SEPTEMBER 22 – 28, 2019

المبادرة السعودية للمطورين

مسار الـJava Script

مشرف المسار:

محمد الأسمري

ملاحظات قبل بدء الدروس:

- على المتدربين نشر كل يوم الجزئية التي تم كتابتها من النص البرمجي في الـ Github تحت Topic بعنوان saudidevorg

كما تم توضيحه في دروس الـ Github سابقاً

- على المتدربين نشر كل يوم مقدار التقدم وصورة لما تم تعلمه وتطبيقه على Twitter تحت الهاشتاقات:

#المبادرة_السعودية_للمطورين

#100_يوم_برمجة

#100DaysOfCode

تمنياتنا لك بالتوفيق

المبادرة السعودية للمطورين

اليوم الرابع والثلاثون

this

في الدرس السابق تعلمنا ان نستخدم this مع الدوال داخل الكائنات كالمثال أدناه

```
var person = {
  firstName: "Mohammed",
  lastName: "Alali",
  age: 30,
fullName : function() {
    return this.firstName + " " + this.lastName;
}
};

document.write(person.fullName());
```

واليوم سوف نتعلم أكثر عن this

القيمة الحقيقة لها هي تحديد كيف يتم استدعاء الدالة حتى نصل للنتيجة التي نريدها كالمثال السابق والمثال أدناه

```
var test = {
  prop: 42,
  func: function() {
    return this.prop;
  },
};

document.write(test.func());
```

ما الذي يحدث عند استدعاءها لوحدها كالتالي:

```
var x = this;

document.write(x);
```

سيكون استدعاء للـ global object والذي سيكون هنا [object Window]

هل الـ window المقصود فيها نظام التشغيل؟؟

لا، هي كائن المتصفح وكذلك ليست كائن للجافاسكريبت

ما هو كائن المتصفح، باختصار هي التي تظهر object window في المتصفح وأشهرها alert() وprompt() والتي نستخدمها مع برمجة الجافا سكريبت

دعونا نجرب هذا المثال أيضاً:

```
document.write(this === window); //  true وذلك لأننا تأكدنا أنها نفسها سترجع

a = 37;

document.write("<br>"+window.a); // 37


this.b = "it is mine";

document.write("<br>"+window.b);

document.write("<br>"+b);
```

كذلك حتى لو استخدمناها لإرجاع فنكشن سترجع global object window

```
function myFunction() {
  return this;
}
document.write(myFunction())
```

اليوم الخامس و الثلاثون

Inheritance

الوراثة هي طريقة لتوريث الخصائص والوظائف من كائن إلى آخر فعلى سبيل المثال لو أردنا أن ننشئ أكثر من كائن يشتركون في نفس الخصائص والوظائف فبدل من تكرار الإنشاء نقوم بإنشاء فنكشن واحد ليورث كل شي للبقية

تخيلوا أننا نريد أن ننشئ مستخدمين لنظام معين، الشيء المشترك بينهم الاسم والرقم الخاص وغيرها

```
function user(id, FirstName, LastName){
   this.id = id;
   this.FirstName = FirstName;
   this.LastName = LastName;
}

var Mohammed = new user(1001, "Mohammed", "Alali");
var Fahad = new user(1002, "Fahad", "Saad");

document.write(Mohammed.id+" "+Mohammed.FirstName+" "+Mohammed.LastName);
```

اليوم السادس والثلاثون

Deleting Properties

وتعمل على حذف الخصائص فقط على سبيل المثال

```
var person = {
  firstName: "Mohammed",
  lastName: "Alali",
  age: 30,
fullName : function() {
    return this.firstName + " " + this.lastName;
}
};

document.write(person.firstName+"<br>");
delete person.firstName;
document.write(person.firstName+"<br>");
```

اليوم السابع والثلاثون

Testing Properties

هي طريقة لاختبار الخصائص هل هي موجودة أم لا فعلى سبيل المثال

```
var person = {
  firstName: "Mohammed",
  lastName: "Alali",
  age: 30,
fullName : function() {
    return this.firstName + " " + this.lastName;
}
};
document.write("firstName" in person);
```

سنلاحظ أنها سترجع قيمة true، جرب أن تغير firstName الى اسم غير موجود، سترجع false

أيضاً نستطيع أن نستخدم hasOwnProperty() للتأكد فعلى سبيل المثال

```
var person = {
  firstName: "Mohammed",
  lastName: "Alali",
  age: 30,
fullName : function() {
    return this.firstName + " " + this.lastName;
}
};
document.write(person.hasOwnProperty("firstName"));
```

جرب وابحث عن propertyIsEnumerable

اليوم الثامن و الثلاثون

Property Getters and Setters

نستطيع أن نغير قيمة المتغيرات داخل object معين من خلال getter & setter وهي التي تسمح لنا بتعريف الـ Property Accessors

ملحوظة:

Property Accessors هي التي تعطينا صلاحية الوصول لأي متغير داخل object معين

مثال:

```
// Create an object:
var person = {
 firstName: "Mohammed",
 lastName: "Alali",
 language: "EN",
 get langGet(){              // get إنشاء
 return this.language; },          // يكون بين الدالتين فاصلة وليس فاصلة منقوطة
 set langSet(value) {      // set إنشاء
 this.language = value; }
};
// Set a property using setter
person. langSet = "AR";
// Display data from the object using getter
document.getElementById("demo").innerHTML = person. langGet;
```

اليوم التاسع والثلاثون
واليوم الأربعون

نـهاية الاسبوع السادس

أنشئ برنامجاً

نريد أن ننشئ برنامجا للمعلمين فنسند كل مادة للمعلم الخاص بها

- أنشئ كائن (Object) يحتوي على الخصائص التالية:
Subject – CoordinatorName – NOofHours – NoofStudents
ثم عبئ القيم كل مرة بخصائص مختلفة باستخدام getter & setter

- أنشئ دالة (Function) يحتوي على الخصائص التالية:
Subject – CoordinatorName – NOofHours – NoofStudents
ثم عبئ القيم كل مرة بخصائص مختلفة باستخدام الـ Inheritance

هل لاحظت الفرق بين الطريقتين ؟ أيهما أسهل وأيهما تفضل ^^