



# تطبيقات الويب Web Applications

2023-2024

Dr.-Eng. Samer Sulaiman

<https://manara.edu.sy/>

## مفردات المنهاج

- أساسيات برمجة تطبيقات الويب
- تطبيقات الويب الستاتيكية
  - أساسيات لغة الـ HTML
  - أساسيات الـ CSS
  - أساسيات "JS" الـ JavaScript
- تطبيقات الويب الديناميكية
  - ربط صفحات الويب بقواعد البيانات
  - لمحة عن لغة الـ PHP

<https://manara.edu.sy/>

# أساسيات تصميم الويب



## • أساسيات الـ JavaScript:

### • استخدام الدالات (التوابع) (Functions):

Name	Description
getDate()	The day of the month as an integer from 1 to 31
getDay()	The day of the week as an integer where 0 is Sunday and 1 is Monday
getHours()	The hour as an integer between 0 and 23
getMinutes()	The minutes as an integer between 0 and 59
getMonth()	The month as an integer between 0 and 11 where 0 is January and 11 is December
getSeconds()	The seconds as an integer between 0 and 59
getTime()	The current time in milliseconds where 0 is January 1, 1970, 00:00:00
getFullYear()	The year, but this format differs from browser to browser

<https://manara.edu.sy/>

# أساسيات تصميم الويب



## • أساسيات الـ JavaScript:

### • استخدام الدالات (التوابع) (Functions):

#### • بعض الدالات المعيارية:

#### • Writing the Date to Your Web Page

• هنا سيتم انشاء غرض من النوع Date على الشكل التالي:

```
• var now = new Date();
```

• يتم استخدام هذا الغرض للوصول الى التوابع الخاصة

```
• var the_year = now.getFullYear();
```

• مثال:

```
• <script type = "text/javascript">
var now = new Date();
var yyyy = now.getFullYear();
var mm = now.getMonth() + 1;
if (10 > mm) mm = '0' + mm;
var dd = now.getDate();
if (10 > dd) dd = '0' + dd;
document.write(dd + '-' + mm + '-' + yyyy);
</script>
```

<https://manara.edu.sy/>

# أساسيات تصميم الويب



## • أساسيات الـ JavaScript:

### • التعامل مع الكائنات (Objects):

- يمكننا تعريف والتصريح عن الـ Object بأكثر من طريقة:
  - عن طريق استخدام الـ {curly braces}.
  - أو عن طريق استخدام الـ new (والتي يمكن استخدامها عند التصريح عن كائن عن طريق دالة)
- يوجد أكثر من طريقة لاسترجاع القيم من الـ Object
  - عن طريق اسم الـ object ومن ثم نقطة ومن ثم الـ key
  - مثال: `varObject.keyName`
  - أو باستخدام الأقواس [ ] بحيث يوضع ضمنهم اسم المكون (أو الخاصية)

```
// Object Example
varObject = {firstName:"Anees", age:24, carType: "Still Without Car -_-"};
varObject['carType']; (or) varObject.carType;
```

<https://manara.edu.sy/>

# أساسيات تصميم الويب



## • أساسيات الـ JavaScript:

### • التعامل مع الكائنات (Objects):

- يمكن تشبيه الكائن بالـ class في لغات البرمجة، وعليه يمكن تضمين الكائن متحولات (تسمى خصائص) بالإضافة إلى التتابع (الدوال)

```
// Object Example
varObject = {firstName:"Anees", carType: "Still Without Car -_-",
  age: function() {return 23 + 1;} }; ← تعريف دالة
var CurrentAge = varObject.age(); ← استدعاء الدالة
```

- مثال:
 

```
var myObject = {};
myObject.myFunction = function() {};
myObject.myArray = [];
myObject.myString = 'string';
myObject.myNumber = 33;
myObject.myDate = new Date();
myObject.myNull = null;
myObject.myObject = {};
myObject.myMath_PI = Math.PI;
```

<https://manara.edu.sy/>

# أساسيات تصميم الويب



- أساسيات الـ JavaScript:
  - التعامل مع الكائنات (Objects):
    - إنشاء كائنات مركبة ومعقدة:
      - يمكن تضمين كائن ضمن كائن آخر
      - مثال:
- ```
var object1 = {
  object1_1: {
    object1_1_1: {foo: 'bar'},
    object1_1_2: {}
  },
  object1_2: {
    object1_2_1: {},
    object1_2_2: {}
  }
};
console.log(object1.object1_1.object1_1_1.foo);
```

<https://manara.edu.sy/>

# أساسيات تصميم الويب



- أساسيات الـ JavaScript:
  - التعامل مع الكائنات (Objects):
    - إنشاء كائنات مركبة ومعقدة:
      - يمكن تضمين كائن ضمن كائن آخر
      - مثال:
- ```
<script>
var myArray = [[[]]];
var myFunction = function() {
var myFunction = function() {
var myFunction = function() {
};};
var foo = [{
foo: [{
bar: {
say: function() {
return 'hi';
}
}}];
console.log(foo[0].foo[0].bar.say());
</script>
```

