

# BASICS OF JAVASCRIPT

## What is JavaScript?

JavaScript is a **dynamic** computer programming language. It is lightweight and most commonly used as a part of web pages, whose implementations allow **client-side** script to interact with the user and make dynamic pages. It is an interpreted programming language with object-oriented capabilities.

Client-side JavaScript is the most common form of the language.

يعنى ال javascript هى بتتعامل مع ال client-side بتاعك اللى هو مثلا ال browser و ده ليه مميزت كتير جدا فانه مثلا انت لو عايز تعمل check هل ال user وهو بيدخل ال data بتاعته فال form ال mail بتاعه كان valid ولا لا فبذل ما تروح لل server ذات نفسه و تعمل ال check ده و طبعا هيحصل load كتير لما تروح لل server يتأكد و يرجع انت ممكن تعمل بال javascript وال validation ده يحصل فال browser بتاعك فقلل من ثانيه مثلا عشان كده هو client-side

وممكن من خلالها تخلى ال web page بتاعتك dynamic يعنى مثلا ممكن لما ال user ي click على اى button او يعمل اى event يروح مثلا يعمل tag معين فال html او يخفى حاجه انت مش عايزها .

JavaScript was first known as **LiveScript**, but **Netscape** changed its name to JavaScript.

وده كان موجود فى وقتها 3 languages وهما ال **java,scheme,self** و كان ال design بتاع كل language منهم مختلف تمام عن التانيه فاخدوا من ال java ال **syntax** وخذ من ال scheme ال **concept** بتاع ال **function in a first-class object** و خد من ال self ال **oop prototype based** و جمع التلاته دول مع بعض و طلع حاجه اسمها **livescript** ومنها المنظمه العالميه **ECMA** سمته **ECMAScript** وده المفروض الاسم الصح بتاعها حاليا و لكن الاسم السوقى لغايه دلوقتى هو **javascript** و بدأ ينزل منها versions لغايه

## يعني ايه OOP Prototype-based ؟

ال OOP ده اختصار **Object-Oriented programming** ده زى **technique** كده فى لغات برمجه بت support ال **technique** ده ولغات لأ وده تقدر من خلاله ت **simulate** ال **reality** يعني نحاكى الواقع لان مثلا بيكون عندك حاجه اسمها **class** ده فى ال **common properties** بتاعه الحاجه اللى بتمثلها دى و فيه ال **Methods** اللى بتقدر تعملها زى المثال اللى كان فال session لو عايز تمثل الدكتور عموما فالدكتور ليه id,name,salary,position وليه برودو **Methods** او وظائف بيعملها ده ال **class** عمتا بيبقى حاجه مش ملموسه لما مثلا اقول دكتور احمد ال id بتاعه ١٢٣ وال salary بتاعه ١٢٣ وال position بتاعه المدير مثلا فكده انا اخدت object من ال **class** ده .

فبكره فى فال OOP نوعين حاجه اسمها **class-based** وده اللى هو الشرح اللى فات اللى هو **class** واخذ منه **objects** و حاجه اسمها **Prototype-based** يعنى نموذج مبدأى ده بقى بيبقى عندى object و واخذ منه **objects** و ال object ده بيبقى حاجه ملموسه عادى زى مثلا ترييزه معينه موجوده وانت ت customize حاجات عليها عشان تطلع شكل جديد .

وبكره ال javascript هي **OOP prototype-based** .

## ويعنى ايه functional programming language اللى اتخدت من ال Scheme ؟

هنا انا اقول على اللغه اللى بستخدمها دى هي functional programming language لما تكون ال function دى هي **first-class object** يعنى تكون حاجه مهمه اوى فيه طب امتى اقول ان ال function فاللغه دى هي first-class object ؟ لو اتحقق عندى اربع شروط

١ - لما اقدر اعرف ال function بتاعتي ف variable عادى زى مثلا ال integers انا

ممکن تحطه ف variable x=3

٢ - و ممكن ابعتها ك parameter ل function مثلا

٣ - و ممكن يكون property ف object زى مثلا ممكن من ضمن ال properties

بتاعه object هو ال age فممكن احط فيه 24 مثلا

٤ - و ممكن اعمله return من function

فعندى ال integer مثلا is a first-class object عندى فاللغه اللى شغال بيها لانه بيحقق الاربع شروط دول وهكذا

فانا هجيب ال **function** بردو عندى فاللغه بتاعتى و اطبق عليها الاربع شروط دول لو اتحققوا يبقى ال **function** عندى عنا **is a first-class object** و تبقى اللغه دى **functional programming language** .

