



## الفصل التاسع: مهارات JQuery

| الصفحة | العنوان   |
|--------|---|
| 3      | <b>1. التعامل مع عناصر الوثيقة</b>              |
| 4      | <b>1.1</b> التعامل مع نموذج كائن الوثيقة DOM    |
| 11     | <b>2. التعامل مع الأحداث</b>                    |
| 12     | <b>1.2</b> التعامل مع الأحداث                   |
| 22     | <b>3. Ajax: Asynchronous JavaScript and XML</b> |
| 23     | <b>1.3</b> التعامل مع تقنية Ajax                |
| 23     | <b>2.3</b> تحميل البيانات                       |
| 24     | <b>3.3</b> إرسال البيانات إلى المخدم            |
| 29     | <b>4.3</b> وظائف JQuery AJAX                    |
| 30     | <b>5.3</b> أحداث JQuery AJAX                    |
| 31     | <b>4. التأثيرات Effects</b>                     |
| 32     | <b>1.4</b> التأثيرات                            |
| 35     | <b>2.4</b> طرق التأثيرات JQuery                 |
| 36     | <b>3.4</b> مكتبة واجهة المستخدم UI              |

## الكلمات المفتاحية

عناصر الوثيقة، الأحداث، Ajax، التأثيرات.

## الملخص

نستعرض في هذا الفصل مجموعة من المهارات التي توفرها JQuery.

## الأهداف التعليمية:

يتعرف الطالب في هذا الفصل على:

- التعامل مع عناصر الوثيقة.
- التعامل مع الأحداث.
- استخدام تقانات Ajax.
- التأثيرات.

## المخطط

يضم فصل مهارات JQuery 4 وحدات (Learning Objects) هي:

- التعامل مع عناصر الوثيقة
- التعامل مع الأحداث
- Ajax: Asynchronous JavaScript and XML
- التأثيرات Effects

## التعامل مع عناصر الوثيقة

### الأهداف التعليمية

- التعرف على مختلف طرق التعامل مع عناصر الوثيقة.

## التعامل مع نموذج كائن الوثيقة DOM

تُمكن المكتبة JQuery من التعامل مع عناصر نموذج كائن الوثيقة DOM بسهولة.

### التعامل مع المحتوى

تسمح الوظيفة `html()` بالحصول على المحتوى `html`.

مثال:

يُبين المثال التالي استخدام الطريقة `html()` للحصول على المحتوى، واستخدام الطريقة `text(val)` لإسناد قيمة للنص.

```
<!doctype html>
<html>
<head>
<title>the title</title>
<script type="text/javascript"
src="jquery/jquery-1.10.2.min.js"></script>
<script type="text/javascript" language="javascript">
$(document).ready(function() {
    $("div").click(function () {
        var content = $(this).html();
        $("#result").text( content );
    });
});
</script>
<style>
    #division{ margin:10px;padding:12px;
                border:2px solid #666;
                width:60px;
            }
</style>
</head>
<body>
<p>Click on the square below:</p>
<span id="result"> </span>
<div id="division" style="background-color:blue;">
    This is Blue Square!!
</div>
</body>
</html>
```

يُظهر المتصفح أولاً:

Click on the square below:



وبعد النقر على العنصر div:

Click on the square below:

This is Blue Square!!



## استبدال عناصر DOM

تسمح الوظيفة `replaceWith( content )` باستبدال عنصر بـ HTML أو بعناصر أخرى.

مثال:

يُبين المثال التالي استخدام الطريقة `replaceWith( content )`:

```
<!doctype html>
<html>
<head>
<title>the title</title>
  <script type="text/javascript"
    src="jquery/jquery-1.10.2.min.js"></script>
  <script type="text/javascript" language="javascript">
    $(document).ready(function() {
      $("div").click(function () {
        $(this).replaceWith("<h1>jQuery is Great</h1>");
      });
    });
  </script>
  <style>
    #division{ margin:10px;padding:12px;
               border:2px solid #666;
               width:60px;
            }
  </style>
</head>
<body>
  <p>Click on the square below:</p>
  <span id="result"> </span>
  <div id="division" style="background-color:blue;">
    This is Blue Square!!
  </div>
</body>
</html>
```

يُظهر المتصفح أولاً:

Click on the square below:



وبعد النقر على العنصر div:

Click on the square below:

## JQuery is Great

### حذف عناصر DOM

تسمح الوظيفة `remove( expr )` بحذف جميع العناصر المحددة.

مثال:

يُبين المثال التالي استخدام الطريقة `remove( expr )`:

```
<!doctype html>
<html>
<head>
<title>the title</title>
  <script type="text/javascript"
    src="jquery/jquery-1.10.2.min.js"></script>
  <script type="text/javascript" language="javascript">
    $(document).ready(function() {

      $("div").click(function () {
        $(this).remove( );
      });

    });
  </script>
  <style>
    .div{ margin:10px;padding:12px;
          border:2px solid #666;
          width:60px;
        }
  </style>
</head>
<body>
  <p>Click on any square below:</p>
  <span id="result"> </span>
  <div class="div" style="background-color:blue;"></div>
  <div class="div" style="background-color:green;"></div>
  <div class="div" style="background-color:red;"></div>
</body>
</html>
```

يُظهر المتصفح أولاً:

Click on any square below:



النقر على أي div سيؤدي إلى إخفائه:

Click on any square below:



### إدراج عناصر DOM

تسمح الوظيفة (content) after بإدراج محتوى بعد العناصر المحددة، والوظيفة (content) before بإدراج محتوى قبل العناصر المحددة.

