;  
***x*** = ***y*** = (3 + 5) \* 4 - 6; *// 8 \* 4 - 6 // 32 - 6 // 26****console***.log(***x***, ***y***);

*// More operators****x*** \*= 2; *// x = x \* 2****console***.log(***x***);  
***x*** += 10; *// x = x + 10****console***.log(***x***);  
***x***--; *// x = x-1****console***.log(***x***);

و طبعا انا مش محتاج اشرح ++ و =+ و الكلام ده يعنى المفروض انه سهل يعنى الكومنتات شرحاها لوحدها اصلا

Mebo

######################################################################  
If statement  
------------------------------

دى اف الشرطيه بمعنى انا هيتحقق شرط معين و يدينى true فاالاسكوب بتاع الشرط ده هيتحقق و الباقى شكرا   
و ادى مثال

**var *firstName*** = **'John'**;  
**var *civilStatus*** = **'single'**;  
  
**if** (***civilStatus*** === **'married'**) {  
 ***console***.log(***firstName*** + **' is married!'**);  
} **else** {  
 ***console***.log(***firstName*** + **' will hopefully marry soon :)'**);  
}  
*// John will hopefully marry soon***var *isMarried*** = **true**;  
**if** (***isMarried***) {  
 ***console***.log(***firstName*** + **' is married!'**);  
} **else** {  
 ***console***.log(***firstName*** + **' will hopefully marry soon :)'**);  
}  
*// John is married!***var *massMark*** = 78; *// kg***var *heightMark*** = 1.69; *// meters***var *massJohn*** = 92;  
**var *heightJohn*** = 1.95;  
  
**var *BMIMark*** = ***massMark*** / (***heightMark*** \* ***heightMark***);  
**var *BMIJohn*** = ***massJohn*** / (***heightJohn*** \* ***heightJohn***);  
  
**if** (***BMIMark*** > ***BMIJohn***) {  
 ***console***.log(**'Mark\'s BMI is higher than John\'s.'**);  
} **else** {  
 ***console***.log(**'John\'s BMI is higher than Marks\'s.'**);  
}  
*// Mark's BMI is higher than John's.*

Mebo

######################################################################  
Boolean Logic  
------------------------------

هنا بيعرفيك استخدام And , OR  
&&  
||  
فى الجافا سكريبت مع الايف كونيديشن و من الاخر And لازم الاتنين يبقوا ترو عشان تديك ترو غير كده بتدى فولس و OR لازم الاتنين فولس عشان يديك فولس غير كده كله ترو

امثله للتوضيح

**var *firstName*** = **'John'**;  
**var *age*** = 20;  
  
**if** (***age*** < 13) {  
 ***console***.log(***firstName*** + **' is a boy.'**);  
} **else if** (***age*** >= 13 && ***age*** < 20) { *// true && false = false* ***console***.log(***firstName*** + **' is a teenager.'**);  
} **else if** (***age*** >= 20 && ***age*** < 30) { *// true && true = true* ***console***.log(***firstName*** + **' is a young man.'**);  
} **else** { *// as there is true in last condition so now it became false* ***console***.log(***firstName*** + **' is a man.'**);  
}

Mebo

######################################################################  
Ternary operator  
------------------------------

دى بدل ما بكتب الايف كونديشن فى اقواس و هيصه بختصرها و اكتبها فى سطر واحد اهه يا اسطى

**var *age*** = 14;  
**if** (***age*** >= 18) {  
 **var *drink*** = **'beer'**;  
} **else** {  
 **var *drink*** = **'juice'**;  
}  
***console***.log(***drink***);  
  
**var *drink*** = ***age*** >= 18 ? **'beer'** : **'juice'**;  
***console***.log(***drink***);

هتا بقوله لو اكبر من 18 يشرب بيره و خد الكلمه خزنها بقه فى الفاريبول درينك و لو اصغر يبقى عصير ههههههه  
و فى طريقه تانيه للتسجيل فى الكونسول بدل الفاريبول

**var *firstName*** = **'John'**;  
**var *age*** = 14;  
*// Ternary operator****age*** >= 18 ? ***console***.log(***firstName*** + **' drinks beer.'**) : ***console***.log(***firstName*** + **' drinks juice.'**);

من الاخر علامه الاستفهام دى يعنى هل ال 14 اكبر من 18 او بتسوايها فلو الاجابه بنعم الى هى true الى بيرجع بقه الى بعد علامه الاستفهام على طول اما لو لأ الى هى false ساعتها بترجع الى بعد علامه : على طول

Condition ? return if true : return if false

Mebo

######################################################################  
For Loop   
------------------------------

دى بقه تم اختراعها عشان لو انت عاوز تكرر كتابه حاجه عدد معين دون ما تكتبها فى كل مره فعملوا حاجه عامله ذى الدايره الكهربيه كده تفضل تلف لحد ما الحاله الى المفروض تقف فيها تيجى فتروح واقفه فهنا فى المثال ده انا بقوله لف على الجمله الى فى الكونسول و اكتبها طول ما ال I اقل من 10 و بما انها 0 ففكل لفه زودها واحد عشان ميفضلش يلف للابد يعنى

for(i = 0 ; i<10 ; i++){

console.log("Hello Number : " + i);

}

Mebo

######################################################################  
while Loop   
------------------------------

هو هو ذى ال for لوب بس هنا بتتكتب بطريقه تانيه بقوله طالما الكونديشن بيطلعلك ترو فخليك شغال و اول ما يبقى ب فولس خلاص اوقف اللوب يا معلم

let i = 0 ;

while( i<10 ){

console.log("-Hello Number : " + i);

i++;

}

Mebo

######################################################################  
Do while Loop   
------------------------------

دى ذى الى فوقيها بس الفرق ان هنا بقوله اعمل اول مره من غير اى كونديشنز انت اعملها و خلاص و بعد كده فكر فى الكونديشن بقه

let i = 0 ;

do {

console.log("--Hello Number : " + i);

i++;

}while( i<0 )

الكونديشن هنا بقوله اعمل الى جوه اللوب فى حاله ان ال I اصغر من 0 فده حاجه مستحيله تحصل و بالتالى اللوب مش هتشتغل بس بالرغم من كده هى كتب فى النتيجه --Hello Number 0 و ده لان انا قولتله اول مره خش من غير ما تشغل دماغك بالكونديشن بينما لو كان قايلى وايل الاول و مفيش do كان ساعتها هيخرج من اللوب و مش هيكتب اى حاجه

Mebo

######################################################################  
continue Vs Break   
------------------------------

Break: دى يا سيدى لو حصلت وسط لوب و الكونديشن بتعها اتحقق هتخرجك بره اللوب خالص و مش هتكمل شغل

for (i = 1; i <= 10; i++) {

if (i % 3 == 0) {

break;

}

console.log("Hello Number : " + i);

}

هنا هيجبلك رقم 1 و 2 و بعد كده هيقفل اللوب عشان دخل نفذ ال break و خرج من اللوب خالص و اشتغل بره اسكوب اللوب بقه   
ملحوظه هامه البريك بيطلعك بره الاسكوب الخاص بيه يعنى لو جوه لوب هيخرجك من اللوب لو جوه فانكشن هيخرج من الفانكشن و اى سكوب تانى الكود هيمشى فيه بيمشى فيه عادى

Continue: دى بقه يا سيدى ذى فكره skip الى هو لو الكونديشن بتعها اتحقق و مشى عليها هيروح خارج بره اللوب و راجع تانى شغال عادى المهم يعمل سكيب للحاله دى فى الوقت الحالى

for (i = 1; i <= 10; i++) {

if (i % 3 == 0) {

continue;

}

console.log("Hello Number : " + i);

}

فهنا على سبيل المثال انا قايله اى رقم بيقبل القسمه على 3 اعمل كونتنيو و بالتالى اى رقم بيقبل القسمه على 3 اول ما الاى تخده ساعتها مش هيمر على الجمله بتاعت الكونسول و هيمر فى الجمله الى بعدها الى مش هتخش فى كونتينو و عشان كده هيطلع النتيجه 1 ,2,4,5,7,8,10

Mebo

######################################################################  
Naming Loop   
------------------------------

هنا بنسمى اللوب عشان نعرف نخرج من انهى واحد و ده بيتعمل لما بنكون عاملين Nested loop ومش عارفين اللوب لو اتعمله بريك او كونتينو المفروض يخرج من انهى واحد

for (i = 1; i <= 10; i++) {

for (j = 1; j <= 10; j++) {

if (j % 3 == 0) {

break ;

}

console.log("Hello Number : " + j);

}

}

دى العاديه الى مفيهاش اسماء للوب هنا البريك لما هيحصل هيخرج بره الفور لوب الى جوه و يرجه للى بره يشتغل تانى و هكذا لحد ما ال 10 بتوع ال I يخلصوا لانى عامل البريك على اللوب الى جوه

يعنى النتيحه انه هيعد 1, 2 عشر مرات

outerloop:

for (i = 1; i <= 10; i++) {

innerloop:

for (j = 1; j <= 10; j++) {

if (j % 3 == 0) {

break outerloop;

}

console.log("Hello Number : " + j);

}

}

على الرقم انى قايله الكونديشن هيحصل لما الاينر لوب يكون ال j بتقبل القسمه على 3 و ده معناه ان المفروض يلف 10 مرات بتوع ال inner وفى الاوتر كل اما الكونديشن تتحقق يروح راجع منفذ ال outer الا ان بسبب تسميه اللوب باسماء معينه و لما جيت اعمل البريك عملته للوب الخارجى فهو من اول مره هيحصل فيها بريك الكود هيرمى بره سكوب اللوب الخارجى على طول

يعنى النتيحه انه هيعد 1, 2 مره واحده بس و بعد كده هيخرج بره اللوب الكبير و الصغير خالص

Mebo

######################################################################  
Switch statement  
------------------------------

دى ذى الايف كونديشن بالظبط بس بفروقات بسيطه ان هنا فى السويتش انا بديله الى عاوز اعمل عليه اتشيك هو بقه بيخش كيس كيس يشوف هل الانبوت الى انا مديهوله هو هو نفس الكيس ولا لأ لو هو نفس الكيس بيروح منفز الكود بتاع الكيس و طبعا فى بريك عشان ميعدش يشوف باقى الكيسيسز الى بعديها على الفاضى و يضيع وقت

**var *firstName*** = **'John'**;  
**var *job*** = **'instructor'**;  
**switch** (***job***) {  
 **case 'teacher'**:  
 **case 'instructor'**:  
 ***console***.log(***firstName*** + **' teaches kids how to code.'**);  
 **break**;  
 **case 'driver'**:  
 ***console***.log(***firstName*** + **' drives an uber in Lisbon.'**);  
 **break**;  
 **case 'designer'**:  
 ***console***.log(***firstName*** + **' designs beautiful websites.'**);  
 **break**;  
 **default**:  
 ***console***.log(***firstName*** + **' does something else.'**);  
}

هنا بقوله ان الوظيقه انستراكتور فلما دخل السويتش لفى ان فى كيس تيتشار و معاها كيس انستراكتور فدى هى الى هتتاخد على طول كانه هنا بيقول فى حاله انه مدرس أو محاضر نفذ الكلام و طبعا ده حصل لانى محطتش break لتيتشار و بالتالى وضع ال Break فى اخر كل كيس مهم جدا جدا جدا عشان لو محطتهاش هتلاقى اتنين تلاته اربع كيسيز مع بعضهم شغالين اشطه هههههههههه  
  
يبقى فى حاله انه محاضر او مدرس نفذ الكود بتعها و فى بريك هيوقفنى فمش هيخش فى مصمم او سواق خالص الا لو كان الانبوت فعلا سواق او مصمم