فال **javascript** عندى **functional programming language** لان ال **function** فيها **is a first-class object**

---

- **Syntax:**

الاول كده انت ممكن تكتب **internal javascript** عن طريق انك تيجى قبل قفله ال **body** وتعمل كده

```
<script>
    document.write("Hello");
</script>

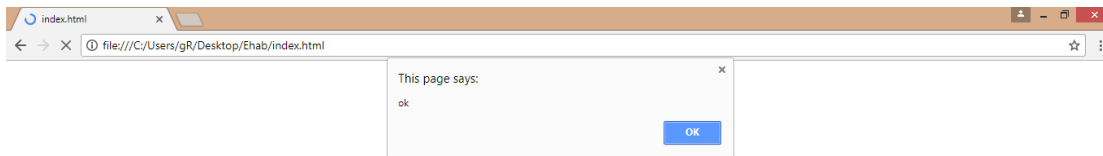
</body>
</html>
```

او ممكن تعمل **external** عن طريق انك تكتب ال **script** بتاعك قبل قفله ال **body** و تحط فيه ال **src** اللى هترحله وفى ال **sheet** ده بتكتب الكود عادى

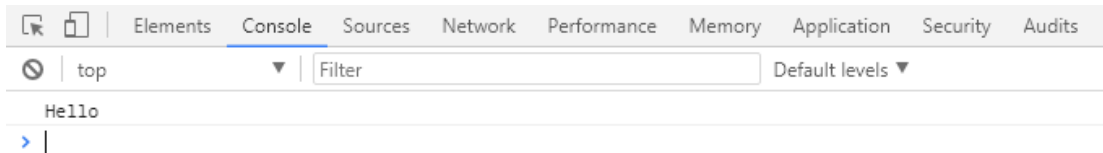
```
<script src="main.js"></script>
</body>
</html>
```

و عندنا كذا طريقه نعمل بيها **output** يعنى لو عايزين نشوف ال **result** بتاعتنا ايه او نتأكد من حاجه وهكذا

- **Alert()** : دى بخلى ال **output** يظهرلى عن طريق **message** من ال **browser**  
زى كده **alert("OK")**



- `Console.log()` : وده اهمهم , هنا انا بطلع حاجه ليا انا ك developer يعنى ال user مش بيشفها فبيبقى زى test ليا انا كده , بعمل inspect و بدوس على ال tab ال console مثلا `console.log("Hello")`



