

القسم الأول- التقنيات من جهة الزبون Client Side

الفصل السادس- أساسيات JQuery

3.....	ماهي JQuery
3.....	تنزيل المكتبة
4.....	استخدام مكتبة JQuery
4.....	استخدام المكتبة من رابط عام CDN
5.....	استدعاء وظائف JQuery
6.....	استخدام خطاطات مخصصة
7.....	المحددات SELECTORS
8.....	مثال: استخدام محدد المؤثر
9.....	مثال: استخدام محدد الصف
11.....	مثال: تحديد جميع العناصر *
13.....	التعامل مع الواصفات
13.....	قراءة/إسناد قيمة واصفة
14.....	تطبيق الأنماط
14.....	بعض طرق الواصفات
17.....	HTML و TEXT
18.....	أسلوب الصفحات المتتالي
18.....	التعامل مع خصائص CSS
19.....	تطبيق أكثر من خاصية CSS
20.....	تغيير عرض وارتفاع العناصر
21.....	وظائف JQuery CSS
22.....	التعامل مع نموذج كائن الوثيقة DOM
22.....	التعامل مع المحتوى
23.....	استبدال عناصر DOM
24.....	حذف عناصر DOM
26.....	إدراج عناصر DOM
27.....	طرق التعامل مع العناصر

ماهي JQuery

هي مكتبة تحوي مجموعة من الوظائف المكتوبة بلغة JavaScript. قام ببنائها John Resig عام 2006 بهدف: الكتابة أقل، الفعل أكثر Write less, do more. مكتبة مجانية وفي عام 2021 تم استخدامها في 77.8% من أشهر 10 ملايين موقع حول العالم.

تُسهّل هذه المكتبة التعامل مع مختلف مكونات صفحة HTML. من أهم الميزات التي تقدمها هذه المكتبة:

- التعامل مع العناصر DOM manipulation: تُسهّل الوصول إلى مختلف العناصر وتعديل المحتوى باستخدام محرك تحديد مفتوح المصدر يُدعى Sizzle.
- معالجة الأحداث Event handling: تُقدّم طرق أنيقة لالتقاط أحداث المستخدم مثل النقر على رابط مثلاً، دون الحاجة لخلط كود HTML نفسه مع معالجات الأحداث.
- دعم Ajax support: تُساعد في تطوير صفحات غنية وتفاعلية باستخدام تكنولوجيا Ajax.
- الإحياء Animations: توفر مجموعة من إمكانيات الإحياء الجاهزة.
- الخفة Lightweight: لايتجاوز حجم المكتبة 200K.
- دعم المتصفحات Cross Browser Support: تعمل على جميع المتصفحات المشهورة.
- أحدث التكنولوجيا Latest Technology: تدعم التكنولوجيا الحديثة مثل CSS3 و XPath.
- التعامل مع JSON.

تنزيل المكتبة

يُمكن تنزيل إصدار المكتبة الأخير مجاناً من الرابط jquery.com.

(jquery-x.x.x.min.js).

سنستخدم في الأمثلة النسخة jquery-3.6.0.min.js

استخدام مكتبة JQuery

يمكنك تضمين المكتبة في صفحة HTML كما يلي:

```
<!DOCTYPE HTML>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <title>The jQuery Example</title>
  <script type="text/javascript"
    src="jquery/jquery-3.6.0.min.js"></script>
  <script type="text/javascript">
    // you can add our javascript code here
  </script>
</head>
<body>
  .....
</body>
</html>
```

استخدام المكتبة من رابط عام CDN

يمكن تضمين المكتبة من خلال موقع مشاركة عام (CDN (Content Delivery Network كما يلي:

```
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js">
</script>
```

أو

```
<script src="https://code.jquery.com/jquery-3.6.0.js"
  integrity="sha256-H+K7U5CnX11h5ywQfKtSj8PCmoN9aaq30gDh27Xc0jk="
  crossorigin="anonymous">
</script>
```

استدعاء وظائف JQuery

بما أن معظم ما نفعله عند استخدام JQuery هو التعامل مع نموذج كائن الوثيقة (DOM)، فإننا بحاجة للتأكد من أننا نُضيف الأحداث عند جاهزية هذا النموذج.

