Advanced JavaScript Summery

Mebo

#######################################################################  
Name Value Pairs

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

**var *address*** = **'100 Main St Cairo '**;  
***console***.log(***address***);

ده معناه ان عباره عن فاريبول بياخد كذا قيمه بداخله و لكن فى الاخر هو قيمه واحده محطوطه جوه سترينج

Mebo

#######################################################################  
object

-----------

من منظور الى فات هو عباره عن مجموعه من الفاريبول او النيم فاليو بيرز المنظمه و كل واحد محطوط فى بروبرتى و فل الفل و بيتكتب كده

**var *address*** = {  
 **Number**:100,  
 **Street**:**'Main st'**,  
 **City** : **'Cairo'**,  
 **apartment**:{  
 **Floor** : 5,  
 }  
}  
***console***.log(***address***.**Number** +**' '**+ ***address***.**Street** +**' '**+ ***address***.**City** );  
***console***.log(**'Floor Num : '**+***address***.**apartment**.**Floor**);

واخد بالك انى عملت الاوبجيكت جواه شويه بروبرتيس و جواه كمان بروبرتى من نوع اوبجيكت جواه بروبرتيس تانيه و بناديه بشكل تسلسلى

#######################################################################  
Global Object => window

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

**var *Myname*** = **'Mahmoud Badawy'**;  
***console***.log(***window***); *// Global Obj Contain Every Thing in JS****console***.log(**this**); *// if it is Global like now it refer to Window Obj***function** *GetName*(MynameParam) {  
 ***console***.log(**'My name is : '**+ MynameParam);  
}  
*// you see i can Call any Variable or function By this or window****console***.log(***window***.***Myname*** + **' By window'**);  
***console***.log(**this**.***Myname*** + **' By this'**);  
***window***.*GetName*(**'Mostafa Mahmoud By window'**);  
**this**.*GetName*(**'Mostafa Mahmoud By this '**);

ويندوز او زيس الى بتعوض عنها فى حاله انها جلوبال بالشكل الى انت شايفه فوق ده بيبقوا شايلين كل البروبتيس و البلت ان فانكشنز الى اتخلقت فى الدنيا كلها و الفاريبولز و الفانكشنز الى انا عاملهم و كل حاجه عشان كده هتلاحظ انك تقدر تكتب اى فانكشن او اى فاريبول قبلها كلمه window فى الجافا سكريبت

#######################################################################  
Hoisting

-----------

دى معناها ان الجافا سكريبت بتاخد الفاريبولز و الفانكشنز الى انت بتعرفها و تديها قيمه و تخزنها بس هتلاحظ ان انت المفروض تعرف الاول الحاجه و بعد كده تستخدمها مش العكس فمثلا اول مثال ده سليم و هيديك الاوت بوت ذى الفل

**var *Myname*** = **'Mahmoud badawy'**;  
**function** *MyFunc*() {  
 ***console***.log(**'Hello from MyFunc'**);  
}  
  
***console***.log(***Myname***); *// output : Mahmoud badawy  
MyFunc*(); *// output : Hello from MyFunc*

بس هنا هيبان انك لازم تعرف الفاريبول قبل ما تناديها فى اى مكان بينما الفانكشن ممكن تناديها من فوقها و من تحتها و هفهم الوضع ده ليه بعد الكود ده

***console***.log(***Myname***); *// output : --> undefined <---  
MyFunc*(); *// output : Hello from MyFunc***var *Myname*** = **'Mahmoud badawy'**;  
**function** *MyFunc*() {  
 ***console***.log(**'Hello from MyFunc'**);  
}

لاحظ ان الكود طلع اوت بوت الفاريبول الى اتنده قبل ما يتكريت ب undefined و معملش ايرور فده معناها انه بيقولك انا مخزن الفاريبول ده بس انت مستخدمه قبل ما تديله قيمه   
اما الفانكشن فعادى مش مشكله عارف ليه بقه لان هو اصلا بياخد اى definition للفاريبول او الفانكشنز و يطلعها فوق و ده الى وراء الكواليس فى اللغه و ده الى بيخلى الفاريبول مش undefined و الفانكشن شغاله عادى فيبقى الكود الى فوق بيساوى بالظبط الكود ده

**var *Myname*** ; *// undefined***function** *MyFunc*() {  
 ***console***.log(**'Hello from MyFunc'**);  
}  
  
***console***.log(***Myname***); *// output : --> undefined <---****Myname*** = **'Mahmoud badawy'**; *// Mahmoud badawy  
MyFunc*(); *// output : Hello from MyFunc*

اما بقه فى حاله انك معرفتش الفاريبول اصلا من اساسه لا فوق ولا تحت فمن الاخر هيديك ايرور لو ندهتلها اصلا هيقولك مش موجوده اصلا ذى الكود ده كده مثلا

**function** *MyFunc*() {  
 ***console***.log(**'Hello from MyFunc'**);  
}  
  
***console***.log(***Myname***); *// output : --> undefined <---****Myname*** = **'Mahmoud badawy'**; *// Mahmoud badawy  
MyFunc*(); *// output : Hello from MyFunc*

Uncaught ReferenceError: Myname is not defined

معلومع على الماشى لازم تبقى عارف ان ال JS بترن الكود line by line

#######################################################################  
undefined قيمه ال

-----------

دى قيمه خاصه موجوده فى الجافا سكريبت معناها ان الفاريبول موجوده بس مفيهاش قيمه بس كده

كود يوضح الكلام ده بقه و الاوت بوت بتاعه

**var *Myname***;  
***console***.log(***Myname***); *// output : --> undefined <---***if** (***Myname*** === **undefined**)  
{  
 ***console***.log(**'Myname is undefined'**);  
}  
**else**{  
 ***console***.log(**'Myname is defined'**);  
}

ملحوظه اوعى عشان تقارن ولا عشان تعمل اى حاجه تروح حاطط جوه اى فاريبول القيمه undefined عشان بتعمل مشاكل على المدى البعيد فانت خد حظرك و متحطهاش فى فاريبول على الاطلاق

#######################################################################  
JS is **Single Threaded Synchronous Execution**

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

ده معناها ان الجافا سكريبت : بتعمل الشغل الى فى ايدها تخلصه و تشوف غيره تنفذه يعنى تنفذ الامر ده دلوقتى خلصته تشوف غيره انما **مش** بتشتغل فى كذا امر مع بعض و تخلصهم مع بعض الا فى حالات معينه

#######################################################################  
Execution Stack

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

*/\* start Global scope\*/***function** *a*() {  
 */\* start Func a() scope\*/  
 b*(); *// 2* **var** c ; *// 4  
 /\* end Func a() scope\*/*}  
**function** *b*() {  
 */\* start Func b() scope\*/* **var** d ; *// 3  
 /\* end Func b() scope\*/*}  
*a*(); *// 1***var *d***; *// 5  
/\* end Global scope\*/  
// it is saving Excution like stack  
/\*\*  
 \* 3- b() function  
 \* 2- a() function  
 \* 1- Global Excution  
 \* \*/  
// then you can see it Remove excution after save by this order  
/\*\*  
 \* 1- b() function as it Run and Complete firstly  
 \* 2- a() function as it Run and Complete Secondly  
 \* 3- Global Excution as it Run and Complete in the last   
 \* you can understanc it by Order Numbers i put in code   
 \* \*/*

#######################################################################  
Variable Environment

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

*/\* start Global scope\*/***function** *b*() {  
 */\* start Func b() scope\*/* **var** Myvar ;  
 ***console***.log(Myvar); *// Order => (3) => output : undefined  
 /\* end Func b() scope\*/*}  
  
**function** *a*() {  
 */\* start Func a() scope\*/* **var** Myvar = 7 ;  
 ***console***.log(Myvar); *// Order => (2) => output : 7  
 b*();  
 */\* end Func a() scope\*/*}  
  
**var *Myvar*** = 5 ;  
***console***.log(***Myvar***); *// Order => (1) => output : 5  
a*();  
***console***.log(***Myvar***); *// Order => (4) => output : 5  
// as it is in Global scope so it take it's variable from Global scope  
  
/\* end Global scope\*/  
/\*\*  
 \* the Output is   
 \* 5 from Global scope  
 \* 7 from a scope  
 \* undefined from b scope  
 \* 5 from Global scope   
 \* \*/*

من الاوت بوت هتفهم ان اللوكال سكوب لا يؤثر ابدا على الجلوبال سكوب و ان حتى لو الفاريبولز متشابهه فكل واحد جوه الاسكوب بتاعه فريد و كانه واحد تانى خالص و بيستجل فى الميمور باسم غير التانى عشان كده لما بحتاج اى واحد فيهم لازم بناديه فى الاسكوب بتاعه و لو نديت فى سكوب مش بتاعه فبيجيب الفاريبول الى الخاص بالاسكوب الى انا فيه يا اسطى مش عشان مثلا سكوب فانكشن a كان قبل اظهار الفاريبول فيروح متغير لأ هتلاقى ان اخر حاجه جابلك الفاريبول الى فى الجلوبال برضوا و ملوش دعوه بالى جره للفريبول فى الفانكشن ايه لانه معتبره واحد تانى خالص   
  
بص اعتبر ان الاسكوب ده اسم اب فممكن يبقى عيال كتير بنفس الاسم بس ابوهم اسمه متغير و بالتالى هقول انا اعرف

* محمود جلوبال عشان انا فى سكوب الجلوبال
* محمود a عشان هو فى سكوب فانكش ال a اللوكال ههههه
* محمود b عشان هو فى سكوب فانكشن ال b اللوكال برضوا

معاك 3 عيال باسم واحد بس ابهتهم مختلفه هل دول يبقوا واحد ؟ لأ طبعا و بالتالى كل اما تروح تاندى لواحده و هو فى الاسكوب بتاع ابوه هيخرجلك محمود الى فى الاسكوب ده بس مش اى حد تانى و بالتالى التلاته مختلفين عن بعض تماما جوه الميمورى انما ظاهريا هما ليهم اسم واحد الى هو محمود حد فاهم حاجه :D

بس ملحوظه مهمه لو انت بقه ناديت على محمود ده فى فانكش a و اصلا فانكشن a ده مجبش عيال اسمهم محمود ساعتها ايه الى بيحصل ؟؟ محمود جلوبال بيرد عشان ده بيعوض عن اى محمود مش موجود على طول

و ده الى هتفهمه من الكود الى جاى ده

*/\* start Global scope\*/***function** *b*() {  
 */\* start Func b() scope\*/* ***console***.log(***Myvar***); *// Order => (3) => output : 5 from Global scope  
 // as the is no Define for Myvar here  
 /\* end Func b() scope\*/*}  
  
**function** *a*() {  
 */\* start Func a() scope\*/* **var** Myvar = 7 ;  
 ***console***.log(Myvar); *// Order => (2) => output : 7  
 b*();  
 */\* end Func a() scope\*/*}  
  
**var *Myvar*** = 5 ;  
***console***.log(***Myvar***); *// Order => (1) => output : 5  
a*();  
***console***.log(***Myvar***); *// Order => (4) => output : 5  
// as it is in Global scope so it take it's variable from Global scope  
  
/\* end Global scope\*/  
/\*\*  
 \* the Output is  
 \* 5 from Global scope  
 \* 7 from a scope  
 \* 5 from Global scope as the is no Define for Myvar in (b) scope  
 \* 5 from Global scope  
 \* \*/*

#######################################################################  
Scope Chain