مثال:

يُبين المثال التالي استخدام الطريقة (content) before:



```
<!doctype html>
<html>
<head>
<title>the title</title>
  <script type="text/javascript"
    src="jquery/jquery-1.10.2.min.js"></script>
  <script type="text/javascript" language="javascript">
    $(document).ready(function() {

      $("div").click(function () {
        $(this).before('<div class="div"></div>' );
      });

    });
  </script>
  <style>
    .div{ margin:10px;padding:12px;
          border:2px solid #666;
          width:60px;
        }
  </style>
</head>
<body>
  <p>Click on any square below:</p>
  <span id="result"> </span>
  <div class="div" style="background-color:blue;"></div>
  <div class="div" style="background-color:green;"></div>
  <div class="div" style="background-color:red;"></div>
</body>
</html>
```

يُظهر المتصفح أولاً:

Click on any square below:



النقر على أي div سيؤدي إلى ظهور div جديد قبله:

Click on any square below:



## طرق التعامل مع العناصر

يُبين الجدول التالي **طرق التعامل** مع عناصر الوثيقة:

للحصول على أمثلة إضافية يُمكن زيارة الرابط:

<http://www.tutorialspoint.com/jquery/jquery-dom.htm>

| Method                          | Description  |
|---------------------------------|--|
| <b>after( content )</b>         | Insert content after each of the matched elements.                               |
| <b>append( content )</b>        | Append content to the inside of every matched element.                           |
| <b>appendTo( selector )</b>     | Append all of the matched elements to another, specified, set of elements.       |
| <b>before( content )</b>        | Insert content before each of the matched elements.                              |
| <b>clone( bool )</b>            | Clone matched DOM Elements, and all their event handlers, and select the clones. |
| <b>clone( )</b>                 | Clone matched DOM Elements and select the clones.                                |
| <b>empty( )</b>                 | Remove all child nodes from the set of matched elements.                         |
| <b>html( val )</b>              | Set the html contents of every matched element.                                  |
| <b>html( )</b>                  | Get the html contents (innerHTML) of the first matched element.                  |
| <b>insertAfter( selector )</b>  | Insert all of the matched elements after another, specified, set of elements.    |
| <b>insertBefore( selector )</b> | Insert all of the matched elements before another, specified, set of elements.   |

|                               |  |
|-------------------------------|--|
| <b>prepend( content )</b>     | Prepend content to the inside of every matched element.  |
| <b>prependTo( selector )</b>  | Prepend all of the matched elements to another, specified, set of elements.                          |
| <b>remove( expr )</b>         | Removes all matched elements from the DOM.   |
| <b>replaceAll( selector )</b> | Replaces the elements matched by the specified selector with the matched elements.                   |
| <b>replaceWith( content )</b> | Replaces all matched elements with the specified HTML or DOM elements.                               |
| <b>text( val )</b>            | Set the text contents of all matched elements.   |
| <b>text( )</b>                | Get the combined text contents of all matched elements.  |
| <b>wrap( elem )</b>           | Wrap each matched element with the specified element.  |
| <b>wrap( html )</b>           | Wrap each matched element with the specified HTML content.   |
| <b>wrapAll( elem )</b>        | Wrap all the elements in the matched set into a single wrapper element.                              |
| <b>wrapAll( html )</b>        | Wrap all the elements in the matched set into a single wrapper element.                              |
| <b>wrapInner( elem )</b>      | Wrap the inner child contents of each matched element (including text nodes) with a DOM element.     |
| <b>wrapInner( html )</b>      | Wrap the inner child contents of each matched element (including text nodes) with an HTML structure. |

## التعامل مع الأحداث

### الأهداف التعليمية

- التعامل مع الأحداث المختلفة.

## التعامل مع الأحداث

تُمكن المكتبة JQuery من إنشاء صفحات ويب ديناميكية عن طريق استخدام الأحداث المختلفة مثل النقر وتحريك الفأرة فوق العناصر أو ضغط مفاتيح لوحة المفاتيح وغيرها. يُمكنك ربط وظيفة لتُنفذ عند وقوع الحدث الموافق. تُدعى هذه الوظيفة بمعالج الحدث Event Handler.

### ربط العناصر بمعالجات الأحداث

تسمح الوظيفة `bind()` بربط عنصر/حدث/وظيفة.