**var *firstName*** = **'John'**;  
**var *age*** = 56;  
**switch** (**true**) {  
 **case *age*** < 13:  
 ***console***.log(***firstName*** + **' is a boy.'**);  
 **break**;  
 **case *age*** >= 13 && ***age*** < 20:  
 ***console***.log(***firstName*** + **' is a teenager.'**);  
 **break**;  
 **case *age*** >= 20 && ***age*** < 30:  
 ***console***.log(***firstName*** + **' is a young man.'**);  
 **break**;  
 **default**:  
 ***console***.log(***firstName*** + **' is a man.'**);  
}

هنا بقوله الانبوت للسويتش عباره عن true و الكيسيز عباره عن مقارانات و بالتالى هو فى كل كيس هيقارن و الى هيطلع ترو فى الاول هيتنفذ و طالما فيه break يبقى كل الى بعده بلح بقه

Mebo

######################################################################  
Truthy and Falsy values and equality operators  
----------------------------------------------------------

***// falsy values: undefined, null, 0, '', NaN  
// truthy values: NOT falsy values***

ده يعنى فيه قيم لو هى الى طلعت نتيجه شرط فكان النتيجه تساوى false و اى حاجه غيرها بقه بتعتبر true

مثال على الكلام الى فات ده

**var *height***;  
***height*** = 23;  
**if** (***height*** || ***height*** === 0) { *// 23 so it is like true* ***console***.log(**'Variable is defined'**);  
} **else** {  
 ***console***.log(**'Variable has NOT been defined'**);  
}  
  
*// Equality operators***if** (***height*** == **'23'**) { *// it is true as 23 is equal to '23' with ==* ***console***.log(**'The == operator does type coercion!'**);  
}  
  
*// Equality operators***if** (***height*** === **'23'**) { *// it is false as 23 is not equal to '23' with ===* ***console***.log(**'The == operator does type coercion!'**);  
}

اول حاله هو بيقول 23 ولا 0 فطبعا 23 و ده معاناه انها كانك بتقول true و بالتالى دخل فى اول شرطه و ساب التانى   
الحاله التانيه بقه هنا مقارنه ب == الى بتقارن بالقيمه بس و عشان كده رجع true و اشتغلت   
الحاله التالته بيقارن ب === الى بتقارن بالقيمه و النوع و بما ان رقم لا يساوى سترينج فرجع false فمشتغلتش بقه

ده مثال على معظم الى فات بص عليه

**var *scoreJohn*** = (189 + 120 + 103) / 3;  
**var *scoreMike*** = (129 + 94 + 123) / 3;  
**var *scoreMary*** = (97 + 134 + 105) / 3;  
***console***.log(***scoreJohn***, ***scoreMike***, ***scoreMary***);  
  
**if** (***scoreJohn*** > ***scoreMike*** && ***scoreJohn*** > ***scoreMary***) {  
 ***console***.log(**'John\'s team wins with '** + ***scoreJohn*** + **' points'**);  
} **else if** (***scoreMike*** > ***scoreJohn*** && ***scoreMike*** > ***scoreMary***) {  
 ***console***.log(**'Mike\'s team wins with '** + ***scoreMike*** + **' points'**);  
} **else if** (***scoreMary*** > ***scoreJohn*** && ***scoreMary*** > ***scoreMike***) {  
 ***console***.log(**'Mary\'s team wins with '** + ***scoreMary*** + **' points'**);  
} **else** {  
 ***console***.log(**'There is a draw'**);  
}  
  
  
**if** (***scoreJohn*** > ***scoreMike***) {  
 ***console***.log(**'John\'s team wins with '** + ***scoreJohn*** + **' points'**);  
} **else if** (***scoreMike*** > ***scoreJohn***) {  
 ***console***.log(**'Mike\'s team wins with '** + ***scoreMike*** + **' points'**);  
} **else** {  
 ***console***.log(**'There is a draw'**);  
}

Mebo

######################################################################  
Functions   
-----------------------

الفانكشن هى من الاخر كده بتخلى كود معين اقدر اعمله 200 الف مره بمجرد انى اناديله بس كده و يمكن مناداه الفانكشن من اى مكان سواء فوقها او تحتها لان بمجرد ما بعمل الفانكشن بتروح متسجله فى الميمورى على طول و الكومبيلر بيقرأ اى فانكشن داخل الصفحه أولا ثم يمشى على الكود بقه سطر سطر عادى

**function** *calculateAge*(birthYear) {  
 **return** 2018 - birthYear;  
}  
  
**var *ageJohn*** = *calculateAge*(1990);  
**var *ageMike*** = *calculateAge*(1948);  
**var *ageJane*** = *calculateAge*(1969);  
***console***.log(***ageJohn***, ***ageMike***, ***ageJane***);

هنا الفانكشن بترجع القيمه فى return و بالتالى اقدر ارجع اى فاليو فيها و استقبلها فى فاريبول و بعد كده اظهرها فى كونسول او اى حاجه

**function** *yearsUntilRetirement*(year, firstName) {  
 **var** age = 21;  
 **var** retirement = (2018-year) - age;  
   
 **if** (retirement > 0) {  
 ***console***.log(firstName + **' retires in '** + retirement + **' years.'**);  
 } **else** {  
 ***console***.log(firstName + **' is already retired.'**)  
 }  
   
}  
  
*yearsUntilRetirement*(1990, **'John'**);  
*yearsUntilRetirement*(1948, **'Mike'**);  
*yearsUntilRetirement*(1969, **'Jane'**);

هنا بقه بترتن فى الكونسول جوه الفانكشن و بالتالى مينعش اخزنها فى حاجه ههههههه و بالتالى التعامل مع الكونسول بس و بنادى الفانكشن و بحطهاش فى حاجه ذى ما انت شايف اهه و هى بتظهر لوحدها فى الكونسول   
طبعا طريقه الريتيرن احسن 100 مره عشان تتحكم فيها برحتك

Mebo

######################################################################  
Function Statements and Expressions  
----------------------------------------------

*// Function declaration***function** *whatDoYouDo*() {***console***.log(**"Hello Badawy from Normal Function"**)}  
*whatDoYouDo*();  
*// Function expression***var** *whatDoYouDo* = **function**(){***console***.log(**"Hello Badawy from Function expression"**)}  
*whatDoYouDo*();

ذى ما انت شايف فى طريقيتين اهم لكتابه الفانكشن العاديه الطريقه الاولى انك بتكتبها عادى خالص ذى اى لغه و الطريقه التانيه انك بتكتب الفانكشن و تخزن الريتيرن بتعها فى فاريبول الفاريبول ده بيبقى هو اسم الفانكسن لما تيجى تستدعيها و فى الغالب مينقعش تستخدم كونسول جواها عشان هتعمل مشكله فيما بعد و المفروض تتكتب ريتيرن احسن ذى كده

*// Function declaration***function** *whatDoYouDo*() {**return "Hello Badawy from Normal Function"**;}  
***console***.log(*whatDoYouDo*());  
*// Function expression***var** *whatDoYouDo* = **function**(){**return "Hello Badawy from Function expression"**}  
***console***.log(*whatDoYouDo*());

و ادى مثال بجمع فيه الفانكشن و السويتش بقه ههههههه و ببينلك فيه فايده نداء الفانكشنز بدل ما اعدل اكتب 200 سطر

*// Function expression***var** *whatDoYouDo* = **function**(job, firstName) {  
 **switch**(job) {  
 **case 'teacher'**:  
 **return** firstName + **' teaches kids how to code'**;  
 **break**;  
 **case 'driver'**:  
 **return** firstName + **' drives a cab in Lisbon.'**;  
 **break**;  
 **case 'designer'**:  
 **return** firstName + **' designs beautiful websites'**;  
 **break**;  
 **default**:  
 **return** firstName + **' does something else'**;  
 }  
}  
***console***.log(*whatDoYouDo*(**'teacher'**, **'John'**));  
***console***.log(*whatDoYouDo*(**'designer'**, **'Jane'**));  
***console***.log(*whatDoYouDo*(**'retired'**, **'Mark'**));

عاوزك تاخد بالك تانى ان الفانكشن انا شيلت اسمها و خليته فاريبول اخزن فيه نتيجه ارجاع الفانكشن و لما باجى استدعى الفانكشن بستدعيه باسمها الى هو فاريبول اهه ههههههههه :D

**ملحوظه اخرى مهمه جدا :** الفانكشن الى بتتكتب بالشكل ده الى هى بتبقى متسجله فى فاريبول دى ياباشا مينفعش استدعيها قبل انشائها لانها هى بقت بالشكل ده عباره عن فاريبول و بالتالى مينفعش اناديها قبل ما اعملها لا لازم اناديها بعد ما اكتبها يعنى لازم النداء بتعها يبقى تحتها و لو جربت كتبت النداء بتعها قبلها هتطلعلك undefined

دى الحاله العاديه

console.log(X()); // Hello From Func

function X (){

return 'Hello From Func' ;

}

و دى الحاله التانيه الى بخزن الفانكشن فى فاريبول لو عملت النداء فى الاول هتضرب

console.log(X()); // Error Uncaught ReferenceError: X is not defined

let X = function (){

return 'Hello From Func' ;

}

اما لو النداء فى الاخر هيشتغل عادى

let X = function (){

return 'Hello From Func' ;

}

console.log(X()); //Hello From Func

Mebo

######################################################################  
Function with Parameters  
----------------------------------------------

و ده انى و انا بنادى على الفانكشن ابعت براميترز اقدر استخدمها جواها و ادى الكود ذى ما انت شايف و عامل للبراميترز default عشان لو حاجه اتبعتت فاضيه ولا حاجه متخربش الدنيا

let X = function ( x = 1 , y = 2){

return x+y ;

}

console.log(X(5,6)); // 11

Mebo

######################################################################  
Function Arguments  
----------------------------------------------

دى حاجه built in فى اللغه بيتخزن فيها البراميتر الى باعتها فى call و على فكره انا فى ال call للفانكشن ممكن احط براميتر بزياده و فى الكولينج هياخد فقط الى هو محتاجه فمثلا فى المثال الاتى هياخد اول اتنين بس و الباقى اقدر استخدمهم عن طريق الاريجومينتس عادى خالص و دلوقتى هوريك اذاى ابعت عدد براميتر اكتر من لما يكون الفانكشن محتاج و اذاى استخدم الارجيومنتس جوه الفانكشن   
استخدام الارجيومنتس الاساسى ان ده بديل ال overloading انى اقدر اعمل عدد لا نهائى من الفانكشنز بدون الحاجه لعمل اوفلودينج يعنى مش لازم اكتب 500 فانكشن عشان مثلا كل مره محتاج ازود براميتر ولا حاجه و دى طبعا ميزه فشيخه فى ال JS

دى الفانكشن العاديه بس زايده فيها براميتر عن ما هى محتاجه و هتلاقيها شغاله و ذى الفل و هتاخد فقط اول اتنين براميتر و الباقى هتتجاهله و تخزنه فى الاريجومنتس

function X ( x = 1 , y = 2){

return x+y ;

}

console.log(X(5,6,7,6,9,8)); // 11

دى بقه لما استخدمت الاريجومنتس فى انى اعرف عدد البراميتر الفعلى الى مبعوت

و كمان اجيب كل واحد فيهم بكام

function X(x = 1, y = 2) {

console.log(arguments) // [5,6,7,6,9,8]

console.log('Number of Arguments : ' + arguments.length); //6

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

console.log( 'Argument ' + (i+1) + ' is : ' + arguments[i]); // 5,6,7,6,9,8

}

return x + y; // 5+6 = 11

}

console.log('Result is : ' + X(5,6,7,6,9,8));