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

بص الى انا فهمته ان ياعم جوه الفانكشن ده سكوب لوكال --- بره الفانكشن ده يبقى بالنسبه للفانكشن دى سكوب جلوبال عليها ---- لو فى فاريبول سكوب لوكال و بنفس الاسم بتاع الفاريبول الاسكوب جلوبال الفانكشن هتفضل اللوكال طبعا حتى لو كان انت مستدعى فاريبول قبل ما تكريتها فهتديك undefined و مش هتاخد من الى بره الجلوبال اما لو ماشيه بالترتيب الصحيح فهتاخد اللوكال طبعا اما لو الفاريبول الجلوبال دى دخلت جوه الفانكشن لقت استدعاء بس و مفيش انشاء لفاريبول بنفس الاسم ساعتها الجلوبال هى الى هتشتغل

مثال على جلوبال موقوف حالها و مش شغاله عشان اللوكال شغال غصب عن عين التخين و فى نفس الوقت جلوبال صغير جوه اللوكال بيوريك انك تقدر تعمل جلوبال على مستوى الفانكشن لو جطيت فانكشن تانيه جواها و لاحظ ان الفانكشن الى داخل فانكشن دى لما بتيجى تناديها بتناديها فى اللوكال سكوب بتاع الفانكشن الاب مش الجلوبال سكوب العام

*/\* start Global scope\*/***var *Myvar*** = 5 ; *// it will work if there is not any variable in local Scopes with same name***function** *a*() {  
 */\* start Func a() scope\*/* **function** *b*() {  
 */\* start Func b() scope\*/* ***console***.log(Myvar); *// 3- 7 from the local Myvar in a() which is Global for b()  
 /\* end Func b() scope\*/* }  
 ***console***.log(Myvar); *// 1- undefined as it has local Var but it is before the variable* **var** Myvar = 7 ; *// it is local for func a() Global for Func (b)* ***console***.log(Myvar); *// 2- 7 s it has local Var but it is after the variable  
 b*(); *// can call b() here as it is it's Global Scope in the func a()  
 /\* end Func a() scope\*/*}  
*a*();  
*// we can not Call b(); here it will make Error  
//so output will be 7  
/\* output is :  
\* undefined  
\* 7  
\* 7  
\* \*/*

خد بالك بقه انك تقدر تحط فانكشن جوه فانكشن جوه فانكشن جوه فانكشن بس استدعاء كل واحده بيبقى فى ابوها و مينفعش فى جدها بقه هههههههههه

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

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

الفرق بين التلاته دول ايه بقه   
بص يا سيدى ال var ممكن تعرفها جوه IF/for و تناديها بره ال IF/for و تلاقيها شغاله عادى خالص هى معتبره اقواس ال if ملهاش لازمه & تانى حاجه ملهاش نوع معين يعنى بتعرف الحاجه و تعملها انشياليز ب سترينج او نامبر او اراى او اى حاجه

/\*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

if(true)

{

var x = 5;

}

console.log(x);

ال let ليها سكوب اسمه البلوك سكوب الى هو اى قوس يبقى لوكال سكوب بالنسبه ليها من الاخر

if(true)

{

let MyNum = 5;

}

console.log(MyNum);

من الاخر ال var مش معترفه باى قوس غير اقواس الفانكشن غير كده لأ على عكس let اى قوس بالنسبلها ده سكوب و مسمياه بلوك سكوب

فرق كمان

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 احبره يتعامل معاه   
فهتلاقىه بيتعامل مع الفاريبول كسترينج و كانمبر و كبوليان و كأراى و ادى مثال اهه

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";

#######################################################################  
Asynchronous Callbacks

و مناداه الايفينت اذاى فى الجافا سكريبت

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

بص يا سيدى الموضوع و ما فيه ان عندنا نوعين من الفانكشن Asynchronous Synchronous Vs و الفرق بينهم هنوضحه دلوقتى

أولا ال synchronous دى الفانكشنز العاديه الى هو الفانكشن بتتعمل الكومبيلر يخش يحطها فى الاستاك و لما اعملها كول و اناديها تروح تتنادى و يفضل الكومبيلر شغال جواها لحد ما يخلص و لما يخلص خالص يخرج منها فمثلا لو انا عامل فانكشن رفع فيديو مثلا و الفيديو ده كبير جدا فكده الفانكشن هتاخد وقت كبير جدا على ما يخرج منها و يشتغل على حاجه تانيه و بالتالى كده شكرا الفانكشن دى مش نافعه تتعمل بالطريقه دى خالص لان كده انا وقفت الموقع تماما و شليته

طب ايه حل المشكله دى لما تقبالنا ؟؟

الحل اننا نستخدم Asyncronous ليه بقه ؟؟ لان الطريقه دى بتشتغل ذى ال Threads كده الى هو الكومبيلر هيخش فى الفانكشن دى و مقسم نفسه كذا thread فهيلاقى الفانكشن دى هتطول فهيخلى thread يشتغل عليها و يروح يكمل شغل فى السطور الى بعد الفانكشن دى عادى خالص ب thread تانى و لما الشغل يخلص فى الفانكشن تماما يروح ساعتها مرجع النتيجه بتعتها عادى خالص

مثال على كده من الحياه دلوقتى انا محمود اهه عاوز اجيب بيتزا فرحت اتصلت على محل البيتزا الى هو عملت call يعنى و قولت للراجل اعملى بيتزايه عشان عاوز اكل فراح الراجل ابتدى شغل فى البيتزا و ده يعنى الفانكشن ابتدت تشتغل هل انا بقه هفضل ذى الاهبل مستنيه كده لحد ما يخلص ؟؟ لأ طبعا هروح العب شويه على الكمبيوتر و اضيع الوقت فى اى شغل تانى الى هو يعنى كانى رحت اعمل فانكشن تانيه ملهاش علاقه بالفانكشن الاولانيه الى هو عمل البيتزا و بخلص كل الكلام ده عادى و الراجل شغال بيعمل البيتزا و اول ما بتاع البيتزا يخلص عملها فهيروح الراجل مكلمنى فى التليفون يقولى البيتزايه خلصت و ده معناه ان الراجل هيرد عليا بمكلمه ان البيتزا خلصت يعنى هيدينى النتيجه فى callback ابعتهالى ولا تيجى تستلمها يعنى من الاخر كده هنستلم البيتزا خلاص هناكل ؟؟ لأ طبعا الموضوع لسه مخلصش لان النتيجه الى رجعلى دى هتتحط فى طابور اسمه callback Queue و ده ليه ؟؟ ده يا سيدى ان مش معنى ان الاكل جه فروح اكل ذى الجحش لأ هخلص الى فى ايدى الاول الى هو الفانكشن الى كنت شغال عليها الى هو لو بعلب جيم مثلا هكمله و بعد كده انزل اخد النتيجه عشان جه دورها بعد ماخلصت لعب و كده خلاص هعد اكل و يبقى كده الاسينكرينس فانكشن خلصت ههههههههههههههههههههههه و ارجع اكمل حياتى الطبيعه الى هى باقى الكود يعنى ههههههههههه تخيل بقه انك طالب بيتزا و فراخ و كشرى بقه هههههههه

سؤال محيرنى بقه هو بالشرح ده انا كده فهمت ان الفانكشن الى انا مخليها فيها Async مش بترجع النتيجه بشكل فورى و بتفضل شغاله فيها لحد ما تخلص و بتشغل معاها الكود الى بعده صح , طب ما كده لو الكود الى بعدها معتمد على نتيجتها هيشتغل على نتيجه لسه مرجعتش صح كده ؟؟ بص يا سيدى احنا بنخلى الى بشتغل بس هو الى ملوش علاقه تماما بنتيجه الفانكشن اصلا عشان ميأثرش على نتيجه الفانكشن و بنخلى اى حاجه هتشتغل فى ال Async تبقى حاجه ملهاش علاقه بيها من قريب او من بعيد للعلم بالشى فى فانكشنز كتير جاهزه فى ال JS شغاله Async ذى setTimeout , SetInterval , Events  
و فى طبعا فانكشنز انت بقه بتبنيها بأيدك

بص يا سيدى ده كود بيوضح ال Syncronous Func هنا انا عامل فانكشن بتستغرق 3 ثوانى عشان تخلص فهنا على الرغم من انها بعد فانكشن الكليك هاندلار الى هو المفروض انى لما ادوس على اى مكان فى الصفحه تظهرلى الرساله بتاعت كليك ايفنت الا ان انا لو دوست فى اى وقت قبل ما ال 3 ثوانى يخلصوا مفيش رساله هتظهر خالص و ده معناه ان فانكشن ال waitThreeSecond عطلت كل الصفحه لحد ما تخلص و ده مثال حى على المشكله الى ال delay و العطله الى الفانكشن دى تسببت فيها

*// long running function***function** *waitThreeSeconds*() {  
 **var** ms = 3000 + **new *Date***().getTime();  
 **while** (**new *Date***() < ms){}  
 ***console***.log(**'finished function'**);  
}  
  
**function** *clickHandler*() {  
 ***console***.log(**'click event!'**); *// 3- output : click event!  
 // but it will Shown in browser after Click on Doc*}  
  
*// listen for the click event****document***.addEventListener(**'click'**, *clickHandler*);  
  
*waitThreeSeconds*(); *// 1- output : finished function****console***.log(**'finished execution'**); *// 2- output : finished function*

و هيكون حل الشمكله دى ايه بقه اننا نستخدم ال Asyncronous

تعالى بقه نشوف الكود بال Asyncrouns

بص ياعم الحج الكود عشان يبقى Async لازم يبقى فيه Wait و انا بصراحه هأجل موضوع شرح الكود لحد ما تفهم البروميس عشان تربط و تفهم الدنيا أحسن

#######################################################################  
promise

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

دى معناها وعد و دى شغاله على طول مع أوامر Async و عمتا طريقه شغلها كالأتى

الفانكشن بتشتغل و بالمفروض يترجع نتيجه يا اما ترجع ايرور

لو رجعت نتيجه بترجعها فى ال Resolve لو رجعت ايرور بترجعه فى reject اعتبرها ذى الفانكشن العاديه السليم يرجه فى Return و البايظ يرجع فى ال throw Exeption هنا بقه Resolve , Reject

طريقه كتابتها بقه كده

اول حاجه بتروح تعمل البروميس نفسه و جواه بتعمل الفانكشنز الى انت عاوز تنفذها و جوه الفانكشن الى انت عاوز تنفذها دى هتكتب فانكشن الريسولف و هتبعت فيها براميتار و الريجيكت برضوا هتبعت فيها براميتارز عشان بعد كده هتستخدمهم و هيبقى الكتابه كده

ده كده انا عملت فاريبول من نوع بروميس فيه بقوله خش نفذلى الفانكشن الى جواك دى بعد ثانيتين و لو الفانكشن نجحت نفذلى ال Resolve و لو فشلت نفذلى ال Reject

  var MyPromise = new Promise(

            //Arrow Function for Naming Resolve and Reject functions

            (MyResolve, MyReject) => {

                setTimeout(function () {

                    var Result = "Promise Logic Here";

                    // var Result = null; // Remove comment and Try

                    if (Result != null) {

                        // if  Success

                        MyResolve("Hello From Resolve");

                    } else {

                        // if  Error

                        MyReject("Hello From Reject");

                    }

                }, 2000)

            }

        );