- `Document.write()` : هنا انا بكتب فال document بتاعتي ذات نفسها

`document.write("Hello")`

وممكن اكتب **tag** من ال **html** عادي وهو هيفهمه زى كده

`document.write("<p>test</p>")`

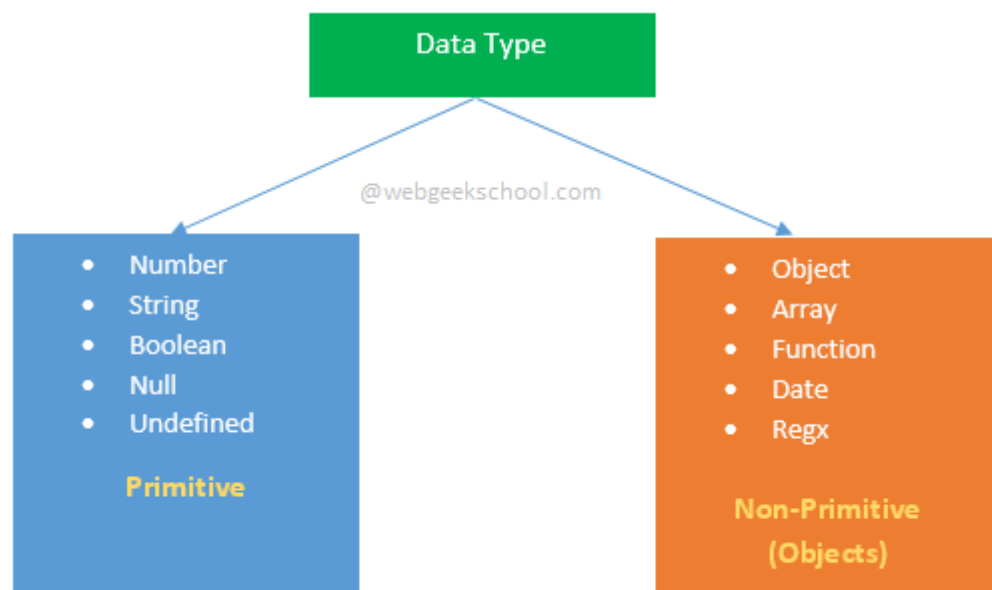
فكأني كتبت فال HTML كده `<p>test</p>`

- `Document.getElementById("Hena bktb el ID elli 3iz`

`twslslo").innerHTML="ay7aga"`

هنا انت بتحدد ال element اللي ليه id معين

## • DataTypes:



JavaScript Data Type

فأى لغة برمجة عندنا حاجة اسمها **datatype** يعنى انت فالتطبيعى بتكتب data فمينفعش تكتب اى data وخلاص لازم تكتب ال type بتاع ال data ده ايه عشان ال data دى اكيد طبعا بتتخزن فال **memory** عندك فلازم ال memory تعرف **مساحة** المكان اللى هتجزه لل data بتاعتك دى ايه .

بس فى لغات برمجة بتبقى **weakly typed languages** يعنى فال javascript هى weakly typed language لان لما باجى اعرف data **مش بحدد** ال **type** بتاعه , هو ب detect ال type لوحده مثلا كده **var x="test";** كده عرف ان ال x دى شيله data من النوع **string** ولو مشيت شويه فالكود و كتبت **x= 3** مش هيعترض لانه بردو هيعتبر ال x هنا كده **number** وفى **functions** جهزه موجوده اسمها **typeof(x)** هنا هيجبك ال type بتاع ال variable اللى قولتله عليه

### - Boolean:

ده بياخد قيمتين بس **true , false**

### - Undefined :

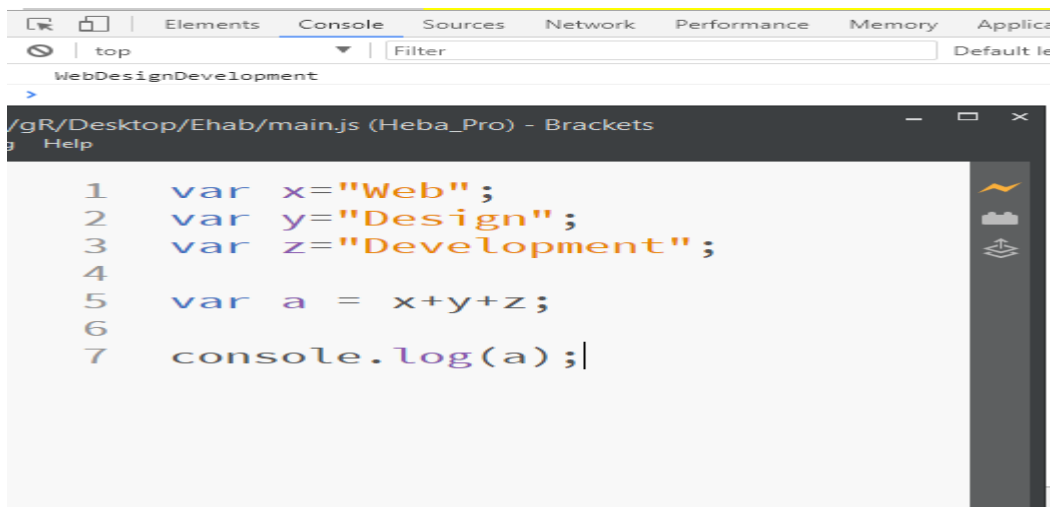
هنا لو انا عرفت **variable** و مدتلوش قيمة زى مثلا **var x;** وجبت عملت كده **console.log(typeof(x))** هيقولى **undefined** عشان مفهوش قيمة .