ذى ما انت شايف اهه مستخدمتش اى براميتر اصلا و مع ذلك قدرت اعمل كول بكذا شكل و بعت ارجيومنتس بالعبيط و كل حاجه فل الفل و كل كول جاب نتيجه مختلفه و استخدم عدد مختلف من البراميتر مع انه اصلا الفانكشن مفيهاش اى نيله فى البراميترز و ده كله لانى استخدمت الارجيومنتس الى تم تخزينها

function X() {

let Result=0;

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

Result+=arguments[i];

console.log('Result is : ' + arguments[i]);

}

console.log('------------------------');

return Result;

}

console.log('Result is : ' + X(5)); // 5

console.log('##########################');

console.log('Result is : ' + X(5,6)); // 11

console.log('##########################');

console.log('Result is : ' + X(5,6,7)); // 18

console.log('##########################');

console.log('Result is : ' + X(5,6,7,6)); // 24

console.log('##########################');

console.log('Result is : ' + X(5,6,7,6,9)); // 33

انا كمان اقدر ابعت Array أو Object فى ال call و لما اجى اشوفهم بالارجيومنتس ساعتها المفروض انى عشان اوصلهم احط الانديكس بتعهم عشان اشوفهم كويس ذى كده

function X() {

console.log(arguments[0]);

}

X([5,6,7,6,9]); // Array

X({

name:'Mahmoud',

Age:25

}); // Object

و ممكن كمان ابعت فى الكول Array with Object عادى خالص و معاهم براميتر عادى كمان وكل واحد فيهم هيبقى فى انديكس خاص بيه افتح الكونسول و هتفهم كلامى

function X() {

console.log(arguments);

}

X(2,

[5, 6, 7, 6, 9], {

name: 'Mahmoud',

Age: 25

}

); // normal var and Array and Object

Mebo

######################################################################  
Higher Order Function  
----------------------------------------------

ده يعنى فانكشن بتعمل ريتيرن لفانكشن تانى جواها و بتبقى كده

function X() {

return (function () {

return 15 ;

})() // Iam Anonymous Func

}

console.log(X());

لاحظ ان الفانكشن الى جوه الفانكشن مش هتشتغل الا لما اخليها self Invoke و ده معناه انى اخليها بتنادى على نفسها و تشتغل لوحدها و دى ليها طريقه معينه هتعرفها بعد شويه

function X() {

return (function () {

console.log('Iam Anonymous Func') ;

})()

}

X();

Mebo

######################################################################  
functions Types  
----------------------------------------------

الفانكشن العاديه

function MyFunc() {

console.log('Iam Anonymous Func');

}

MyFunc();

الفانكش ال Anonymous الى بخزنها فى فاريبول و ملهاش اسم و بتاخد اسمها بعد كده فى الكول بالفاريبول

let MyFunc = function () {

console.log('Iam Anonymous Func');

}

MyFunc();

ال arrow Function و دى يا سيدى التطور الطبيعى لل anonymous هو لغى اسم الفانكشن و لغى كلمه فانكشن دى اصلا و بقى شكلها كده

let MyFunc = () => {console.log('Iam Arrow Func');}

MyFunc();

و لو حابب تبعت اى براميتر بتبعته جوه القوس الى قبل السهم بالشكل ده

let MyFunc = (num) => {console.log('Iam Arrow Func with : ' + num);}

MyFunc(5);

و ممكن تتكتب البراميتر من غير اقواس لو هى براميتر واحد بس ذى ما بتشوف فى انجولار

let MyFunc = Res => {console.log('Iam Arrow Func with : ' + Res);}

MyFunc(5);

و ممكن من غير اقواس فى الناحيه اليمين برضوا

let MyFunc = Res => console.log('Iam Arrow Func with : ' + Res)

MyFunc(5);

ال Pure function دى الانكشن النضيفه الى هى من الاخر بتستخدمش اى حاجه من براها بمعنى اصح بتاخد الانبوت بتعها من اللبراميتر و مش بتاخد اى قيمه من القيم الجلوبال تستخدمها لان لو دخل فيها حاجه من الحاجات الجلوبال ممكن تصعب الموضوع فى حته الديباجينج و فايده البيور فانكشن انها بتسهل عليك الديباجينج جدا

دى فانكشن بيور اهه

function MyFunc(num) {

console.log(num);

}

MyFunc(5);

دى بقه Not a pure Function و غير مستحب استخدامها بصراحه

let x = 6

function MyFunc() {

console.log(x);

}

MyFunc();

Mebo

######################################################################  
Scopes  
----------------------------------------------

بص يا سيدى الاسكوب بيبقى الفاريبول ليه الاسكوب بتاعه داخل القوس الكبير بتاعه بيبقى لوكال بالنسبه للقوس الى هو فيه و جلوبال بالنسبه للابناء

let x = 6

function MyFunc() {

let y = 7;

console.log(x + y);

}

console.log(x); // 6 as it Global

console.log(y); // error undefined as it is Local in Function Scope

MyFunc();

هنا الاتنين هيدوا ايرور لان انا حددت سكوب معين لل X يبقى نافع فيه و اى حاجه براه مش هتشوفه من الاخر كده الفاريبول بيمشى من جوه لبره و اول ما بيلاقى فى اقواس مبيقدرش يتقرأ براها بقه ساعتها بينما اى حاجه موجوده جوه جنبه او تحته ساعتها بيدخل جواها عادى يعنى من الاخر الاب بيدى للابناء انما الاب بيخدش من الابناء حاجه ده ملخص الكلام و شكرا على كده

{

let x = 6

function MyFunc() {

let y = 7;

console.log(x + y);

}

}

console.log(x); // error undefined as it is Local in {} Scope

console.log(y); // error undefined as it is Local in Function Scope

MyFunc();

Mebo

######################################################################  
Arrays  
----------------------------------------------

بص دى ياعم بخزن فيها كميات كبيره من المعلومات بدل ما كل شويه اروح اخزن فى فاريبول فقالك لا انا اعمل وعاء يشيل حبه حلوين و لما احتاج حاجه فيهم اجبها برقم هديهالها و ده رقم الانديكس يعنى كان واحد راح يعد فى فندق ادوله غرفه برقم لما يحبوا يخدموه يكموا الغرفه بتعته على اساس رقمها او عنوانها عندهم   
الانديكس يبدأ من ال 0

اذاى بقه اعمل اراى بص يا سيدى الكلام ده اتشرح فى دوكمنتيشن تانيه لما اتهرى فهنا الشرح هيبقى بسيط و مش هشرح الفانكشنز الخاصه بالاراى لانى شرحها فى مكان تانى

*// Initialize new array***var *names*** = [**'John'**, **'Mark'**, **'Jane'**];  
**var *years*** = **new *Array***(1990, 1969, 1948);  
*// Different data types***var *john*** = [**'John'**, **'Smith'**, 1990, **'designer'**, **false**];

فى طريقتين لانشاء الاراى بس يفضل الطريقه الى من غير new هههههههه ذى ما انت شايف فى الجافا سكريبت الاراى عامله ذى الليست ممكن تاخد قيم من نوع واحد و ممكن تبقى منوعات ذى ما انت شايف

اذاى انادى على حاجه جوه الاراى عشان اشوفها بقه ...... باسم الاراى و رقم الانديكس الى هو عنوان العنصر داخل الاراى

***console***.log(***names***[2]); *// show element in index 2 => Jane****console***.log(***names***); *// show all elements in Array names****console***.log(***names***.length); *// show the length of array*

لما اعوز اضيف بقه للاراى فى كذا طريقه

الطريقه الاولى انى بكتب الاندكس الى هحط فيه الحاجه بس غالبا لو المكان ده فيه حاجه هيروح عامل اوفرايت عليها و يمسح القديم و يكتب الجديد ففى الغالب بشوف اللينس بتاع الاراى و هو ده الى بحط فيه على اساس ان اللينس هيساوى رقم اخر انديكس +1

***names***[1] = **'Ben'**;  
***names***[***names***.length] = **'Mary'**;

الطريقه التانيه بالفانكشنز

*// Different data types***var *john*** = [**'John'**, **'Smith'**, 1990, **'designer'**, **false**];

***john***.push(**'blue'**);  
***john***.unshift(**'Mr.'**);  
***console***.log(***john***);

بوش بتضيف فى الاخر و انشيفت بتضيف فى الاول

لما اعوز امسح حاجه بقه بالفانكشز برضوا و مشروح الكلام ده بالتفصيل فى دوكمنت تانى

***john***.pop();  
***john***.shift();

طيب لما عوز اعمل سيرش ساعتها بتسيرش بفانكشن search او indexOf

**var *isDesigner*** = ***john***.indexOf(**'designer'**) === -1 ? **'John is NOT a designer'** : **'John IS a designer'**;  
***console***.log(***isDesigner***);

Mebo

##################################################

Array Properties

-------------------------

arrName.length;

بتعد العناصر الى جوه الاراى

var arrName = ["a","b","c","d","e"]; // array

console.log(arrName.length); // 5

Mebo

##################################################

arrName.toString();

بيحول الارى لاسترينج يا عمنا

var arrName = ["a","b","c","d","e"]; // array

console.log(arrName.toString()); // "a,b,c,d,e"

Mebo

##################################################

arrName.join(" \* ");

بيحول الارى لاسترينج يا عمنا

و بيحط بين كل عنصر و عنصر النجمه الى انا حاططها كعلامه فصل دى و لو حطيط & هيحط بين كل عنصر و عنصر & هو فيه شبه من ال split

بس الفرق بقه ان واحد بيحول لاسترينج من اراى الى هو الجوين ده

و الاسبليت بقه العكس :D

var MyArray = ["a","b","c","d","e"]; // array

console.log(MyArray.join("-")); // "a-b-c-d-e"

Mebo

##################################################

arrName.splice(2, 0, "Lemon", "Kiwi");

ده بيستقطع و بيضيف فى نفس الوقت

هنا معناها ابدأ من اول انديكس 2

متمسحش حاجه

ضيف عنصرين جدا بقه فى المكان الى انت واقف فيه العنصر هما لمون و كيوى

لو حبيت تمسح حاجه بقه بتغير الصفر لعدد العناصر الى عاوز تمسحها من بعد الانديكس 2 الى انت محدده ده لو مش عاوز تضيف اى عناصر جديده و تمسح بس متكتبش اى عناصر جديده يعنى امسح اللمون و الكيوى

var MyArray = ["a","b","c","d","e"]; // array

console.log(MyArray); // ["a","b","c","d","e"]

MyArray.splice(2, 0, "Lemon", "Kiwi") //add 2 Elements from index 2 & delete 0 Elements

console.log(MyArray); //  ["a", "b", "Lemon", "Kiwi", "c", "d", "e"]

MyArray.splice(4, 2) // Will Remove 2 Elements from index 4

console.log(MyArray); //  ["a", "b", "Lemon", "Kiwi", "e"]

Mebo

##################################################

arrName.fill(“stringValue”, start index , End index);

دى مهمتها استبدال عناصر الاراى بالاسترينج الى انا محدده من الاندكس الفلانى الى الانديكس الفلانى الى انا بحددهم برضوا ببدايه و نهايه ذى ما انت شايف

var MyArray = ["a","b","c","d","e"]; // array

console.log(MyArray); // ["a","b","c","d","e"]

MyArray.fill("MB", 1, 3) // Replcae Elements to "MB" String from index 1 to 3

console.log(MyArray); // ["a", "MB", "MB", "d", "e"]

Mebo

##################################################

MyArray.copyWithin( start of paste index , start of copy index , end of copy index)