طيب دلوقتى عشان ينفذ ال Resolve or Reject المفروض انا كده هاخد الفاريبول بتاعت البروميس و اشغل عليها اتنين فانكشن تانين then , Catch ذى try , Catch ال Resolve تقابلها ال try و ال reject تقابلها ال Catch و ادى ياعم باقى الكود عشان البروميس تشتغل بقه

     MyPromise.then(

            // Arrow Func (call Back) Take Only One Param

            // so If you need Alot of Variable Send all in one Object

            (Only\_1\_Param) => {

                console.log(Only\_1\_Param);

            }

        ).catch(

            (Only\_1\_Param) => {

                console.log(Only\_1\_Param);

            }

        )

#######################################################################  
Nested Promises

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

لو عاوز تعمل بروميس لما تخلص تشغل البروميس الى بعدها و هكذا تعملها اذاى ؟؟ بالنيستيد بروميس

طب و دى تتكتب اذاى ؟؟ بتتكتب كده يا اسطى

 var MyPromise = new Promise(

            (MyResolve, MyReject) => {

                setTimeout(function () {

                    var Result = "Promise Logic Here";

                    if (Result != null) {

                        MyResolve("Hello From Resolve");

                    } else {

                        MyReject("Hello From Reject");

                    }

                }, 2000)

            }

        );

        var MyPromise\_2 = new Promise(

            (MyResolve, MyReject) => {

                setTimeout(function () {

                    var Result = null; // Remove comment and Try

                    if (Result != null) {

                        MyResolve("Hello From Resolve 2 ");

                    } else {

                        MyReject("Hello From Reject 2 ");

                    }

                }, 4000)

            }

        );

        MyPromise.then(

            (Only\_1\_Param) => {

                console.log(Only\_1\_Param);

                MyPromise\_2.then(

                    (Only\_1\_Param) =>{

                        console.log(Only\_1\_Param);

                    }

                ).catch(

            (Only\_1\_Param) => {

                console.log(Only\_1\_Param);

            })}

).catch(

            (Only\_1\_Param) => {

                console.log(Only\_1\_Param);

            }

        )

ذى ما انت شايف عمنا 2 بروميس و ناديت على الاولى بعد ثانيتن و بعد كده ناديت على التانيه بعد ثانتين كمان عشان كده قولتله اندهها بعد 4 ثوانى من بدايه التشغيل

طيب انا بصراحه الطريقه دى فى كتابه الكود مش عجبانى و حاسسها ملعبكه حاجه بسيطه اى نعم هى سهله و حلوه بس الاحسن ان احنا نستدعى الاتنين برومايسيس بطريقه السلسله و دى عن طريق انى ارجع فى ال then بتاعت الاولى البروميسيس التانيه فى return و ده معناه انى كانى بقوله خلى كل الجزء ده بيساوى Promise\_2 و بالتالى انت دلوقتى تقدر تستدعى ياباشا ال then بتاعت بروميس 2 و تخلى كله ليه كاتش واحده هيرجع فيها الرساله بتعته باختلافهم فى التعريف و ادى طريقه الكتابه الحلوه

 MyPromise.then(

            (Only\_1\_Param) => {

                console.log(Only\_1\_Param);

                return MyPromise\_2

            }

        ).then(

            (Only\_1\_Param) => {

                console.log(Only\_1\_Param);

            }

        ).catch(

            (Only\_1\_Param) => {

                console.log(Only\_1\_Param);

            }

        )

#######################################################################  
Promises With Async \_ Await

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

دى طريقه بقه احسن من الاتنين الى فاتوا فى كتابه البروميس لانها بتختصر كتابه كود كتير

دلوقتى انت عامل اتنين بروميس و عاوز تشغل الاولى ثم التانيه دى هتعملها اذاى بسهوله فشيخه ؟؟

كل الى هتعمله انك هتعمل فانكشن و هتديها data Key اسمها Async و البروميس الى عاوز تشغلها هتكتب قبلها await و تكتب اسمها هتلقيها اشتغلت و مهمه ال Await ان هى بتخلى البروميس تشتغل و متخرجش منها الا لما تخلصها و بعد ما تخلصها تشتغل فى البروميس التانيه و ادى مثال بالكود اهه

     async function MyFunc(){

            let  MyProm\_1 = await MyPromise;

            console.log(MyProm\_1);

            return MyPromise\_2;

     }

     MyFunc().then((msg)=>{console.log(msg);}

     ).catch((err)=>{console.log(err)});

بص يا سيدى الفكره انى عملت ويت للبروميس الاول بحيث انه مفيش اى حاجه تتنفذ الا لما هو يخلص و فعلا اول ما الثانيتن الاولنين خلصوا راح ظاهر الكونسول الى انا مرجه فيه نتيجه البروميس الاول و الى هى يا اما ريسولف يا اما ريجيكت طيب و بعدها عملت ايه ؟؟ عملت ريتيرن ؟؟ طب عملت ريتيرن لأيه ؟؟ عملت ريتيرن للبروميس التانى و ده عشان يبقى كانى بقوله ان الفانكشن دى نتيجتها فى الاخر انها ترجع البروميس التانى و بالتالى ساعتها اقدر اقوله الفانكشن دوت then دوت catch الى هو كانى بقوله اعملى بقه ثين و كاتش بتوع البروميس التانيه الى نتجت عن الكول للفانكشن الى انا فيها عامل ويتنج و بشغل البروميس الاولى

يارب تكون فهمت بعد العك ده كله :D :D

#######################################################################  
JS Dynamic Typing

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

ده معناها ان var بتعرف اى فاريبول على انها Object و فى اى مكان فى الكود تقدر تغيير نوعها عادى خالص على عكس لو بتستخدم let و محددلها نوع ساعتها متقدرش تغير النوع الا بطرق معينه معروفه

و ده مثال يفهمك الوضع

**var *Myvar*** = **true**;  
***console***.log(***Myvar***); *//output true****Myvar*** = **'Mahmoud'**;  
***console***.log(***Myvar***); *//output Mahmoud****Myvar*** = 20;  
***console***.log(***Myvar***); *////output 20  
  
/\* as the var Type is Dynamic So No Error and output will be :   
\* true  
\* Mahmoud  
\* 20  
\* \*/*

اما لو ستاتيك مكنش هيرضى ياخد اى حاجه الا من نوعه بس او من نوع تانى معين بعد البارسينج ولازم يبقى ينفع يتعمله بارسينج كمان يعنى مينفعش اجيب محمود و قوله حوله لرقم مثلا هيشتمنى اكيد يعنى :D

**let *Myvar***:**boolean** = **true**;  
***console***.log(***Myvar***); ***Myvar*** = **'Mahmoud'**;  
***console***.log(***Myvar***); ***Myvar*** = 20;  
***console***.log(***Myvar***);

الكود ده هيدى ايرور عشان انا استخدمت let و قولت ان نوعه Boolean و بالتالى بقى   
static Type و بالتالى مينفعش اجى احط فيه اى بتنجان كده و خلاص

#######################################################################  
Primative Types

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

انواع الفاريبول من الاخر

* Undefined
* Null
* True OR False
* Num
* String
* Symbol و دى تبع ال ECS 6

#######################################################################  
Opertators

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

الجمع و الطرح و القسمه و الضرب و المودلاس   
ولازم تبقى عارف انك تقدر تستخدم < و > بس ده ساعتها بيعتبر كانك عامل اف كونديشن و بيطلع Boolean

ده جدول توضيحى شامل

<https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Operators/Operator_Precedence>

**var *Myvar*** = 5 > 4;  
***console***.log(***Myvar***); *//output true*

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

طيب الكود ده نتيجته ايه ؟

**var *Myvar*** = 5 + 4 \* 2;  
***console***.log(***Myvar***); *//output 13*

و ده لانه بيعمل الضرب و القسمه قبل الجمع و الطرح

طب لو استخدمنا القوس فى الجمع النتيجه تتغير ايوه طبعا لانه بيشوف الى جوه القوس ينفذه و بعد كده ينفذ عليه الشغل الى ناقص

**var *Myvar*** = (5 + 4) \* 2;  
***console***.log(***Myvar***); *//output 18*

طيب شوف المثال الى جاى ده هيوريك الفاريبول بيخزن فى بعضه اذاى

**var *a*** = 2, ***b***=3 , ***c***=4;  
***a***=***b***=***c***; *// from Rigth to left C=4 so b = 4 So a = 4****console***.log(***a***); *//output 4****console***.log(***b***); *//output 4****console***.log(***c***); *//output 4*

#######################################################################  
Concatenation between String and Num

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

***console***.log(1+**'2'**); *//output 12****console***.log(**'1'**+2); *//output 12****console***.log(**'1'**+**'2'**); *//output 12****console***.log(*parseInt*(**'1'**)+*parseInt*(**'2'**)); *//output 3*

لاحظ ان اى حاجه استرينج + رقم بتبقى كانك حطيت كلام جنبه رقم عادى خالص   
بينما لو حولت الرقم الاسترينج ده لرقم فعلا بالبارسينج هتلاقيه اعتبره انتيجر و جمعه عادى خالص و اداك نتيجه جمع صحيحه

#######################################################################  
Comparison Operators

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

فى اكبر و اصغر و الكلام العبيط ده و عمتا كله بيستخدم مع الايف كونديشن بس هنيجى بقه للى محتاجين نفهمه صح و هكتب الكود يوضح ايه الى بيحصل

***console***.log(-4 < 2); *//output true****console***.log(-4 > 2); *//output false****console***.log(**true** < 2); *//output true****console***.log(**true** > 2); *//output false****console***.log(**true** == 1); *//output true****console***.log(**true** === 1); *//output false****console***.log(**false** == 0); *//output true****console***.log(**false** === 0); *//output false****console***.log(***Number***(**false**) === 0); *//output true*

***console***.log(***Number***(**false**) === ***Number***(0)); *//output true****console***.log(*parseInt*(**false**) === 0); *//output false parseint does not make false as num****console***.log(*parseInt*(**false**)); *//output NAN****console***.log(*parseInt*(0)); *//output 0****console***.log(*parseInt*(**false**) === *parseInt*(0)); *//output false as false will be NAN after parse to int  
// and 0 paresed to int to 0 so NAN is not equal to NAN*

ذى ما انت شايف الترو بوليان فلو قارنتها بواحد ب == بيدى واحد عادى و بيدى ترو انما اول ما هجى اقوله ان المقارنه ب === هيقولى فولس لان ساعتها دى رقم و دى بوليان لانه بيعتبر الفولس ب 0 و الترو ب 1 بس النوع مختلف و انت عارف === لازم النوع يبقى نفس النوع

طيب الزيرو هنا مصنف انه رقم اذا اعمل للفولس برضوا بارس للرقم فهيدينى فعلا ترو بينما لو عملتله بارس انتيجر هيدى فولس لان الصفر اصلا انا مش معرفه على انه انتيجر لا ا الجافا سكريبت معتبراه رقم و خلاص

و عمتا انت لو حاولت تحول false لانتيجر هيديك NAN مش هيديك رقم اصلا انما لما بتحوله لرقم بيتحول ل 0

ده جدول جميل تعرف منه كل حاجه فى الكلام ده

<https://developer.mozilla.org/en-US/docs/Web/JavaScript/Equality_comparisons_and_sameness>

#######################################################################  
 in JS default Value is always inside function Not in param

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

**function** *MyFunc*(Name) {  
 **Name = Name || 'Mahmoud Badawy';**  
 ***console***.log(Name)  
}  
  
*MyFunc*(); *//output : Mahmoud Badawy  
MyFunc*(**'Mostafa'**); *//output : Mostafa*