إذا أردت أن يعمل حدث في الصفحة، فيجب استدعائه داخل `$(document).ready()`.
كل شيء داخلها سوف يُحمّل حال تحميل عناصر الوثيقة.
نقوم باستخدام حدث جاهزية الوثيقة كما يلي:

```
$(document).ready(function () {
    // do stuff when DOM is ready
});
```

في المثال التالي نقوم بتسجيل حدث النقر على أي مقطع div في الصفحة. سيؤدي النقر إلى ظهور رسالة ترحيب.

```
<!DOCTYPE HTML>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <title>The jQuery Example</title>
  <script type="text/javascript"
    src="jquery/jquery-3.6.0.min.js"></script>
  <script type="text/javascript" language="javascript">
    $(document).ready(function () {
      $("div").click(function () {
        alert("Hello world!");
      });
    });
  </script>
</head>
<body>
  <div >
    Click on this to see a dialogue box.
  </div>

  <div >
    Click also on this to see a dialogue box.
  </div>

</body>
</html>
```

يُظهر المتصفح عند النقر الرسالة الموافقة:

Click on this to see a dialogue box.
Click also on this to see a dialogue box.



استخدام خطوط مخصصة

من الأفضل كتابة تعليماتك المخصصة في ملف جافا. مثلاً:

```
/* Filename: custom.js */
$(document).ready(function() {
    $("div").click(function() {
        alert("Hello world!");
    });
});
```

ومن ثم تضمين الملف في الصفحة:

```
<!DOCTYPE HTML>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>The jQuery Example</title>
    <script type="text/javascript"
        src="jquery/jquery-3.6.0.min.js"></script>
    <script type="text/javascript"
        src="jquery/custom.js"></script>
</head>
<body>
    <div>
        Click on this to see a dialogue box.
    </div>

    <div>
        Click also on this to see a dialogue box.
    </div>

</body>
</html>
```

المحددات Selectors

توفر مكتبة JQuery إمكانية الوصول للعناصر في نموذج كائن الوثيقة (DOM) بسرعة وسهولة.

يبدأ المحدد بـ `$()` ويمكن أن يستخدم:

- اسم المؤثر Tag Name: مثلاً `$('p')` يُحدّد جميع الفقرات في المستند.
- معرف مؤثر Tag ID: مثلاً `$('#some-id')` يُحدد العنصر الوحيد الذي له المعرف **some-id**.

- صف مؤثر Tag Class: مثلاً `$('.some-class')` يُحدد جميع المؤثرات من الصف **some-class**.

تُبين الأمثلة التالية أهم إمكانيات التحديد:

<code>\$('*')</code>	تحديد جميع عناصر الوثيقة
<code>\$('p')</code>	تحديد جميع الفقرات
<code>\$('div, p')</code>	تحديد العناصر من النوع p أو div
<code>\$("p > *")</code>	تحديد جميع العناصر أبناء فقرة
<code>\$("#specialID")</code>	تحديد عنصر له المعرف المحدد specialID
<code>\$('div#yourid')</code>	تحديد عنصر من النوع div له المعرف yourid
<code>\$(".specialClass")</code>	تحديد العناصر من الصف المحدد specialClass
<code>\$("li:not(.myclass)")</code>	تحديد جميع العناصر من النوع li والتي ليست من الصف myclass
<code>\$("a#specialID.specialClass")</code>	تحديد الروابط لها المعرف specialID والصف specialClass
<code>\$("p a.specialClass")</code>	تحديد الروابط من الصف specialClass ضمن فقرة
<code>\$("ul li:first")</code>	تحديد العنصر الأول فقط من قائمة ul

\$("#container p")	تحديد الفقرات الموجودة تحت عنصر له المعرف container
\$("#li > ul")	تحديد جميع العناصر ul الموجودة تحت li
\$("#li:eq(2)")	تحديد العنصر الثالث في قائمة li
\$("#li:lt(2)")	تحديد العناصر قبل العنصر الثالث في قائمة li
\$("#li:gt(1)")	تحديد العناصر بعد العنصر الثاني في قائمة li
\$("#li:even")	تحديد العناصر ذات الفهرس الزوجي في قائمة li
\$("#li:odd")	تحديد العناصر ذات الفهرس الفردي في قائمة li
\$("#li:visible")	تحديد العناصر الظاهرة li
\$("#li:hidden")	تحديد العناصر المخفية li
\$("#:radio")	تحديد جميع أزرار الراديو في نموذج
\$("#:checked")	تحديد جميع صناديق التحقق التي لها القيمة checked
\$("#:text")	تحديد صناديق النص
\$("#div, p")	تحديد جميع العناصر من النوع div أو من النوع p
\$("#p.small")	تحديد الفقرات من الصف small
\$("#a[@href*=domain.com]")	تحديد جميع الروابط التي تحوي domain.com
\$("#input[@name^=myname]")	تحديد العناصر من النوع input والتي اسمها يبدأ بـ myname
\$("#input[@name=myname]")	تحديد العناصر من النوع input والتي اسمها يساوي myname