من الاخر كده انا بقوله هاتلى العناصر الفلانيه من الانديكس 2 الى الانديكس 4 و اعملها كوبى و حطهم بقه من بدايه الانديكس رقم واحد بحيث يحلوا محل العناصر الى من اول العنصر رقم واحد لحد ما يخلصوا الى هو عند العنصر 3 لانى واخد عنصرين مكبيرهم و ده مثال تفهم منه

var MyArray = ["a","b","\*","\*","e"]; // array

console.log(MyArray); // ["a", "b", "\*", "\*", "e"]

MyArray.copyWithin( 1 ,2, 4) // copy elem from index 2 to 4 and paste it from index 1 to 3

console.log(MyArray); // ["a", "\*", "\*", "\*", "e"]

Mebo

##################################################

arrName.slice(3);

دى يعنى هاتلى العناصر من الاول لحد الاندكس رقم تلاته امسحهم بقه و هاتلى الباقى من الاراى

var MyArray = ["a","b","c","d","e"]; // array

console.log(MyArray.slice(3)); // ["d", "e"]

arrName.slice(3,6);

دى يعنى هاتلى العناصر الى ما بين 3 الى 6 و الباقى امسحه يا حج

var MyArray = ["a","b","c","d","e","F","G"]; // array

console.log(MyArray.slice(3,6)); // ["d", "e","F"]

Mebo

##################################################

arr1.concat(arr2, arr3);

ده بقه هيضيف عناصر اراى 1 على 2 على 3 و هيعمل احلى تجميعه بقه حطها فى اى اراى فاريبول بقه و استخدمها كله فى بعضع يا كبير :D

var arr1 = ["a","b","c"]; // array

var arr2=["d","e","F"]

var arr3=["G"] // or you can right it like string arr3="G"

console.log(arr1.concat(arr2, arr3));//["a", "b", "c", "d", "e", "F", "G"]

Mebo

##################################################

arrName.reverse();

بتعكس الترتيب بتاع عناصر الاراى

var arr1 = ["a","Apple","c"]; // array

var x = arr1.reverse()

console.log(x); // ["c", "Apple", "a"]

Mebo

##################################################

arrName.find(predicate Function);

بيعمل سيرش جوه الاراى اول قيمه فقط بتحقق الشرط الى جواه الفانكشن الى جواها و هى ذى ما انتوا شايفين دى بتتكتبش مجرد حاجه بيدور عليها لا ده انت بتعمل جوه فانكشن الفايند دى برديكيت فاكشن الى هو بتدور جواها بفانكشن خاصه بالبحث و ادى مثال

var arr1 = [1,2,3,4,5,6,7,8,9,10]; // array

var x = arr1.find((Value,index)=>(

Value>3 && index>5

))

console.log(x); // 7

ده عباره عن ايه ؟؟

الاراى اهه و انا بقوله خشلى فى الاراى دورلى على عنصر يكون القيمه بتعته اكبر من 3 و الانديكس بتاعه اكبر من 5 فده بينطبق على العنصر 7

البرديكيت فانكشن دى عامله ذى حوارات ال LINQ فى السى شارب

الفانيد بترجع أول عنصر بيحقق الشرط الى هى قايله عليه جواها طب لو عاوز مجموعه العناصر الى بتحقق الشرط تستخدم ايه بقه ؟؟؟

**Filter**

var arr1 = [1,2,3,4,5,6,7,8,9,10]; // array

var x = arr1.filter((Value,index)=>(

Value>3 && index>5

))

console.log(x); // [7, 8, 9, 10]

Mebo

##################################################

arrName.Every(“Predicate Func”) VS arrName.some(“Predicate Func”)

Every لو كل العناصر حققت الشرط هيطلعلك True غير كده هطلعلك false

Some لو عنصر واحد حقق الشرط هيطلعلك True و الفولس دى مش هتشوفها الا لو كل العناصر منفعتش بنفعه و محققتش الشروط

var arr1 = [1,2,3,4,5,6,7,8,9,10]; // array

var x = arr1.every((Value,index)=>(

Value>3 && index>5

))

console.log(x); // false

var arr1 = [1,2,3,4,5,6,7,8,9,10]; // array

var x = arr1.some((Value,index)=>(

Value>3 && index>5

))

console.log(x); // true

Mebo

##################################################

arrName.indexOf("Mebo");

بيعمل سيرش جوه الاراى اول قيمه بتساوى الى جوه الانديكس اوف و بيجيب الاندكس بتعها على طول واضح هو بيجيب ايه من الاسم اهه خلى بالك الفانكشن حساسه و اى تغيير فى الحروف كابتل او سمول بيفرق جامد

var arrName = [1,2,3,"Mebo",5,6,7,8,9,10]; // array

console.log(arrName.indexOf("Mebo")); // 3 found

console.log(arrName.indexOf("MEbo")); // -1 not found

و كذلك

arrName.lastIndexOf("Mebo");

بس دى بتمشى من اخر الاراى لاوله بس غالبا دى معموله عشان البرفورمنس لانها بتطلع نفس النتايج برضوا

var arrName = [1,2,3,"Mebo",5,6,7,8,9,10]; // array

console.log(arrName.lastIndexOf("Mebo")); // 3 found

console.log(arrName.lastIndexOf("MEbo")); // -1 not found

Mebo

##################################################

arrName.includes(“Mebo”);

دى العنصر موجود فى الاراى هترجع ترو مش موجود هترجع فولس

var arrName = [1,2,3,"Mebo",5,6,7,8,9,10]; // array

var x = arrName.includes("Mebo");

console.log(x); // true

var x = arrName.includes("MEbo");

console.log(x); // false

Mebo

##################################################

arrName.sort();

بترتب الاسترينج الى جوه الاراى

A-Z

لو عاوز ترتب ارقام يبقى ليها فانكشن تانيه ابحث عنها بقه

عشان هتضيف جوه السورتينج شويه حاجات عشان ترتب ارقام مش سترينج

var arrName = ["Mahmoud","Ahmed","Badawy","Said","Fouad"]; // array

console.log(arrName.sort()); //  ["Ahmed", "Badawy", "Fouad", "Mahmoud", "Said"]

لو جيت ترتب الارقام هتلاقى الفانكشن خرفت لانه بيرتب الارقام على انها استرينج فبخلى ال 1 جنبه 10 و ال 100 عشان كلهم بيبدأوا ب 1 ذى ما انت شايف كده

var arrName = [5,1,3,4,2,9,7,6,10,8]; // array

console.log(arrName.sort()); // [1, 10, 2, 3, 4, 5, 6, 7, 8, 9]

طب حل المشكله دى ايه ؟؟

الكومبير فانكشن بقه الى جوه السورتينج و دى شغلها بيبقى كده بتقارن العنصر الاول بالعنصر التانى و بتاخد الى بيحقق الشرط الى جواها

var arrName = [5, 1, 3, 4, 2, 9, 7, 6, 10, 8]; // array

var SortedArray = arrName.sort(function (FirstElem, SecElem) {

if (FirstElem > SecElem) {

return 1; // return FirstElem

} else if (FirstElem < SecElem) {

return -1; // return SecElem

} else {

return 0; // two Elem are Equal

}

})

console.log(SortedArray); // [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

arrName.reverse();

دى بتعكس ترتيب العناصر

var arrName = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]; // array

var SortedArray = arrName.reverse()

console.log(SortedArray); // [10, 9, 8, 7, 6, 5, 4, 3, 2, 1]

و ده لينك عشان تفهم منه ترتيب الارقام فى الاراى يا اسطى

https://www.w3schools.com/js/js\_array\_sort.asp

Mebo

##################################################

Array.isArray(arrName);

دى بتشوف هل الاراى دى اراى ولا لأ هههههههههههههههه

var arrName = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]; // array

console.log(Array.isArray(arrName)); // true

console.log(Array.isArray("arrName")); // false as it is String

Mebo

##################################################

ده بقه الكومبيلت ريفرنس للريجيولار اكسبريشن

https://www.w3schools.com/jsref/jsref\_obj\_regexp.asp

و ذاكره لو مفهمتش من فيديوهات أسامه الزيرو طبعا

Mebo

##################################################

arrName.toString()

دى بتحول الاراى ل String

arrName.Join(“علامه”)

دى بتحول الاراى ل String بس الفرق انها بتضيف بين كل عنصر و عنصر علامه انا عاوها بمزاجى

var arrName = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]; // array

console.log(arrName.toString()); // "1, 2, 3, 4, 5, 6, 7, 8, 9, 10"

console.log(arrName.join(" -> ")); // "1 -> 2 -> 3 -> 4 -> 5 -> 6 -> 7 -> 8 -> 9 -> 10"

arrName.Split(“علامه”)

دى عكس الجوين بقه بتحول الاسترينج لاراى و بتاخد كل اليمينت لما تشوف العلامه الى انا حططها على سبيل المثال انا هقوله كل اما تشوف علامه # فى الاسترنجايه دى خدلى الكلام الى قبلها فى المنت و هكذا لحد ما تخلص السطر كله فلو مثلا موجود علامتين وسط الكلام هتلاقى انه كون أراى مكون من 3 عناصر ذى كده

var arrName = "Mahmoud#Ahmed#Badawy"; // String

console.log(arrName.split("#")); //["Mahmoud", "Ahmed", "Badawy"]

Mebo

######################################################################  
Looping Throw Elements  
----------------------------------------------

مره هجيبها بطريقه ال for With Length و مره عن طريق ال For Of

var arr1 = [1,2,3,4,5,6,7,8,9,10]; // array

///////////////////// for ////////////////////////

for(var i=0;i<arr1.length;i++){

console.log(arr1[i]); // 1 to 10

}

////////////////////// for Of ///////////////////////

for(let x of arr1){

console.log(x); // 1 to 10

}

Mebo

######################################################################  
forEach  
----------------------------------------------

دى يا اسطى بتمسك الاراى كده تخدها تفنطها و تشتغل عليها بمزاجها يعنى هى بتاخد كل عنصر من عناصر الاراى تدخله جواها تعمل عليه بروسيسنج معينه بحيث تطلعه بالشكل المطلوب

////////////////////// foreach ///////////////////////

arr1=[1,2,3,4,5,6,7,8,9,10]

arr1.forEach( (Elem,Index)=>{

console.log(Elem +" ==> "+Index)

})

/\*\*

\* Elem ==> index

\*1 ==> 0

to

\*10 ==> 9

\* \*/

////////////////////// foreach ///////////////////////

arr1=[3,7,1,5]

arr1.forEach( (Elem)=>{

console.log(Elem)

})

/\*\*

\* 3

\* 7

\* 1

\* 5

\*/

من عيوبها انها مش بتعمل ريتيرن لحاجه و عشان كده اخترعوا ال Map

Mebo

######################################################################  
map  
----------------------------------------------

دى بقه بتعمل نفس الى بتعمله ForEach بس بتتميز عنها انها ممكن تعمل جواها بروسيسينج على الاراى و ترجع الناتج فى اراى جديده ذى مثلا انا عاوز ازود على كل عناصر الاراى بتعتى 1 فالماب هتوفرلى انى ازود علي كل عنصر واحد و ارجع الاراى الجديده يعنى هحفظ الناتج الجديد فى اراى جديده بالنتايج الجديده وده لانها بتدعم ال Return و الكونسول برضوا على فكره

////////////////////// map ///////////////////////

arr1=[3,7,1,5]

var NewArr = arr1.map( (Elem)=>{

return Elem+1;

})

console.log(NewArr); //[4, 8, 2, 6]

Mebo

######################################################################  
reduce  
----------------------------------------------