معنى السطر الى انا معلم عليه ده الاسم يساوى الاسم لو موجود اما لو مش موجود فهيبقى undefined و بالتالى فى المقارنه undefined ولا محمود بدوى طبعا الكومبيلر هياخد محمود بدوى   
اما لو فى اسم ذى مصطفى فى البرامتر فهيبقى مصطفى ولا محمود بدوى فهياخد الاختيار الاول بغض النظر هو مصطفى ولا محمود بدوى ههههههههههه و بالتالى القيمه الافتراضيه فى الجافا سكريبت بتتعمل ذى ما انت شايف كده حاجه شلل الصراحه لان جربت اعملها فى البراميتر منفعتش

#######################################################################  
 Objects

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

بعمل حاجه من نوع اوبجيكت و املاها و انادى على الى جواها كانى بالظبط بنادى على بروبرتى جوه كلاس و هتلاحظ ان الاوبجيكت بيشل كل حاجه سواء   
بروبيرتيس او فانكشنز و هنا اوريك التعامل مع الاوبجيكت اذاى تعمل اوبجيكت و اذاى تحط جواه داتا بشكل سهل و اذاى بتنادي اى حاجه جوه الاوبجيت و اذاى بتنادى بره الاوبجيكت بطريقتين اقرأ الكومنتات كويس جدا لانها موضحه حاجات كتير جدا هتفهمها منها على سبيل المثال انت لو عملت لنفس الاوبجيكت ستراكتشار جديد بتنسى القديم خالص و بتناديش منه حاجه عشان ميطلعلكش ايرور و بتنادى من الجديد بس

**var *person\_Obj*** = {  
 **Name** : **'Mahmoud Badawy'**,  
 **Age** : 15 ,  
 **Address** : {  
 **FloorNum** : 5 ,  
 **Streat** : **'Mebo St'** ,  
 **City** : **'Cairo'**,  
 **Country** : **'Egypt'**,  
 } ,  
 NameFunc : **function** Myname( person\_Obj)  
 {  
 **return this**.**Name** ;  
 }  
}  
  
***console***.log(***person\_Obj***.NameFunc()); *// Mahmoud Badawy****console***.log(***person\_Obj***.**Name**); *// Mahmoud Badawy****console***.log(***person\_Obj***.**Age**); *// 15****console***.log(***person\_Obj***.**Address**); *// Object data****console***.log(***person\_Obj***.**Address**.**Country**); *// Egypt****person\_Obj***.**Name** = **'Mesho'**;  
***console***.log(***person\_Obj***.**Name**); *// Mesho  
  
//it is like that you do new object with New structure so you  
// can not Call any property from line before this Creation of new obj Structure****person\_Obj*** = {**MyName**:**'Kamal'** ,  
 **Nationality** : **'Egyption'**,  
}; *// set property on the fly****console***.log(***person\_Obj***.**Name**); *// undefined as it is setted after this Line in New Obj****console***.log(***person\_Obj***.**MyName**); *// Kamal****person\_Obj***[**'Name'**] = **'Marzok'**; *// we can set an get it like Array****console***.log(***person\_Obj***.**Name**); *// Marzok****console***.log(***person\_Obj***[**'Name'**]) *// Marzok  
//cal Func  
/\* they will make Error as it is not in new onj Structure  
// so i make it is Commented  
console.log(person\_Obj.Address.Country);  
console.log(person\_Obj.NameFunc());  
console.log(person\_Obj.NameFunc({Name:'Kamal' ,  
 Nationality : 'Egyption',  
})); // Kamal  
\*/***function** *SayHello*(Person) {  
 ***console***.log(Person.Name);  
}  
*SayHello*(***person\_Obj***); *// Marzok  
SayHello*({  
 **Name** : **'Fat7y'** ,  
}); *// Fat7y as i set it in the fly*

#######################################################################  
 Objects in side Object by two ways

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

**var *MyObj*** = {  
 **Name**:{  
 **FirstName** : **'Mahmoud'**,  
 **LastName** : **'Badawy'**,  
 },  
};  
***console***.log(***MyObj***.**Name**.**FirstName** + **' '** + ***MyObj***.**Name**.**LastName**);

ذى ما انت شايف اوبجيكت جوه اوبجيكت و شايف اذاى بنديه و طبعا باين ان الدينا سهله فشخ

تعالى بقة اوريك الطريقه التانيه لعمل الكلام ده الى هى اوبجيكت جواه اوبجيكت بعامله على انه بروبرتي بكريته على الماشى و انا شغال

**var *MyObj*** = {}; *// Great Empty object (MyObj)****console***.log(***MyObj***.**Name**) *// undefined as still no taken any Values even Empty Value****MyObj***.**Name** = {}; *// Great Name Obj that is property of (MyObj) and Take Empty****MyObj***.**Name**.**FirstName** = **'Mahmoud'**; *// here we Create FN prop in Name Obj and Give it Value****MyObj***.**Name**.**LastName** = **'Badawy'**; *// here we Create LN prop in Name Obj and Give it Value****console***.log(***MyObj***.**Name**.**FirstName** + **' '** + ***MyObj***.**Name**.**LastName**);  
  
*/\*\* output  
 \* undefined  
 \* Mahmoud Badawy  
 \* \*/*

الطريقه الاولى من وجهه نظرى احلى بلاش عك و لعبه بقه :D :D

#######################################################################  
 Object to JSON and JSON to Object

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

**var *MyObj*** = {  
 **Name** : **'Mahmoud Badawy'**,  
 **Age** : 15 ,  
 **IsMarried** : **true**,  
 **Address** : {  
 **City** : **'Cairo'** ,  
 **Country** : **'Egypt'** ,  
 } ,  
};  
  
*// JSON.stringify(ObjVar) Convert Object Data To Json Format***var *JsonFormat\_Data*** = ***JSON***.stringify(***MyObj***);  
***console***.log(***JsonFormat\_Data***);  
*// JSON.parse(JsonData) Convert Json Data To Object Format***var *ObjFromat\_data*** = ***JSON***.parse(***JsonFormat\_Data***);  
***console***.log(***ObjFromat\_data***);

هنا بتعلم اذاى تحول الابويجيكت لجيسون فورمات و العكس عشان بعد كده هتستخدم الكلام ده و معاه سيرياليز بقه و هتعيش

*/\*\*  
 \* output  
 \* {"Name":"Mahmoud Badawy","Age":15,"IsMarried":true,"Address":{"City":"Cairo","Country":"Egypt"}}  
 \* {Name: "Mahmoud Badawy", Age: 15, IsMarried: true, Address: {…}} Object  
 \*/*

#######################################################################  
Functions are Objects

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

الفانكشن فى جافا سكريبت ممكن تعاملها على انها نوع خاص جدا من ال object بيتحطلها بروبرتى بالدوت و اعمله سيت ذى كده ده الى هيفهمك فعلا بس مهما ضفتلها هو بيضفلها فى الميمورى بس و الاستراكتشار مش بيتغير ابدا بتاع الفانكشن من جوه

**function** *Myfunc* () {  
 **return 'Hello world'**;  
}  
*Myfunc*.**Name** = **'Mahmoud'**;  
*Myfunc*.**Age** = 20;  
***console***.log(*Myfunc*); *// output : A structure if Function****console***.log(*Myfunc*()); *// output : Hello world****console***.log(*Myfunc*.**Name**); *// output : Mahmoud****console***.log(*Myfunc*.**Age**); *// output : 20*

#######################################################################  
Prototype

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

بص يا سيدى البروتوتايب دى بيتم استخدمها مع الاوبجيكت الناتج عن فانكشن فى الغالب و فى الغالب انا بستخدمه لما اكون عاوز اخزن حاجه فى الكلاس و تكون ثابته فى الاوبجيكتات الى انا بنشأها من الكلاس بتاعى و فى اى وقت اقدر اوصلها بحاجه اسمها \_\_proto\_\_ لان البتاعه دى بتتنشأ مع اى أوبجيكت بيتعمل جديد و بيترمى فيها كل البروبارتيس المعموله عن طريق ال protoype و ادى كود يوضح الامر

    <script>

    function Person(Name,Age){

        this.name = Name;

        this.age = Age;

    }

    var Obj\_1 = new Person("Mahmoud", 25);

    var Obj\_2 = new Person("Mostafa", 2);

    Person.prototype.address = "Cairo";

    console.log(Obj\_1);

    console.log(Obj\_1.address); //Cairo

    console.log(Obj\_2.address); //Cairo

    /\*

    Person

        age: 25

        name: "Mahmoud"

        \_\_proto\_\_:

            address: "Cairo"

            constructor: ƒ Person(Name,Age)

            \_\_proto\_\_: Object

    \*/

    </script>

لاحظ بقه ان البروتوتايب الى انت عملته للفانكشن او للأوبجيكت Person راح اتخزن فى حاجه اسمها \_\_proto\_\_

و دى الى بتخزن اى حاجه نوعها بروتوتايب و بما انها خزنت البروتوتايب ده فى اوبجيكت البرسون فكده هتلاقى الادريس هو هواه نفس الادريس فى الاتنين اوبجيكت الى معمولين لان الاتنين بيشاوروا على نفس ال \_\_proto\_\_

و عشان اغير واحد عن التانى لازم اعمل كل ادريس فى كل اوبجيكت على حدى و هيبقى كده كانى انشات بروبرتى من جديد جوه الاوبجيكت و لما اقرا منه هلاقيه مختلف بس لو جيت اجيب ال \_\_proto\_\_ للاتنين أوبجيكت برضوا هلاقى ان العنوان كايرو متغيرش و الاتنين نفس الحاجه و محصل تغيير فيهم و ده لان غيرت فى الاوبجيكت و ضيفتله ادريس كبروبرتى و مجتش ناحيه البروتوتايب فى حاجه كل الى انا عملته انى خليته ان الاوبجيكت زاد عليه ادريس مش اكتر فلما راح لقاها فى الاوبجيكت قراها و بس

بص ملخص الكلام عشان تفهم اكتر الكومبيلر بيخش على الاوبجيكت لو انا بستدعى فيه بروبارتى معينه يشوفها لو موجوده من ضمن البروبرتيس العاديه بيروح مشغلها و شكرا على كده طب فى حاله ملقهاش ؟؟ بيروح رايح مضور فى ال \_\_proto\_\_ لو لقها بيشغلها و يطلع قيمتها ملقهاش بقه بيعمل ايه ؟؟ ساعتها بيقولك البروبرتى دى مش موجوده ياعم بطل بيض

و عشان كده هنا فى الكود ده هو اول مره دخل فى الاوبجيكت ملقاش الادرس فى الاوبجيكت راح رايح جايبه من البروتوتايب الى هو موحد فى الاتنين اوبجيكت عشان كده ادانا العنوان للاتنين كيرو و لما جيت انا حطيتله ضمن الاوبجيكتس الادريس Address ساعتها دخل لقى الادريس فى كل اوبجيكت مختلف راح باعتهم مختلفين لان كل واحد اصبح بيشاور على مكانه فى الاوبجيكت بتاع و ساعتها رجع كيرو محمود و كيرو مصطفى و مدخلش اصلا للبروتوتايب يجيب منه حاجه فهمت كده ؟؟ ايوه ياعم فهمت جك غزه فى معاميعك

    <script>

    function Person(Name,Age){

        this.name = Name;

        this.age = Age;