<https://manara.edu.sy/>

# أساسيات تصميم الويب



- أساسيات الـ JavaScript
- التعامل مع الكائنات (Objects):
- مثال:

```
<script>
var cody = new Object();
cody['living'] = true;
cody['age'] = 33;
cody['gender'] = 'male';
cody['getGender'] = function() {return cody.gender;};
console.log(cody['living'], cody['age'], cody['gender'], cody['getGender']());
cody['living'] = false;
cody['age'] = 99;
cody['gender'] = 'female';
cody['getGender'] = function() {return 'Gender = ' + cody.gender;};
console.log(cody);
</script>
```

<https://manara.edu.sy/>

# أساسيات تصميم الويب



- أساسيات الـ JavaScript
- استخدام المصفوفات Array:

المصفوفات هي نوع من أنواع البيانات صممت خصيصا لحفظ أكثر من قيمة داخل متغير واحد ويمكن تعريف المصفوفة بطريقتين:

```
var x = [1, 5, 54, 14];
console.log("x[1] = " + x[1]); // result is 5
console.log("x[2] = " + x[2]); // result is 54
// --
```

```
var x = [1, 5, 54, 14]; // إعطاء القيم مباشرة
// or
var z = [
1,
2,
3
];
```

أكثر من سطر.. لكن يتم تجاهل ذلك.. ويتم تجاهل المسافات..

```
var a = [];
a[0] = "anees ";
a[1] = "HIKMAT ";
a[2] = "Abu-hamid ";
console.log(a[0] + a[1] + a[0] + a[2]);
```

إعطاء القيم بعد تعريف المصفوفة

Var X= [ 1, 3, 5, 6];

Var X= new array(1, 3, 5, 6);

<https://manara.edu.sy/>

# أساسيات تصميم الويب



## • أساسيات الـ JavaScript:

### • استخدام المصفوفات Array:

- يمكن استخدام الدالة length لتحديد حجم المصفوفة المستخدمة عن طريق ربطها بالـ (.)
- يمكننا إضافة عنصر جديد للمصفوفة عن طريق:
  - تحديد عدد العناصر ومن ثم وضع عدد العناصر الناتج كـ key أو index للقيمة الجديدة
  - أن تحدد الـ key بشكل يدوي مباشرة.
  - باستخدام الـ push
  - لإضافة عنصر جديد لآخر المصفوفة وناتج الإرجاع سيكون طول المصفوفة الجديدة
  - باستخدام الـ unshift
  - تقوم بإضافة عنصر للمصفوفة ولكن بالموقع الأول لها ومن ثم تقوم بإزاحة كافة العناصر خطوة إلى الأمام وناتج الإرجاع سيكون طول المصفوفة الجديدة
  - باستخدام الـ splice
    - تقوم بإضافة والحذف بذات الوقت
    - يمكنها تحديد الموقع المراد إضافة العنصر إليه
    - تمتلك ثلاثة بارامترات: (arrayName.splice(l, R, V):
      - l: وهو الموقع المراد بدء الإضافة من عنده
      - R: عدد العناصر المراد حذفها والتي تأتي بعد إضافة العناصر
      - V: القيمة/القيم المراد إضافتها
- إذا كانت قيمته 2 فسيقوم بحذف عنصرين من المصفوفة وهما العنصرين اللذان يأتيان بعد ما سيتم إضافته على المصفوفة
- تجرب الكود وإظهار النتيجة

```
var x = [1, 5, 54, 14];
x.splice(1, 0, "anees");
console.log(x.toString());
x.splice(2, 2, "taher", "saed");
console.log(x.toString());
x.splice(0, 1, "ahmad", "muath", "Rashad");
console.log(x.toString());
```

<https://manara.edu.sy/>

# أساسيات تصميم الويب



## • أساسيات الـ JavaScript:

### • استخدام المصفوفات Array:

- طرق الحذف:
  - pop(): يتم حذف آخر عنصر
  - shift(): يتم حذف أول عنصر
  - splice(): نضع الـ Parameters التالية أو R فقط
- العمليات على المصفوفات:
  - reverse: تقوم على طباعة عناصر المصفوفة بشكل عكس .
  - لا تملك أي parameters
  - sort: تستخدم لترتيب المصفوفة، والترتيب إما أن يكون هجائي أو رقمي، ويمكن أن يكون تنازلي أو تصاعدي

```
var x = [1, 5, 54, 14, 12, 125, 5124, 11234];
console.log("Array: " + x.toString());
x.splice(6, 2);
console.log("x.splice(6, 2): " + x.toString());
x.pop();
console.log("x.pop(): " + x.toString());
x.shift();
console.log("x.shift(): " + x.toString());
```

```
var x = [1, 5, 54, 14];
var y = ["anees", "hikmat", "anees", "abu-hmaid"];
console.log(x.reverse().toString());
console.log(y.reverse().toString());
```

```
var y = ["anees", "hikmat", "anees", "abu-hmaid"];
console.log(y.sort().toString());
console.log(y.sort().reverse().toString());
```