دى ياعم بتمشى على كل عناصر الاراى تقسمهم لعنصر سابق و عنصر حالى و لو عاوز تعمل عليهم بروسيسنج بقه بتعمله و ترجعلك الناتيج النهائى فمثلا انا هستخدمها فى جمع كل عناصر الاراى مع بعض هتخش على اول عنصر تخليه عنصر سابق و تانى عنصر هو عنصر حالى و تجمعهم على بعض و يبقى كده ناتيج جمعهم هو عنصر سابق و تروح متحركه للعنصر التالت و يبقى ده العنصر الحالى فتجمع العنصر السابق الى هو ناتج جمع الاتنين الى فاتوا + العنصر الحالى و ترجع نتيجه جمع التلاته مع بعض و تحطهم فى فاريبول جديد و يبقىده دلوقتى العنصر السابق و العنصر الرابع هو العنصر الحالى و هكذا بقه لحد ما تخلص شغل على الاراى كلها

arr1=[3,7,1,5]

var NewArr = arr1.reduce( (Prev, Current)=>{

return Prev + Current;

})

console.log(NewArr); // 16

/\*\*

\* 3 + 7 = 10

\* 10 + 1 = 11

\* 11 + 5 = 16 so final Result is 16

\*/

Mebo

######################################################################  
Multi Dimension Array  
----------------------------------------------

دى يعنى من الأخر الاراى تبقى جواها اراى تانيه طيب و بتتكتب اذاى دى و بيتقرى من جواها اذاى   
بتقول اسم الاراى و الانديكس بتعها و ليكن الانديكس الاول و جواه اراى ده طيب اقرأ الى جواه اذاى ؟؟ اروح عامله انه يقرأ اندكس كمان عشان يجيب الفاليو بتعته كانك بتقول انا عاوز محمود الى فى شارع رقم 1 جوه البيت رقم 4

و ادى مثال

var arr=[

[1,2] ,

[3,4] ,

[5,6]

];

for (var i =0 ; i < arr.length ; i++ ) {

for (var j =0 ; j < arr[i].length ; j++ ) {

console.log(arr[i][j]); // 1,2,3,4,5,6

}

}

////////////////////// و ممكن نكتبها كده //////////////////////

arr.forEach(element => {

element.forEach(Elem=>{

console.log(Elem); // 1,2,3,4,5,6

})

});

Mebo

######################################################################  
Destructing Assignment for Array   
----------------------------------------------

ده انه اخد كل عنصر فى الاراى اسجله فى فاريبول بشكل مباشر من غير ما اعد فتت فيها و ده مثال للتوضيح

var arr=[1,2,3,4];

var v1= arr[0];

var v2= arr[1];

var v3= arr[2];

var v4= arr[3];

console.log(v1,v2,v3,v4);

////////////////////// Destructing Assignment ///////////////////////

var arr=[1,2,3,4];

var [v1,v2,v3,v4] = arr;

console.log(v1,v2,v3,v4);

Mebo

######################################################################  
Destructing Assignment for Obj   
----------------------------------------------

نفس فكره الى قبله بس الفكره هنا الى الفاريبول لازم تبقى بتساوى نفس اسم البروبارتى عشان تعرف تعمل معاها مابينج بسهوله

var obj={num1 : 1 , num2 :2 , num3 : 3, num4 : 4};

var v1= obj.num1;

var v2= obj.num2;

var v3= obj.num3;

var v4= obj.num4;

console.log(v1,v2,v3,v4);

////////////////////// Destructing Assignment ///////////////////////

var obj ={num1 : 1 , num2 :2 , num3 : 3, num4 : 4};

var {num1,num2,num3,num4} = obj;

console.log(num1,num2,num3,num4);

Mebo

######################################################################  
Spread Operator   
----------------------------------------------

ده يا سيدى بيفرد الاراى بحيث بدل ما استخدم عنصر عنصر لوحده لأ انا ممكن استخدمهم كلهم من غير ما اضطر اكتب الانديكسيس بتعتهم و من الاخر كده مش هقدر افهم الا بمثال

بص ياعم الحج انا عندى فانكشن بتاخد 3 براميترز و عندى اراى و الاراى فيها 3 عناصر فعشان ابعتهم المفروض هكتب كل عنصر بالانديكس بتاعه فلقوا ان الموضوع ده مرهق خصتا لو انا عاوز استخدم اراى كبيره على فانكشن مثلا فيها 20 براميتر او 100 او 1000 فيعملوا ايه ؟؟

عملوا الاسبريد براميتر و ادى مثال يا اسطى هتحس بعده ان الدنيا سهله بالاسبريد بدل العك الى كان بيتعمل الاول

var arr=[ 1 , 2 , 3, 4];

function MyFunc(a,b,c,d){

console.log(a+b+c+d);

}

MyFunc(arr[0],arr[1],arr[2],arr[3]); // 10

////////////////////// Spread Operator ///////////////////////

MyFunc(...arr); // 10

Mebo

######################################################################  
Spread Operator for Variable  
----------------------------------------------

var arr=[ 1 , 2 , 3, 4];

var [a,b,...c] = arr;

console.log(a); // 1

console.log(b); // 2

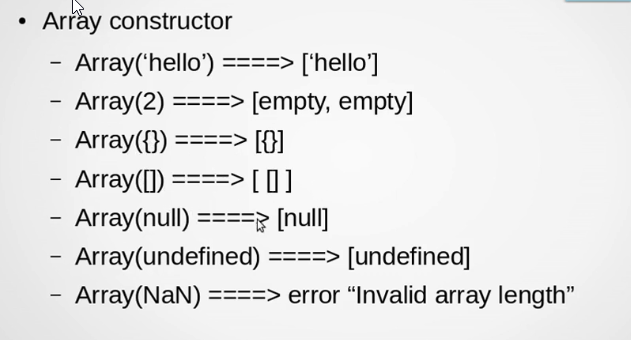
console.log(c); // [3,4]

ذى ما انت شايف هو بياخد اول رقم من الاراى يحطه فى اول فاريبول و تانى رقم فى تانى فاريبول و بعد كده لما لقى ال c معمولها سبريد اوبريتور عرف ساعتها ان ده فاريبول عباره عن اراى هيتملى فيها باقى عناصر الاراى و عشان كده السى اخدت الرقمين 3 و 4 فى اراى لوحدها و الملحوظه المهمه ان الاسبريد اوبريتور فى الحاله دى بيتعمل فى اخر عنصر فى الارارى مش فى النص ولا فى الاول عشان ده هيبقى كده غلط

Mebo

######################################################################  
Array Constructor   
----------------------------------------------

بيتكتب ذى طريقه كتابه اى اوبجيكت فى كلاس بس الفرق ان داخل الكونستراكتور بتحط فيه العناصر الخاصه بالاراى الى بتتكون بس من عيوبه انه لو كتبت اراى فيها عنصر واحد رقمى هيعتبره ده ال size بتعها او length و هيقولك انها طولها بيساوى الرقم الى انت حاطه فيها ده و الاراى فاضيه لسه اما لو حطيتله سترينج مش رقم هيحطه فى اراى و شكرا



var arr= new Array(1,2,3);

console.log(arr); // [1,2,3]

var arr= new Array(5);

console.log(arr); // Empty Array [empty × 5] => length: 5

طب ايه حل مشكله العنصر الواحد الى بيجيب Empty ده ؟؟

الحل انى يا كبير استخدم Array.of و دى بتعمل اراى من اى حاجه تديهالها يعنى اديتها رقم اشطه اتنين اشطه اى حاجه هتحطها فيها هتعملك منها اراى و خلاص

var arr= Array.of(5);

console.log(arr); // [5]

arr2 = [5,6,4]

var arr= Array.of(arr2);

console.log(arr); // [5,6,4]

Mebo

######################################################################  
Array from   
----------------------------------------------

دى يا سيدى عباره عن فانكشن فى الاراى كونستراتور دى من الاخر بتستخلص الاراى من اى حاجه عباره عن اوبجيكت او سترينج او من الاخر اى حاجه ليها Length اديتها اراى هتفردهولك و تعمل منه اراى اديتها استرينج هتفردهولك حروف و تعمل منه اراى اديتها طول و و قولتها املهولى من فانكشن كل مهمتها ترجع الانديكس الناتج من اروفانكشن هتشتغلك على الموضوع ده و تجيبلك منه انديكسات و هتعملك عليهم بروسيس كمان

var arr2 = [5,6,4]

var arr= Array.from(arr2);

console.log(arr); // [5,6,4]

var arr= Array.from("Hello");

console.log(arr); // ["H", "e", "l", "l", "o"]

var arr= Array.from({length:5} , (elem,index)=>index+1);

console.log(arr); // [1, 2, 3, 4, 5]

أخر واحده دى بيتم استخدامها فى انى اعمل مجموعه عناصر بسرعه كده من غير تعب يعنى :D بفانكشن بسيطه مع كتابه اللينس بتاع الاراى يعمل احلى شغل تخيل مثلا انك كتبت مليون فى اللينس هيروح من غير اى تغيير فى الكود هو بس هتكتب مليون فى الطول هتلاقى المليون عنصر اتعملوا من غير ما تكتب حرف

Mebo

######################################################################  
Arrays are By Reference values as it considered Object   
----------------------------------------------

الاراى تعتبر نوع من انواع الاوبجيكت و بالتالى هى الفاليو بتعتها بتشتغل باى ريفرنس يعنى لو انا سويت اتنين اراى ببعض و جيت غيرت فى واحده منهم اى قيمه لاى عنصر ساعتها هيأثر فى التانيه على طول

var arr2 = [5,6,4]

var arr= arr2;

arr[0] = 10;

console.log(arr2); //  [10, 6, 4]

Mebo

######################################################################  
Type Conversion  
----------------------------------------------

بص على الصفحه دى هتفهم كل حاجه الموضوع مش محتاج شرح اصلا

<https://www.w3schools.com/js/js_type_conversion.asp>

<https://www.w3schools.com/js/js_number_methods.asp>

و دول اهم الفانكشنز المعروفه

* Number()
* parseInt()
* parseFloat()
* toString()
* Number()

Mebo

######################################################################  
Objects  
----------------------------------------------

الفرق بين الاوبجيكت و الاراى ان الابوجيكت بيهتمش بالترتيب و الاندكس و الكلام ده انما فى الاراى لا بيهتم بالترتيب و الانديكس و التفاصيل دى و كمان الاوبجيكت فى استدعائها فيها شبه من ال Associtive array   
بالمناسبه الاوبجيكت بيتحط فيه بروبرتيس من اى نوع عادى خالص و طريقه كتابته من جوه عامله ذى بتاعت جيسون مع بعض الاختلافات البسيطه

Mebo

######################################################################  
Objects and properties  
----------------------------------------------

اذاى بعمل اوبجيكت اذاى بقه ؟  
اول طريقه اهه

*// Object literal***var *john*** = {  
 **firstName**: **'John'**,  
 **lastName**: **'Smith'**,  
 **birthYear**: 1990,  
 **family**: [**'Jane'**, **'Mark'**, **'Bob'**, **'Emily'**],  
 **job**: **'teacher'**,  
 **isMarried**: **false**};  
*// How to show values of properties in Object****console***.log(***john***.**firstName**);  
***console***.log(***john***[**'lastName'**]);  
**var *x*** = **'birthYear'**;  
***console***.log(***john***[***x***]);

هنا عملت الاوبجيكت و خزنت جواه حاجات و بعد كده عاوز اظهرها الطريقه الاولى فى الاظهار انى اعملها ذى اى اوبجيكت جواه بروبرتى فبناده باسم الاوبجيكت دوت البروبرتى   
الطريقه التانيه بعماله ذى الاسوسيتيف اراى تماما فبحط اسم البروبرتى فى سترينج و بعد كده بعامل الاوبجيكت كانها واراى و الاسترينج بتاع البروبرتى ده عباره عن انديكس   
الطريقه التالته ذى التانيه بس الفرق انى حطيت الاسترينج جوه فاريبول و استخدمت الفاريبول بس كده   
لما اعوز اعمل اى تعديل على اى حاجه داخل الاوبجيكت بعمل كده