    }

    var Obj\_1 = new Person("Mahmoud", 25);

    var Obj\_2 = new Person("Mostafa", 2);

    Person.prototype.address = "Cairo";

    console.log(Obj\_1);

    console.log(Obj\_1.address); //Cairo

    console.log(Obj\_2.address); //Cairo

    Obj\_1.address = "Cairo Mahmoud";

    Obj\_2.address = "Cairo Mostafa";

    console.log(Obj\_1.address); // Cairo Mahmoud

    console.log(Obj\_2.address); //Cairo Mostafa

    console.log(Obj\_1.\_\_proto\_\_.address); // Cairo

    console.log(Obj\_2.\_\_proto\_\_.address); // Cairo

    </script>

مثال تانى بالكلاس العادى

<script>

    class Person{

         Name;

         Age;

        constructor(name , age) {

          this.Name = name;

          this.Age = age;

         }

    }

    var Obj\_1 = new Person("Mahmoud", 25);

    var Obj\_2 = new Person("Mostafa", 2);

    Person.prototype.address ="Cairo";

    console.log(Obj\_1.address); // Cairo

    console.log(Obj\_2.address); // Cairo

    Obj\_1.address = "Cairo Mahmoud";

    Obj\_2.address = "Cairo Mostafa";

    console.log(Obj\_1.address); // Cairo Mahmoud

    console.log(Obj\_2.address); //Cairo Mostafa

    console.log(Obj\_1.\_\_proto\_\_.address); // Cairo

    console.log(Obj\_2.\_\_proto\_\_.address); //Cairo

    </script>

#######################################################################  
Prototype chain

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

ده معناه انى اعمل البروتوتايب ذى ال Nested كده على اساس انى اقدر من الاوبجيكت انادى على بروتاتيب و من البروتوتايب انادى على بروتوتايب و هكذا و ابقى عامل سلسله من الندهات بقه ههههههههههه

و على فكره دى الطريقه المثاليه لعمل الوراثه فى ال Prototype و ده لان فيها بقدر اورث خصائص الاب فى البروتوتايب بتاع الابن و الجد فى الاب و هكذا يعنى كل واحد بيدى لابنه و مينفعش انك تجيب الاب و الجد فى الابن بشكل مباشر عشان هتلاقى الاوبجيكت بتاع الجد عمل اوفر رايت على بتاع الاب و الابن هيورث ساعتها بتاع الجد بس و بالتالى عشان نحل المشكله دى و الابن يورث ابوه و حجه فا احنا عملنا ان الابن يورث ابوه و الاب يكون وارث الجد و بالتالى كده الابن هيجيله خصائص الاب والجد و ادى الكود عشان تفهم بقه كيفيه الوراثه

من الكود هيتضح ان مجرد بس ما احط اوبجيكت الاب فى البروتوتايب بتاع الابن فبكده الابن ورث كل خصائص ابوه

<script>

        function Child(Name,Age) {

            this.ChildName= Name;

            this.Childage= Age;

        }

        function Father(Name , Fav) {

            this.Fathername= Name;

            this.Fatherfav =Fav;

        }

        function Gran\_Father(Sec\_name,Grand\_address) {

            this.grand\_sec\_name= Sec\_name;

            this.grand\_address = Grand\_address;

        }

let Child\_Obj = new Child("Mostafa",2);

let Father\_Obj = new Father("Mahmoud","Foot Ball");

let Gran\_Father\_Obj = new Gran\_Father("Badawy" , "KSA");

Child\_Obj.\_\_proto\_\_ = Father\_Obj;

Father\_Obj.\_\_proto\_\_ = Gran\_Father\_Obj;

console.log(Child\_Obj.ChildName);

console.log(Child\_Obj.Fathername);

console.log(Child\_Obj.grand\_sec\_name);

console.log(Child\_Obj.Childage);

console.log(Child\_Obj.Fatherfav);

console.log(Child\_Obj.grand\_address);

    </script>

ملحوظه مهمه انت لو وارق حاجه من ابوك و عاوز تغيير فى القيمه بتعتها عادى براحتك يا اسطى انت بتورث الحاجه ذى ما هى و لو عاوز تغيير فيها ده بمزاجك بقه لانك بتبقى واخد صوره من الحاجه و تغييرك فى حاجتك الى انت ورثتها مش هيغير من القيمه عند الاب بينما تغيير القيمه عند الاب يأثر فى الى بيورثه الابن ههههه

#######################################################################  
inheritance in Functions

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

هنا بقه انا هشرح اذاى تعمل انهيرتانس من فانكشن لفانكشن تانيه من غير استخدام البروتوتايب و هيبقى الوراثه هنا عن طريق this و call بمعنى انى فى الابن هنادى الاب و ابعتله قيمه this بالاوبجيكت بتاع الابن وبالتالى اى اضافه للبروبرتيس على this الى انا بعتها هى اضافه فى اوبجيكت الابن و ادى الطريقه بالكود يا اسطى

<script>

        function Father(Name, Fav) {

            // this here is Father\_Obj

            this.Fathername = Name;

            this.Fatherfav = Fav;

            this.getInfo = function () {

            console.log(this.Fathername + "  likes " + this.Fatherfav);

            }

        }

        function Child(Name, Age) {

            // this here is Child\_Obj

            this.ChildName = Name;

            this.Childage = Age;

            // I will send it to Father to inhert from him

            Father.call(this, "Ahmed Badawy", "Basket ball");

            // Overloading After Inheritance

            this.getInfo = function (Name, Fav) {

                console.log("Hi From Child");

            }

        }

        var Child\_obj = new Child("Mahmoud", 25);

        var Father\_obj = new Father("Badawy", "Football");

        console.log(Father\_obj);

        console.log(Child\_obj);

        Father\_obj.getInfo();

        Child\_obj.getInfo();

    </script>

ذى ما انت شايف فى فانكشن الابن عملنا فانكشن مهمتها انى ابعت فيها this بتاعت الابن للاب و معاها طبعا البراميتارز بتاعت الفانكشن بتاعت الاب و ده عشان املى بيهم خصائص الاب و انا بورثها و ذى ما انت شايف بقوله خش على فانكشن و ممكن نعتبره كلاس الاب father اعملها call و ابعتلى فيها ابوجيكت الابن فى this عشان لما اروح و املى this بتاعت الاب يبقى بتاعت الابن اخدت مكانها و بالتالى اى خاصيه قبلها this تتورث من الاب للابن على طول

و طبعا انت عارف الاوفرلودينج و هو ان لما الابن يورث فانكشن من الاب يقدر يعمل عليها اى تعديل و هنا بنسميها overloading و ده طبقته فى الجزئيه بتاعت getInfo

شوفت الكلام الجميل الى فوق ده هوريهولك بقه بالكلاس هيبقى سهل جدا و هو عن طريق ال

Class – extends – constractors – super

    <script>

        class Father {

            Fathername;

            Fatherfav;

            constructor(Name, Fav) {

                this.Fathername = Name;

                this.Fatherfav = Fav;

            }

            getInfo = function () {

                console.log(this.Fathername + "  likes " + this.Fatherfav);

            }

        }

        class Child extends Father {

            ChildName;

            Childage;

            constructor(Name, Age, FthrName, Fav) {

                super(FthrName,Fav);

                this.ChildName = Name;

                this.Childage = Age;

            }

            getInfo = function (Name, Fav) {

                console.log("Hi From Child");

            }

        }

        var Father\_obj = new Father("Badawy", "Football");

        var Child\_obj = new Child("Mahmoud", 25, "Boshkash", "TV");

        console.log(Father\_obj);

        console.log(Child\_obj);

        Father\_obj.getInfo();

        Child\_obj.getInfo();

    </script>

ذى ما انت شايف كلاس دى عشان اعرف الكومبيلر ان دى كلاس و اكستيند دى عشان اورث من كلاس لكلاس اخر و داخل كل كلاس بقه بروباريتيس و فيه كونستراكتور و ده بملى فيه البروبارتيس و من الاخر كده بستخدمه فى نقل البيانات بين كلاس وكلاس و

ال super بقه دى بنحطها داخل كونستراكتور الابن الذى ورث و هو لازم تتحط و ببعت فيها اى حاجه عاوز ابعتها كانى بكلم كونستراكتور الاب تماما و ده لان سوبر دى اعتبرها كونستراكتور الاب الى هو الكلاس الى انا وارث منه و العمليه دى بتسمى هنا Dependency injection

#######################################################################  
Functions As (Variable & parameter)

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

ممكن تحط الفانكشن فى فاريبول و لكن ساعتها ميزه انك تنادي الفانكشن قبل عملها هتتلغى تماما و جرب و انت هتفهم

اولا :- الفانكشن العاديه

*Myfunc*(); *// output : Hello world*

**function** *Myfunc* () {  
 ***console***.log(**'Hello world'**);  
}  
*Myfunc*(); *// output : Hello world*

ثانيا : - الفانكشن الى محطوطه فى variable و اذاى اناديها .. عمتا لا يمكن منادتها قبل عملها و عشان كده لو جربت اناديها قبل ما الفانكشن نفسها هتعمل ايرور لان الفاريبول نفسها هتبقى ما زالت شايله undefined و مشلتش الفانكشن نفسها لان ذى ما انت عارف الجافا سكريبت لاين باى لاين

*// Myfunc(); // it will make an Error so i Commented it***var** *Myfunc* = **function**() {  
 ***console***.log(**'Hello world'**);  
}  
*Myfunc*(); *// output : Hello world*

ثالتا :- الفانكشن الى بخزنها فى براميتر اذاى بقه دى اناديها

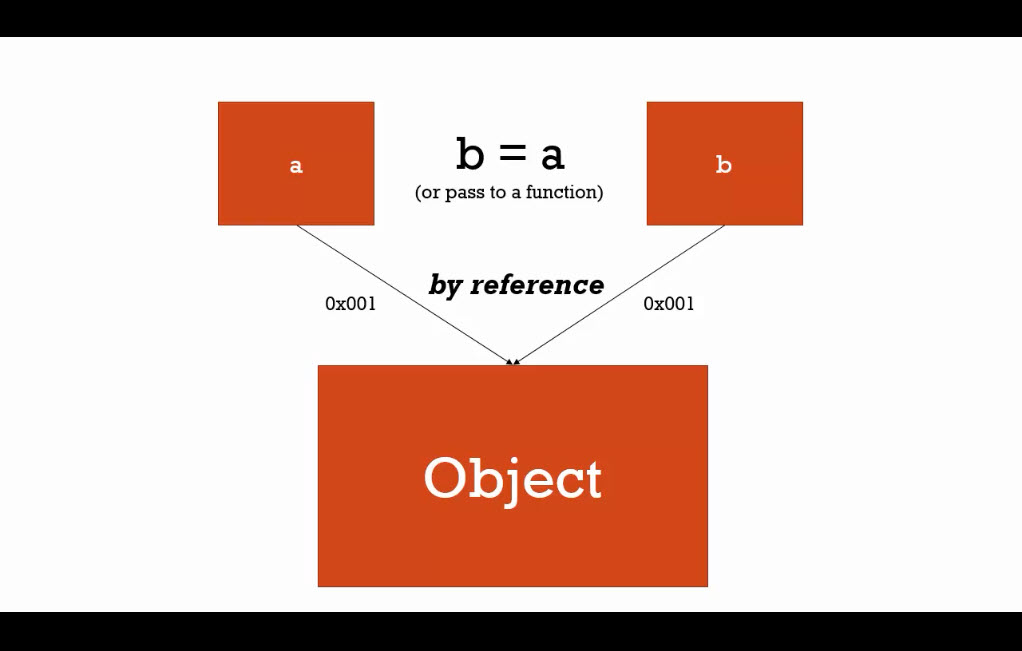
*// Myfunc(); // it will make an Error so i Commented it***function** *funcCreator*(Func) {  
 Func(); *// Func is the Passed Function in PAram and () for Invoce this Passed Function*}  
*funcCreator*(  
 **function** () {  
 ***console***.log(**'Hello Mahmoud'**);  
 }  
);