تصاعدي  
تنازلي

<https://manara.edu.sy/>

# أساسيات تصميم الويب



## • أساسيات الـ JavaScript:

### • استخدام المصفوفات Array:

#### • العمليات على المصفوفات:

• sort: تستخدم لترتيب المصفوفة، والترتيب إما أن يكون هجائي أوقي، ويمكن أن يكون تنازلي أو تصاعدي

• للتعامل مع الأرقام سيتم استخدام function مع الـ sort

```
var x = [1, 5, 54, 14];
console.log(x.sort(function(a, b){return a-b}).toString());
console.log(x.sort(function(a, b){return a+b}).toString());
```

تصاعدي

تنازلي

```
var z = [25, 50, 13, 14];
z = z.sort(function(a, b){return a-b}); // Sort Array ASC
console.log(z.toString());
console.log("Lowest Number is: " + z[0]);
```

• مثال: المطلوب طباعة أكبر رقم موجود داخل

مصفوفة معينة أو طباعة أقل رقم موجود

داخل مصفوفة معينة

```
z = z.sort(function(a, b){return a+b}); // Sort Array DESC
console.log(z.toString());
console.log("Largest Number is: " + z[0]);
```

<https://manara.edu.sy/>

# أساسيات تصميم الويب



## • أساسيات الـ JavaScript:

### • استخدام المصفوفات Array:

#### • العمليات على المصفوفات:

```
var newArray = x.concat(y);
console.log(newArray.toString());
var newArray2 = x.concat(z, newArray);
console.log(newArray2.join(" / "));
```

• concat(): تستخدم لدمج أكثر من مصفوفة معا والنتاج يكون مصفوفة جديدة

• slice(): تقوم على اقتطاع جزء من المصفوفة وإرجاعه على شكل مصفوفة جديدة

• الصيغة العامة: arrayName.slice(start, end);

• start: نقطة البداية وهي قيمة عددية صحيحة اجبارية.

• end: نقطة النهاية وهي قيمة عددية صحيحة اختيارية

• في حال لم توضع هذه القيمة يكون مقدار القطع من نقطة الـ start الى النهاية -1

```
var z = [25, 50, 13, 14, "anees", "hikmat", 1991];
var newArray = z.slice(3, 6);
console.log(newArray.toString());
var newArray2 = z.slice(4);
console.log(newArray2.toString());
```

<https://manara.edu.sy/>

# أساسيات تصميم الويب



## • أساسيات الـ JavaScript:

### • الأحداث Events:

- Click: ينفذ عند النقر
- onchange: ينفذ بمجرد الخروج من العنصر html الخاص به، لكن بشرط أن يكون قد حصل تعبير معين على هذا العنصر
- onmouseover: ينفذ عند الدخول الى العنصر
- onmouseout: ينفذ بمجرد الخروج من نطاق العنصر
- onmousedown: ينفذ بمجرد النقر على زر الفأرة
- onmouseup: ينفذ بمجرد تحرير زر الفأرة
- onload: ينفذ بمجرد أن العنصر تم تحميله.
- onfocus: ينفذ بمجرد تحديد العنصر ويعمل مع `<a>`, `<input>`, `select`

<https://manara.edu.sy/>

# أساسيات تصميم الويب



## • أساسيات الـ JavaScript:

### • الأحداث Events:

- onblur: ينفذ بمجرد خروج التحديد عن العنصر، وهو بنفس مواصفات الـ onfocus ولكن بشكل معاكس
- ondblclick: ينفذ عند النقر نقرا مزدوجا على زر الفأرة
- onmousemove: ينفذ طالما مؤشر الفأرة يتحرك فوق العنصر.
- onkeydown: ينفذ بمجرد الضغط على أي زر من أزرار لوحة المفاتيح.
- onkeypress: بنفس فكرة الـ onkeydown لكن لا يعتبر الـ shift والـ ctrl والـ alt... الخ من الأزرار وينفذ فقط للأرقام والحروف والرموز الخاصة
- onkeyup: ينفذ بمجرد تحرير أي من أزرار لوحة المفاتيح
- الصيغة العامة:

```
object.EventName=function() {myScript};
```

<https://manara.edu.sy/>

# أساسيات تصميم الويب



## • أساسيات الـ JavaScript:

### • الأحداث Events:

#### • مثال:

```
<input type="button" id="btnMouseMove" title="Click Event" value="Click Event" />
<script>
  // Declare Event
  document.getElementById("btnMouseMove").onmouseover = function() {myFunctionOver()};
  document.getElementById("btnMouseMove").onmouseout = function() {myFunctionOut()};

  function myFunctionOver() {
    document.getElementById("btnMouseMove").style.backgroundColor = "red";
    document.getElementById("btnMouseMove").style.color = "blue";
  }
  function myFunctionOut() {
    document.getElementById("btnMouseMove").style.backgroundColor = "green";
    document.getElementById("btnMouseMove").style.color = "red";
  }
</script>
```

<https://manara.edu.sy/>