مثال: استخدام محدد المؤثر

عند تنفيذ المثال التالي، ستظهر 3 رسائل على التوالي تُظهر محتوى الفقرات.

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <title>The jQuery Example</title>
  <script type="text/javascript">
```



```

        src="jquery/jquery-3.6.0.min.js"></script>
<script type="text/javascript" language="javascript">
    $(document).ready(function () {
        var pars = $("p");
        for (i = 0; i < pars.length; i++) {
            alert("Found paragraph: " + pars[i].innerHTML);
        }
    });
</script>
</head>
<body>
    <div>
        <p class="myclass">This is a paragraph.</p>
        <p id="myid">This is second paragraph.</p>
        <p>This is third paragraph.</p>
    </div>
</body> </html>

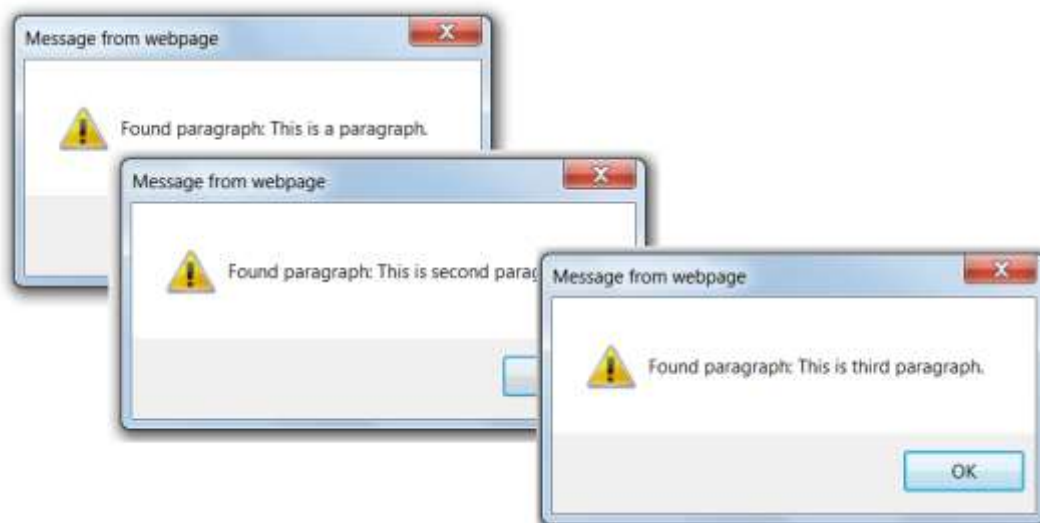
```

حيث يُظهر المتصفح:

This is a paragraph.

This is second paragraph.

This is third paragraph.



مثال: استخدام محدد الصف

نقوم في المثال التالي بتعديل جميع العناصر من الصف small لتصبح من الصف big.

```

<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>The jQuery Example</title>
    <style type="text/css">
        .small {
            font-size: small;

```

```

        color:green ;
    }
    .medium {
        font-size: medium;
        color: blue;
    }
    .big {
        font-size: larger;
        color: red;
    }
</style>

<script type="text/javascript"
    src="jquery/jquery-3.6.0.min.js"></script>
<script type="text/javascript" language="javascript">
    $(document).ready(function () {
        $('.small').toggleClass('big');
    })
</script>
</head>
<body>

    <div class="small" id="divid1">
        <p class="para1" id="pid1">This is first paragraph.</p>
        <p class="para2" id="pid2">This is second paragraph.</p>
        <p class="para3" id="pid3">This is third paragraph.</p>
    </div>
    <br />

    <div class="medium" id="divid2">
        <p>This is second division of the DOM.</p>
        <p>This is second para inside second division.</p>
    </div>
    <br />

    <div class="big" id="divid3">
        <p>This is a para inside third division</p>
    </div>