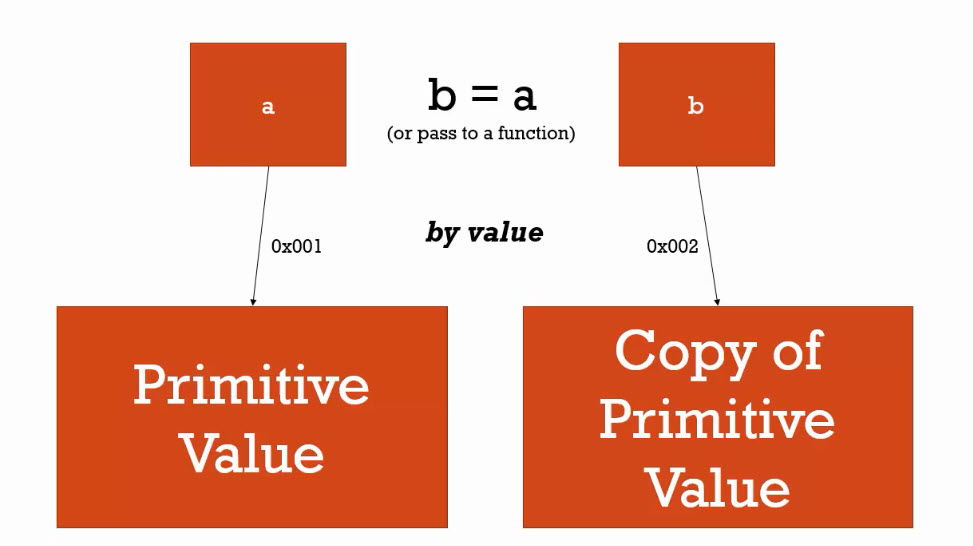
الفكره انى بعمل فانكشن عاديه خالص بخلى البراميتر الى فيها عباره عن فانكشن من غير اسم ذى فانكشن انفوك بس بديهاش القوسين بتوع الانفوك   
داخل الفانكشن الى بنادى بيها بقه بروح حاطط جنب البراميتر قوسين الانفوك عشان ترن جواها بقه بس كده شكرا هى دى الفكره ياعم

#######################################################################  
Value VS Reference

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

فى الغالب ال reference بيجى مع الاوبجيكت خصتا لو هعمل تغيير فى بروبرتى بس يكون بطريقه السيت الى بره الاوبجيكت الى هى بالدوت و ادى مثال للتوضيح هتلاقى ان الفاريبول العادى ده بيدى لبعضه بفاليو   
و بالتالى هتلاقى ان كل فاريبول وخد مكان فى الميمورى فلو حد اتغير التانى بيتغيرش عادى يعنى   
  
على عكس الى بالريفرنس ده كانه عنوان بيتكت انت و اخوك   
فلو فرضا ابوك غير البيت و راح بيت تانى فأوتوماتيك   
انت و اخوك عنوانكوا هيتغير و بالتالى انتوا الاتنين قيمتكوا هتتغير بس لو انت بصين على نفس العنوان يبقى انتوا الاتنين برضوا نفس القيمه هههههههههه  
  
بص الصوره دى هتفهمك





*// by value (primitives)***var *a*** = 3;  
**var *b***;  
***b*** = ***a***; *// a = 3 so as b take a value of a so b = 3****a*** = 2; *// a became value 2****console***.log(***a***); *// 2****console***.log(***b***); *// 3  
  
// by reference (all objects (including functions))***var *c*** = { **greeting**: **'hi'** };  
**var *d***;  
  
***d*** = ***c***; *// as c is Obj so d = c as Obj C & d Refer to data of object C****c***.**greeting** = **'hello'**; *// mutate  
// as c & d Refer to same Memory place in memory  
// so if i change c the d will Changed with c****console***.log(***c***); *// 'hello'****console***.log(***d***); *// became 'hello' like What happend in c  
  
// by reference (even as parameters)***function** *changeGreeting*(obj) {  
 obj.**greeting** = **'Hola'**; *// mutate*}  
  
*changeGreeting*(***d***); *// here we change in d object that refer to same place as c So****console***.log(***c***); *// became 'Hola' like What happend in d****console***.log(***d***); *// 'Hola'  
  
// equals operator sets up new memory space (new address)****c*** = { **greeting**: **'howdy'** };  
***console***.log(***c***); *// howdy****console***.log(***d***); *// 'Hola'  
// as the change happen in the value it self  
// not in the refrence place in memory by Dotted but by Making new c Obj  
// so It is Changed values So they will be diffrent*

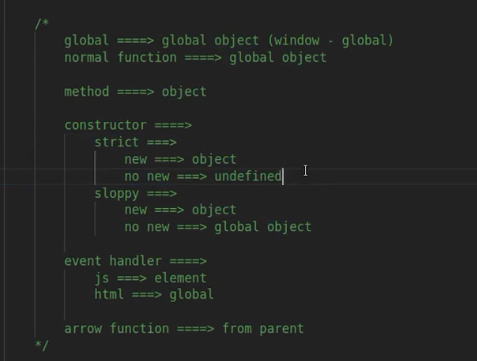
اقرا الكود و الكومنتات كويس جدا

#######################################################################  
this

-----------

تعبر عن حاجتين   
1- لو this موجوده فى مكان جلوبال فهى بتعبر عن ال Global Object ===< window

2- لو جوه Object فهنا ال this بتعوض عن اسم ال object   
الصوره دى بتوضح ال this بتعبر عن ايه فى كل حاجه



هتفهم اكتر من الكود حبه برضوا

*//this here is equal to Global (Window) Object****console***.log(**this**); *// Global this So Output Window Obj  
GetThisValue*();  
**function** *GetThisValue*() {  
 ***console***.log(**this**); *// Global this So Output Window Obj*}  
  
*/\*##################### this inside Object ####################\*/  
//this here is equal to (MyObj) Object***var *MyObj*** = {  
 **Name** :**'Mahmoud'**,  
 **Age** :27,  
  
 MyFunc : **function** () {  
 **this**.**Name** = **'Samy'**;  
 **function** *MYFuncion*(Param) {  
 **var** self = **this** ; *// self here became equal to this*   
 self.**Age** = Param;  
 ***console***.log(self.**Age**);  
 }  
 *MYFuncion*(20); *// 20 as i call function inside function Property* ***console***.log(**this**.**Name**); *// Samy as i change name by this Code this.Name = 'Samy';* },  
}  
  
***MyObj***.MyFunc();

#######################################################################  
this

-----------

لو عاوز بقه ابعت اى قيمه بمزاجى فى ال this تتعمل اذاى دى ؟؟

بفانكشن جاهزه اسمها call او Apply او bind و الفروقات بينهم بسيطه و ادى كود يوضح انى ممكن ابعت اى حاجه فى this طالما باعت الحاجه دى فى ال call

دى الكول و فيها انا ببعت اول حاجه القيمه الى انا عاوزها تعوض عن ال this و هنا خليت الى يعوض عن ال this الاوبجيكت و بعت فيه العنوان و بعد كده بعت البراميتارز الى الكونستراكتور بياخدها كل واحده لواحده

    <script>

        //constractor function

        function Person(nameP, ageP) {

            this.name = nameP;

            this.age = ageP;

        }

        var Obj = {

            address: "Cairo"

        };

        Person.call(Obj,"Mahmoud",25);

        console.log(Obj); //{address: "Cairo", name: "Mahmoud", age: 25}

    </script>

طيب لو عاوز اعمل نفس الكلام ب ال apply ؟؟ هو هو نفس الكلام الفرق بس ان البراميتار الى هبعتها هبعتها فى اراى فهيبقى طريقه كتابتها كده

  Person.apply(Obj,["Mahmoud",25]);

اما فى حاله ال bind هى ذيها ذى ال call بس الفرق ان البايند عشان تشتغل لازم احطها فى فاريبول من نوع فانكشن و اروح اعمل كول للفانكشن الى انا خزنتها فيها دى و بتتكتب كده

 var NewFunc =  Person.bind(Obj,"Mahmoud",25);

 NewFunc();

#######################################################################  
Array is Equal to list

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

فى الجافا سكريبت الاراى ذى الليست بينفع تعبى فيها اى انواع مع بعضيها خلطبيطه يعنى مش لازم كله من نوع واحد

طيب فى عكه كده عاوز اعملها انى احط في الاراى اوبجيكت و فانكشن و انادين الفانكشن الى فى الاراى و احط فيه البراميتر عباره عن الاوبجيكت الى فى الاراى تعملها اذاى دى :D :D الكود اهه يا ناصح هتتعمل كده

**var *My\_Array*** = [  
 5 ,  
 **true**,  
 {  
 **Name**:**'Mahmoud Badawy'** ,  
 **Age** : 15,  
 } ,  
**"StringVar"** ,  
 **function**(Param){  
 **var** Greating = **'Hello'**;  
 ***console***.log(Greating + **' '** + Param)  
 },  
];  
  
*//Call function that Take name From Obj as Param****My\_Array***[4](***My\_Array***[2].**Name**)

هتلاخظ انى بديس الفانكشن ولا الاوبجيكت اى اسامى لان اصلا هما فى اراى بيخدوا index   
و بالتالى ذى ما انت شايف اهه الفانكشن ناديت عليها بالانديكس 4 و عشان اخليها تشتغل عملتها انفوك و ادتها () وجواه برامتير ذى ما هى عاوزه و كان البراميتر ده عباره عن انى بنادى على الابوجيكت بالانديكس الى هو فيه و بعجيها اقوله انا عاوز البروبرتى الفلانيه الى هى الاسم مثلا فبيروح يجيبه و يشغلك الدنيا دى و بس كده

#######################################################################  
arguments

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

دى عباره عن built in array كل الى بتعملها انها بتاخزن جواها اى براميتر بتاعت اى فانكشن و بتستخدم داخل الفانكشن الى انا محتاج اتشك فيها اذا كان البراميتر اتبعتت ولا لأ و على فكره دى ممكن استخدمها فى حاجات كتير اولها انى اقدر استخدمها فى انى اعمل ديفولت لكل البراميتر و هوريك فى تانى مثال المهم ان اول مثال ده بيوضحلك هى شغاله اذاى هنا بقوله لو عدد البراميتر الى داخله الفانكشن اقل من الى موجودين فعلا فقولى ان فى براميتر ناقص

**function** *MyFunc*(FName , LName , Lang) {  
 ***console***.log(***arguments***); *// arguments is an Array that save Function Param in it* **if**(***arguments***.length < 3)  
 {  
 ***console***.log(**'Missing parameter in MyFunc function Call'**);  
 }  
 **else** {  
 ***console***.log(FName + **' '** + LName + **' . My Language is : '** + Lang);  
 }  
  
