

### تفاعيل صفحات الويب

في هذا الفصل نلقى الضوء على لغة JavaScript والتي تعتبر من أبرز لغات برمجة صفحات الويب لإضفاء سمة التفاعلية عليها. سوف نبدأ بتعريف طبيعة هذه اللغة، وموضع كتابة تعليماتها في صفحات HTML. وبعد ذلك سوف نتقل إلى كيفية إنشاء المتغيرات وبناء الجمل الشرطية والدوال، وغيرها الكثير. يهتم الفصل بعرض الطرق المتبعة للتعامل مع الأخطاء، وأبرز الرموز الخاصة التي تستخدمها لغة JavaScript أثناء كتابة التعليقات البرمجية.

#### المحتويات:

- ما هي لغة JavaScript؟
- موضع تعليمات JavaScript داخل صفحات HTML
- المتغيرات Variables
- الجمل الشرطية Conditional Statements
- المعاملات JavaScript Operators
- الصناديق المنبثقة JavaScript Popup Boxes
- الدوال JavaScript Functions
- الحلقات التكرارية JavaScript Loops
- الأحداث JavaScript Events
- اصطياذ الأخطاء Catching Errors
- الرموز الخاصة JavaScript Special Characters



## الأهداف التعليمية:

١. تحديد مجالات استخدام لغة JavaScript.
٢. التمييز بين الأماكن المختلفة لكتابة تعليمات JavaScript البرمجية.
٣. القدرة على إنشاء المتغيرات وبناء الجمل الشرطية والحلقات التكرارية.
٤. استخدام الصناديق المنبثقة popup boxes في صفحات الويب التفاعلية.
٥. تحديد متطلبات استخدام الدوال والأحداث في التعليمات البرمجية.
٦. التأكيد على الإرشادات الخاصة بكتابة تعليمات JavaScript النصية.

### ١- مقدمة إلى لغة JavaScript

تعتبر لغة JavaScript من أشهر لغات البرمجة في بيئة الويب. فهي تستخدم في ملايين الصفحات على الشبكة العنكبوتية؛ بهدف تحسين التصميم، والتحقق من البيانات المدخلة في النماذج form validation، والتعرف على برنامج المتصفح الذي يستخدمه المستخدم، وإنشاء برامج الكوكيز cookies، وغيرها الكثير.

تعمل لغة JavaScript مع جميع برامج التصفح المعروفة؛ مثل Internet Explorer، وMozilla، وFirefox، وNetscape Navigator، وOpera.

وقبل أن نبدأ معا في تعلمها، أحب أن أؤكد لك عزيزي القارئ أنها لغة سهلة،

وسوف تستمتع بها لدرجة كبيرة.

ما الذى يجب أن تعرفه قبل البدء فى تعلم JavaScript؟

يجب أن يكون لديك عزيزى القارئ معرفة أساسية بلغة HTML.

### ما هى لغة JavaScript؟

- تم تصميم هذه اللغة لإضافة التفاعلية interactivity إلى صفحات HTML.
- تعتبر JavaScript لغة تعليمات نصية scripting language.
- تتكون هذه اللغة من أسطر (أكواد) قابلة للتنفيذ على الكمبيوتر.
- عادة ما تدرج أكواد JavaScript مباشرة داخل صفحات HTML.
- هى لغة مفسرة interpreted language (بمعنى أن أكوادها تنفذ دون تجميع compiling مسبق).
- يمكن لأى شخص استخدام لغة JavaScript دون شراء ترخيص لذلك.

### هل Java وJavaScript شيء واحد؟

لا!

تعتبر كل من لغة Java ولغة JavaScript لغتان مختلفتان تماما، من حيث المفهوم والتصميم!

تم تطوير Java بواسطة شركة Sun Microsystems، وهى تعد لغة برمجة قوية وأكثر تعقيدا، مثلها مثل لغات C و C++.

### ما الذى يمكن للغة JavaScript أن تفعله؟

□ تعطى لغة JavaScript مصممي HTML أداة برمجة – إن مؤلفي HTML ليسوا مبرمجين، لكن JavaScript لغة نصوص برمجية ذات قواعد سهلة جدا. وتقريبا يستطيع أى شخص أن يضع جزءا صغيرا من الكود داخل صفحات HTML.

□ تستطيع JavaScript أن تضع نصا ديناميكيا داخل صفحة HTML – إن جملة

JavaScript التالية: `document.write("<h1>" + name + "</h1>")` تستطيع

أن تضع نصا متغيرا داخل صفحة HTML.

▣ تستطيع JavaScript أن تستجيب للأحداث – يمكن ضبط JavaScript لكي

تنفذ عند حدوث شيئا ما، مثلا بعد انتهاء تحميل الصفحة على شاشة

المتصفح، أو عندما يضغط المستخدم على عنصر معين في صفحة HTML.

▣ تستطيع JavaScript قراءة وكتابة عناصر HTML – يمكن للغة JavaScript

قراءة وتغيير محتوى صفحة HTML.

▣ يمكن استخدام JavaScript للتحقق من صحة البيانات – تستطيع

JavaScript أن تتحقق من بيانات النموذج قبل إرسالها للخادم server، الأمر

الذي يوفر إجراء تجهيز غير مطلوب داخل الخادم.

▣ يمكن استخدام JavaScript للتعرف على برنامج المتصفح لدى زائر الصفحة

– حيث يمكن تحميل صفحة مصممة للعرض بشكل جيد في متصفح معين.

▣ يمكن استخدام لغة JavaScript لتخزين واسترجاع معلومات داخل جهاز

الكمبيوتر الخاص بزائر الموقع.

### كيفية كتابة أكواد JavaScript

يستخدم حقل `<script>` في لغة HTML لكتابة أكواد JavaScript داخل

صفحات HTML.

أمثلة:

كتابة نص

كيفية كتابة نص في صفحة HTML.

```
<html>
```

```
<body>
```

```
<script type="text/javascript">
```

```
document.write("Hello World!")
```

```
</script>
```

```
</body>
</html>
```

كتابة نص مع التنسيق

كيفية تنسيق النص في صفحة HTML.

```
<html>
<body>
```

```
<script type="text/javascript">
document.write("<h1>Hello World!</h1>")
</script>
</body>
</html>
```

**كيفية وضع أكواد JavaScript داخل صفحة HTML**

```
<html>
<body>
<script type="text/javascript">
document.write("Hello World!")
</script>
</body>
</html>
```

الكود أعلاه سوف ينتج المخرجات التالية في صفحة HTML:

**Hello World!**

**شرح المثال**

لإدراج كود JavaScript داخل صفحة HTML، نستخدم حقل `<script>` (أيضا استخدم خاصية النوع `type` لتعريف لغة النصوص البرمجية).

ومن ثم، نجربنا الكود التالي `<script type="text/javascript">` و `</script>` أين تبدأ JavaScript وأين تنتهي:

```
<html>
```

```
<body>
<script type="text/javascript">
...
</script>
</body>
</html>
```

وتعتبر جملة `document.write` أحد أوامر JavaScript المعيارية، وتستخدم لكتابة مخرجات على الصفحة.

وبكتابة أمر `document.write` بين حقل `<script type="text/javascript">` وحقل `</script>`، سوف يدركه برنامج المتصفح على أنه أحد أوامر JavaScript، وينفذ سطر الكود. في هذه الحالة سوف يكتب برنامج المتصفح Hello World! على الصفحة.

```
<html>
<body>
<script type="text/javascript">
document.write("Hello World!")
</script>
</body>
</html>
```

ملاحظة: إذا لم نكتب حقل `<script>`، سوف يتعامل برنامج المتصفح مع أمر `document.write("Hello World!")` باعتباره نصا صريحا، وبالتالي سوف يكتب السطر كله على الصفحة.

### إنهاء الجمل البرمجية بشبه الشارحة

في لغات البرمجة الشائعة، مثل C++ و Java، يجب أن ينتهي كل سطر من أسطر الكود بشبه الشارحة `semicolon`.

ومعظم المبرمجين يفعلون ذلك مع JavaScript. لكن بشكل عام يعتبر استخدام شبه الشارحة هنا اختياري. ومع ذلك، يجب استخدامها في حالة ما إذا أردت وضع أكثر من جملة برمجية واحدة في سطر مفرد.