</body></html>

```

ويُظهر المتصفح:

This is first paragraph.

This is second paragraph.

This is third paragraph.

This is second division of the DOM.

This is second para inside second division.

This is a para inside third division

مثال: تحديد جميع العناصر *

نقوم في المثال التالي بإضافة الصف big إلى جميع عناصر الوثيقة:

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <title>The jQuery Example</title>

  <script type="text/javascript"
    src="jquery/jquery-3.6.0.min.js"></script>
  <script type="text/javascript" language="javascript">
    $(document).ready(function () {
      $(".*").addClass("big");
    })
  </script>
  <style type="text/css">
    .big {
      font-size: larger;
      color: red;
    }
  </style>
</head>
<body>
  <p >This is first paragraph.</p>
  <p >This is second paragraph.</p>
  <p >This is third paragraph.</p>
</body></html>
```

حيث يُظهر المتصفح:

This is first paragraph.

This is second paragraph.

This is third paragraph.

التعامل مع الواصفات

تُمكن مكتبة JQuery من التعامل مع الواصفات بسهولة.
لنتذكر أولاً أنه في المثال البسيط التالي:

```

```

يكون اسم المؤثر `img` وتكون الواصفات هي `id`, `src`, `alt`, `class`. كل واصفة هي زوج مفتاح/قيمة.

قراءة/إسناد قيمة واصفة

تسمح الطريقة `attr()` بإيجاد قيمة واصفة العنصر الأول في مجموعة من العناصر المحددة. أو بإسناد قيمة إلى جميع العناصر المحددة.

مثال: نقوم في المثال التالي بإيجاد قيمة المؤثر `em` وإسناد هذه القيمة إلى نص `div` له المعرف `divid`.

```
<!doctype html>
<html>
<head>
<title>the title</title>
<script type="text/javascript"
src="jquery/jquery-3.6.0.min.js"></script>
<script type="text/javascript" language="javascript">

$(document).ready(function() {
  var title = $("em").attr("title");
  $("#divid").text(title);
});

</script>
</head>
<body>
<div>
  <em title="Bold and Brave">This is first paragraph.</em>
  <p id="myid">This is second paragraph.</p>
  <div id="divid"></div>
</div>
</body>
</html>
```

يُظهر المتصفح:

This is first paragraph.

This is second paragraph.

Bold and Brave

تطبيق الأنماط

يُمكن استخدام الطريقة `addClass(classes)` لتطبيق نمط معرف على العناصر المحددة.

مثال:

```
<!doctype html>
<html>
<head>
<title>the title</title>
<script type="text/javascript"
src="jquery/jquery-3.6.0.min.js"></script>
<script type="text/javascript" language="javascript">
$(document).ready(function() {
    $(".em").addClass("selected");
    $("#myid").addClass("highlight");
});
</script>
<style>
    .selected { color:red; }
    .highlight { background:yellow; }
</style>
</head>
<body>
    <em title="Bold and Brave">This is first paragraph.</em>
    <p id="myid">This is second paragraph.</p>
</body>
</html>
```

يُظهر المتصفح

This is first paragraph.

This is second paragraph.

بعض طرق الواصفات

يُبين الجدول التالي بعض الطرق المفيدة التي تسمح بالتعامل مع الواصفات.

الوصف	الطريقة
تعيد قيمة الواصفة للعنصر المحدد.	<code>attr(properties)</code>
تسمح بإسناد قيمة لواصفة.	<code>attr(key, fn)</code>

removeAttr(name)	حذف واصفة معينة من كل العناصر المحددة.
hasClass(class)	تُعيد true إذا كان لأحد العناصر المحددة الصف .class
removeClass(class)	حذف الصف من العناصر المحددة.
toggleClass(class)	إضافة الصف إذا كان غير موجود وحذفه إن كان موجوداً.
html()	إعادة المحتوى html لأول عنصر محدد.
html(val)	إسناد المحتوى html لجميع العناصر المحددة.
text()	إعادة نص جميع العناصر المحددة.
text(val)	إسناد نص إلى جميع العناصر المحددة.
val()	إعادة قيمة أول عنصر محدد.
val(val)	إسناد قيمة الواصفة value لجميع العناصر المحددة إذا كانوا من النوع input. أما مع عنصر select فيتمّ تحديد الخيار option الموافق للقيمة. ومع صندوق تحقق أو زر راديو، فسيتمّ تحديد العنصر الموافق للقيمة.