### - String:

ده لما بكتب **text** عادى مثلا **var x="test"** فهنا ده ال datatype بتاعه **string**

فى **methods** كتير موجوده ممكن اعملها على ال string ده زى مثلا

- **Concatination** : ده كانه بيلزق 2 strings او اكثر ببعض زى كده



The image shows a web browser's developer console with the 'Console' tab selected. The console output shows 'WebDesignDevelopment'. Below the console, a code editor window titled '/gR/Desktop/Ehab/main.js (Heba\_Pro) - Brackets' displays the following JavaScript code:

```
1 var x="Web";
2 var y="Design";
3 var z="Development";
4
5 var a = x+y+z;
6
7 console.log(a);
```

ولو عملت  $x + 3 + 5$  كده هيطلع **Web8** لكن لو كتبت  $x + 5 + 3$  كده هيطلع **x53** لانه اعتبر ال 3 , 5 دول **string** لانهم موجودين بعد **string** .

- **Length** : هنا انا ممكن اجيب طول ال **string** بتاعى مثلا ال **x.length** هيجبلى 3 لان ال **x** متكونه من 3 حروف .

- **indexOf()** : انا مثلا عايزه اعرف مكان كلمه معينه فال **string** دى مكانه رقم كمان او حرف فالمثال اللي عندنا ده مثلا **a.indexOf('b')** هيطلع 2 لان بيعد من اول **zero** .

- **Split("")** : هنا انا مثلا عندي ال **a** عبارة عن **Web Design and Development** فعايزه اقسام الجملة دى لكلمات مثلا فكده عايزه اقسامها عند ال **spaces** الموجوده فبعمل كده **var b = a.split(" ");** عملت بين ال **""** space عشان ده اللي عايزه اقسام بيه

## - Array:

ده من النوع **object** و ممكن احط فيه اكثر من **value** مع بعض

```
var x = ["Web","Design","test"];
```

و لو عايزه مثلا اوصل لكلمه **Design** بس بعمل كده **x[1]** لانه بردو **zero based** يعنى بيعد من **zero**

وفى مجموعه **methods** ممكن اعملها على ال **array** ده زى مثلا

- **Length** : ده بيجبلى طول ال **array** بتاعى مثلا **x.length** ده هيطلع 3 لان فيه 3 **elements** .

- **Concat** : هنا ممكن اخلى كذا **array** مع بعض فلو عندي مثلا

```
var y = ["Hello","world"];
```

فبعمل **var z = x.concat(y)** كده ال **z** فيها

```
["Web","Design","test","Hello","world"]
```

- **Push** : بيضفلى فآخر ال **array** بتاعتي مثلا **y.push("Ahmed")** فكده ال **y** فيها **["Hello","world","Ahmed"]**

- **Pop()** : دی بیشیل من اخر ال Array مثلا **y.pop()** كده انا مسحت من ال y كلمه Ahmed و بقت فيها Web,Design بس .

- **Splice()** : دی لو عايزه تعامل مع اى عنصر فال Array بتاخذ ال **indexOf** ال **element** ده و كام عنصر عايز تحذفه و ال **elements** اللى عايز تضيفها مثلا **z.splice(2,1,"A","B")** كده انا حذف من ال z كلمه **test** وضفت كلمه **A,B** فال z بقت كده **["Web","Design","A","B","Hello","world"]** .

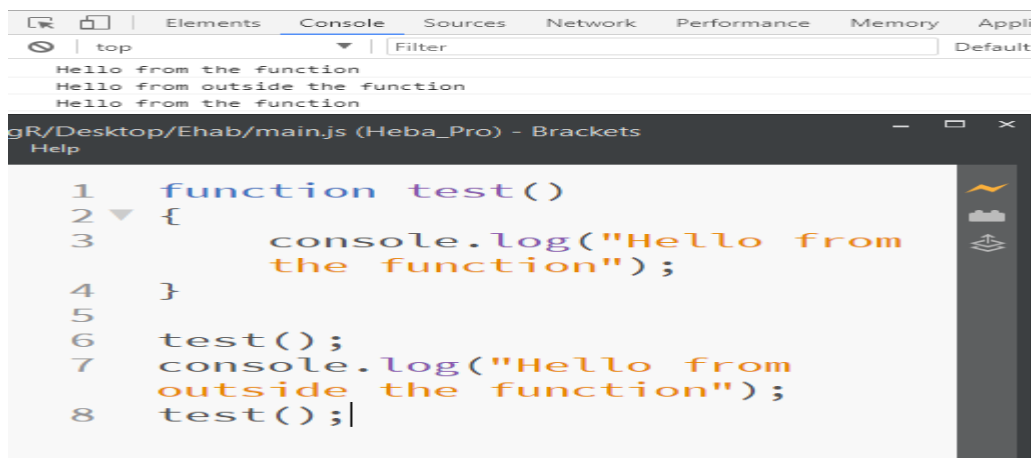
- **Unshift()** : دی زى **push** بس بتضيف من الاول .  
- **Shift()** : دی زى **pop** بتحذف من الاول .