## كيفية التعامل مع برامج التصفح القديمة

إن برامج التصفح التي لا تدعم JavaScript سوف تعرض النصوص البرمجية كمحتوى للصفحة. ولنعها من القيام بذلك، يمكننا استخدام حقل HTML الخاص بالتعليقات comment tag:

```
<script type="text/javascript">
<!--
document.write("Hello World!")
//-->
</script>
```

تعتبر الشرطتان المائلتان للأمام في نهاية سطر الأمر (//) رمز التعليقات في JavaScript. وهذا من شأنه منع مجمع JavaScript من تجميع السطر.

### ٢- موضع كود JavaScript

يتم تنفيذ أكواد JavaScript الموجودة في جسم الصفحة body section أثناء تحميل الصفحة.

أما الأكواد الموجودة في رأس الصفحة head section فيتم تنفيذها فقط عند استدعائها.

### أمثلة

#### رأس الصفحة Head

يتم وضع النصوص البرمجية التي تحتوي على دوال functions داخل رأس الصفحة، حيث يضمن لنا ذلك أن الكود سوف يتم تحميله قبل استدعاء الدالة.

```
<html>
<head>
<script type="text/javascript">
function message()
{
alert("This alert box was called with the onload event")
}
```

```
}  
</script>  
</head>  
  
<body onload="message()">  
  
</body>  
</html>
```

### جسم الصفحة Body

يتم تنفيذ الكود الموجود في جسم الصفحة مباشرة.

```
<html>  
<head>  
</head>  
  
<body>  
  
<script type="text/javascript">  
document.write("This message is written when the page loads")  
</script>  
  
</body>  
</html>
```

### النصوص البرمجية الخارجية External script

المثال التالي يبين لنا كيفية الوصول إلى أكواد محفوظة في ملف خارجي:

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

<p>

The actual script is in an external script file called "xxx.js".

</p>

</body>

</html>

### أين نضع أكواد JavaScript

يتم تنفيذ أكواد JavaScript الموجودة في صفحة HTML مباشرة أثناء تحميل الصفحة على شاشة برنامج المتصفح. لكن ذلك ليس دائما ما نريده. فأحيانا نريد تنفيذ النصوص البرمجية عند تحميل الصفحة، وأحيانا أخرى نريد تنفيذها عندما يتعامل الزائر مع حدث ما.

### النصوص البرمجية في رأس الصفحة Scripts in the Head Section

التعليقات البرمجية التي نريد تنفيذها فقط عند استدعائها، أو عند وقع حدث معين، يجب وضعها في رأس head الصفحة.

```
<html>
```

```
<head>
```

```
<script type="text/javascript">
```

```
....
```

```
</script>
```

```
</head>
```

### النصوص البرمجية في جسم الصفحة Scripts in the Body Section

التعليقات البرمجية التي نريد تنفيذها أثناء تحميل الصفحة، يجب وضعها في جسم body الصفحة.

```
<html>
```

```
<head>
```

```
</head>
```

```
<body>
```

```
<script type="text/javascript">
....
</script>
</body>
```

### النصوص البرمجية في رأس / وجسم الصفحة Scripts in the Head/Body Section

يمكنك كتابة أى عدد من التعليقات البرمجية في صفحة HTML، حيث يمكن كتابة هذه التعليقات في كل من رأس وجسم الصفحة في الوقت ذاته.

```
<html>
<head>
<script type="text/javascript">
....
</script>
</head>
<body>
<script type="text/javascript">
....
</script>
</body>
```

### استخدام أكواد JavaScript خارجية

قد تستدعى الضرورة أحيانا تنفيذ كود معين من أكواد JavaScript في صفحات متعددة؛ ومن ثم يفضل عدم كتابة هذا الكود في كل صفحة.

ولتبسيط ذلك، يمكنك كتابة هذا الكود في ملف مستقل. احفظ هذا الملف بالامتداد .js على النحو التالى (filename.js).

ملاحظة: لا تحتوى التعليقات البرمجية الخارجية على حقل <script>.

وعند الحاجة إلى استخدام ذلك الملف الخارجى، فقط حدده في خاصية src ضمن حقل <script>. أنظر المثال التالى:

```
<html>
<head>
```

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

ملاحظة: تذكر في وضع الإشارة إلى الملف بالضبط في المكان الذي كنت ستكتب فيه التعليقات البرمجية.

### ٣- المتغيرات JavaScript Variables

المتغير ما هو إلا "وعاء container" للمعلومات التي تريد تخزينها.

#### أمثلة

يوضح المثال التالي كيفية استخدام المتغير في تخزين البيانات.

```
<html>
<body>

<script type="text/javascript">
var name = "Hege"
document.write(name)
document.write("<h1>" + name + "</h1>")
</script>
```

**<p>This example declares a variable, assigns a value to it, and then displays the variable.</p>**

**<p>Then the variable is displayed one more time, only this time as a heading.</p>**

```
</body>
</html>
```

قد تتغير قيمة المتغير في النصوص البرمجية. ويتم الإشارة للمتغير عن طريق اسمه. وهناك قواعد لتسمية المتغيرات:

- ▣ أسماء المتغيرات حساسة لحالة الأحرف case sensitive
  - ▣ يجب أن تبدأ بحرف أو الشرطة السفلية underscore
- هام: تتسم لغة JavaScript بحساسيتها لحالة الأحرف. فالمتغير الذي اسمه strname يختلف عن المتغير الذي اسمه STRNAME.

### الإعلان عن المتغير Declare a Variable

يتم الإعلان عن المتغير بواسطة جملة var:

```
var strname = some value
```

كما يمكنك إنشاء متغير بدون جملة var:

```
strname = some value
```

### تعيين قيمة للمتغير Assign a Value to a Variable

يمكنك تعيين قيمة للمتغير على النحو التالي:

```
var strname = "Hege"
```

أو على النحو التالي:

```
strname = "Hege"
```

### فترة حياة المتغيرات Lifetime of Variables

عند الإعلان عن متغير داخل دالة، يمكن الوصول للمتغير فقط في إطار هذه الدالة. وعند الخروج منها، لا يعتبر المتغير موجودا. وهذه المتغيرات يطلق عليها "متغيرات محلية local variables". ويمكن الحصول على متغيرات محلية بنفس الاسم في دوال مختلفة، وذلك بسبب أن كل منها يدرك فقط بواسطة الدالة التي تم فيها الإعلان عن المتغير.

إذا أعلنت عن متغير خارج دالة، تستطيع جميع الدوال الموجودة في الصفحة

الوصول إليه واستخدامه. حيث تبدأ فترة حياة هذه المتغيرات من لحظة الإعلان عنها، وتنتهي عند إغلاق الصفحة.

### عبارات If...Else

تستخدم العبارات الشرطية في JavaScript لتنفيذ أفعال مختلفة استناداً إلى تحقق شروط مختلفة.

#### أمثلة

جملة If

يوضح المثال التالي كيفية كتابة جملة if:

```
<html>
<body>
<script type="text/javascript">
var d = new Date()
var time = d.getHours()
if (time < 10)
{
document.write("<b>Good morning</b>")
}
</script>

<p>
This example demonstrates the If statement.
</p>

<p>
If the time on your browser is less than 10,
you will get a "Good morning" greeting.
</p>
</body>
</html>
```

## جملة If...else

ويوضح المثال التالي كيفية كتابة جملة if...else:

```
<html>
<body>
<script type="text/javascript">
var d = new Date()
var time = d.getHours()
if (time < 10)
{
document.write("<b>Good morning</b>")
}
else
{
document.write("<b>Good day</b>")
}
</script>
<p>
This example demonstrates the If...Else statement.
</p>
<p>
If the time on your browser is less than 10,
you will get a "Good morning" greeting.
Otherwise you will get a "Good day" greeting.
</p>
</body>
</html>
```

## جملة if...else if...else

تستخدم لغة JavaScript جملة if...else if...else وذلك على النحو التالي:

```
<html>
<body>

<script type="text/javascript">
```

```

var d = new Date()
var time = d.getHours()
if (time<10)
{
document.write("<b>Good morning</b>")
}
else if (time>=10 && time<16)
{
document.write("<b>Good day</b>")
}
else
{
document.write("<b>Hello World!</b>")
}
</script>

```

<p>

This example demonstrates the if..else if...else statement.