أمثلة

<code>\$("#myID").attr("custom")</code>	إعادة قيمة الوصفة custom لأول عنصر له المعرف myID.
<code>\$("#img").attr("alt", "Sample Image")</code>	إسناد قيمة الوصفة alt بجميع العناصر img إلى القيمة Sample Image.
<code>\$("#input").attr({ value: "", title: "Please enter a value" });</code>	إسناد القيمة الفارغة إلى جميع العناصر input والعنوان title إلى القيمة Please enter a value.
<code>\$("#a[href^=http://]").attr("target", "_blank")</code>	إسناد الوصفة target إلى القيمة _blank لجميع الروابط التي تبدأ فيها الخاصية href بـ http://.
<code>\$("#a").removeAttr("target")</code>	حذف الوصفة target لجميع الروابط.
<code>\$("#form").submit(function() { \$("#:submit", this).attr("disabled", "disabled"); });</code>	إسناد القيمة disabled إلى الوصفة disabled عند النقر على زر submit.
<code>\$("#p").text()</code>	تُعيد النص لكل الفقرات
<code>\$("#p").text("<i>Hello World</i>")</code>	تقوم بإسناد النص <i>Hello World</i> إلى محتوى جميع الفقرات
<code>\$("#p").html()</code>	تُعيد محتوى HTML لكل الفقرات.
<code>\$("#div").html("Hello World")</code>	تُسند Hello World إلى محتوى html لكل div.
<code>\$("#input:checkbox:checked").val()</code>	تُعيد قيمة أول صندوق تحقق مُحدّد.
<code>\$("#input:radio[name=bar]:checked").val()</code>	تُعيد أول قيمة لمجموعة من أزرار الراديو.
<code>\$("#button").val("Hello")</code>	إسناد القيمة Hello لكل button.

مثال: تغيير قيمة واصفة

```

<!doctype html>
<html>
<head>
<title>the title</title>
  <script type="text/javascript"
                                src="jquery/jquery-3.6.0.min.js"></script>
  <script type="text/javascript" language="javascript">
$(document).ready(function() {
  $("img").attr({
    src: "images/jquery.jpg",
    title: "jQuery",
    alt: "jQuery Logo"
  });
});
</script>
</head>
<body>
<div class="div1" id="divid">
  <p>Following is the logo of jQuery</p>
  
</div>
</body></html>

```

يُظهر المتصفح:

Following is the logo of jQuery



Text و HTML

يقوم المثال التالي بنسخ محتوى HTML الموجود في فقرة معرفها pid1 وإسناده إلى نص فقرة أخرى معرفها pid2.

```

<!doctype html>
<html>
<head>
<title>the title</title>
  <script type="text/javascript"
                                src="jquery/jquery-3.6.0.min.js"></script>
  <script type="text/javascript" language="javascript">

```

```

$(document).ready(function() {
    var content = $("#pid1").html();
    $("#pid2").text(content);
});
</script>
<style>
    .red { color:red; }
    .green { color:green; }
</style>
</head>
<body>
<p class="green" id="pid1">This is <i>first paragraph</i>.</p>
<p class="red" id="pid2">This is second paragraph.</p>
</body>
</html>

```

أسلوب الصفحات المتتالي

تدعم المكتبة JQuery معظم محددات CSS بإصداراتها 1 إلى 3.

التعامل مع خصائص CSS

تُستخدم الطريقة (css(PropertyName, PropertyValue) للتعامل مع الخصائص.

مثال:

يقوم المثال التالي بتلوين العنصر الثالث في قائمة:

```

<!doctype html>
<html>
<head>
<title>the title</title>
    <script type="text/javascript"
    src="jquery/jquery-3.6.0.min.js"></script>
    <script type="text/javascript" language="javascript">

    $(document).ready(function() {
        $("li").eq(2).css("color", "red");
    });
    </script></head>
<body>

```

```

<div>
<ul>
  <li>list item 1</li>
  <li>list item 2</li>
  <li>list item 3</li>
  <li>list item 4</li>
  <li>list item 5</li>
  <li>list item 6</li>
</ul>
</div>
</body>
</html>

```

يُظهر المتصفح:

- list item 1
- list item 2
- list item 3
- list item 4
- list item 5
- list item 6