## - Object :

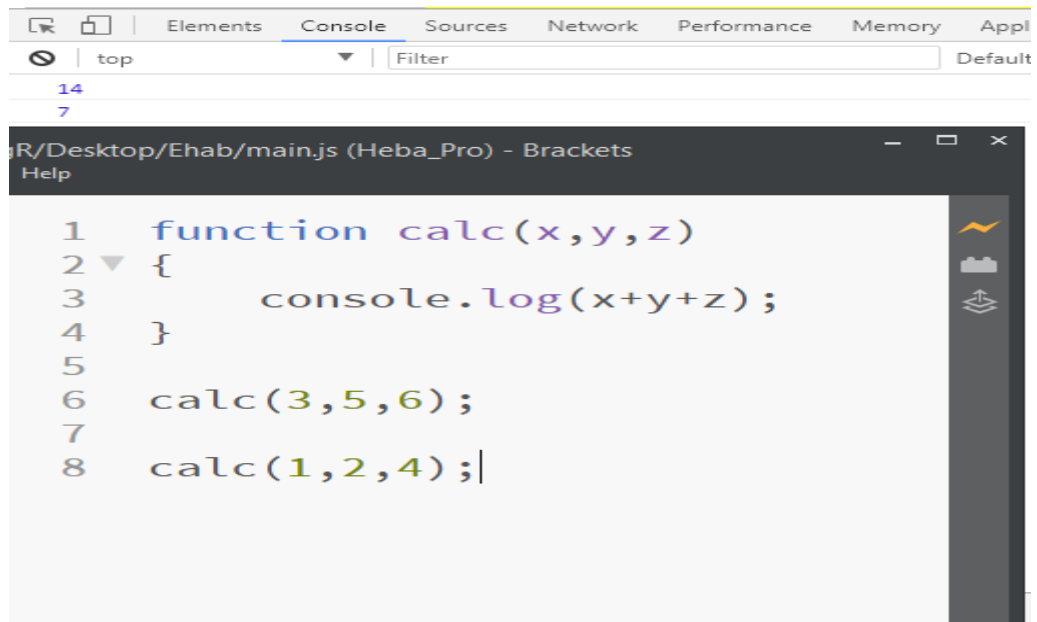
ده طريقه كتابته مختلفه بتبقى كده **var x = {firstName:"Ali",age:12}** ولو عايزه اطبع **firstName** بس بتعمل كده **console.log(x.firstName)** فكه هيطبع **Ali**

## - Function:

دى داله او **method** بتعمل حاجه معينه عباره عن مجموعه من الكود ممكن بتكرر المجموعه دى كل شويه مثلا بس بقيم مختلفه فبدل ما تكتب مجموعه ال code دى كل شويه , انت ممكن تحطها ف **function** و تعملها **call** وقت ما تعوز تستخدمها