</p>

</body>

</html>

### مثال تطبيقي: الرابطة العشوائية

يحتوي المثال التالي على رابطة تشعبية link، وعندما يضغط زائر الصفحة على هذه الرابطة، سوف يذهب إلى موقع الجمعية المصرية للمكتبات والمعلومات [www.elaegypt.com](http://www.elaegypt.com) أو إلى موقع الاتحاد العربي للمكتبات والمعلومات [www.afli.info](http://www.afli.info). فهناك فرصة بنسبة ٥٠٪ لكل موقع.

<html>

<body>

<script type="text/javascript">

var r=Math.random()

```

if (r>0.5)
{
document.write("<a href='http://www.elaegypt.com'> الجمعية
المصرية للمكتبات والمعلومات
</a>")
}
else
{
document.write("<a href='http://www.afli.info'> الاتحاد
العربي
للمكتبات والمعلومات
</a>")
}
</script>

</body>
</html>

```

#### ٤- الجمل الشرطية Conditional Statements

غالبا ما تحتاج أثناء كتابة الأكواد إلى تنفيذ أفعال مختلفة لقرارات مختلفة. وهنا يمكنك الاستعانة بالجمل الشرطية في الكود للقيام بذلك.

توجد الجمل الشرطية التالية في لغة JavaScript:

- جملة if – تستخدم هذه الجملة فقط إذا أردت تنفيذ التعليقات الموجودة في كود معين إذا تحقق شرط ما.
- جملة if...else – تستخدم هذه الجملة لتنفيذ تعليقات برمجية معينة إذا تحقق شرط ما، وتنفيذ تعليقات أخرى إذا لم يتحقق ذلك الشرط.
- جملة if...else if...else – تستخدم هذه الجملة إذا أردت اختيار تنفيذ كود معين (ضمن مجموعة أكواد).
- جملة switch – تستخدم هذه الجملة أيضا إذا أردت اختيار تنفيذ أحد مجموعات الأكواد. (سوف نقوم بشرح هذه الجملة بالتفصيل في الجزء التالي).

## جملة If

يجب استخدام هذه الجملة فقط إذا أردت تنفيذ كود معين في حالة تحقق شرط ما.

### البناء العام:

(الشرط (if

الكود الذي يتم تنفيذه إذا تحقق الشرط }

لاحظ أن if تكتب بحروف صغيرة lowercase letters. ويؤدي استخدام الحروف الكبيرة (IF) uppercase letters إلى ظهور رسالة الخطأ.

### مثال ١:

```
<script type="text/javascript">
//Write a "Good morning" greeting if
//the time is less than 10
var d=new Date()
var time=d.getHours()

if (time<10)
{
document.write("<b>Good morning</b>")
}
</script>
```

### مثال ٢:

```
<script type="text/javascript">
//Write "Lunch-time!" if the time is 11
var d=new Date()
var time=d.getHours()

if (time==11)
```

```
{  
document.write("<b>Lunch-time!</b>")  
}  
</script>
```

ملاحظة: عند المقارنة بين المتغيرات، يجب عليك دائماً استخدام علامتى تساوى (==).

لاحظ عدم وجود جملة else في المثال السابق. وذلك لأنك تريد تنفيذ كود معين فقط إذا تحقق شرط ما.

### جملة If...else

إذا كنت تريد تنفيذ كود معين إذا تحقق شرط ما، وتنفيذ كود آخر إذا لم يتحقق ذلك الشرط، إذن عليك استعمال جملة if...else.

### البناء العام:

```
if (الشرط)  
{  
    يتم تنفيذه إذا تحقق الشرط الكود الذى  
}  
else  
{  
    يتم تنفيذه إذا لم يتحقق الشرط الكود الذى  
}
```

### مثال:

```
<script type="text/javascript">  
//If the time is less than 10,  
//you will get a "Good morning" greeting.  
//Otherwise you will get a "Good day" greeting.  
var d = new Date()  
var time = d.getHours()
```

```
if (time < 10)
{
document.write("Good morning!")
}
else
{
document.write("Good day!")
}
</script>
```

### جملة If...else if...else

يجب استخدام هذه الجملة إذا أردت اختيار تنفيذ كود معين (ضمن مجموعة أكواد).

### البناء العام:

```
if (الشرط الأول)
{
يتم تنفيذه إذا تحقق الشرط الأول الكود الذي
}
else if (الشرط الثاني)
{
يتم تنفيذه إذا تحقق الشرط الثاني الكود الذي
}
else
{
يتم تنفيذه إذا لم يتحقق كل من الشرط الأول والشرط الثاني الكود الذي
}
}
```

### مثال:

```
<script type="text/javascript">
var d = new Date()
```

```
var time = d.getHours()
if (time<10)
{
document.write("<b>Good morning</b>")
}
else if (time>10 && time<16)
{
document.write("<b>Good day</b>")
}
else
{
document.write("<b>Hello World!</b>")
}
</script>
```

### جملة Switch

تستخدم هذه الجملة إذا أردت اختيار تنفيذ أحد مجموعات الأكواد.

مثال توضيحي على عمل جملة switch:

```
<html>
<body>
<script type="text/javascript">
var d = new Date()
theDay=d.getDay()
switch (theDay)
{
case 5:
document.write("<b>Finally Friday</b>")
break
case 6:
document.write("<b>Super Saturday</b>")
break
case 0:
document.write("<b>Sleepy Sunday</b>")
```

**break**

**default:**

```
document.write("<b>I'm really looking forward to this weekend!</b>")
}
</script>
```

<p>This JavaScript will generate a different greeting based on what day it is. Note that Sunday=0, Monday=1, Tuesday=2, etc.</p>

```
</body>
</html>
```

### البناء العام:

**switch(n)**

```
{
```

**case 1:**

تنفيذ الكتلة الأولى من الكود

فاصل

**case 2:**

تنفيذ الكتلة الثانية من الكود

فاصل

**default:**

الكود الذي يتم تنفيذه إذا كانت قيمة  $n$  تختلف عن  $case1$  و  $case2$

```
}
```

طريقة عمل جملة **switch**: في البداية لدينا تعبير معين  $n$  (في معظم الأحيان عبارة عن متغير)، والذي يتم تقييم قيمته. وبعد ذلك يتم مقارنته مع القيمة الخاصة بكل حالة  $case$  في البناء الكودي. فإذا كانت هناك مضاهاة، يتم تنفيذ الكود الموجود داخل الحالة. يفضل استخدام جملة **break** حتى لا ينتقل الكود تلقائياً إلى الحالة التالية. أنظر المثال التالي:

```

<script type="text/javascript">
//You will receive a different greeting based
//on what day it is. Note that Sunday=0,
//Monday=1, Tuesday=2, etc.
var d=new Date()
theDay=d.getDay()
switch (theDay)
{
case 5:
document.write("Finally Friday")
break
case 6:
document.write("Super Saturday")
break
case 0:
document.write("Sleepy Sunday")
break
default:
document.write("I'm looking forward to this weekend!")
}
</script>

```

#### ٥- المعاملات JavaScript Operators

#### المعاملات الحسابية Arithmetic Operators

المعامل	الوصف	مثال	النتيجة
+	الجمع	x=2 y=2 x+y	4
-	الطرح	x=5 y=2 x-y	3

*	الضرب	x=5 y=4 x*y	20
/	القسمة	15/5 5/2	3 2.5
%	Modulus (القسمة باقى)	5%2 10%8 10%2	1 2 0
++	التزايد Increment	x=5 x++	x=6
--	التناقص Decrement	x=5 x--	x=4

### معاملات التعيين Assignment Operators

المعامل	مثال	تساوي
=	x=y	x=y
+=	x+=y	x=x+y
-=	x-=y	x=x-y
*=	x*=y	x=x*y
/=	x/=y	x=x/y
%=	x%=y	x=x%y

### معاملات المقارنة Comparison Operators

المعامل	الوصف	مثال
==	يساوي	5==8 returns false
===	يساوي (مع التحقق من كل من قيمة value ونوع type البيانات)	x=5 y="5"

		$x==y$ returns true $x===y$ returns false
!=	لا يساوي	$5!=8$ returns true
>	أكبر من	$5>8$ returns false
<	أصغر من	$5<8$ returns true
>=	أكبر من أو يساوي	$5>=8$ returns false
<=	أصغر من أو يساوي	$5<=8$ returns true

### المعاملات المنطقية Logical Operators

المعامل	الوصف	مثال
&&	و	$x=6$ $y=3$ $(x < 10 \ \&\& \ y > 1)$ returns true
	أو	$x=6$ $y=3$ $(x==5 \    \ y==5)$ returns false
!	ليس	$x=6$ $y=3$ $!(x==y)$ returns true

### المعامل النصي String Operator

يطلق على البيان النصي في لغات البرمجة كلمة string، على سبيل المثال عبارة "Hello World!". ولربط متغيرين أو أكثر من المتغيرات النصية معا، نستخدم المعامل +.

```
txt1="What a very"  
txt2="nice day!"  
txt3=txt1+txt2
```

هنا المتغير txt3 يحتوي الآن على "What a very nice day!" .