***john***.**job** = **'designer'**;  
***john***[**'isMarried'**] = **true**;  
***console***.log(***john***);

و لو عوزت اضيف بروبرتى داخل الاوبجيكت بالدهلكه كده بره بعملها كده

***john***.**Age** = 15;  
***console***.log(***john***.**Age**);

و فى طريقه تانيه خالص بتعمل بيها الاوبجيكت برضوا و دى طريقه اسمها طريقه نيو بص ده الكود كله و فى اخره طريقه ال new و لاحظ انى حطيت فانكشن هنا عشان تفهم ان الاوبجيكت بيقبل جواه فانكشنز :D

*// Object literal***var *john*** = {  
 **firstName**: **'John'**,  
 **lastName**: **'Smith'**,  
 **birthYear**: 1990,  
 **family**: [**'Jane'**, **'Mark'**, **'Bob'**, **'Emily'**],  
 **job**: **'teacher'**,  
 **isMarried**: **false**,  
 **myfunc** : (**function** () {  
 **return 'badawy'**;  
 })()  
};  
*// How to show values of properties in Object****console***.log(***john***.**firstName**);  
***console***.log(***john***[**'lastName'**]);  
**var *x*** = **'birthYear'**;  
***console***.log(***john***[***x***]);  
***console***.log(***john***.**myfunc**);  
***john***.**job** = **'designer'**;  
***john***[**'isMarried'**] = **true**;  
***console***.log(***john***);  
***john***.**Age** = 15;  
***console***.log(***john***.**Age**);  
  
*// new Object syntax***var *jane*** = **new *Object***();  
***jane***.**firstName** = **'Jane'**;  
***jane***.**birthYear** = 1969;  
***jane***[**'lastName'**] = **'Smith'**;  
***console***.log(***jane***);

طبعا الاحسن تشتغل بالطريقه الاولى يعنى :D

Mebo

######################################################################  
Objects and methods  
----------------------------------------------

**var *john*** = {  
 **firstName**: **'John'**,  
 **lastName**: **'Smith'**,  
 **birthYear**: 1992,  
 **family**: [**'Jane'**, **'Mark'**, **'Bob'**, **'Emily'**],  
 **job**: **'teacher'**,  
 **isMarried**: **false**,  
 calcAge: **function**() {  
 **this**.**age** = 2018 - **this**.**birthYear**;  
 },  
 MycalcAge: **function**() {  
 **return** 2018 - **this**.**birthYear**;  
 },  
 MycalcAgeByCons: **function**() {  
 ***console***.log(2018 - **this**.**birthYear**);  
 }  
};  
  
***john***.calcAge();  
***console***.log(***john***.**age**);  
  
***console***.log(***john***.MycalcAge());  
***john***.MycalcAgeByCons();

اول فانكشن بترجع قيمتها فى بروبرتى معمول in the Fly و بالتالى المفروض بنادى للفانكشن ثم بنادى للبروبرتى الى اتعمل جواها عشان نستدعى الحاجه الى هترجع   
تانى فانكشن معموله ترجع فى الريتيرن و بالتالى هنحطها فى كونسول عشان تظهر فى الكونسول   
ثالث حاجه محطوطه بشكل مباشر فى الكونسول و بالتالى مش محتاجين غير اسم الفانكشن هناديها بيه و خلاص و طبعا كل ده تابع للاوبجيكت جون   
this دى معناها ان الكلام ده تابع للاوبجيكت الى انا فيه يعنى من الاخر البروبرتى ده بتاعت جون الى هو الاوبجيكت الى انا فيه جوه اصلا

Mebo

######################################################################  
this  
--------------

لها معانى كسيره فكل مكان بتتحط فيه بيبقى ليها معنى معين فمثلا فى حاله انى عملت اوبجيكت و الاوبجيكت دى جواه بروبرتيس و فانكشنز لو حبيت استخدم حاجه من الاوبجيكت بداخله فبدا من انى اكتب اسم الاوبجيكت تانى بكتب this   
 ادى مثال قبل استخدام this و بعد استخدامها

var Obj = {

FirstName : "Mostafa",

LastName : "Badawy",

Age : 2,

Address : "Cairo",

FullName : function () {

return `I'm ${Obj.FirstName + " " + Obj.LastName}` ;

}

}

console.log(Obj.FullName());

و لنفترض بقه انى غيرت اسم الاوبجيكت اجرى بقه اغير اسمه فى كل مكان انا مستخدمه فيه ؟؟

لأ طبعا و من هنا جت فكره انهم يعملوا this بحيث لو تم تغيير اسم الاوبجيكت ميأثرش على كودى و ادى الكود ب this و طبعا انا عملت كونسول لل this عشان تلاحظوا انه نسخه من الاوبجيكت و ده لانه جوه الاوبجيكت

var Obj = {

FirstName : "Mostafa",

LastName : "Badawy",

Age : 2,

Address : "Cairo",

FullName : function () {

console.log(this); // Obj Object

return `I'm ${this.FirstName + " " + this.LastName}` ;

}

}

console.log(Obj.FullName());

طيب لز استخدمنا كلمه اوبجيكت جوه فانكشن داخل الفانكشن الموجوده فى الاوبجكت ؟؟

فى الحاله دى هتبقى this الى جوه الفانكشن بنت الفانكشن بنت الاوبجيكت ملهاش علاقه بالاوبجيكت لأ هتبقى بتعبر عن ال Global Object الى هوى window اوبجيكت فى الحاله دى وده بقه عباره عن اوبجيكت شايل كل حاجه فى المشروع أو البراوزر يعتبر

var Obj = {

FirstName : "Mostafa",

LastName : "Badawy",

Age : 2,

Address : "Cairo",

FullName : function () {

console.log(this); // Obj Object

InsideFunc();

function InsideFunc(){

console.log(this); // Window Object

}

return `I'm ${this.FirstName + " " + this.LastName}` ;

}

}

console.log(Obj.FullName());

طيب عشان اعرف استخدم الاوبجيكت داخل الفانكشن الى جوه الانكشن دى اعمل ايه ؟؟ فى طريقتين الطريقه القيدمه و الطريقه الجديد

الطريقه القديمه استخدام that

var Obj = {

FirstName : "Mostafa",

LastName : "Badawy",

Age : 2,

Address : "Cairo",

FullName : function () {

let that = this;

InsideFunc();

function InsideFunc(){

console.log(that); // Obj Object

}

return `I'm ${this.FirstName + " " + this.LastName}` ;

}

}

console.log(Obj.FullName());

الحل التانى استخدام ال Arrow Functions و ساعتها مش هحتاج ولا ذات ولا بتنجان بس خلى بالك ان فى الارو فانكشن عشان الدنيا تبقى سليمه طالما هتعمل اسم الفانكشن فى فاريبول لازم الكولينج يبقى تحت

var Obj = {

FirstName : "Mostafa",

LastName : "Badawy",

Age : 2,

Address : "Cairo",

FullName : function () {

console.log(this); // Obj Object

InsideFunc = () => {

console.log(this); // Obj Object

}

InsideFunc();

return `I'm ${this.FirstName + " " + this.LastName}` ;

}

}

console.log(Obj.FullName());

this لو داخل فانكشن عاديه خالص فى الهواء كده لوحدها هتبقى بتعبر عن Global Object الى هوى window اوبجيكت

InsideFunc();

function InsideFunc(){

console.log(this); // Window Object

}

Mebo

######################################################################  
Set and Get and Delete Object Property  
------------------------------------

var Obj = {

FirstName : "Mostafa",

LastName : "Badawy",

age:2

}

console.log(Obj.FirstName); // Get FirstName From Obj

Obj.LastName = "Ahmed";// Set New LastName For Obj

console.log(Obj.LastName); // Get New LastName From Obj

Obj["LastName"] = "Mahmoud"; // Set New LastName For Obj

console.log(Obj["LastName"]); // Get New LastName From Obj

Obj.Address = "Cairo"; // Add New Property For Obj

console.log(Obj.Address); // Get Address From Obj

console.log("--------------------------------------")

console.log(Obj)

delete(Obj.age); // Delete Address Property From Obj

console.log(Obj)

Mebo

######################################################################  
Looping Throw Property  
------------------------------------

دى ياعم عشان يعمل لوب على خواص الاوبجيكت و يجيب منها قيمتها و هى كما هو مبين بالشكل الاتى

var Obj = {

FirstName : "Mostafa",

LastName : "Badawy",

Age : 2,

Address : "Cairo",

FullName : function () {

console.log(this); // Obj Object

InsideFunc = ()=> {

console.log(this); // Obj Object

}

InsideFunc();

return `I'm ${this.FirstName + " " + this.LastName}` ;

}

}

for(let Prop\_key in Obj){

let Key = Prop\_key; // loop give us Prop Names as Key of Obj Array

let value = Obj[Key]; // we us Obj Array to fild value by Key

console.log(Key + " --> " + value );

}

// Check By In if the Proparty in object or not

console.log( 'FirstName' in Obj ); // True

console.log( 'FName' in Obj ); //False

Mebo

######################################################################  
Comparing Between 2 Obj   
------------------------------------

بص لو جبت اتنين اوبجيكت و اديتهم نفس كل حاجه و جيت اقارن بينهم هيطلعوا مش بيساوا بعض عارف ليه ؟؟

عشان فى الاوبجيكتس المقارنه بين اتنين بتبقى By reference و ده معناه انه هيقارن عناوين كل واحده فيهم و لو لقاهم مختلفين يبقى الاتنين بيساووش بعض و هما فعلا الاتنين عمرهم ما هيساوا بعض فى العناوين لان كل فاريبول بيبقى ليه عنوان خاص بيه و بالتالى عشان اخلى الاتنين يساوا بعض لازم اقوله ان الاوبجيكت ده عنوانه هو هو نفس عنوان التانى طب و تتعمل اذاى دى ؟ دى تتعمل عن طريق انك تقوله التانى بيساوى الاول بس كده و بكده بقوا الاتنين بيشاوروا على نفس العنوان و بالتالى اى تغيير فى قيمه الاول او التانى الاتنين هيتغيروا مع بعض و ده لان الاوبجيكت بيتعامل بال reference

تعالى اديك بقه مثالين على الكلام الى فوق ده المقارنه الى هتدينى اختلاف و المقارنه الى هتدينى مفيش اختلاف

var Obj1 = {

FirstName : "Mostafa",

LastName : "Badawy",

}

var Obj2 = {

FirstName : "Mostafa",

LastName : "Badawy",

}

console.log( Obj1 === Obj2 ); //False

Obj2 = Obj1;

console.log( Obj1 === Obj2 ); //Became True

Obj2.FirstName="Mahmoud Ahmed"

console.table(Obj1); // any changes in Obj2 Will Affect Obj1 as you see

Mebo

######################################################################  
Factory   
------------------------------------

و دى يعنى مصنع و ده عباره عن فانكشن بعمل بيها اوبجيكتس كتير ذى منا عايز و بدل ما اكتب كل مره نفس الاوبجكت بنفس الشكل بس بغير الاسم و القيم قالك لأ انا اعمل هو اوبجيكت واحد احطه جوه فانكشن و امررله القيم كبراميتارز و هو ده الفاكتورى و هو حل مشكله التكرار و انا هوريك المشكله و هوريك الفاكتورى الى هو الحل عشان تشوف الفرق

أدى المشكله

var Obj1 = {

FirstName : "Mostafa",

LastName : "Badawy",

}

var Obj2 = {

FirstName : "Mahmoud",

LastName : "Ahmed",

}

الحل بقه عشان مكررش كل شويه الكود

function FactoryFunc(FName,LName){

return {

FirstName : FName,

LastName : LName ,

}

}

let Obj1 = FactoryFunc("Mostafa","Badawy");

let Obj2 = FactoryFunc("Mahmoud","Ahmed");

console.table(Obj1);

console.table(Obj2);

