OCTOBER 20 – 26, 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

#100يوم_برمجة _

#100DaysOfCode

تمنياتنا لك بالتوفيق

المبادرة السعودية للمطورين

اليوم الثاني والستون

(function)

الفنكشن (function) هي كتلة برمجية يتم تنفيذها مرة أو اكثر من مرة في البرنامج وتستطيع استدعاءها متى ماأردت استخدامها.

هنا في هذا المثال كتبنا فنكشن بداخلها for loop ثم قمنا باستدعائها

```
var o;
function printprops(o) {
for (var p=0; p<o;p++)
{
  document.write("p = " + p +"<br>");
}
}
printprops(5);
```

ولو أردنا الدالة ترجع قيمة فتكون بهذه الطريقة return

```
function distance(x1, y1, x2, y2) {
var dx = x2 - x1;
var dy = y2 - y1;
return Math.sqrt(dx*dx + dy*dy);
}

document.write(distance(2,4,6,8));
```

وكما ذكرنا سابقاً , يجب التركيز في كتابة اسم الفنكشن حيث هناك كلمات محجوزة يجب تجنبها.

والجافاسكريبت تسمح بتداخل الوظائف بمعنى فنكشن داخل فنكشن لاحظ المثال أدناه حيث أردنا حساب وتر المثلث كتبنا فنكشن التربيع داخل فنكشن الوتر ومن ثم رجعنا القيم فيها

```
function hypotenuse(a, b) {
function square(x) { return x*x; }
return Math.sqrt(square(a) + square(b));
}
document.write(hypotenuse(3,4));
```

اليوم الثالث و الستون

الفنكشن (Functions)

استكمال للفنكشن , نستطيع أن نستخدم الفنكشن كقيمة على سبيل المثال

```
function myFunction(a, b) {
  return a * b;
}

var x = myFunction(4, 3);
```

وكذلك نستطيع جعل الفنكشن يستدعي نفسه

جرب هذين المثالين

```
(function s() {
  document.write("Hello! I called myself");
})();
```

وهناك أيضا مايسمى built-in JavaScript function constructor

انظر لفرق الطريقتين

```
var myFunction = function (a, b) {return a * b}
document.write(myFunction(4, 3));

var myFunction = new Function("a", "b", "return a * b");
document.write(myFunction(4, 3));
```

اليوم الرابع والستون

call

هي طريقة الفنكشن من الكائن والتعامل معه كـparameter

```
var person = {
  fullName: function() {
    return this.firstName + " " + this.lastName;
  }
}
var person1 = {
  firstName:"John",
  lastName: "Doe"
}
var person2 = {
  firstName:"Mary",
  lastName: "Doe"
}
document.write(person.fullName.call(person1));
```

وكذلك نستطيع جعل معها Arguments

```
var person = {
  fullName: function(city, country) {
    return this.firstName + " " + this.lastName + "," + city + "," + country;
  }
}
var person1 = {
  firstName:"John",
  lastName: "Doe"
}
document.write(person.fullName.call(person1, "Oslo", "Norway"));
```

اليوم الخامس و الستون

(apply)

هي نفس call ولكن تستطيع استخدماها مع كائن آخر

```
var person = {
 fullName: function() {
   return this.firstName + " " + this.lastName;
 }
}
var person1 = {
 firstName: "Mary",
 lastName: "Doe"
}
person.fullName.apply(person1);
```

والفرق بينهما أن call تأخذ Arguments مفصلة أما apply فيأخذها كمصفوفة على سبيل المثال

```
var person = {
 fullName: function(city, country) {
   return this.firstName + " " + this.lastName + "," + city + "," + country;
 }
}
var person1 = {
 firstName:"John",
 lastName: "Doe"
}
person.fullName.apply(person1, ["Oslo", "Norway"]);
```

اليوم السادس والستون

(JavaScript Closures)

من المهم جداً معرفة استخدام المتغيرات ومواقعها مع الفنكشن ويجب أن نركز ونفرق بين local or global

```
function myFunction() {
 var a = 4; // local
 return a * a;
}

var a = 4; //global
function myFunction() {
 return a * a;
}
```

انظر الى نتيجة counter هنا

```
var counter = 0;

function add() {
 counter += 1;
}

add();
add();
add();
document.write(counter);
```

ثم انظر اليها هنا

```
var counter = 0;

function add() {
 var counter = 0;
 counter += 1;
}
```

```
add();
add();
add();
document.write(counter);
```

ثم لاحظ هنا

```
function add() {
 var counter = 0;
 counter += 1;
 return counter;
}

add();
add();
add();
document.write(add());
```

وايضاً لاحظ هنا

```
var add = (function () {
 var counter = 0;
 return function () {counter += 1; return counter}
})();

add();
add();
add();
document.write(add());
```

اليوم السابع والستون
واليوم الثامن والستون

نـهاية الاسبوع العاشر

ابحث عن الفنكشن ()memorize واستخدمها في كود لارجاع قيمة من الذاكرة