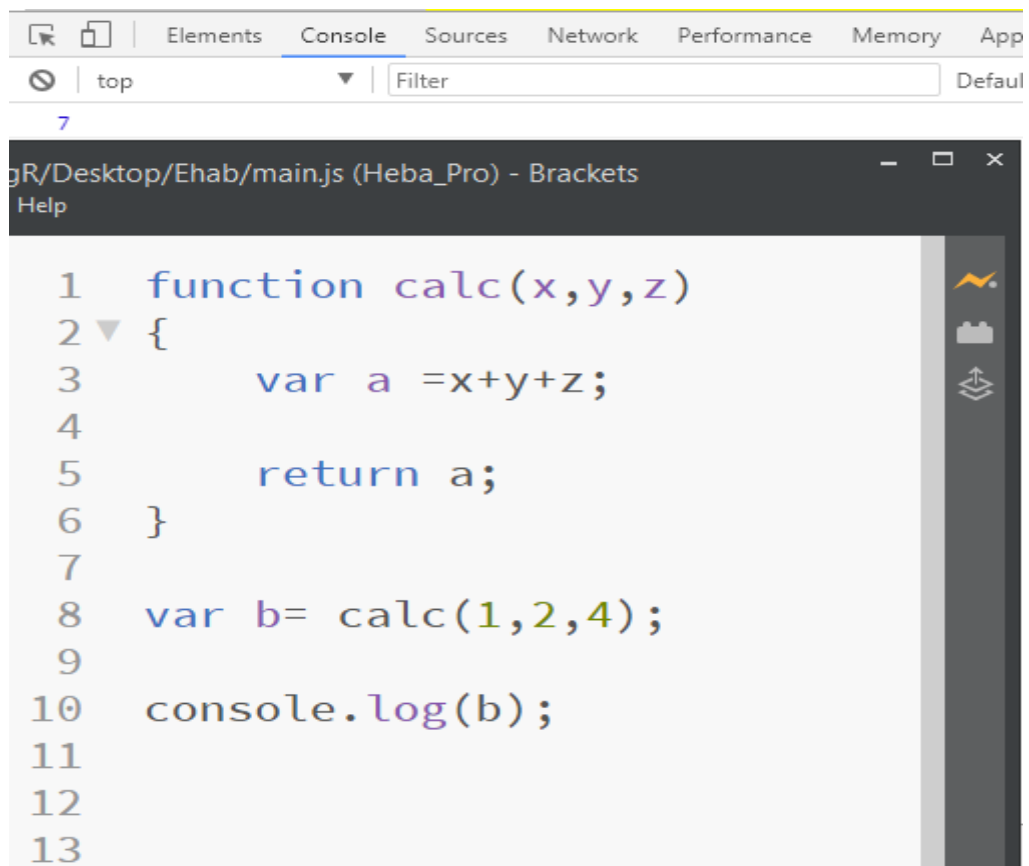


القوسين دول بحط فيهم ال **parameters** اللى عايزه اديها لل **function** يعنى المتغيرات اللى فال **function** دى وهنا فاضين عشان مش محتاحه فالمثال بتاعى ده متغيرات .



```
1 function calc(x,y,z)
2 {
3     console.log(x+y+z);
4 }
5
6 calc(3,5,6);
7 calc(1,2,4);
```

هنا انا عملت لل function متغيرات لان ممكن اكون عايزه كل مره اعمل فيها call لل function ابعثها قيم مختلفه .



```
1 function calc(x,y,z)
2 {
3     var a =x+y+z;
4
5     return a;
6 }
7
8 var b= calc(1,2,4);
9
10 console.log(b);
11
12
13
```