الحل التانى انك تعمل ياباشا كونستراكتور و برضوا ده هيعملك مليون اوبجيكت و هيمررلك ليهم بقيم مختلفه عادى برضوا و ده الشايع فى الاستخدام اصلا

function ClassName(FName,LName){

this.FirstName = FName,

this.LastName = LName

}

let Obj1 = new ClassName("Mostafa","Badawy");

let Obj2 = new ClassName("Mahmoud","Ahmed");

console.table(Obj1);

console.table(Obj2);

طبعا لما يكون فى كلاس هيبقى الدنيا عسل على الاخر

الفرق بين ال Factory VS Constructor

الفاكتورى بتعمل ريتيرن فقط للاوبجيكت و بالتالى مفيش اى ربط بين الاوبيجكتس الى طالعه من و بعضها

ما الكونستراكتور بيعمل انستانس من الاوبجيكت و ده يعنى وريث خواص الاوبجيكت و بالتالى فى رابط بين كل الاوبجيكتس و ده لانهم الاتنين من نفس الكونستراكتور و ابناء لاب واحد الى هو الكلاس

الاستخدام الامثل انك تستخدم طبعا الكونستراكتور و قليل جدا لما تحتاج تستخدم الفاكتورى يكاد يكون معدوم

Mebo

######################################################################  
By Value   
------------------------------------

و ده الى فى معظم الفاريبول العاديه ذى النامبرز و الاسترينج فمثلا هنقول و ليكن معايا اتنين فاريبول و عملنا اساين من واحد لواحد عشان نقوله ان قيمه ده بتساوى قيمه ده ففى الحاله دى تغيير قيمه اى واحد فيهم مش هتأثر خالص فى التانى و ده لان الاتنين مش ليهم نفس العنوان برضوا و بالتالى هيشاورا على قيم مختلفه و اهه ذى ما انت شايف فى الكود

var x = 5;

var y = 2;

console.log(x); // 5

x=y;

console.log(x); // 2 as it take variable from y

y=7;

console.log(x); // 2 as it still have 2 as it does not refer to same addres of y

Mebo

######################################################################  
By Reference   
------------------------------------

و ده معناه ان هخلى اتنين يشاوروا على نفس العنوان و بالتالى اى تغيير فى الاولى هيأثر فى التانيه و ده هتلاقيه موجود فى الاوبجيكتس و الاراي و الليستيس

و ادى مثال

var Obj1 = {

FirstName : "Mostafa",

LastName : "Badawy",

}

var Obj2 = {

FirstName : "Mostafa",

LastName : "Badawy",

}

Obj2 = Obj1;

Obj2.FirstName="Mahmoud Ahmed"

console.table(Obj1); // any changes in Obj2 Will Affect Obj1 as you see

Mebo

######################################################################  
Named Parameters   
------------------------------------

بص يا سيدى الفانكشنز ممكن تبعتلها براميتارز ذى ما انت عارف بس دلوقتى لقوا ان ممكن الواحد لو فى 10 براميتر انا باعتهم ممكن اتلغبط و ابعت برامتيزر فى اماكن غلط فقالك لأ انا و انا بعمل calling ممكن اسمى كل براميتر باسمه عشان ميحصلش لغبطه فحتى لو انا كتبته غلط مش هيبقى فى مشكله و ده مثال

function MyFunc(x, y) {

console.log(x, y);

}

MyFunc(x=5,y=3);

// or

MyFunc(y=5,x=3);

لقوا ان الموضوع ده مرهق برضوا قالك طب بقولك ايه احنا نبعت بدل ما نبعت 20 حاجه كده و نكتب و نوجع ايدينا فى الكود طب ما انا ابعت براميتر اوبجيكت جواه كل البراميترز الى هحتاجها و استقبلها جواه و العب فيه براحتى ذى كده

function MyFunc(obj) {

let My\_x = obj.x;

let My\_y = obj.y;

console.log(My\_x, My\_y);

}

MyFunc({x:3,y:5});

// or

MyFunc({y:5,x:3});

قالك لا ياعم الموضوع برضوا مرهق انا اعمل احسن من ده كله انا اعمل اوبجيكت و استقبله فى اوبجيكت من البراميترز و ده هيبقى كده و هيبقى اسهل فى الكتابه

function MyFunc(obj) {

let {x,y} = obj; // Obj must be same of what in call to map

console.log(x, y);

}

MyFunc({x:3,y:5});

// or

MyFunc({y:5,x:3});

بس خلى بالك انك لازم تعمل براميترز الاوبجيكت الى جوه الفانكش بنفس اسامى البروبرتيس الى انت بعتها فى الاوبجيكت و ات بتعمل كول عشان كده هكتبه كده let {x,y} = obj;

و لو غيرته مش هيعمل Map مش هيبعت البيانات اصلا

Mebo

######################################################################  
Named Parameters   
------------------------------------

let num = 55.634;

console.log(num.toExponential()); // 5.5634e+1

console.log(num.toFixed()); // 56

console.log(num.toFixed(1)); // 55.6

console.log(num.toFixed(2)); // 55.63

console.log(num.toFixed(5)); // 55.63400

console.log(num.toPrecision(2)); // 56

console.log(num.toPrecision(3)); // 55.6

console.log(num.toPrecision(4)); // 55.63

console.log(num.toString(2)); //Binary Num

console.log(num.toString(8)); //Octal Num

toExponential : دى معناها حولى الرقم الى رقم exponential

55.634 = 5.5634 \* 10 = 5.5634e+1

3478.541 = 3.478541 \* 1000 = 3.478541e+3

As 10 = e+1 and 100 = e+2 and 1000 = e+3 in Exponentials

toFixed: دى يا سيدى انت بتشوف انت عاوز تظهر كام ديجيت بعد الدوت و تكتبه فى البراميتر الخاص بالفانكشن و هى بتنفذ و لو انت مبعتش اى برامتير هى بتعتبرك بعت صفر و بالتالى مبعتلهاش اى حاجه مش هتظهر اى كسور بعد الدوت و لو بعتلها 1 هتظهرلك ديجيت واحد بس بعد الدوت و لو بعتلها 9 هتظهرلك 9 ديجيت بعد الدوت بحيث ان لو فى ديجيت فاضيه هى هتخليها صفر ههههههههه شوف فوق و هتفهم

toPrecision : دى يا حج بقه بتشوف انت عاوز تظهر كام ديجيت من اول الرقم من ناحيه الشمال و بتقرب الرقم لاقرب قيمه كمان و هى لو مبعتلهاش اى قيمه بترجع الرقم ذى ما هو و لو بعتلها 1 فعلا بترجع الرقم كله فى ديحيت واحد و لو قولتلها اتنين هترجعولك فى اتنين و لو قولتلها تلاته هترجعهولك كله فى تلاته و هكذا

toString : دى يا سيدى بترجع الرقم على هيئه استرينج و لو انت اديتها ارقم بتبقى الارقام دى عباره عن نظام الارقام يعنى ممكن تبعت فيها 2 مثلا فهترجعلك الرقم على انه Binary و لو بعتلها 8 هترجعهولك ثمانى و هكذا

Mebo

######################################################################  
Special Numbers   
------------------------------------

Infinity ما لا نهايه & سالب Infinity   
NAN

اذاى اعمل انفنيتى ده ؟؟

console.log(1/0); // Infinty

console.log(-1/0); // -Infinty

console.log(Math.pow(2,1024)); // Infinty

console.log(-Math.pow(2,1024)); // -Infinty

طب اذاى اعمل ال NAN ؟

console.log(Infinity/Infinity); // NAN

console.log(3\*"Hi"); // NAN

console.log(3/"Hi"); // NAN

اذاى اعرف الحاجه الى قدامى دى NAN ولا لأ ؟؟ عن طريق فانكشن ال IsNaN الى بترجع ترو او فولس

let CheckNaN = isNaN(Infinity/Infinity)

console.log(CheckNaN);

Mebo

######################################################################  
Special Numbers   
------------------------------------

فى الجافا سكريبت الكسور جمعها مش بيبقى ذى الطبيعى الى نعرفه لان ذى ما انت شايف فى الكود اهه هتلاقى ان 0.1+0.2 != 0.3

let Num1 = 0.1 + 0.2

let Num2 = 0.3

console.log(Num1); // 0.30000000000000004

console.log(Num2); // 0.3

console.log(Num1===Num2); // False

طب نحل المشكله دى اذاى ؟؟

نصحح الارقام و نجمعها و بع ما نجمعها على بعض نرجعها لاصلها تانى

ذى ما انت شايف كده هنا و ممكن نستخدم ال toFixed برضوا

الحل الاول

let Num1 = 0.1\*100 + 0.2\*100; // 30

let Num2 = 0.3 \* 100; // 30

Num1 =Num1/100;

Num2 =Num2/100;

console.log(Num1); // 0.3

console.log(Num2); // 0.3

console.log(Num1===Num2); // True

الحل التانى

let Num1 = 0.1+0.2

let Num2 = 0.3

Num1=Num1.toFixed(1);

Num2=Num2.toFixed(1);

console.log(Num1); // 0.3

console.log(Num2); // 0.3

console.log(Num1===Num2); // False

Mebo

######################################################################  
Template Literals  
------------------------------------

فاكر الكونكاتيونيشن الى فى الاسترينج ؟؟ ايوه هيبقى كده

let x = "Hello";

let y = "Mahmoud Badawy";

console.log(x + " " + y);

طيب هما لقوا ان كل مره يبقى لازم احط سبيس و اظبط الاسترينج عشان يطلع ذى ما انا عاوز قالك طب ما احنا نعمل حاجه تسهل علينا الموضوع بحيث تبقى حاجه كده ارمى فيها القيم جوه الاسترينج و يبقى الدنيا متظبطه مع نفسها من غير كوناكتيونيشن و نيله و ده خلاهم يعملوا حاجه اسمها template Literals و طريقه كتابتها كده

let x = "Hello";

let y = "Mahmoud Badawy";

console.log(`${x} ${y} , I'm Here Do not Worry`);

واخد بالك انى بكتب بين قوسين ليس هما سينجل كوتس او دابل و لكن هى العلامه الى بتتكتب لما بدوس على حرف ال ذ الزرار الى فوق ال Tab من الاخر بس و احنا محولين انحليزى الى هى ` `

و ميزته بقه انك ممكت تعمل جواه عمليات حسابيه كمان على سبيل المثال اهه

let x = "Hello";

let y = 5;

console.log(`${x} ID : ${y\*5} , I'm Here Do not Worry`);

و كمان تقدر تعمل جواه If statement ذى ما انت شايف كده

let x = "Hello";

let y = 25;

console.log(`${x} ${y > 20 ? "Mahmoud":"Mostafa"} , I'm Here Do not Worry`);

Mebo

######################################################################  
string Functions  
--------------------

length  
--------------------------------------------------------------------------------------------------------

var str = "Please locate where 'locate' occurs!";

console.log(str.length); //36

دى فانكشن بتجيب طول الاسترنج بالارقام

Mebo

--------------------------------------------------------------------------------------------------------

str.indexOf(" locate");

دى بقه للبحث فى الاسترينج هل الاسترانجايه الى انا مدهاله جوه القوس موجوده فى الاسترنجايه الاصليه ولا لأ لو موجوده بيطلع هى فى انهى انديكس و عشان تبقى فاهم هو بياخد أول حاجه بيشوفها لو فى نسختين من الحاجه يعنى لو مش موجوده بيرجع -1

var str = "Please locate where locate as you have occurs!";

console.log(str.indexOf("locate")); // 7

Mebo

--------------------------------------------------------------------------------------------------------

str.lastIndexOf(" locate");