ولكى تضيف مسافة بين المتغيرين النصيين، فقط ضع مسافة في التعبير، أو في أحد الجمل النصية، على النحو التالي:

```
txt1="What a very"  
txt2=" nice day!"  
txt3=txt1+" "+txt2  
or  
txt1="What a very "  
txt2="nice day!"  
txt3=txt1+txt2
```

الآن يحتوي المتغير txt3 على "What a very nice day!" .

### المعامل الشرطي Conditional Operator

تحتوي لغة JavaScript أيضا على معامل شرطي يقوم بتعيين قيمة للمتغير طبقا لشرط ما.

**البناء العام:**

```
variablename=(condition)?value1:value2
```

**مثال:**

```
greeting=(visitor=="PRES")?"Dear President ":"Dear "
```

إذا كانت قيمة المتغير visitor تساوى PRES، إذن ضع الجملة النصية "Dear President" في المتغير المسمى greeting. أما إذا كانت قيمة المتغير visitor لا تساوى PRES، فعليك (أى الكمبيوتر) أن تضع الجملة النصية "Dear" في المتغير المسمى

.greeting

## الصناديق المنبثقة JavaScript Popup Boxes

في لغة JavaScript يمكنك إنشاء ثلاثة أنواع من الصناديق المنبثقة: صندوق التحذير، وصندوق التأكيد، وصندوق الحث.

### صندوق التحذير Alert Box

غالباً ما نستخدم صندوق التحذير عند الحاجة إلى التأكد من وصول المعلومات للمستخدم.

عندما يبتثق صندوق التحذير، يجب على المستخدم الضغط على زر "OK" للاستمرار.

### البناء العام:

```
alert("sometext")
```

مثال:

```
<html>
<head>
<script type="text/javascript">
function disp_alert()
{
alert("I am an alert box!!")
}
</script>
</head>
<body>
```

```
<input type="button" onclick="disp_alert()" value="Display
alert box" />
```

```
</body>
</html>
```

### صندوق التحذير مع فاصل سطري Alert box with line breaks

```
<html>
<head>
<script type="text/javascript">
function disp_alert()
{
  alert("Hello again! This is how we" + '\n' + "add line breaks to an
alert box!")
}
</script>
</head>
<body>

<input type="button" onclick="disp_alert()" value="Display
alert box" />

</body>
</html>
```

### صندوق التأكيد Confirm Box

غالبا ما يستخدم صندوق التأكيد عند الحاجة إلى تحقق المستخدم من، أو قبول، شيء ما.

عندما ينبثق صندوق التأكيد، يجب على المستخدم إما الضغط على زر "OK" أو زر "Cancel" للاستمرار.

إذا ضغط المستخدم على زر "OK"، يسترجع الصندوق القيمة (صح true). أما إذا ضغط المستخدم على زر "Cancel"، فإن القيمة المسترجعة هي (خطأ false).

## البناء العام:

```
confirm("sometext")
```

مثال:

```
<html>
<head>
<script type="text/javascript">
function disp_confirm()
{
var r=confirm("Press a button")
if (r==true)
{
document.write("You pressed OK!")
}
else
{
document.write("You pressed Cancel!")
}
}
</script>
</head>
<body>

<input type="button" onclick="disp_confirm()" value="Display
a confirm box" />

</body>
</html>
```

## صندوق الحث Prompt Box

غالبا ما يستخدم صندوق الحث عند الحاجة إلى إدخال قيمة معينة بواسطة المستخدم قبل الدخول إلى الصفحة.

عندما ينبثق صندوق التأكيد، يجب على المستخدم إما الضغط على زر "OK" أو زر "Cancel" للاستمرار بعد إدخال القيمة المطلوبة.

إذا ضغط المستخدم على زر "OK"، يسترجع الصندوق القيمة المدخلة. أما إذا ضغط المستخدم على زر "Cancel"، فإن القيمة المسترجعة هي (فارغ null).

### البناء العام:

```
prompt("sometext","defaultvalue")
```

### مثال:

```
<html>
<head>
<script type="text/javascript">
function disp_prompt()
{
var name=prompt("Please enter your name","Harry Potter")
if (name!=null && name!="")
{
document.write("Hello " + name + "! How are you today?")
}
}
</script>
</head>
<body>

<input type="button" onclick="disp_prompt()" value="Display a
prompt box" />

</body>
</html>
```

تعرف الدالة على أنها كتلة من الكود يمكن إعادة استخدامها، ويتم تنفيذها بواسطة حدث ما، أو عند استدعاء الدالة.

### أمثلة:

يوضح المثال التالي كيفية استدعاء الدالة:

```
<html>
<head>

<script type="text/javascript">
function myfunction()
{
alert("HELLO")
}
</script>

</head>
<body>

<form>
<input type="button"
onclick="myfunction()"
value="Call function">
</form>

<p>By pressing the button, a function will be called. The function
will alert a message.</p>

</body>
</html>
```

والمثال التالي يبين كيفية تمرير متغير إلى الدالة، واستخدام هذا المتغير داخل الدالة:

```
<html>
```

```
<head>
```

```
<script type="text/javascript">
function myfunction(txt)
{
alert(txt)
}
</script>
```

```
</head>
```

```
<body>
```

```
<form>
```

```
<input type="button"
onclick="myfunction('Hello')"
value="Call function">
</form>
```

<p>By pressing the button, a function with an argument will be called. The function will alert this argument.</p>

```
</body>
```

```
</html>
```

وفي المثال التالي نرى معا كيفية تمرير متغيرات إلى الدالة، واستخدام هذه المتغيرات داخل الدالة:

```
<html>
```

```
<head>
```

```
<script type="text/javascript">
function myfunction(txt)
{
alert(txt)
}
</script>
```

```
</head>
```

```
<body>
```

```
<form>
```

```
<input type="button"
onclick="myfunction('Good Morning!)"
value="In the Morning">
```

```
<input type="button"
onclick="myfunction('Good Evening!)"
value="In the Evening">
</form>
```

```
<p>
```

When you click on one of the buttons, a function will be called. The function will alert the argument that is passed to it.

```
</p>
```

```
</body>
```

```
</html>
```

أما المثال التالي فيوضح كيفية استرجاع قيمة عن طريق الدالة:

```
<html>
```

```
<head>
```

```
<script type="text/javascript">
function myFunction()
{
return ("Hello, have a nice day!")
}
</script>
```

```
</head>
```

```
<body>
```

```
<script type="text/javascript">
document.write(myFunction())
</script>
```

<p>The script in the body section calls a function.</p>

<p>The function returns a text.</p>

```
</body>
</html>
```

أما المثال التالي فيبين كيفية قيام الدالة بإيجاد نتيجة تعبيرين، واسترجاع النتيجة.

```
<html>
<head>
<script type="text/javascript">
function product(a,b)
{
return a*b
}
</script>
</head>
```

```
<body>
<script type="text/javascript">
document.write(product(4,3))
</script>
```

<p>The script in the body section calls a function with two parameters (4 and 3).</p>

<p>The function will return the product of these two parameters.</p>

```
</body>
</html>
```

## الدوال JavaScript Functions

إذا أردت أن يقوم برنامج المتصفح browser بتنفيذ التعليمات البرمجية بمجرد تحميل الصفحة، يمكنك كتابة تلك التعليمات كدالة.

تحتوي الدالة على كود يتم تنفيذه فقط بواسطة حدث ما، أو عند استدعاء الدالة. يمكنك استدعاء الدالة من أي مكان في الصفحة (أو حتى من أية صفحة أخرى في حالة ما إذا كانت الدالة موجودة في ملف js خارجي).