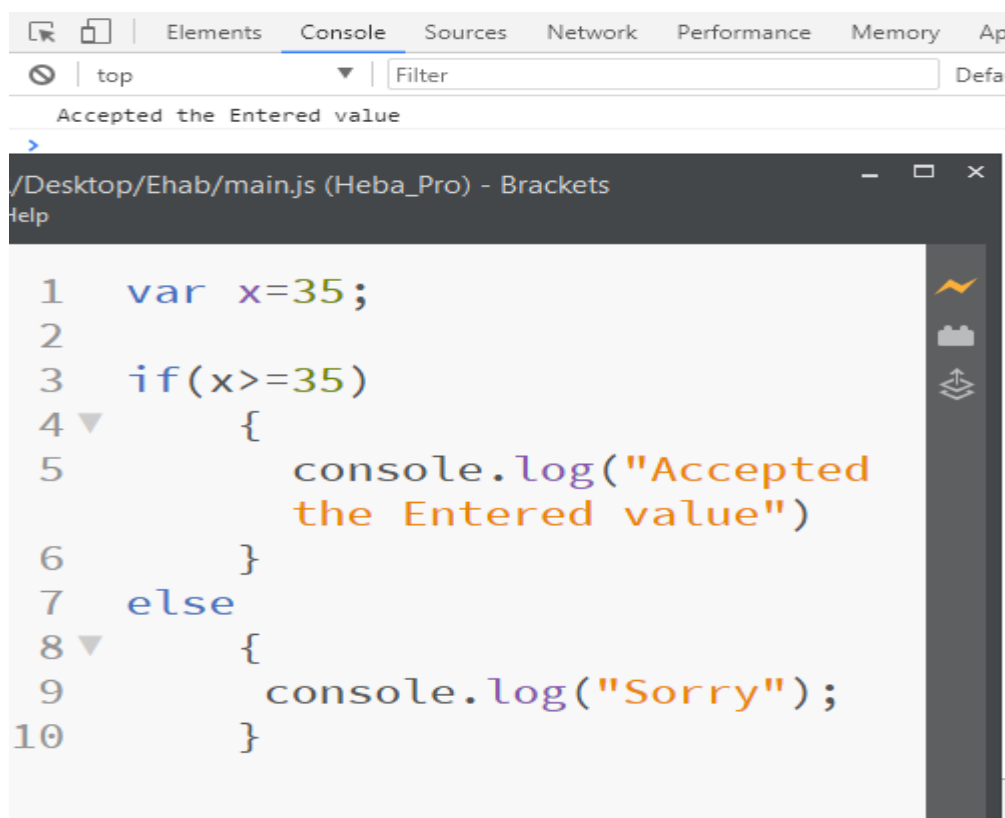
هنا انا ممكن اعمل call ال function بتاعتي و اخليها ترجعلي يعني تعمل **return** لقيمة مثلا وهنا لازم ال function دي اعملها call عن طريق **variable** زي هنا مثلا عن طريق ال b عشان ال variable ده يثيل القيمة اللي راجعه من ال function دي .

**ملحوظة :** انا لو جيت بره ال function دي واستخدمت ال **a** هيجبلي **error** لان ال **a** ده ال **scope** بتاعها او حدودها هي ال function دي بس وده اسمه كده **local variable** لكن لو **variable** بره ال function عادي فده كده اسمه **global variable** .

- **Condition:**

هنا انا ممكن اعمل **check** على شرط معين لو اتحقق هأخذ قرار ما و لو متحققش هأخذ قرار تاني فكده احنا بنتكلم عن اتحقق يعني الشرط ده كان صح او كان **true** او متحققش فبكده الشرط ده كان غلط او كان **false**

والشرط هنا بعمله عن طريق ال **If Condition**



```
1  var x=35;
2
3  if(x>=35)
4  {
5      console.log("Accepted
6          the Entered value")
7  }
8  else
9  {
10     console.log("Sorry");
11 }
```

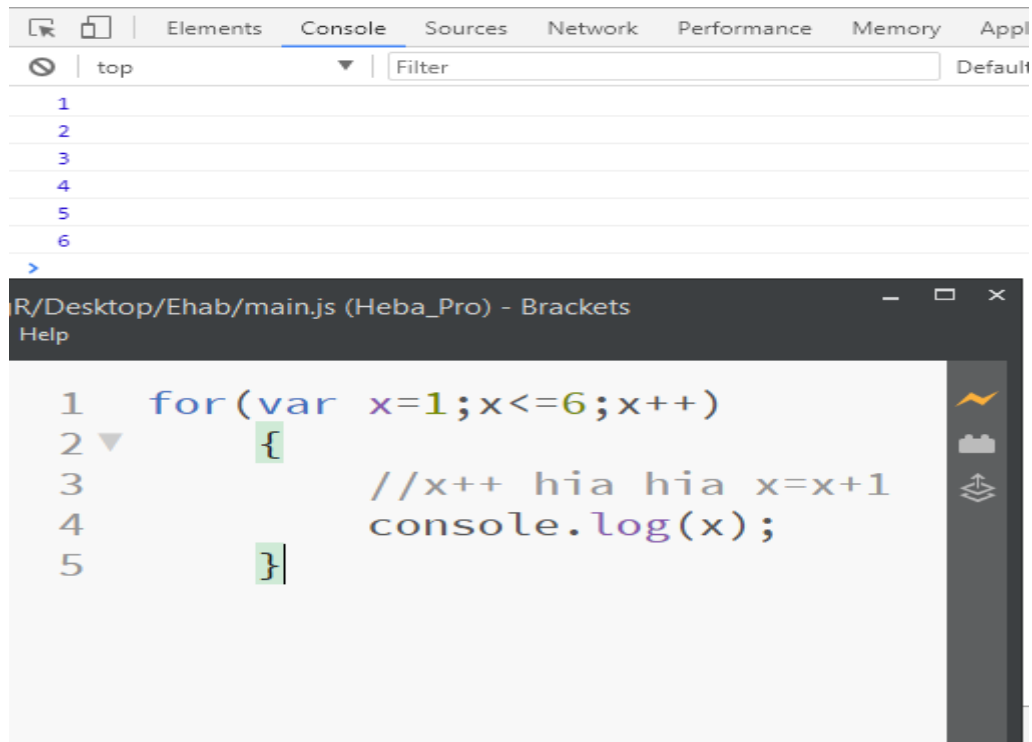
انا لو عملت فال **check** بتاعي كده **x==35** فكده كاني مسأله هلي ال **x** فعلا = 35 ؟  
لكن لو عملت كده **x=== 35** فكده انا بت **check** هل ال **x** فعلا = 35 و ليها نفس ال **datatype** ؟ اللي هي ال **number**