ذى الى فوق بس بتمشى فى بحثها من ورا لقدام

var str = "Please locate where locate as you have occurs!";

console.log(str.lastIndexOf("locate")); // 20

Mebo

--------------------------------------------------------------------------------------------------------

str.search("locate");

ذى انديكس اوف بس هنا بتقبل بقه ريجويلار اكسبريشن

var str = "Please locate where locate as you have occurs!";

console.log(str.search("locate")); // 7

Mebo

--------------------------------------------------------------------------------------------------------

str.includes("locate");

ذى بيبحث عن الحاجه و لو لقاها بيرجع true غير كده بيرجع false

var str = "Please locate where locate as you have occurs!";

console.log(str.includes("locate")); // true

Mebo

--------------------------------------------------------------------------------------------------------

str.slice(start, end);

دى بتستقطع حته من الاسترنجايه بتديله بدايتها و نهايتها و هى بتطلعهالك لان كل حرف فى الاسترينج ليه رقم انديكس و بالتالى لما بديله رقم بدايه كلمه و نهايتها هيطلعهالى بالسلايس

var str = "Please locate where locate as you have occurs!";

console.log(str.slice(2, 18)); // ease locate wher

لو اديته البدايه بس يبقى النهايه بتعته باى ديفولت كده هو اخر الاسترنج بقه لدام انت محددتش النهايه فين

Mebo

--------------------------------------------------------------------------------------------------------

str.substring(start, end);

ذى صحبتها الى فوقها بالظبط الى هى سلايس دى

var str = "Please locate where locate as you have occurs!";

console.log(str.substring(2, 18)); // ease locate wher

Mebo

--------------------------------------------------------------------------------------------------------

str.substr(start, length);

ذى صحبتها الى فوقها بس الفرق انى بدل ما اديله النهايه بديله طول الاسترنجايه الى يخدها من اول ما يبتدى

var str = "Please locate where locate as you have occurs!";

console.log(str.substr(2, 10)); // ease locat

Mebo

--------------------------------------------------------------------------------------------------------

str.replace("locate", "W3Schools");

دى بتبدل كلمه فى الاسترينج الاصلى بكلمه تانيه غيرها و المكانين جوه القوص بيقبلوا ريجيولار اكسبريشن و على فكره هى بتبحث عن الكلمه الى هتبدلها و أول كلمه بتحقق الى هى عيزاه بتبدلها و الباقى لأ الا لو انت بقه قولتله جلوبال فى الريجبولار اكسبريشن بقه

var str = "Please locate where locate as you have occurs!";

console.log(str.replace("locate", "W3Schools"));

// Please W3Schools where locate as you have occurs!

دلوقتى انا عملتها بقه بالريجولار اكسبريشن خليتها مش كيس سينسيتيف و كمان خليتها جلوبال و كمان خليتها مالتى لاين يعنى التغيير هيحصل فى اى حاجه مكتوبه اسمها locate حتى لو مكتوبه بشكل كابتل فى سمول و عجن ذى كده

var str = "Please LocaTE where LOcatE as you have occurs!";

var RegExpPattern = /locate/gmi;

// i --> insenetive case i=I

// m --> MultiLines

// g --> global

console.log(str.replace( RegExpPattern, "W3Schools"));

// Please W3Schools where W3Schools as you have occurs!

Mebo

##################################################

ده ريفرنس جميل **للرجيولار اكسبريشن** بقه   
<https://www.w3schools.com/jsref/jsref_obj_regexp.asp>

--------------------------------------------------------------------------------------------------------

str.toUpperCase();

بيحول الحروف الاسترينج الى UpperCase حروف كابتل كلها

Mebo

##################################################

str.toLowerCase();

بيحول الحروف الاسترينج الى LowerCase

Mebo

--------------------------------------------------------------------------------------------------------

text1.concat("elkalb lolo ", text2);

طبعا انتوا عارفين ان الفانكشن دى بتعمل كونكاتيونيشن و بتضيف استرنجايه لاسترنجايه و كمان لو اراى بتضيفهم لبعض

var text1 = "Hi";

var text2 = " Mahmoud";

console.log(text1.concat(text2)); // Hi Mahmoud

Mebo

--------------------------------------------------------------------------------------------------------

var txt = "a,b,c,d,e"; // String

txt.split(",");

سبليت دى بتقسم الاسترنجايه لاراى فيها عناصر على حسب العلامه او الكلمه الى انا حاططه اجواها فمثلا انا هنا حاططله فاصله هو كل ما هيشوف الفاصله دى هياخد الى قبل الفاصله و يحطها كعنصر فى الاراى و يخش على الى بعده و هكذا

var txt = "a,b,c,d,e"; // String

let array = txt.split(",");

console.log(array); // [a,b,c,d,e] Array

Mebo

--------------------------------------------------------------------------------------------------------

Join(Separetor)

دى يا سيدى لو فى اراى عندك و انت عاوز تجمع بين كل عناصره فى استرنجايه واحده و تحط بينهم علامه معينه او متحط بتعملها كده

var array = ["a","b","c","d","e"]; // array

console.log(array); // [a,b,c,d,e] array

console.log(array.join("-")); // a-b-c-d-e string

Mebo

######################################################################  
Math Functions and Object  
-------------------------------------

كلها عباره عن فانكشن بتساعد فى العمليات الحسابيه دى صفحه اقرأ برضوا الى جواها عشان الموضوع مش محتاج شرح و هيصه

<https://www.w3schools.com/js/js_math.asp>

اهم الفانكشنز فيهم

Round – Floor – random– min – max

Math.round(4.7); // returns 5

Math.round(4.4); // returns 4

دول بتقرب الحاجه بضمير و عدل و حكمه هههههههههه

Mebo

##################################################

Math.ceil(4.4); // returns 5

دى ملهاش فيه بقه بتقرب لفوق و خلاص حتى معناها سقف ههههههههههه

Mebo

##################################################

بتعلى

Math.floor(4.7); // returns 4

دى ملهاش فيه بقه بتقرب لتحت و خلاص حتى معناها أرضيه ههههههههههه

Mebo

##################################################

Math.min(0, 150, 30, 20, -8, -200); // returns -200

مش محتاجه شرح

Mebo

##################################################

Math.max(0, 150, 30, 20, -8, -200); // returns 150

مش محتاجه شرح

Mebo

##################################################

Math.random();

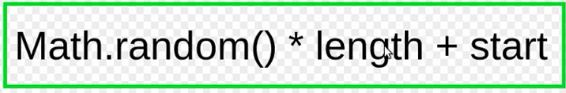
دى بترجع رقم عشوائى ما بين ال 0 و ال 1

طيب لو عاوزه ما بين 0 و 99 اضرب فى 100 و اعملها ماث دوت فلور ذى دى كده

Math.floor(Math.random() \* 100);

طب لو عاوزها من 1 ل 100 زود عليهم الاتنين 1 يعنى زوده من بره يبقى كده

Math.floor(Math.random() \* 100) + 1;



طيب لو عاوزه من 0 ل 100 يبقى زياده الواحد تبقى على 100 يبقى كده يعنى

Math.floor(Math.random() \* 101) ;

Mebo

######################################################################  
throw – try & catch & finally  
-------------------------------------

Throw دى يا معلم ذى ال return بالظبط بس الفرق ان ال Return بترجع الناتج عن عمل الفانكشن اما ال throw بترجع الايرور الى حصل اثناء عمل الفانكشن و ادى طريقه كتبته و ليكن انا هعمل حاجه المفروض انها تدينى ايرور ذى قسمه الاعداد على صفر

فى حاله الريتيرن مش هيبان اى ايرور ذى ما انتوا شايفين و هيرجع الرساله الهبله دى و خلاص

function Dividing(num1,num2){

    if (num2==0){

        return 'Can not Divide on Zero';

    }

    else{

        return num1/num2;

    }

}

console.log(Dividing(5,0)); // Can not Divide on Zero

اما فى حاله استخدام ال throw هيعمل ايرور فعلا

function Dividing(num1,num2){

    if (num2==0){

        throw new Error('Can not Divide on Zero');

    }

    else{

        return num1/num2;

    }

}

console.log(Dividing(5,0)); // error : Uncaught Can not Divide on Zero

طيب ايه بقه ال try Catch دى ؟؟

دى يا سيدى بتتكتب عشان الموقع لو فيه ايرور ميوقفش الدنيا كلها بسببه و يطلع ايرور فى المكان و خلاص هو بيشتغل اذاى بقه ؟؟

بص يا سيدى فكرته انه بيتحط الكود جوه ال try بحيث الكومبيلر يخش يشوف هل فيه ايرور ولا لأ لو فيه ايرور بيروح مطلعلك رساله الايرور الى هو اول ما بيشوف الايرور داخل ال try بيروح جارى رايح لل catch يقولها الحقى يا حاجه ده فى ايرور جايلى من الكود الى كنت بجرب عليه خدى اظهريه بقه و ده العمليه الى بتحصل بين التراى و الكاتش طب ايه دخل ال throw دلوقتى فى الموضوع بص يا سيدى التراى بتروح تجرب على الكود الى جواها طب دلوقتى انا لو بنادى جوه ال try دى على فانكشن يبقى هروح انادى فانكشن بره ال try اصلا و كده هى مش تبع ال try طب عشان اخليها تبع ال try اعمل ايه ؟؟

اروح حاطط فيها ال throw عشان بمجرد ما يحصل ايرور داخل الفانكشن تروح بعتاه لل try الى هى جواها و تقولها اديه و نبى للكاتش بتعتك تظهر الايرور

طب ايه finally دى ؟ دى يا سيدى منطقه محايده اى حاجه انت عاوز تظهرها سواء فى ايرور او مفيش ايرور بتكتبها فيها ذى رساله ان الكود خلص مثلا انا فى الغالب لما بستخدمها بستخدمها فى قفل الكونيكشن بتاع الداتا بيز فى السى شارب :D

ده بقه مثال كامل على ال Throw => try => catch => finally

try{

    console.log(Dividing(5,0));

}

catch(Code\_Err) {

    console.log(Code\_Err.message + "\n" + Code\_Err.type);

}

function Dividing(num1,num2){

    if (num2==0){

        var err = new Error();

        err.message='num2 can not be Zero';

        err.type = 'Divide on Zero';

        throw err;

    }

    else{

        return num1/num2;

    }

}

//Result

/\*num2 can not be Zero

Divide on Zero \*/

Mebo

#######################################################################  
strict Mode

----------------------------

دى يا سيدى معموله عشان تمشى كود الجافا سكريبت على ستنادرد محترم كده الى هو مثلا مينفعش اعمل فاريبول من غير ما اديله var او let و تانى حاجه ان مثلا مينفعش اعمل اتنين براميتر فى النفانكشن بنفس الاسم و ثالت حاجه ان مثلا مينفعش اعمل كونستراكتور من اوبجيكت من غير ما اكتب new ودول الحاجات الى انا فاكرها حتى الان

دى الامثله الى ممكن فى حاله الاستريكت مود يدوا ايرور بس مش هيدوا ايرور ولا هيعملوا اى بتنجان فى حاجه ال sloppy mode

<script>

      "use strict"

       x = 5;

       console.log(x);

</script>

و

    <script>

      "use strict"

       function Myfunc(x,x){

        console.log(x);

       }

       Myfunc(10,30);

    </script>

و

    <script>

      "use strict"

       //constractor function

       function Myfunc(){

        this.x=5;

       }

      var Obj =  Myfunc();

    </script>

كل دول هيخبطوك فى وشك ايرور لو شغلت ستريكت مود و هيعدوا ذى السكينه فى الحلاوه و ده حاجه طبعا غلط لو فى الاسلوبى مود