يتم تعريف الدوال في بداية الصفحة، وذلك في رأس <head> الصفحة.

مثال:

```
<html>
<head>
<script type="text/javascript">
function displaymessage()
{
alert("Hello World!")
}
</script>
</head>
<body>
<form>
<input type="button" value="Click me!"
onclick="displaymessage()" >
</form>
</body>
</html>
```

في حالة عدم كتابة السطر: alert("Hello world!!") الموجود بالمثل السابق داخل الدالة، سوف يتم تنفيذه بمجرد الوصول إليه أثناء تحميل الصفحة. لكن المثل السابق تم كتابته بحيث لا ينفذ هذا السطر قبل ضغط المستخدم على الزر. فقد قمنا بإضافة الحدث onClick للزر الذي سوف يقوم بتنفيذ الدالة

displaymessage() عند الضغط على الزر.

سوف تتعرف لاحقا بشكل أكثر على الأحداث events في لغة JavaScript.

## بناء الدالة

الشكل العام لإنشاء الدالة على النحو التالي:

```
function functionname(var1,var2,...,varX)
{
    أسطر الكود
}
```

يعتبر var1 و var2 متغيرات أو قيم يتم تمريرها للدالة. أما الأقواس المتعرجة { و } فهي تحدد بداية ونهاية الدالة.

ملاحظة: يجب أن تحتوى الدالة على الأقواس () بعد اسم الدالة، حتى إذا لم تكن هناك أية متغيرات أو قيم يتم تمريرها إليها:

```
function functionname()
{
    أسطر الكود
}
```

ملاحظة: لا تنسى عزيزى القارئ أهمية الحروف الكبيرة capitals في لغة JavaScript. فكلمة function يجب كتابتها بحروف صغيرة lowercase letters، وخلاف ذلك يؤدي إلى ظهور رسالة خطأ. كما يجب أن تلاحظ أيضا أنه يجب عليك أن تستدعى الدالة بنفس الحروف الموجودة في اسم الدالة.

## جملة return

تستخدم جملة return لتحديد القيمة المسترجعة من الدالة. لذا يجب على الدوال التى سوف تسترجع قيمة ما، أن تستخدم جملة return.

مثال:

تقوم الدالة التالية باسترجاع نتيجة ضرب رقمين (a و b):

```
function prod(a,b)
{
x=a*b
return x
}
```

وعندما تقوم باستدعاء الدالة السابقة، يجب تمرير القيمتين:

```
product=prod(2,3)
```

القيمة المسترجعة من الدالة prod() هي ٦، وسوف يتم تخزينها في المتغير product.

## ٧- الحلقة التكرارية JavaScript For Loop

تستخدم الحلقات التكرارية في تنفيذ كود معين لعدد من المرات، أو حتى يتحقق شرط ما.

**أمثلة:**

### الحلقة التكرارية For

تستخدم الحلقة التكرارية For لتنفيذ كود معين لعدد من المرات، وذلك على النحو التالي:

```
<html>
<body>

<script type="text/javascript">
for (i = 0; i <= 5; i++)
{
document.write("The number is " + i)
document.write("<br />")
}
```

```
}  
</script>
```

```
<p>Explanation:</p>
```

```
<p>This for loop starts with i=0.</p>
```

```
<p>As long as <b>i</b> is less than, or equal to 5, the loop will  
continue to run.</p>
```

```
<p><b>i</b> will increase by 1 each time the loop runs.</p>
```

```
</body>
```

```
</html>
```

ويوضح المثال التالي كيفية استخدام الحلقة التكرارية for مع الحقول الخاصة  
بالرؤوس headings في لغة HTML:

```
<html>
```

```
<body>
```

```
<script type="text/javascript">
```

```
for (i = 1; i <= 6; i++)
```

```
{
```

```
document.write("<h" + i + ">This is header " + i)
```

```
document.write("</h" + i + ">")
```

```
}
```

```
</script>
```

```
</body>
```

```
</html>
```

## الحلقات التكرارية JavaScript Loops

أثناء كتابة الكود غالبا ما تحتاج إلى تنفيذ تعليقات برمجية معينة أكثر من مرة.

وبدلا من تكرار كتابة الكود نفسه بعدد المرات التي تريد تنفيذه، يمكننا استخدام الحلقات التكرارية loops لتحقيق ذلك.

في لغة JavaScript يوجد نوعان من الحلقات التكرارية:

- الحلقة for – تنفيذ الكود عدد معين من المرات.
- الحلقة while – تكرار تنفيذ الكود حتى يتحقق شرط معين.

### الحلقة For

تستخدم هذه الحلقة عندما نعلم مقدما عدد المرات التي يجب فيها تنفيذ الكود.

البناء العام:

```
for (var=startvalue;var<=endvalue;var=var+increment)
{
    سيتم تنفيذها التعليقات البرمجية التي
}
```

### مثال:

الشرح: سوف نقوم في المثال التالي بتعريف حلقة تكرارية تبدأ من  $i=0$ . وسوف تستمر هذه الحلقة في التنفيذ طالما كانت قيمة  $i$  أقل من أو تساوى ١٠. تزداد قيمة  $i$  بواحد في كل مرة يتم فيها تنفيذ الحلقة التكرارية.

ملاحظة: يمكن لمعامل التزايد increment parameter أن يكون سلبيا، وكذلك علامة  $=>$  يمكن أيضا أن تكون أية علامة من علامات المقارنة.

```
<html>
<body>
<script type="text/javascript">
var i=0
for (i=0;i<=10;i++)
{
document.write("The number is " + i)
```

```
document.write("<br />")
}
</script>
</body>
</html>
```

### النتيجة :

The number is 0  
The number is 1  
The number is 2  
The number is 3  
The number is 4  
The number is 5  
The number is 6  
The number is 7  
The number is 8  
The number is 9  
The number is 10

### الحلقة التكرارية JavaScript While Loop

تستخدم الحلقات التكرارية في تنفيذ الكود عدد معين من المرات، أو حتى يتحقق شرط معين.

### الحلقة التكرارية while

يبين المثال التالي كيفية كتابة الحلقة التكرارية while. وتستخدم هذه الحلقة في تنفيذ الكود حتى يتحقق شرط معين.

```
<html>
<body>

<script type="text/javascript">
i = 0
while (i <= 5)
```

```
{  
document.write("The number is " + i)  
document.write("<br>")  
i++  
}  
</script>
```

<p>Explanation:</p>

<p><b>i</b> is equal to 0.</p>

<p>While <b>i</b> is less than , or equal to, 5, the loop will continue to run.</p>

<p><b>i</b> will increase by 1 each time the loop runs.</p>

</body>

</html>

### الحلقة التكرارية Do while

يبين المثال التالي كيفية كتابة الحلقة التكرارية do...while. وتستخدم هذه الحلقة في تنفيذ الكود حتى يتحقق شرط معين. ونلاحظ أن هذه الحلقة دائما يتم تنفيذها على الأقل مرة واحدة، حتى لو لم يتحقق الشرط؛ وذلك نظرا لأن الكود يتم تنفيذه قبل اختبار الشرط.

<html>

<body>

```
<script type="text/javascript">
```

```
i = 0
```

```
do
```

```
{
```

```
document.write("The number is " + i)
```

```
document.write("<br>")
```

```
i++  
}  
while (i <= 5)  
</script>
```

<p>Explanation:</p>

<p><b>i</b> equal to 0.</p>

<p>The loop will run</p>

<p><b>i</b> will increase by 1 each time the loop runs.</p>

<p>While <b>i</b> is less than , or equal to, 5, the loop will continue to run.</p>

```
</body>  
</html>
```

### الحلقة التكرارية while

تستخدم الحلقة التكرارية while في تكرار تنفيذ الكود بشكل مستمر طالما كان الشرط صحيح.

```
while (var<=endvalue)  
{  
الكود المطلوب تكرار تنفيذه  
}
```

ملاحظة: يمكن استخدام أى من علامات المقارنة، وليس فقط <=.