تطبيق أكثر من خاصية CSS

يُبين المثال التالي تطبيق أكثر من خاصية:

```

<!doctype html>
<html>
<head>
<title>the title</title>
  <script type="text/javascript"
    src="jquery/jquery-3.6.0.min.js"></script>
  <script type="text/javascript" language="javascript">
    $(document).ready(function() {
      $("li").eq(2).css({"color": "red",
                        "background-color": "green"});
    });
  </script>
</head>
<body>

```

```

<div>
<ul>
  <li>list item 1</li>
  <li>list item 2</li>
  <li>list item 3</li>
  <li>list item 4</li>
  <li>list item 5</li>
  <li>list item 6</li>
</ul>
</div>
</body>
</html>

```

يُظهر المتصفح:

- list item 1
- list item 2
- **list item 3**
- list item 4
- list item 5
- list item 6

تغيير عرض وارتفاع العناصر

يُبين المثال التالي كيفية تغيير عرض وارتفاع العناصر:

```

<!doctype html>
<html>
<head>
<title>the title</title>
  <script type="text/javascript"
    src="jquery/jquery-3.6.0.min.js"></script>
  <script type="text/javascript" language="javascript">

    $(document).ready(function() {
      $("div:first").width(100);
      $("div:first").css("background-color", "blue");
    });

```

```

</script>
<style>
div{ width:70px; height:50px; float:left; margin:5px;
    background:red; cursor:pointer; }
</style>
</head>
<body>
<div></div>
<div>d</div>
<div>d</div>
<div>d</div>
<div>d</div>
</body>
</html>

```

يُظهر المتصفح:



وظائف JQuery CSS

يُبين الجدول التالي أهم الوظائف للتعامل مع أسلوب الصفحات المتتالي:

Method	الوصف
css(name)	يُعيد خاصية الأسلوب لأول عنصر محدد.
css(name, value)	يُسند قيمة لخاصية لجميع العناصر المحددة.
css(properties)	يُسند قيمة/خاصية لجميع العناصر المحددة.

height(val)	يُسند الارتفاع لجميع العناصر المحددة.
height()	يُعيد قيمة الارتفاع لأول عنصر محدد.
width(val)	يُسند العرض لجميع العناصر المحددة.
width()	يُعيد قيمة العرض لأول عنصر محدد.

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-3.6.0.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-3.6.0.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-3.6.0.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 `after(content)` بإدراج محتوى بعد العناصر المحددة، والوظيفة `before(content)` بإدراج محتوى قبل العناصر المحددة.

مثال:

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

```
<!doctype html>
<html>
<head>
  <title>the title</title>
  <script type="text/javascript"
    src="jquery/jquery-3.6.0.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 (المرجع) التالي طرق التعامل مع عناصر الوثيقة:

للحصول على أمثلة إضافة يُمكن زيارة الرابط:

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

Method	Description
after(content)	Insert content after each of the matched elements.
append(content)	Append content to the inside of every matched element.
appendTo(selector)	Append all of the matched elements to another, specified, set of elements.

before(content)	Insert content before each of the matched elements.
clone(bool)	Clone matched DOM Elements, and all their event handlers, and select the clones.
clone()	Clone matched DOM Elements and select the clones.
empty()	Remove all child nodes from the set of matched elements.
html(val)	Set the html contents of every matched element.
html()	Get the html contents (innerHTML) of the first matched element.
insertAfter(selector)	Insert all of the matched elements after another, specified, set of elements.
insertBefore(selector)	Insert all of the matched elements before another, specified, set of elements.
prepend(content)	Prepend content to the inside of every matched element.
prependTo(selector)	Prepend all of the matched elements to another, specified, set of elements.
remove(expr)	Removes all matched elements from the DOM.
replaceAll(selector)	Replaces the elements matched by the specified selector with the matched elements.
replaceWith(content)	Replaces all matched elements with the specified HTML or DOM elements.
text(val)	Set the text contents of all matched elements.
text()	Get the combined text contents of all matched elements.
wrap(elem)	Wrap each matched element with the specified element.
wrap(html)	Wrap each matched element with the specified HTML content.
wrapAll(elem)	Wrap all the elements in the matched set into a single wrapper element.
wrapAll(html)	Wrap all the elements in the matched set into a single wrapper element.
wrapInner(elem)	Wrap the inner child contents of each matched element (including text nodes) with a DOM element.
wrapInner(html)	Wrap the inner child contents of each matched element (including text nodes) with an HTML structure.