## • LOOP :

هنا انا لما اكون عايزه اكرر حاجه كذا مره بدل مثلا لو ٣ مرات فبدل ماكتبها ٣ مرات انا ممكن اعمل loop تكرر لي ده ٣ مرات . وال loop ليها كذا شكل او كذا طريقه

### 1- For loop:

هنا انا بقول لل loop بتاعتك فين بدايتك و فين نهايتك و ايه ال condition اللي كل مره بت check بيه هل تكمل ولا لا و بيكون بنهم ;



### 2- While:

دي عامله زي if condition كده بس ال if كانت بتتعمل مره واحده بس لكن ال loop ممكن تتكرر طول ما الشرط ده true فبالتالي لازم حاجه تكسر الشرط ده و تخليه false عشان متدخلش ف infinity loop

### 3- Do..while:

دى بتعمل اللى كانت ال **while** بتعمله بالظبط , بس ال **while** من اول مره الشرط لو متحققش مبيتنفذش ال Code اللى فيها اصلا , لكن ال **do...while** على الاقل **بتنفذ** ال code **مره واحده بس حتى لو الشرط متحققش** وده لانها بت **do** الاول بعد كده بتعمل ال **check** بتاعها

هنا على الاقل طبع لا **hello** مره **وحده** بس مع ان ال **while** هينزل مش هيلقيها بتحقق الشرط ده .

وده مثال انا ممكن اعمل **select** فيها مثلا من 1 لحد 100 فهكتب ال **<select>** عادى واللى عايزه اكرره هو ال **option** فهدخله جوه **for loop** بتمشى من 1 لحد 100

```
1 document.write("<select>");
2
3 for(var x=1;x<=100;x++)
4 {
5     document.write("
6         <option>"+x+"
7     </option>")
8 }
9 document.write("</select>")
```

***END OF JS basics***

***Plugins***

ال plugin بيبقى عباره عن code كبير جدا لل js من خلاله بتعمل شكل معين لل element بتاعك مثلا بس اللي بيبقى صعب انك توصله هو اسم ال plugin ذات نفسه لكن استخدامه بيبقى سهل جدا لان اللي عملوا ال plugin ده اول حاجه بيرفعوه على github بعد كده لو الاستخدام زاد عليه بيعملوا site ليهم و بيحطوا ال plugin ده عليه فمثلا اخدنا كذا plugin منهم ال **nicescroll , owl carousal , skitter slider , wow**

كل plugin من دول فال documentation بتاعته بيقولك بتستخدمه ازاي و بتنزل ايه و بتعتمد على ايه .

وكل واحد فيهم بردو ليه ال fire function بتاعتها لازم تحطها فال js بتاعك

وده اللينكات بتعتهم

## 1-nicescroll

<https://github.com/inuyaksa/jquery.nicescroll>

<https://nicescroll.areaaperta.com/>

## 2-owl carousal

<https://owlcarousel2.github.io/OwlCarousel2/>

<https://github.com/OwlCarousel2/OwlCarousel2>

## 3-skitter slider

<https://github.com/thiagosf/skitter>

<https://skitter-slider.net/>

## 4-wow.js

<http://mynameismatthieu.com/WOW/>

<https://github.com/matthieua/WOW>