### مثال:

الشرح: يقدم المثال التالى حلقة تكرارية تبدأ عندما تكون قيمة i تساوى صفر. وسوف تستمر الحلقة التكرارية فى التنفيذ طالما كانت i أقل من أو تساوى ١٠. كما

أن قيمة  $i$  سوف تتزايد بمقدار ١ في كل مرة يتم فيها تنفيذ الحلقة التكرارية.

```
<html>
<body>
<script type="text/javascript">
var i=0
while (i<=10)
{
document.write("The number is " + i)
document.write("<br />")
i=i+1
}
</script>
</body>
</html>
```

**النتيجة :**

```
The number is 0
The number is 1
The number is 2
The number is 3
The number is 4
The number is 5
The number is 6
The number is 7
The number is 8
The number is 9
The number is 10
```

### الحلقة التكرارية do...while

تعتبر الحلقة التكرارية do...while شكلا مختلفا من الحلقة التكرارية while. فهذه الحلقة دائما يتم فيها تنفيذ الكود مرة واحدة، ثم تتكرر الحلقة طالما كان الشرط صحيح. ومن هنا نقول أن هذه الحلقة التكرارية تنفذ مرة واحدة على الأقل، حتى

لو كان الشرط غير صحيح؛ وذلك نظرا لتنفيذ الكود قبل اختبار الشرط.

```
do
{
    سيتم تنفيذها التعليقات البرمجية التي
}
while (var<=endvalue)
```

**مثال:**

```
<html>
<body>
<script type="text/javascript">
var i=0
do
{
document.write("The number is " + i)
document.write("<br />")
i=i+1
}
while (i<0)
</script>
</body>
</html>
```

**النتيجة:**

**The number is 0**

**جملتي Continue و Break**

تحتوي لغة JavaScript على جملتين تستخدمان بشكل خاص داخل الحلقات التكرارية، وهما جملة break وجملة continue.

**أمثلة:**

**جملة Break**

تستخدم هذه الجملة لإيقاف تنفيذ الحلقة التكرارية.

```
<html>
<body>
<script type="text/javascript">
var i=0
for (i=0;i<=10;i++)
{
if (i==3){break}
document.write("The number is " + i)
document.write("<br />")
}
</script>
<p>Explanation: The loop will break when i=3.</p>
</body>
</html>
```

**النتيجة :**

The number is 0  
The number is 1  
The number is 2

**جملة Continue**

تستخدم هذه الجملة لإيقاف تنفيذ الحلقة التكرارية الحالية، والاستمرار مع القيمة التالية.

```
<html>
<body>
<script type="text/javascript">
var i=0
for (i=0;i<=10;i++)
{
if (i==3){continue}
document.write("The number is " + i)
document.write("<br />")
}
```

```
}  
</script>
```

**<p>Explanation: The loop will break the current loop and continue with the next value when i=3.</p>**

```
</body>  
</html>
```

### النتيجة:

**The number is 0  
The number is 1  
The number is 2  
The number is 4  
The number is 5  
The number is 6  
The number is 7  
The number is 8  
The number is 9  
The number is 10**

### جملة For...In

تستخدم جملة for...in للتكرار بين عناصر المصفوفة array، أو بين الخصائص المكونة لكائن object معين. ويتم تنفيذ الكود في هذه الجملة مرة واحدة لكل عنصر/خاصية.

### البناء العام:

```
for (variable in object)  
{  
    سيتم تنفيذه الكود الذي  
}
```

يمكن للمتغير أن يكون عنصرا في مصفوفة، أو خاصية لكائن معين.

يوضح المثال التالي كيفية استخدام جملة for...in للتكرار بين عناصر المصفوفة:

```
<html>
<body>
<script type="text/javascript">
var x
var mycars = new Array()
mycars[0] = "Saab"
mycars[1] = "Volvo"
mycars[2] = "BMW"

for (x in mycars)
{
document.write(mycars[x] + "<br />")
}
</script>
</body>
</html>
```

#### ٨- الأحداث JavaScript Events

تعرف الأحداث بأنها أفعال يمكن استيعابها/استكشافها بواسطة لغة البرمجة. ففي لغة JavaScript لدينا القدرة على إنشاء صفحات ويب ديناميكية. ويمكن لكل عنصر في صفحة الويب أن يضم أحداث events تستطيع بدورها استدعاء دوال functions لغة JavaScript. على سبيل المثال، يمكننا استخدام الحدث onClick لزر ما، للإشارة إلى تنفيذ دالة معينة عندما يضغط المستخدم على الزر. ونقوم بتعريف الأحداث في حقول HTML.

#### أمثلة على الأحداث:

- النقر بالفأرة A mouse click.
- تحميل صفحة ويب أو صورة A web page or an image loading.
- تمرير الفأرة فوق نقطة ساخنة في صفحة الويب Mousing over a hot spot on the web page.
- اختيار صندوق إدخال في نموذج HTML Selecting an input box in an HTML form.
- إرسال النموذج Submitting an HTML form.
- الضغط على مفتاح A keystroke.

## مرجع الأحداث JavaScript Event Reference

تستخدم الأحداث عادة مع الدوال، حيث لا يمكن تنفيذ الدالة قبل وقوع الحدث.

استحدثت الإصدار الرابعة من لغة HTML القدرة على تنفيذ الأحداث لأفعال معينة على شاشة برنامج المتصفح؛ كأن تبدأ تنفيذ تعليمات برمجية بلغة JavaScript عندما يضغط المستخدم على عنصر ما في الصفحة. وسوف نعرض فيما يلي قائمة بالخصائص التي يمكن إدراجها في حقول HTML لتعريف أفعال الأحداث.

الخاصية	يقع الحدث عند...
onabort	تعطل تحميل الصورة
onblur	فقدان عنصر ما التركيز focus
onchange	تغيير المستخدم لمحتوى الحقل
onclick	النقر بالفأرة على كائن
ondblclick	النقر المزدوج بالفأرة على كائن
onerror	حدوث خطأ ما أثناء تحميل ملف أو صورة
onfocus	حصول العنصر على التركيز focus
onkeydown	الضغط على مفتاح في لوحة المفاتيح
onkeypress	الضغط على مفتاح في لوحة المفاتيح أو وضعه لأسفل
onkeyup	إطلاق مفتاح في لوحة المفاتيح
onload	الانتهاء من تحميل الصفحة أو الصورة
onmousedown	الضغط على زر الفأرة
onmousemove	تحريك الفأرة
onmouseout	تحريك الفأرة بعيداً عن الكائن
onmouseover	تحريك الفأرة فوق العنصر
onmouseup	إطلاق زر الفأرة

الضغط على زر إعادة الضبط Reset	onreset
تغيير حجم النافذة أو الإطار	onresize
اختيار النص	onselect
الضغط على زر إرسال Submit	onsubmit
خروج المستخدم من الصفحة	onunload

### الحدثان onload وonUnload

يقع الحدثان onload وonUnload عندما يدخل المستخدم على الصفحة أو يخرج منها. وغالبا ما يستخدم الحدث onload للتحقق من نوع برنامج المتصفح الذي يستخدمه المستخدم، وكذلك رقم إصدارته. وهو ما يمكننا من تحميل النسخة الملائمة من الصفحة، استنادا لهذه المعلومات.

وغالبا أيضا ما يستخدم الحدثان onload وonUnload عند التعامل مع ملفات الكوكيز cookies التي يجب ضبطها عندما يدخل المستخدم إلى الصفحة أو يغادرها. على سبيل المثال: يمكنك إظهار صندوق منبثق popup يسأل المستخدم على اسمه عند بداية دخوله إلى الصفحة. ويتم تخزين الاسم بعد ذلك في ملف الكوكي. وفي كل مرة لاحقة يعود فيها المستخدم للصفحة، يمكنك إظهار صندوق منبثق، يعرض مثلا: "مرحبا على شاكر!".

### الأحداث onFocus, onBlur and onChange

غالبا ما تستخدم الأحداث onFocus وonBlur وonChange عند التحقق من حقول البيانات في نموذج form ما.