كما يُبين المثال التالي والذي يربط كل عنصر من النوع `div` بمعالج لحدث النقر `:click`:

```
$('div').bind('click', function (event) {
    alert('Hi there!');
});
```

ولها ثلاث معاملات:

- نوع الحدث `eventType`: اسم الحدث مثل `click` و `submit`.
  - بيانات الحدث `eventData`: معامل اختياري يحوي البيانات المراد تمريرها لمعالج الحدث.
  - معالج الحدث `handler`: وظيفة سيتم تنفيذها في كل مرة يقع الحدث.
- كما تسمح الوظيفة `unbind()` بإزالة معالج الحدث.

مثال:

يُبين المثال التالي استخدام الطريقة `bind()` لربط كل العناصر من النوع `div` مع معالج لحدث النقر `:click`.

```
<!doctype html>
<html>
<head>
  <title>the title</title>
  <script type="text/javascript"
    src="jquery/jquery-1.10.2.min.js"></script>
  <script type="text/javascript" language="javascript">
    $(document).ready(function () {
      $('div').bind('click', function (event) {
        alert('Hi there!');
      });
    });
  </script>
  <style>
    .div {
      margin: 10px;
      padding: 12px;
      border: 2px solid #666;
      width: 60px;
    }
  </style>
</head>
```

```
<body>
  <p>Click on any square below to see the result:</p>
  <div class="div" style="background-color:blue;">ONE</div>
  <div class="div" style="background-color:green;">TWO</div>
  <div class="div" style="background-color:red;">THREE</div>
</body>
</html>
```

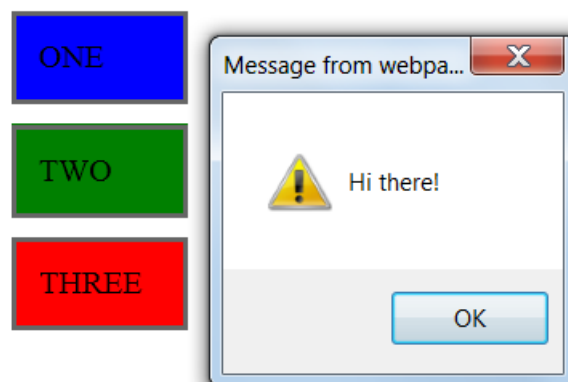
حيث يُظهر المتصفح:

Click on any square below to see the result:



وعند النقر على أي div تظهر الرسالة الموافقة.

Click on any square below to see the result:



أنواع الأحداث

يُبين الجدول التالي أنواع الأحداث:

| Event Type | الوصف   |
|------------|---|
| blur       | يحدث عندما يفقد العنصر التركيز.                 |
| change     | يحدث عندما يتغير العنصر.                        |
| click      | يحدث عند النقر بالفأرة على العنصر.              |
| dblclick   | يحدث عند النقر المزدوج بالفأرة على العنصر.      |
| error      | يحدث عند حصول خطأ ما (في التحميل مثلاً).        |
| focus      | يحدث عندما يصبح التركيز على العنصر.             |
| keydown    | يحدث عند ضغط مفتاح من لوحة المفاتيح.            |
| keypress   | يحدث عند ضغط مفتاح ورفع مفتاح من لوحة المفاتيح. |
| keyup      | يحدث عند رفع مفتاح من لوحة المفاتيح.            |
| load       | يحدث عند تحميل العنصر.                          |
| mousedown  | يحدث عند ضغط زر من أزرار الفأرة.                |
| mouseenter | يحدث عند دخول الفأرة في منطقة العنصر.           |
| mouseleave | يحدث عند خروج الفأرة من منطقة العنصر.           |
| mousemove  | يحدث عند تحريك الفأرة فوق منطقة العنصر.         |
| mouseout   | يحدث عند حركة الفأرة خروجاً من العنصر.          |
| mouseover  | يحدث عند حركة الفأرة فوق العنصر.                |
| mouseup    | يحدث عند تحرير زر الفأرة.                       |
| resize     | يحدث عند تغيير حجم النافذة.                     |
| scroll     | يحدث عند التمرير في النافذة.                    |
| select     | يحدث عند تحديد نص.                              |
| submit     | يحدث عند إرسال بيانات النموذج.                  |
| unload     | يحدث عند تفريغ النموذج.                         |

## الكائن Event

يكون لمعالج الحدث معامل وحيد<sup>1</sup> يُدعى event.

يوفر هذا الكائن مجموعة من الخصائص والطرق وبعض طرق معالجة الأحداث.

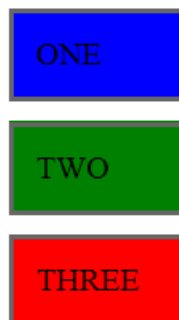
مثال:

يُبين المثال التالي استخدام بعض الخصائص للكائن event.

```
<!doctype html>
<html>
<head>
  <title>the title</title>
  <script type="text/javascript"
    src="jquery/jquery-1.10.2.min.js"></script>
  <script type="text/javascript" language="javascript">
$(document).ready(function() {
  $('div').bind('click', function( event ){
    alert('Event type is ' + event.type);
    alert('pageX : ' + event.pageX);
    alert('pageY : ' + event.pageY);
    alert('Target : ' + event.target.innerHTML);
  });
});
</script>
<style>
  .div{ margin:10px;padding:12px;
    border:2px solid #666;
    width:60px;
  }
</style>
</head>
<body>
  <p>Click on any square below to see the result:</p>
  <div class="div" style="background-color:blue;">ONE</div>
  <div class="div" style="background-color:green;">TWO</div>
  <div class="div" style="background-color:red;">THREE</div>
</body>
</html>
```