}  
  
*MyFunc*();  
*MyFunc*(**'Mahmoud'**);  
*MyFunc*(**'Mahmoud'**,**'Badawy'**);  
*MyFunc*(**'Mahmoud'**,**'Badawy'**,**'English'**);

هنا بقه انا هخلى كل براميتر الارجيومنت يعمله ديفولت فاريبول و على فكره اقدر استخدمها فى الاوفرلودينج برضوا :D :D

**function** *MyFunc*(FName , LName , Lang) {  
 ***console***.log(***arguments***); *// arguments is an Array that save Function Param in it* **if**(***arguments***[0]== **undefined**)  
 {  
 ***arguments***[0] = **'Mahmoud'**;  
 FName = ***arguments***[0];  
 }  
 **if**(***arguments***[1]== **undefined**)  
 {  
 ***arguments***[1]=**'Badawy'**;  
 LName = ***arguments***[1];  
 }  
 **if**(***arguments***[2]== **undefined**)  
 {  
 ***arguments***[2]=**'English'**;  
 Lang = ***arguments***[2];  
 }  
 ***console***.log(FName + **' '** + LName + **' . My Language is : '** + Lang);  
}  
  
*MyFunc*(); *// Mahmoud Badawy . My Language is : English  
  
MyFunc*(**'Mostafa'** , **'Mahmoud Badawy'** , **'Arabic'**);  
*//Mostafa Mahmoud Badawy . My Language is : Arabic*

#######################################################################  
Overloading Function

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

اتنين فانكشن بنفس الاسم بس ببراميترات مختلفه بس كده و انا من وجهه نظرى ان كود الديفولت الى فوقه ده يعتبر بعمل اوفرلودينج بشكل متحايل على اللغه

**function** *MyFunc*(FName , LName , Lang) {  
 FName = FName || **'Mahmoud'**;  
 LName = LName || **'Badawy'**;  
 Lang = Lang || **'English'**;  
 ***console***.log(FName + **' '** + LName + **' . My Language is : '** + Lang);  
}  
  
*MyFunc*(); *// Mahmoud Badawy . My Language is : English  
MyFunc*(**'Karim'**); *// Karim Badawy . My Language is : English  
MyFunc*( **'Karim'** ,**'Elkerdasy'**); *// Karim Elkerdasy . My Language is : English  
MyFunc*(**'Mostafa'** , **'Mahmoud Badawy'** , **'Arabic'**);  
*//Mostafa Mahmoud Badawy . My Language is : Arabic*

اهه نفس الفانكشن و بستخدم فيها كذا براميتر عادى و طبعا التغيير جوه هيبقى بسيط فشخ

#######################################################################  
dangerous Return Semi Colon

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

خد بالك ان الريتيرن مينفعش تتكتب فى سطر و الى هترجعه فى سطر لان اللغه اوتوماتيك بتحط بعد السطر ; فساعتها الدنيا هتخرب و عشان تفهم جرب الكودين دول

**function** *MyFunc*() {  
**return** {

**Name** : **'Mahmoid'** , **Age** : 15}  
}  
***console***.log(*MyFunc*());

شوف بقه لما اعمل الريتيرن و الداتا الى المفروض يرجعها فى سطر هيديك انه error على طول

**function** *MyFunc*() {  
**return** {  
 Name : **'Mahmoid'** , Age : 15   
 } ;  
}  
***console***.log(*MyFunc*());

لانه اعتبر ان الكود مكتوب كده و اسوأ كمان

**function** *MyFunc*() {  
**return** ;  
}  
***console***.log(*MyFunc*());

#######################################################################  
interface

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

فكره الانترفيس انها بتمشيك على قواعد و قوانين الى عامل الانترفيس يعنى قالك تعمل اوبجيكت فيه 4 بروبرتيس اسمهم كذا يبقى لازم يتعملوا قالك اعملى 5 فانكشن مثلا و اسمهم كذا و بروبرتيس و اسمها كذا برضوا لازم تعملها كده

**class** Myclass {  
 *// Object Internal interface* **person** : {  
 **Name**: **string** ,  
 **Age**: **number** ,  
 **address** : **string** ,  
 **Job** : **string** ,  
 *//Methods* Myfunc\_1();  
 Myfunc\_2();  
 }  
  
 **constructor**(){  
 *// so we have to set all property and Methods here   
 // if you miss any one this will make Error in code* **this**.**person** = {  
 **Name**: **'Mahmoud'** ,  
 **Age**: 27 ,  
 **address** : **'Cairo'** ,  
 **Job** : **'Software Engineer'** ,  
 Myfunc\_1() {  
 **return 'hello from Myfunc\_1'**;  
 }  
 ,  
 Myfunc\_2() {  
 **return 'hello from Myfunc\_2'**;  
 }  
 }  
 }  
}

حاول تمسح الى معلم عليهم بالاصفر و اقرأ بقه الايرور الى طلعلك هتلاقيه بيقولك ان البروبرتيس ناقصه

دى بقه لو عاوز تعمل الانترفيس بره و تستقبله فى اوبجيكت يا كبير :D

**class** Myclass {  
 *// Object from External interface  
 //here we take from interface to person Object* **person** : person\_Intergace;  
  
 **constructor**(){**this**.**person** = {  
 **Name**: **'Mahmoud'** ,  
 **Age**: 27 ,  
 **address** : **'Cairo'** ,  
 **Job** : **'Software Engineer'** ,  
  
 Myfunc\_1() {  
 **return 'hello from Myfunc\_1'**;  
 }  
 ,  
 Myfunc\_2() {  
 **return 'hello from Myfunc\_2'**;  
 }  
 }  
 }  
}  
*// External interface*  
**interface** person\_Intergace {  
 *//properties* **Name**: **string** ,  
 **Age**: **number** ,  
 **address** : **string** ,  
 **Job** : **string** ,  
 *//Methods* Myfunc\_1();  
 Myfunc\_2();  
}

طيب لو عاوز اعمل من الانترفيس اراى اوف اوبجيكتس :D :D

**class** Myclass {  
 **person** : person\_Intergace[];  
   
 **constructor**(){  
 **this**.**person** = [{  
 **Name**: **'Mahmoud'** ,  
 **Age**: 27 ,  
 **address** : **'Cairo'** ,  
 **Job** : **'Software Engineer'** ,  
 },  
 {  
 **Name**: **'Mostafa'** ,  
 **Age**: 1 ,  
 **address** : **'Cairo'** ,  
 **Job** : **'Google Engineer'** ,  
 }  
 ];  
 }  
}  
  
**interface** person\_Intergace {  
 *//properties* **Name**: **string** ,  
 **Age**: **number** ,  
 **address** : **string** ,  
 **Job** : **string** ,  
  
}

الطريقه الاشهر فى عمل ال interface و انى اعمل للانترفيس امبلمينت من اى كلاس ما

**class** Myclass **implements** person\_Intergace{  
 **Name** = **'Mahmoud'**;  
 **Age** = 27;  
 **address** = **'Cairo'**;  
 **Job** = **'SWE'**;  
   
 Myfunc\_1() {  
 **return 'hello from Myfunc\_1'**;  
 }  
 Myfunc\_2() {  
 **return 'hello from Myfunc\_2'**;  
 }  
   
}  
  
**interface** person\_Intergace {  
 *//properties* **Name**: **string** ;  
 **Age**: **number** ;  
 **address** : **string** ;  
 **Job** : **string** ;  
  
 Myfunc\_1();  
 Myfunc\_2();  
}

جرب فى الكلاس امسح اى فانكش او بروبرتى هتلاقى ايرور على طول يا اسطى :D :D

#######################################################################  
IIFE (self Invoke)

----------

دى طريقه نداء الفانكشن بشكل حالى يعنى حالا او دلوقتى بمجرد ما اعملها رن

و من هنا انا هوريد كل اشكال كتابه الفانشكن عشان تعرفهم و منها اخر اتنين هما ال IIFE

*// function By statement***function** *GreatingOne* (Name){  
 ***console***.log(**'Hello '** + Name)  
};  
*GreatingOne*(**'Mahmoud'**);  
  
*// function By Expression***var** *GreatingTwo* = **function** (Name){  
 ***console***.log(**'Hello '** + Name)  
};  
*GreatingTwo*(**'Mahmoud'**);  
  
*// function By Imediatelly invoke Function Expression (IIFE)***var *GreatingThree*** = **function** (Name){  
 **return** (**'Hello '** + Name)  
}(**'Mahmoud'**);  
***console***.log(***GreatingThree***);  
  
*// Best (IIFE)*(**function**(Name){  
 ***console***.log(**'Hello '** + Name)  
}(**'Mahmoud'**));

اخر فانكشن دى بتعمل انفوك من غير ما حد يبصلها ههههههه شغاله مع نفسها بمجر ما ارن الكود بتشتغل من غير ما حد يناديها يعنى Self Invoke من الاخر

#######################################################################  
IIFE safe Code

----------

اذاى بقه اخلى الفاريبول الجلوبال ميأثرش على الفانكشن من جوه و كمان يبقى تاثير الى جوه الفانكشن اقوى من الجلوبال

**var *greating*** = **'Hola '**;  
*// Best (IIFE) Not Safe Code*(**function** (Name) {  
 **var** greating = **'Hello '**;  
 ***console***.log(greating + Name);  
}(**'Mahmoud'**)); *// Hello Mahmoud****console***.log(***greating*** + **'Mostafa'**); *// Hola Mostafa***var *greating*** = **'Hellllllllllllo '**;  
*// Best (IIFE) Safe Code*(**function**(Global,Name){  
 **var** greating = **'Hello '**;  
 Global.**greating** = greating ;  
 ***console***.log(greating + Name)  
}(***window***,**'Mahmoud'**)); *// Hello Mahmoud****console***.log(***greating*** + **'Mostafa'**) *// Hello Mostafa*

لاحظ ان فى الكود الاول جوه الفانكشن احنا عملنا جريتينج جديده عشان الفانكشن متتاثرش بالجلوبال و اى حاجه بره الفانكشن الجريتينج اخدت الفاريبول الجلوبال

اما فى الحاله التانيه انا دخلت الجلوبال فاريبول جوه الفانكشن عن طريق الاوبجيكت window او this الى بيعبر عن الجلوبال سكوب و اى فاريبول جواه يعتبره بروبرتى بتاعه المهم بعد ما دخلناه جوه خليناه بيساوى Hello للى واخداها من ال greating الى فى اللوكال سكوب و بالتالى اصبح البروبرتى الى جوه اللوكال سكوب بيساوى Hello و اتغير بشكل جلوبال فلما اجى اندهه بره الفانكشن هتلاقيه ادانى Hello بدلا من Hellllllllllllllllllllllllllo  
تمام ذى تغيير اى بروبرتى فى اى مكان بيبقى تغيير بشكل جلوبال عادى خالص الفكره هنا انى بحمى الكود من ان يحصل اى تغيير فى الفاريبول من اى مكان تانى :D :D   
مهمه الكود التانى انه بيصعب اى تغيير خارجى جلوبال جوه الفانكشان و براها كمان :D :D

#######################################################################  
Clouser

----------

بص ده من فوايده انه بيخليك تنادى على الفانكشن اللوكال الى جوه فانكشن اصلا كجلوبال و بالتالى لو هعمل زرار عداد كاونتر و دخلتله الكاونتر بزيروا هيتعامل على انه من بره عادى و هتلاقى ان الزيرو بيزيد 1 و 2 و 3 كل دوسه و ده هورهولك فى اخر مثال بس هخدك واحده واحده فى الكلوشير افهمهولك