فيما يلي مثال توضيحي على كيفية استخدام الحدث onChange، وفيه يتم استدعاء الدالة checkEmail() عندما يغير المستخدم محتوى الحقل:

```
<input type="text" size="30"
id="email" onchange="checkEmail()">;
```

### الحدث onSubmit

يتم استخدام الحدث onSubmit للتحقق من جميع حقول النموذج قبل إرساله. فيما يلي مثال توضيحي على كيفية استخدام الحدث onSubmit، وفيه يتم استدعاء الدالة checkForm() عندما يضغط المستخدم على زر الإرسال في النموذج. وفي حالة عدم قبول البيانات الموجودة في الحقول، يجب إلغاء عملية الإرسال. تقوم الدالة checkForm() باسترجاع القيمة صح true أو خطأ false. وإذا كانت القيمة المسترجعة صحيحة true، يتم إرسال النموذج. وبخلاف ذلك سوف تلغى عملية الإرسال:

```
<form method="post" action="xxx.htm"
onsubmit="return checkForm()">
```

### الحدثان onMouseOver و onMouseOut

غالباً ما يستخدم الحدثان onMouseOver و onMouseOut في إنشاء أزرار متحركة animated.

فيما يلي مثال توضيحي على كيفية استخدام الحدث onMouseOver، وفيه يظهر صندوق تحذير alert box عند وقوع الحدث onMouseOver:

```
<a href="http://www.aks4lis.net"
onmouseover="alert('An onMouseOver event');return false">

</a>
```

### ٩- جملة Try...Catch

تسمح جملة try...catch باختبار الكود من أجل اكتشاف الأخطاء.

**أمثلة:**

كيفية استخدام جملة try...catch:

```
<html>
<head>
<script type="text/javascript">
var txt=""
```

```

function message()
{
try
{
addlert("Welcome guest!")
}
catch(err)
{
txt="There was an error on this page.\n\n"
txt+="Error description: " + err.description + "\n\n"
txt+="Click OK to continue.\n\n"
alert(txt)
}
}
</script>
</head>
<body>
<input type="button" value="View message"
onclick="message()" />
</body>
</html>

```

وفيما يلي مثال توضيحي آخر على كيفية استخدام جملة try...catch مع صندوق تأكيد confirm box:

```

<html>
<head>
<script type="text/javascript">
var txt=""
function message()
{
try
{
addlert("Welcome guest!")
}

```

```

catch(err)
{
    txt="There was an error on this page.\n\n"
    txt+="Click OK to continue viewing this page,\n"
    txt+="or Cancel to return to the home page.\n\n"
    if(!confirm(txt))
    {
        document.location.href="http://www.aks4lis.net/"
    }
}
}
</script>
</head>
<body>
<input      type="button"      value="View      message"
onclick="message()" />
</body>
</html>

```

### اصطياد الأخطاء Catching Errors

عند تصفح صفحات الويب على الإنترنت، رأينا جميعاً صندوق JavaScript التحذيري، الذي يخبرنا بأن هناك خطأ وقت التنفيذ runtime error، ويسألنا ما إذا كنا نريد أن نتخلص من الأخطاء "Do you wish to debug?". وتفيد رسائل الخطأ هذه المبرمجين أكثر ما تفيد المستخدمين. فعندما يرى المستخدمون مثل هذه الأخطاء، عادة ما يغادرون الصفحة أو الموقع.

في هذا الجزء سوف نتعرف معاً عزيزي القارئ على كيفية تتبع الأخطاء ومعالجتها، وذلك حتى لا تفقد زائري موقعك الخاص على الويب.

توجد طريقتان لاصطياد الأخطاء في صفحة الويب:

▣ استخدام جملة try...catch (متاحة في برنامج IE5+، وبرنامج Mozilla 1.0،

وبرنامج Netscape 6).

■ استخدام الحدث onerror. وهو الحل القديم لاصطياد الأخطاء (متاح منذ الإصدار الثالثة لبرنامج Netscape).

### جملة Try...Catch

تسمح جملة try...catch باختبار الكود من الأخطاء. يحتوي الجزء الذي يلي كلمة try على الكود المطلوب تنفيذه، أما الجزء الذي يلي كلمة catch فيحتوي الكود المطلوب تنفيذه إذا حدث خطأ ما.

### البناء العام:

```
try
{
الكود المطلوب تنفيذه
}
catch(err)
{
الكود المطلوب تنفيذه عند وقوع خطأ
}
```

لا حظ أن جملة try...catch تكتب بحروف صغيرة. حيث يسبب استخدام الحروف الكبيرة ظهور رسالة خطأ.

### مثال ١:

يوضح المثال التالي تعليقات برمجية يفترض أنها تعرض رسالة " Welcome guest! عندما تضغط على زر. ومع ذلك يوجد خطأ typo في الدالة message(). فكلمة alert() مكتوبة خطأ، هكذا addlert()؛ ومن ثم يقع الخطأ:

```
<html>
<head>
<script type="text/javascript">
```

```

function message()
{
  addlert("Welcome guest!")
}
</script>
</head>

<body>
<input type="button" value="View message"
onclick="message()" />
</body>

</html>

```

وحتى تسلم من حدوث مثل هذه الأخطاء، يمكنك استخدام جملة `try...catch`. يقوم المثال التالي بالعرض ذاته الموجود في المثال السابق، وهو عرض رسالة `alert()` "Welcome guest!"، لكن مع استخدام جملة `try...catch`. ونظراً لأن `alert()` مكتوبة بشكل خاطئ، يقع الخطأ. إلا أنه في هذه المرة يقوم الكود الموجود بعد كلمة `catch` باصطياد الخطأ، وتنفيذ التعليقات البرمجية الخاصة بالتعامل مع ذلك الخطأ. يعرض الكود رسالة تخبر المستخدم ما الذي حدث:

```

<html>
<head>
<script type="text/javascript">
var txt=""
function message()
{
  try
  {
    addlert("Welcome guest!")
  }
  catch(err)
  {

```

```

txt="There was an error on this page.\n\n"
txt+="Error description: " + err.description + "\n\n"
txt+="Click OK to continue.\n\n"
alert(txt)
}
}
</script>
</head>

<body>
<input type="button" value="View message"
onclick="message()" />
</body>

</html>

```

## المثال ٢:

يستخدم المثال التالي صندوق تأكيد لعرض رسالة تحبب المستخدمين أنه بإمكانهم الضغط على زر موافق للاستمرار في عرض الصفحة، أو الضغط على زر إلغاء للذهاب إلى الصفحة الرئيسية. إذا استرجع صندوق التأكيد القيمة false، فإن المستخدم ضغط على زر إلغاء. ومن ثم يوجه الكود المستخدم إلى الصفحة الرئيسية. أما إذا كانت القيمة المسترجعة true، فلن يقوم الكود بأى شيء:

```

<html>
<head>
<script type="text/javascript">
var txt=""
function message()
{
try
{
addalert("Welcome guest!")
}
}

```

```

catch(err)
{
txt="There was an error on this page.\n\n"
txt+="Click OK to continue viewing this page,\n"
txt+="or Cancel to return to the home page.\n\n"
if(!confirm(txt))
{
document.location.href="http://www.aks4lis.net/"
}
}
}
</script>
</head>
<body>
<input type="button" value="View message"
onclick="message()" />
</body>
</html>

```

### الحدث onerror

سوف نقوم بشرح الحدث onerror بعد قليل، لكن الآن سوف نتعلم معا عزيزي القارئ كيفية استخدام جملة throw لإنشاء استثناء exception. يمكن استخدام جملة throw مع جملة catch...try.

### جملة Throw

تسمح جملة throw بإنشاء استثناء. وعند استخدامها مع جملة catch...try، يمكنك التحكم في تدفق البرنامج، وتخليق رسائل أخطاء بدقة.

### البناء العام:

#### throw(exception)

قد يكون الاستثناء قيمة نصية، رقمية، أو منطقية، أو كائن ما.

لا حظ أن كلمة throw تكتب بحروف صغيرة. حيث يسبب استخدام الحروف الكبيرة ظهور رسالة خطأ.