حيث يُظهر المتصفح:

Click on any square below to see the result:

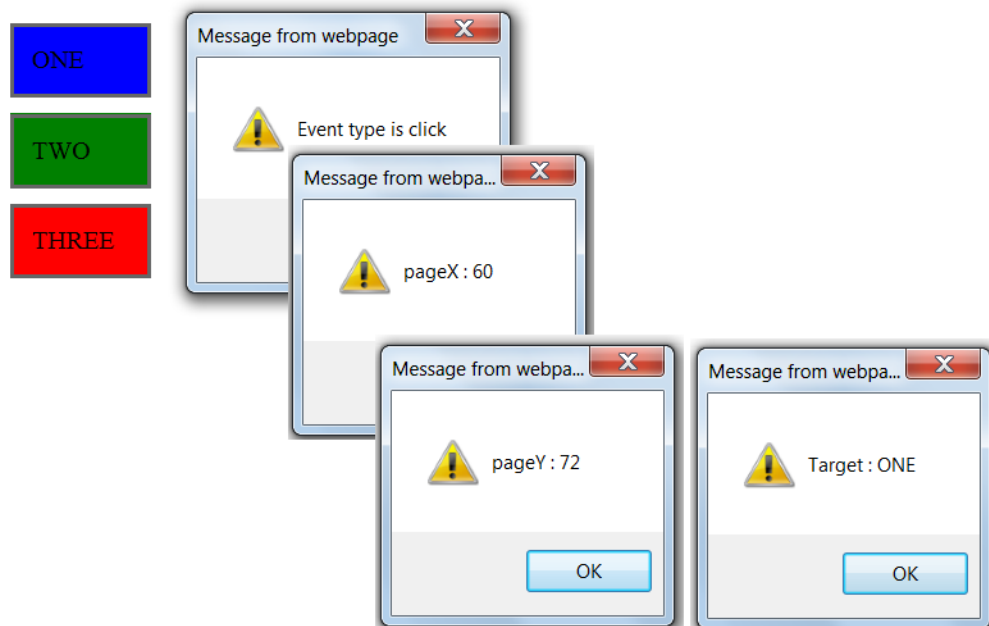


1 في الكثير من الأحيان لا يُستخدم



وعند النقر على أي div تظهر الرسائل الموافقة.

Click on any square below to see the result:



### الطرق المساعدة للأحداث Event Helper Methods

توفر المكتبة مجموعة من الطرق المساعدة التي يمكن أن تُستخدم لإطلاق حدث أو للربط مع أي نوع حدث من الأنواع السابقة.

### طرق إطلاق الأحداث Trigger Methods

يُبين المثال التالي كيفية إطلاق حدث الخروج blur لجميع الفقرات.

```
$("p").blur();
```

### طرق الربط Binding Methods

يُبين المثال التالي ربط حدث النقر click مع مقطع div:

```
$("div").click(function () {
    // do something here
});
```

مثال:

```

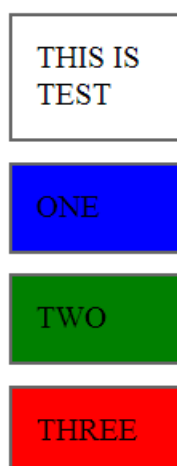
<!doctype html>
<html>
<head>
  <title>the title</title>
  <script type="text/javascript"
    src="jquery/jquery-1.10.2.min.js"></script>
  <script type="text/javascript" language="javascript">
    $(document).ready(function () {

      $("div").click(function () {
        var content = "<b></b>";
        $(this).wrapInner(content);
      });
    });
  </script>
  <style>
    .div {
      margin: 10px;
      padding: 12px;
      border: 2px solid #666;
      width: 60px;
    }
  </style>
</head>
<body>
  <p>Click on any square below to see the result:</p>
  <div class="div" id="destination">THIS IS TEST</div>
  <div class="div" style="background-color:blue;">ONE</div>
  <div class="div" style="background-color:green;">TWO</div>
  <div class="div" style="background-color:red;">THREE</div>
</body>
</html>

```

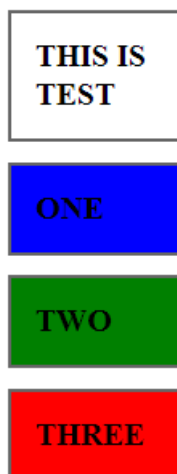
يُظهر المتصفح

Click on any square below to see the result:



وعند النقر على أي مقطع سنلاحظ احتواء النص الداخلي:

Click on any square below to see the result:



### خصائص الحدث

يُبين الجدول التالي **خصائص الحدث**:

للحصول على أمثلة إضافة يُمكن زيارة الرابط:

<http://www.tutorialspoint.com/jquery/jquery-events.htm>

| Property       | Description  |
|----------------|--|
| <b>altKey</b>  | Set to true if the Alt key was pressed when the event was triggered, false if not. The Alt key is labeled Option on most Mac keyboards.              |
| <b>ctrlKey</b> | Set to true if the Ctrl key was pressed when the event was triggered, false if not.  |
| <b>data</b>    | The value, if any, passed as the second parameter to the bind() command when the handler was established.  |
| <b>keyCode</b> | For keyup and keydown events, this returns the key that was pressed.   |
| <b>metaKey</b> | Set to true if the Meta key was pressed when the event was triggered, false if not. The Meta key is the Ctrl key on PCs and the Command key on Macs. |

|                      |   |
|----------------------|---|
| <b>pageX</b>         | For mouse events, specifies the horizontal coordinate of the event relative from the page origin.   |
| <b>pageY</b>         | For mouse events, specifies the vertical coordinate of the event relative from the page origin.   |
| <b>relatedTarget</b> | For some mouse events, identifies the element that the cursor left or entered when the event was triggered.   |
| <b>screenX</b>       | For mouse events, specifies the horizontal coordinate of the event relative from the screen origin.   |
| <b>screenY</b>       | For mouse events, specifies the vertical coordinate of the event relative from the screen origin.   |
| <b>shiftKey</b>      | Set to true if the Shift key was pressed when the event was triggered, false if not.  |
| <b>target</b>        | Identifies the element for which the event was triggered.   |
| <b>timeStamp</b>     | The timestamp (in milliseconds) when the event was created.   |
| <b>type</b>          | For all events, specifies the type of event that was triggered (for example, click).  |
| <b>which</b>         | For keyboard events, specifies the numeric code for the key that caused the event, and for mouse events, specifies which button was pressed (1 for left, 2 for middle, 3 for right) |

## طرق الحدث

يُبين الجدول التالي **طرق الحدث**:

للحصول على أمثلة إضافة يُمكن زيارة الرابط:

<http://www.tutorialspoint.com/jquery/jquery-events.htm>

| Method                                 | Description   |
|--|---|
| <b>preventDefault()</b>                | Prevents the browser from executing the default action.   |
| <b>isDefaultPrevented()</b>            | Returns whether event.preventDefault() was ever called on this event object.  |
| <b>stopPropagation()</b>               | Stops the bubbling of an event to parent elements, preventing any parent handlers from being notified of the event. |
| <b>isPropagationStopped()</b>          | Returns whether event.stopPropagation() was ever called on this event object.                                       |
| <b>stopImmediatePropagation()</b>      | Stops the rest of the handlers from being executed.   |
| <b>isImmediatePropagationStopped()</b> | Returns whether event.stopImmediatePropagation() was ever called on this event object.                              |

## طرق التعامل مع الأحداث

يُبين الجدول التالي طرق **التعامل مع الأحداث**:

للحصول على أمثلة إضافة يُمكن زيارة الرابط:

<http://www.tutorialspoint.com/jquery/jquery-events.htm>

| Method                                 | Description   |
|--|---|
| <b>bind( type, [data], fn )</b>        | Binds a handler to one or more events (like click) for each matched element. Can also bind custom events.             |
| <b>die( type, fn )</b>                 | This does the opposite of live, it removes a bound live event.  |
| <b>hover( over, out )</b>              | Simulates hovering for example moving the mouse on, and off, an object.   |
| <b>live( type, fn )</b>                | Binds a handler to an event (like click) for all current – and future – matched element. Can also bind custom events. |
| <b>one( type, [data], fn )</b>         | Binds a handler to one or more events to be executed once for each matched element.                                   |
| <b>ready( fn )</b>                     | Binds a function to be executed whenever the DOM is ready to be traversed and manipulated.                            |
| <b>Toggle ( fn, fn2, fn3,... )</b>     | Toggle among two or more function calls every other click.  |
| <b>trigger( event, [data] )</b>        | Trigger an event on every matched element.  |
| <b>triggerHandler( event, [data] )</b> | Triggers all bound event handlers on an element.  |
| <b>unbind( [type], [fn] )</b>          | This does the opposite of bind, it removes bound events from each of the matched elements.                            |

## Ajax: Asynchronous JavaScript and XML

### الأهداف التعليمية

- استخدام تقانات Ajax.

## التعامل مع تقنية Ajax

تحتوي المكتبة JQuery الكثير من الأدوات للتعامل مع تقنية Ajax<sup>2</sup> والتي تسمح بتحميل بيانات من المخدم بدون عملية إنعاش refresh للصفحة.

### تحميل البيانات

تسمح الطريقة load بتحميل البيانات ولها الشكل العام:

```
[selector].load( URL, [data], [callback] );
```

حيث:

- URL عنوان المخدم الذي سيُرسل الطلب له (صفحة مثلاً).
- data معامل اختياري. البيانات المرة للطلب.
- callback معامل اختياري. وظيفة سيتم تنفيذها بعد تحميل النتائج.