اول حاجه انا عندى فانكشن جواها فانكشن تانيه

    <script>

         function  outer() {

            var counter = 0;

            function inner () {

                counter += 1;

            }

   inner ();

            console.log(counter);

        };

        outer(); //0

        outer(); //0

        outer(); //0

        outer(); //0

    </script>

الى حصل ايه بقه يا معلم ؟؟ بص يا سيدى هو دلوقتى انا عملت فانكشن اسمها outer الفانكشن جواها فاريبول اسمه counter الفاريبول ده عرفته و بعد كده رحت عامل فانكشن جوه الفانكشن الاوتر و مسميها inner و بعد كده قولت انادى فانكشن الاينار دى عشان ازود الكاونتر منها و بعد كده رحت اشوف الكاونتر ده هيزيد كل مره هنادى فيها الفانكشن الاوتر ولأ لأ ايه الى حصل بقه ؟؟ لقيت انه زود 1 بس على الرغم انى استدعين كذا فانكشن اوتر الا انه رجع كلهم بنفس النتيجه و معملتش عداد كاونتر طب ايه السبب ؟؟

بص يا سيدى طريقه الكتابه دى الى هى العاديه خلت الكود يمشى اذاى ؟؟

اولا الكومبيلار دخل على اول فانكشن الى هى اوتر لقى فيها فاريبول كاونتر راح واخدها فى الميمور بصفر و بعد كده لقى فانكشن راح مخزنها لوقت عوزه و بعد كده لقى نداء على الاينر فانكشن راح رايح منفذها و خلى الكاونتر من صفر بقى واحد ثم بعد كده اظهر قيمه الكاونتر فى الكونسول و بعد كده لقى ايه ؟؟ لقى ان فانكشن الاوتر خلصت خلاص راح فهم كده بقه ان المفروض يروح يمسح الجاربيدج من الميمورى و ده معناه ان راح مسح البيانات الخاصه بالفانكشن دى من الميمورى و بالتالى رجع الفاريبول كاونتر تانى لصفر فلما جيت انادى للفانكشن مره تانيه راح دخل فى الفانكشن من تانى بس كانه داخلها أول مره لأن هو خلاص مسح اى حاجه ليها علاقه بيها فى الميمورى مع نهايتها المره الى فاتت

طيب راحوا عشان يحلوا المشكله دى و يخلوه ميمسحش الجاربيدج من الميمورى و يختفظ بقيمه اى حاجه جوه الفانكشن بحيث الكاونتر هنا ويبقى بيعد و الصفر يبقى واحد ثم اتنين ثم تلاته و هكذا من غير ما يتمسح و يرجع صفر تانى  
   
 راحوا عاملين حاجه اسمها clouser و دى مهمتها ان الجاربيدج ميتمسحش من الميمورى و قيم الفاريبول الجلوبال الى فى الفانكشن الاساسيه outer تفضل موجوده فى الميمورى بأخر قيمه هى وصلت ليها حتى بعد انتهاء العمل فيها و طريقه كتابتها انى **برجع الفانكشن الداخليه الى هى ال inner فى return** فبكده خلاص انا كده بقوله اى حاجه تابعه للفانكشن الاساسيه outer متمسحهاش من الميمورى ياعم الحج و خليها محتفظه باخر قيمه اتخزنت فيها

    <script>

       function  outer() {

            var counter = 0;

            return function () {

                counter += 1;

                return counter ;

            }

        };

        var x =  outer();

        console.log(x()) ; //1

        console.log(x()) ; //2

        console.log(x()) ; //3

        console.log(x()) ; //4

    </script>

خد بالك برضوا ان انا عشان اشغل الفانكشن الاوتر صح كان لازم و اجبارى اخزنها فى فاريبول و اروح ارن الفاريبول عشان يبقى كانى عملها سيلف انفوك عشان ترجعلى القيم و مترجعليش الفانكشن ذات نفسها و تعالى بقه نشرح اذاى اعمل الكلوشر مع السيلف انفوك لان الاتنين لازم يكونوا فى كود واحد مع بعض على طول فانا فهمتك فوق الطريقه السهله و جه وقت الطريقه الى لفه شويه بسيطه ههههههههه

عشان الدنيا تبقى سهله فانا هبتدى الشرح من السيلف انفوك الاول قم اديك المثال بالكلوشر مع السيلف انفوك كامل

اول كوده شوفه و اشرحه

**var *SayHello*** = **function** (greating) {  
 **return** greating; *// Hello*}(**'Hello '**);  
***console***.log(***SayHello***);

هنا عامل فانكشن بتعمل SelfInvoking بتاخد براميتر اسمه جريتينح و من المعروف ان اى فانكشن بتعمل سيلف انفوكينج و بتحتاج براميتر فانا بحط البراميترز بتعتها برضوا فى قوس الانفوك و لذلك انا حطيت Hello فى قوس ال invoking وذى ما انت شايف الانفوكينج فانكشن بترجع قيمه القيمه بخدها احطها فى فاريبول اسمه SayHello الى بعد كده بعرضها فى الكونسول

الكود التانى اهه بقه

**var *SayHello*** = **function** (greating) {  
 **return function** (Name) {  
 ***console***.log(greating + Name);  
 };  
}(**'Hello '**);  
***SayHello***(**'Mostafa'**);

ده بقه الكود التانى الى هو فى فيه فانكشن انفوك بتدخل براميتر جريتينج الى هى Hello و جواها بقه بيعمل ريتيرن لفانكشن و الفانكشن الى جوه دى بتحتاج عشان تشتغل اسم عشان تطلعلك فى الاخر Hello Mostafa مثلا

اذا كده الفانكشن الاولى بترجع اصلا فانكشن و بالتالى الفاريبول SayHello دلوقتى المفروش انه من نوع فانككشن عشان هيرجع فيها حاجه من نوع فانكشن اذا انا كده اقدر اتعامل مع الفاريبول على انها فانكشن و بالتالى اعتبر ان دى اسم فانكشن و ممكن ادخل جواها براميترز و فعلا حصل و فعلا شغلت الفانشكن الى جوه و ده بقه الى بيتسمى Clouser و هو انك تشغل فانكشن لوكال جوه فانكشن على انها جلوبال

الكود التالت

**function** *great*(greating) {  
 **return function** (Name) {  
 ***console***.log(greating + Name);  
 };  
};  
**var *SayHello*** = *great*(**'Hello '**);  
***SayHello***(**'Mostafa'**);

دى اسهل طريقه تفهم الدنيا بيها يا باشا بقه

الكود الرابع

(**function** (greating) {  
 **return function** (Name) {  
 ***console***.log(greating + Name); *// Hello Mostafa* }  
}(**'Hello '**)(**'Mostafa'**));

ده بقه يا معلم بوريك اذاى اعمل انفوك لفانكشن جوه فانكشن :D :D و ذى ما انت شايف ببعت البراميترز اذاى

من اهم اكواد توضح شغل ال clouser

**var *add*** = (**function** () {  
 **var** counter = 0;  
 **return function** () {  
 counter += 1;  
 ***console***.log(counter)  
 }  
})();  
***add***(); *// 1****add***(); *// 2****add***(); *// 3*

ذى ما انت شايف ان الكاونتر لوكال سكوب و بيساوى صفر يعنى المفروض فى العادى انه كل مره هنادى الفانكشن هيبقى ب بصفر و لكن لأ هو اعتبره بيناديه كل اما يتغير من ال Reference بتاعه على اساس ان فى فانكشن تانيه جوه الفانكشن دى هى الى مخليه يتخزن فى مكان واحد فى الميميورى و هو بيتعامل مع عنوان الكاونتر و بيزود عليه فبيظهر انه كل مره بيزيد واحد   
و اسهلهالك خالص

الفانكشن الى جوه معتبره ان الكاونتر بالنسبالها فاريبول جلوبال و بالتالى ال ++ الى جوه الفانكشن شغاله فل الفل

<!DOCTYPE **html**>  
<**html**>  
<**body**>

<**button type="button" onclick="***myFunction*()**"**>Count!</**button**>  
<**p id="demo"**>0</**p**>

<**script**>  
 **var *add*** = (**function** () {  
 **var** counter = 0;  
 **return function** () {counter += 1;  
 **return** counter;}  
 })();  
  
 **function** *myFunction*(){  
 ***document***.getElementById(**"demo"**).innerHTML = ***add***();  
 }  
</**script**>  
</**body**>

ده بقه كود بيوريك تستخدم الكوجر اذاى عمليا

The counter is protected by the scope of the anonymous function, and can only be changed using the add function.

ده بقه كود كلوشر برضوا بيوريك فيه انه بيسجل بالريفرنس

**function** *add*() {  
 **var** arr = [];  
 **for** (**var** i = 0; i < 3 ; i++ ) {  
 arr.push( **function** () {  
 ***console***.log(i)  
 });  
 }  
 **return** arr ;  
};  
  
**var *fs*** =*add*();  
***fs***[0](); *// 3****fs***[1](); *// 3****fs***[2](); *// 3*

بص ياعم دلوقتى انت جوه الفانكشن add فى array الاراى دى بقه انا بخزن في كل انديكس فانكشن تمام كده

داخل الفانكشن الى بخزنها جوه الاراى دى بقه المفروض انى هعرض فيها ال I الى هى من الاخر ذى الكاونتر كده الى فات هيعد يتغير يتغير لحد ما يستقر على انه بقى بكام بقى ب 3 و بالتالى هو هيريفير او هيشاور للعنوان الى فيه القيمه دى الى هى تلاته لانه بيتغير بالريفرنس

و بالتالى فى الفانكشن add حصل انى خزنت 3 فانكشنز فى الاراى اذا المفروض انا هعمل ريتيرن لايه ؟؟

للاراى طيب دلوقتى انا جيت حطيت الريتينرن الى خارج من ال Add() فى فاريبول اذاى اصبج الفاريبول   
fa عباره عن اراى جواه انديكسيس كل انديكس جواه فانكشن و عشان كده انا ندهت للفانكشنز بالشكل الى الى انت شايفه تحت خالص ده و كله فى الاخر هيطلع 3 عشان بيجيب اخر رقم وصله ال i عشان بيبص على العنوان مش القيمه و لما بيبص على العنوان بيروح واخد قيمه العنوان و بالتالى لازم كله يساوى حاجه واحده

#######################################################################  
funcational Programming Libraries

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

بص يا سيدى دى مواقع لبعض المكاتب المهمه جدا جدا للجافا سكريبت لان الفانكشنز الى فيها معموله Not immutable و بالتالى استخدمها للميمورى و الوقت قليل جدا فبتخلى مشروعك سريع جدا لما بتستخدمها و كمان فى فانكشنز كتير built in معموله فيهم مش موجوده فى الجافا سكريبت الاساسيه

<https://lodash.com/>

<https://immutable-js.github.io/immutable-js/docs/#/>

<https://swannodette.github.io/mori/>

<https://underscorejs.org/>

المكاتب دى كلها بتساعد انى لما يكون عندى داتا كبيره فى اراى انى استخدمها بدون ما اثر على البرفورمانس و ده لانهم بيستخدموا فيها ال Shared Data structure concept

#######################################################################  
Best Extension For VS Code

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

اكستنشن عشان اعمل سيرفر لوكال للتجارب على الجافا سكريبت   
live server  
و دى اكستنشن مهمه لانجيولار   
angular files

و ده عشان تنظيم الكود

Beautify