### مثال ١:

يوضح المثال التالي كيفية استخدام جملة throw، وفيه يقوم البرنامج بالتحقق من قيمة المتغير x. فإذا كانت القيمة أكبر من ١٠ أو أقل من ٠، سوف يتم إلقاء (throw) رسالة الخطأ. يتم اصطياد الخطأ عن طريق جملة catch، ومن ثم تعرض الرسالة الملائمة:

```
<html>
<body>
<script type="text/javascript">
var x=prompt("Enter a number between 0 and 10:", "");
try
{
if(x>10)
    throw "Err1"
else if(x<0)
    throw "Err2"
else if(isNaN(x))
    throw "Err3"
}
catch(er)
{
if(er=="Err1")
    alert("Error! The value is too high")
if(er == "Err2")
    alert("Error! The value is too low")
if(er == "Err3")
    alert("Error! The value is not a number")
}
</script>
</body>
</html>
```

## الحدث onerror

لقد تعرفنا معاً عزيزي القارئ على كيفية استخدام جملة try...catch لاصطياد الأخطاء في صفحات الويب. وحين الوقت الآن للتعرف على الحدث onerror والذي - كما ذكرنا من قبل - يعتبر الحل القديم المستخدم لنفس الغرض (اصطياد الأخطاء في صفحات الويب).

ينطلق تنفيذ هذا الحدث عند وقوع خطأ ما في تنفيذ الكود.

وحتى تتمكن من استخدام الحدث onerror، يجب عليك إنشاء دالة للتعامل مع الخطأ. وبعد ذلك تقوم باستدعاء الدالة، التي يجب أن تحتوي على ثلاثة عناصر: رسالة الخطأ msg، وعنوان الصفحة url، والسطر line الذي وقع فيه الخطأ.

### البناء العام:

```
onerror=handleErr
function handleErr(msg,url,l)
{
//التعليقات التالية تنفذ عند وقوع الخطأ//
return true or false
}
```

تحدد القيمة المسترجعة بواسطة الحدث onerror ما إذا كان برنامج المتصفح سوف يعرض رسالة خطأ قياسية. فإذا كانت القيمة false، يعرض برنامج المتصفح رسالة خطأ قياسية. أما إذا كانت القيمة المسترجعة true، فلن يعرض برنامج المتصفح رسالة خطأ قياسية.

### مثال: الحدث onerror (مثال مع خطأ).

يوضح المثال التالي كيفية استخدام الحدث onerror لاصطياد الأخطاء في صفحة الويب:

```
<html>
<head>
```

```

<script type="text/javascript">
onerror=handleErr
var txt=""
function handleErr(msg,url,l)
{
txt="There was an error on this page.\n\n"
txt+="Error: " + msg + "\n"
txt+="URL: " + url + "\n"
txt+="Line: " + l + "\n\n"
txt+="Click OK to continue.\n\n"
alert(txt)
return true
}
function message()
{
adddlert("Welcome guest!")
}
</script>
</head>
<body>
<input type="button" value="View message"
onclick="message()" />
</body>
</html>

```

### الرموز الخاصة JavaScript Special Characters

في لغة JavaScript يمكنك إضافة رموز خاصة إلى النص بواسطة استخدام علامة الشرطة المائلة للخلف للـ backslash.

#### إدراج الرموز الخاصة

تستخدم الشرطة المائلة للخلف ( \ ) لإدراج الفاصلة العلوية apostrophes، والأسطر الجديدة، وعلامات التنصيص quotes، وغيرها من الرموز إلى النص. أنظر الكود التالي:

```
var txt="مرافق المعلومات العربية. نحن الجيل الجديد من "سدنة المعرفة" في"
document.write(txt)
```

في لغة JavaScript تبدأ الجملة النصية وتنتهي بعلامتي تنصيص، أو حتى علامة واحدة. وهذا يعنى أن الجملة النصية في الكود أعلاه سوف تبتتر هكذا: نحن الجيل الجديد من.

لعلاج هذه المشكلة، يجب أن تضع علامة \ قبل كل علامة تنصيص في "سدنة المعرفة". وهذا من شأنه أن يحول علامتي التنصيص إلى رموز نصية عادية:

```
var txt="مرافق المعلومات نحن الجيل الجديد من \"سدنة المعرفة\" في"
"العربية."
document.write(txt)
```

الآن سوف تظهر الجملة النصية هكذا: نحن الجيل الجديد من "سدنة المعرفة" في مرافق المعلومات العربية.

وإليك مثال آخر:

```
document.write ("You \& me are readings books!")
```

ونتيجة الكود السابق على النحو التالي:

**You & me are reading books!**

يبين الجدول التالي الرموز الخاصة الأخرى التي يمكنك إضافتها إلى النصوص باستخدام الشرطة المائلة للخلف:

الكود	المخرجات
'	علامة التنصيص المفردة
"	علامة التنصيص المزدوجة
\&	الواوية
\\	الشرطتان المائلتان للخلف
\n	سطر جديد

## إرشادات عامة JavaScript Guidelines

هناك بعض الأمور الهامة التي يجب أن تعرفها وأنت تكتب التعليقات البرمجية بلغة JavaScript:

### □ لغة JavaScript حساسة لحالة الأحرف

على سبيل المثال الدالة "myfunction" ليست "myFunction"، والمتغير "myVar" ليس هو المتغير "myvar".

وهذا يعنى أن لغة JavaScript حساسة لحالة الأحرف؛ ومن ثم يجب عليك أن تراقب استخدام الحروف الكبيرة بشدة عند إنشاء أو استدعاء المتغيرات، أو الكائنات، أو الدوال.

### □ المسافات البيضاء White Space

تتجاهل لغة JavaScript المسافات الإضافية. ومن ثم يمكنك إضافة مسافات بيضاء للكود حتى تجعله أكثر قابلية للقراءة. على سبيل المثال يعتبر السطران التاليان متساويان:

```
name="Hege"  
name = "Hege"
```

### □ تجزئ سطر الكود Break up a Code Line

يمكنك تجزئ سطر الكود داخل جملة نصية within text string باستخدام الشرطة المائلة للخلف. فالمثال التالي سوف يعرض بشكل صحيح:

```
document.write("Hello \  
World!")
```

لكن لا يمكنك تجزئ سطر الكود على النحو التالي:

```
document.write \  
("Hello World!")
```

## □ التعليقات Comments

يمكن إضافة التعليقات أو الملاحظات إلى أكواد JavaScript بواسطة استخدام الشرطتان الأماميتان // :

```
//this is a comment  
document.write("Hello World!")
```

أو عن طريق استخدام علامة /\* / وعلامة /\* (وهذا يقوم بإنشاء تعليق متعدد الأسطر):

```
/* This is a comment  
block. It contains  
several lines */  
document.write("Hello World!")
```

## تدريبات عملية

١. حدد ثلاثة مجالات مختلفة لاستخدام لغة JavaScript في صفحات الويب.
٢. حدد صحة أو خطأ العبارات التالية:
  - أ- تعتبر لغة Java ولغة JavaScript شيء واحد.
  - ب- يمكن استخدام JavaScript للتحقق من صحة البيانات.
  - ج- نستخدم حقل `<script>` لإدراج كود JavaScript داخل صفحة HTML.
  ٣. ما المقصود بفترة حياة المتغيرات Lifetime of Variables؟
  ٤. اشرح الكود التالي:

```
<script type="text/javascript">
//Write a "Good evening" greeting if
//the time is greater than 12
var d=new Date()
var time=d.getHours()

if (time>12)
{
document.write("<b>Good evening</b>")
}
</script>
```

٥. ما الفرق بين جملة if..else if..else وجملة switch؟
٦. متى نستخدم الحلقة التكرارية for؟ ومتى نستخدم الحلقة التكرارية while؟

٧. وضح، من خلال برنامج بسيط، فكرة استخدام جملة try...catch؟
٨. في لغة JavaScript يمكنك إضافة رموز خاصة إلى النص بواسطة استخدام علامة -----.
٩. في لغة JavaScript تعتبر الدالة myfunction والدالة myFunction مختلفتان. لماذا؟
١٠. كيف يمكنك إضافة تعليقات comments إلى أسطر الكود؟

## المصادر

1. دبلوم علوم الحاسب - أساسيات برمجة الانترنت **Internet Programming**. تأليف: مركز التعريب والبرمجة. الدار العربية للعلوم، ٢٠٠٤. ٣٦٠ ص. (يغطي القسم الأول لغة JavaScript).
2. David Flanagan. *JavaScript : the definitive guide*. 3rd ed. Beijing ; Sebastopol, CA : O'Reilly, c1998.
3. Thomas Powell, Fritz Schneider. *JavaScript : the complete reference*. New York : McGraw-Hill, 2001.
4. Jerry Lee Ford Jr. *Learn JavaScript in a weekend [electronic resource]*. 2nd ed. Boston, Mass. : Premier Press, c2004.