مثال:

```
<!doctype html>
<html>
<head>
  <title>the title</title>
  <script type="text/javascript"
    src="jquery/jquery-1.10.2.min.js"></script>
  <script type="text/javascript" language="javascript">
$(document).ready(function() {
  $("#driver").click(function(event){
    $('#stage').load('jquery/result.html');
  });
});
</script>
</head>
<body>
  <p>Click on the button to load result.html file:</p>
  <div id="stage" style="background-color:blue;">
    STAGE
  </div>
  <input type="button" id="driver" value="Load Data" />
</body></html>
```

يحتوي الملف result.html:

```
<h1>THIS IS RESULT...</h1>
```



يُظهر المتصفح أولاً:

Click on the button to load result.html file:



وبعد النقر على الزر:

Click on the button to load result.html file:



## إرسال البيانات إلى المخدم

يُبين المثال التالي استخدام الطريقة load() مع إرسال بيانات إلى المخدم:

```
<!DOCTYPE html>
<html>
<head>
  <title>the title</title>
  <script src="Scripts/jquery-1.10.2.js"></script>
  <script type="text/javascript" language="javascript">
    $(document).ready(function () {
      $("#driver").click(function (event) {
        var name = $("#name").val();
        $("#stage").load('result.aspx', { "name": name });
      });
    });
  </script>
</head>
<body>

  <p>Enter your name and click on the button:</p>
  <input type="input" id="name" size="40" /><br />
  <div id="stage" style="background-color:green;">
    STAGE
  </div>
  <input type="button" id="driver" value="Show Result" />
</body></html>
```

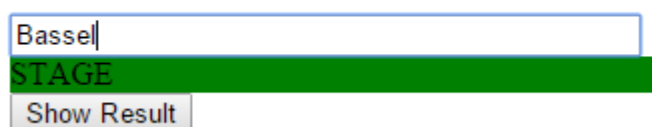
يحتوي ملف المخدم result.aspx التعليمات التالية:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

public partial class result : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        string v;
        v= Request.Params["name"];
        if (v != null)
        {
            Response.Write("Welcome ");
            Response.Write(v);
        }
    }
}
```

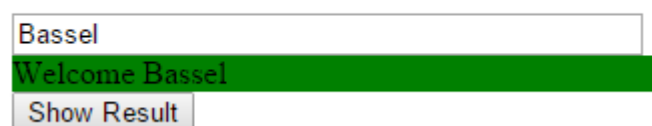
يُظهر المتصفح أولاً:

Enter your name and click on the button:



وبعد النقر على الزر:

Enter your name and click on the button:



## الحصول على بيانات JSON

تُستخدم الوظيفة `getJSON()` عندما يقوم المخدم بإرجاع البيانات كسلسلة <sup>3</sup>JSON.

تأخذ هذه الوظيفة الشكل العام التالي:

```
[selector].getJSON( URL, [data], [callback] );
```

حيث:

- URL عنوان المخدم الذي سيُرسل الطلب له (صفحة مثلاً).
- data معامل اختياري. البيانات المُررة للطلب.
- callback معامل اختياري. وظيفة سيتم تنفيذها بعد تحميل النتائج وحيث يكون المعامل الأول لها البيانات المعادة من المخدم JSON.

مثال:

ليكن لدينا صفحة المخدم json.aspx والتي تُعيد بيانات بشكل json:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.Script.Serialization;
public partial class json : System.Web.UI.Page
{
    class Data
    {
        public string Name { set; get; }
        public string Age { set; get; }
        public string Sex { set; get; }
    }
    protected void Page_Load(object sender, EventArgs e)
    {
        List<Data> data = new List<Data>();
        data.Add(new Data()
        {
            Name = "Bassel",
            Age = "45",
            Sex = "Male"
        });
        data.Add(new Data()
        {
            Name = "Sami",
            Age = "34",
            Sex = "Male"
        });
        var json = new JavaScriptSerializer().Serialize(data);
        Response.Clear();
        Response.ContentType = "application/json; charset=utf-8";
        Response.Write(json);
        Response.End();
    }
}
```

يُمكن استخدام الطريقة `getJSON()` لإظهار البيانات:

```
<!DOCTYPE html>
<html>
<head>
  <title>JSON</title>
  <script src="Scripts/jquery-1.10.2.js"></script>
  <script type="text/javascript" language="javascript">
    $(document).ready(function () {
      $("#driver").click(function (event) {
        $.getJSON('json.aspx', function (jd) {
          $('#stage').html('<p>' + jd.length + '</p>');
          for (var i = 0; i < jd.length; i++) {
            $('#stage').append('<p> Name: ' + jd[i].Name + '</p>');
            $('#stage').append('<p> Age : ' + jd[i].Age + '</p>');
            $('#stage').append('<p> Sex: ' + jd[i].Sex + '</p>');
          }
        });
      });
    });
  </script>
</head>
<body>
  <p>Click on the button to load json data:</p>
  <div id="stage" style="background-color:blue;">
    STAGE
  </div>
  <input type="button" id="driver" value="Load Data" />
</body></html>
```

يُظهر المتصفح أولاً:

Click on the button to load json data:



STAGE

Load Data

وبعد النقر على الزر:

Click on the button to load json data:



2

Name: Bassel

Age : 45

Sex: Male

Name: Sami

Age : 34

Sex: Male

Load Data

## وظائف JQuery AJAX

يُبين الجدول التالي **طرق التعامل** مع الأحداث:

للحصول على أمثلة إضافية يُمكن زيارة الرابط:

<http://www.tutorialspoint.com/jquery/jquery-ajax.htm>

| Methods and Description   |
|---|
| <b>jQuery.ajax( options )</b><br>Load a remote page using an HTTP request.  |
| <b>jQuery.ajaxSetup( options )</b><br>Setup global settings for AJAX requests.  |
| <b>jQuery.get (url, [data], [callback], [type])</b><br>Load a remote page using an HTTP GET request.  |
| <b>jQuerygetJSON (url, [data], [callback] )</b><br>Load JSON data using an HTTP GET request.  |
| <b>jQuery.getScript (url, [callback])</b><br>Loads and executes a JavaScript file using an HTTP GET request.  |
| <b>jQuery.post (url, [data], [callback], [type])</b><br>Load a remote page using an HTTP POST request.  |
| <b>load (url, [data], [callback])</b><br>Load HTML from a remote file and inject it into the DOM.   |
| <b>serialize( )</b><br>Serializes a set of input elements into a string of data.  |
| <b>serializeArray( )</b><br>Serializes all forms and form elements like the .serialize() method but returns a JSON data structure for you to work with. |

## أحداث JQuery AJAX

يُبين الجدول التالي **طرق التعامل** مع الأحداث:  
للحصول على أمثلة إضافة يُمكن زيارة الرابط:

<http://www.tutorialspoint.com/jquery/jquery-ajax.htm>

| Methods and Description   |
|---|
| <b>ajaxComplete (callback )</b><br>Attach a function to be executed whenever an AJAX request completes.                           |
| <b>ajaxStart (callback)</b><br>Attach a function to be executed whenever an AJAX request begins and there is none already active. |
| <b>ajaxError (callback)</b><br>Attach a function to be executed whenever an AJAX request fails.                                   |
| <b>ajaxSend (callback)</b><br>Attach a function to be executed before an AJAX request is sent.                                    |
| <b>ajaxStop (callback)</b><br>Attach a function to be executed whenever all AJAX requests have ended.                             |
| <b>ajaxSuccess (callback)</b><br>Attach a function to be executed whenever an AJAX request completes successfully.                |

## التأثيرات Effects

### الأهداف التعليمية

- استخدام التأثيرات المختلفة.



## التأثيرات

تُمكن المكتبة JQuery من القيام بعمليات تحريك وإحياء ممتعة.

### إظهار وإخفاء العناصر

تسمح الوظيفة `show()` بإظهار العناصر كما تسمح الوظيفة `hide()` بإخفائها.

لكلا الوظيفتين المعاملين التاليين:

- **Speed** ويأخذ قيمة رقمية بالميلي ثانية أو أحد القيم النصية: ("slow", "normal", "fast")
- **callback** وهو معامل اختياري يُحدد وظيفة للتنفيذ بعد كل عملية انتهاء للتأثير.

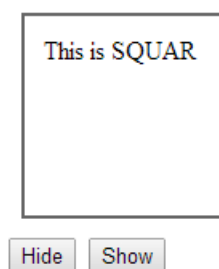
مثال:

```
<!doctype html>
<html>
<head>
<title>the title</title>
  <script type="text/javascript"
    src="jquery/jquery-1.10.2.min.js"></script>
  <script type="text/javascript" language="javascript">

    $(document).ready(function() {

      $("#show").click(function () {
        $(".mydiv").show( 1000 );
      });
      $("#hide").click(function () {
        $(".mydiv").hide( 1000 );
      });
    });
  </script>
  <style>
    .mydiv{ margin:10px;padding:12px;
      border:2px solid #666;
      width:100px;
      height:100px;
    }
  </style>
</head>
<body>
  <div class="mydiv">
    This is SQUAR
  </div>
  <input id="hide" type="button" value="Hide" />
  <input id="show" type="button" value="Show" />
</body>
</html>
```

يُظهر المتصفح أولاً:



وبعد النقر على الزر hide يختفي العنصر :div



## قلب العناصر

تسمح الوظيفة toggle() بقلب حالة ظهور العنصر .

للوظيفة المعاملين التاليين:

- Speed وهو معامل اختياري ويأخذ قيمة رقمية بالملي ثانية أو أحد القيم النصية ("slow", "normal", "fast" or "fast").
- callback وهو معامل اختياري يُحدد وظيفة للتنفيذ بعد كل عملية انتهاء للحركة.

مثال:

```
<!doctype html>
<html>
<head>
<title>the title</title>
<script type="text/javascript"
src="jquery/jquery-1.10.2.min.js"></script>
<script type="text/javascript" language="javascript">
$(document).ready(function() {
    $(".clickme").click(function(event){
        $(".target").toggle('slow', function(){
            $(".log").text('Transition Complete');
        });
    });
});
</script>
<style>
.clickme{ margin:10px;padding:12px;
border:2px solid #666;
width:100px;
height:50px;
}
</style>
</head>
<body>
<div class="content">
<div class="clickme">Click Me</div>
```

```
<div class="target">
  
</div>
<div class="log"></div>
</body>
</html>
```

يُظهر المتصفح أولاً:

Click Me



وبعد النقر على المقطع div الأول يختفي المقطع الثاني:

Click Me

Transition Complete

عند إعادة النقر على المقطع div الأول يظهر المقطع الثاني:

Click Me



Transition Complete

وهكذا دواليك.

## طرق التأثيرات JQuery

يُبين الجدول التالي **طرق التأثيرات** في المكتبة JQuery.

للحصول على أمثلة إضافية يُمكن زيارة الرابط:

<http://www.tutorialspoint.com/jquery/jquery-effects.htm>

| Methods and Description  |
|--|
| <code>animate( params, [duration, easing, callback] )</code><br>A function for making custom animations.   |
| <code>fadeIn( speed, [callback] )</code><br>Fade in all matched elements by adjusting their opacity and firing an optional callback after completion.  |
| <code>fadeOut( speed, [callback] )</code><br>Fade out all matched elements by adjusting their opacity to 0, then setting display to "none" and firing an optional callback after completion. |
| <code>fadeTo( speed, opacity, callback )</code><br>Fade the opacity of all matched elements to a specified opacity and firing an optional callback after completion.                         |
| <code>hide( )</code><br>Hides each of the set of matched elements if they are shown.   |
| <code>hide( speed, [callback] )</code><br>Hide all matched elements using a graceful animation and firing an optional callback after completion.   |
| <code>show( )</code><br>Displays each of the set of matched elements if they are hidden.   |
| <code>show( speed, [callback] )</code><br>Show all matched elements using a graceful animation and firing an optional callback after completion.   |
| <code>slideDown( speed, [callback] )</code><br>Reveal all matched elements by adjusting their height and firing an optional  |

|   |
|---|
| callback after completion.  |
| <p><code>slideToggle( speed, [callback] )</code></p> <p>Toggle the visibility of all matched elements by adjusting their height and firing an optional callback after completion.</p>     |
| <p><code>slideUp( speed, [callback] )</code></p> <p>Hide all matched elements by adjusting their height and firing an optional callback after completion.</p>                             |
| <p><code>stop( [clearQueue, gotoEnd ] )</code></p> <p>Stops all the currently running animations on all the specified elements.</p>   |
| <p><code>toggle( )</code></p> <p>Toggle displaying each of the set of matched elements.</p>   |
| <p><code>toggle( speed, [callback] )</code></p> <p>Toggle displaying each of the set of matched elements using a graceful animation and firing an optional callback after completion.</p> |
| <p><code>toggle( switch )</code></p> <p>Toggle displaying each of the set of matched elements based upon the switch (true shows all elements, false hides all elements).</p>              |
| <p><code>jQuery.fx.off</code></p> <p>Globally disable all animations.</p>   |

## مكتبة واجهة المستخدم UI

توفر هذه **المكتبة** مجموعة إضافية من التأثيرات المخصصة والتي نعرضها في الجدول المرجع التالي:  
للحصول على أمثلة إضافية يُمكن زيارة الرابط:

<http://www.tutorialspoint.com/jquery/jquery-effects.htm>

يجب تنزيل المكتبة من الرابط:

<http://jqueryui.com/download/>

| Methods and Description   |
|---|
| <p><b>Blind</b></p> <p>Blinds the element away or shows it by blinding it in.</p>   |
| <p><b>Bounce</b></p> <p>Bounces the element vertically or horizontally n–times.</p> |
| <p><b>Clip</b></p> <p>Clips the element on or off, vertically or horizontally.</p>  |
| <p><b>Drop</b></p> <p>Drops the element away or shows it by dropping it in.</p>     |
| <p><b>Explode</b></p> <p>Explodes the element into multiple pieces.</p>             |
| <p><b>Fold</b></p> <p>Folds the element like a piece of paper.</p>                  |
| <p><b>Highlight</b></p> <p>Highlights the background with a defined color.</p>      |
| <p><b>Puff</b></p> <p>Scale and fade out animations create the puff effect.</p>     |
| <p><b>Pulsate</b></p> <p>Pulsates the opacity of the element multiple times.</p>    |
| <p><b>Scale</b></p> <p>Shrink or grow an element by a percentage factor.</p>        |
| <p><b>Shake</b></p> <p>Shakes the element vertically or horizontally n–times.</p>   |
| <p><b>Size</b></p> <p>Resize an element to a specified width and height.</p>        |
| <p><b>Slide</b></p>   |

Slides the element out of the viewport.

**Transfer**

Transfers the outline of an element to another.

## اقتراحات وتمارين

1. قم بالدخول إلى الرابط التالي وحل جميع التمارين التفاعلية الموجودة:

<http://jqexercise.droppages.com/>

2. قم بالدخول إلى الرابط التالي وحل جميع التمارين:

<http://courses.cs.washington.edu/courses/cse190m/12su/sections/section07.shtml#>