

المرجع العربي لمكتبة

jQuery

إعداد: المهندس إسماعيل عنجريني

البريد الإلكتروني: ismael.enjreny@gmail.com

آخر تعديل على الوثيقة: الخميس، 26 آذار، 2009

الإصدار: 1.0

11	أساسيات jQuery (jQuery Core)
11	التابع jQuery أو اختصاراً (\$).....
11	jQuery(expression, [context])
12	jQuery(html, [ownerDocument])
12	jQuery(elements)
13	jQuery(callback)
13	أمثلة على التتابع الأساسية في jQuery
15	المحددات في jQuery
15	المحددات الأساسية Basics
15	#id
16	المحدد element
17	المحدد .class
18	المحدد *
19	المحدد selector1, selector2, selectorN
20	محددات التسلسل الهرمي Hierarchy
20	المحدد ancestor descendant
22	المحدد parent > child
23	المحدد prev + next
24	المحدد prev ~ siblings
26	المرشحات الأساسية Basic Filters
26	المرشح :first
27	المرشح :last
27	المرشح :not(selector)
29	المرشح :even
30	المرشح :odd
31	المرشح :eq(index)
31	المرشح :gt(index)
32	المرشح :gt(index)
33	المرشح :header
34	المرشح :animated
35	مرشحات المحتوى Content Filters
35	المرشح :contains(text)
36	المرشح :empty

37	:has(selector) المرشح
38	:parent المرشح
39	Visibility Filters مرشحات الرؤية أو الظهور
39	:hidden المرشح
40	:visible المرشح
42	Attribute Filters السمات
42	[attribute] المرشح
43	[attribute=value] المرشح
44	[attribute!=value] المرشح
45	[attribute^=value] المرشح
45	[attribute\$=value] المرشح
46	[attribute*=value] المرشح
47	[attributeFilter1][attributeFilter2][attributeFilterN] المرشح
48	Child Filters مرشحات العناصر الفرعية
48	:nth-child(index/even/odd/equation) المرشح
49	:first-child المرشح
51	:last-child المرشح
53	:only-child المرشح
55	Forms النماذج
55	:input
56	:text
58	:password
60	:radio
60	:checkbox
60	:submit
61	:image
61	:reset
61	:button
61	:file
61	:hidden
61	Form Filters مرشحات النماذج
61	:enabled المرشح
62	:disabled المرشح
63	:checked المرشح

64	المرشح selected
66	ملاحظة
66	Attributes السمات
66	attr(name)
67	attr(properties)
69	attr(key, value)
70	attr(key, fn)
71	removeAttr(name)
72	Class (كلاسات) الصفوف
72	addClass(class)
73	hasClass(class)
74	removeClass(class)
75	toggleClass(class)
76	toggleClass(class, switch)
77	HTML التوابع الـ
77	html()
79	html(val)
79	Text النصية التوابع
79	text()
80	text(val)
81	Value القيمة التوابع
81	val()
83	val(val)
84	val(val)
86	توابع العبور
86	توابع الترشيح
86	eq(index)
87	hasClass(class)
89	filter(expr)
90	filter(fn)
92	is(expr)
94	map(callback)
95	not(expr)
96	slice(start, [end])

99	توابع البحث
99	add(expr)
100	children([expr])
103	closest([expr])
104	find(expr)
105	next([expr])
106	nextAll([expr])
107	offsetParent()
107	parent([expr])
108	prev([expr])
110	prevAll([expr])
111	siblings([expr])
113	Chaining Functions توابع التسلسلية
113	andSelf()
114	end()
117	Manipulation Functions توابع التعامل مع المحتوى
117	Changing Contents توابع تغيير المحتوى
117	html()
118	html(val)
119	text()
120	text(val)
121	Inserting Inside توابع إدراج المعلومات في داخل العناصر
121	append(content)
122	appendTo(selector)
123	prepend(content)
124	prependTo(selector)
125	Inserting Outside توابع إدراج المعلومات خارج العناصر
125	after(content)
126	before(content)
127	insertAfter(selector)
128	insertBefore(selector)
129	Inserting Around (التطويق) توابع الإحاطة
129	wrap(html)
130	wrap(elem)

131	wrapAll(html)
132	wrapAll(elem)
133	wrapInner(html)
134	wrapInner(elem)
135	توابع الاستبدال
135	replaceWith(content)
136	replaceAll(selector)
137	توابع إدالة العناصر Removing
137	empty()
138	remove([expr])
139	توابع النسخ Copying
139	clone()
140	clone(bool)
141	توابع الأنماط المتعاقبة CSS Functions
141	CSS
141	css(name)
142	css(properties)
144	css(name, value)
145	توابع التموضع Positioning Functions
145	offset()
146	position()
147	scrollTop()
148	scrollTop(val)
149	scrollLeft()
150	scrollLeft(val)
151	توابع العرض والارتفاع Height and Width Functions
151	height()
153	height(val)
154	width()
156	width(val)
157	innerHeight()
158	innerWidth()
159	outerHeight(margin)
160	outerWidth(margin)

161Events Functions	توابع الأحداث
161Page Load Functions	توابع تحميل الصفحة
161	ready(fn)
162Event Handling Functions	توابع التعامل مع الحدث
162	bind(type, data, fn)
165	one(type, [data], fn)
167	trigger(event, [data])
170	triggerHandler(event, [data])
172	unbind([type], [fn])
174Live Events Functions	توابع الأحداث المباشرة
174	live(type, fn)
176	die(type, fn)
178Interaction Helpers Functions	توابع التفاعل مع العناصر
178	hover(over, out)
180	toggle(fn1, fn2, [fn3], [fn4], ...)
181Event Helpers Functions	توابع الأحداث
181	blur()
182	blur(fn)
183	change()
183	change(fn)
184	click()
185	click(fn)
186	dblclick()
186	dblclick(fn)
188	error()
188	error(fn)
188	focus()
188	focus(fn)
189	keydown()
190	keydown(fn)
190	keypress()
190	keypress(fn)
192	keyup()
193	keyup(fn)

193	load(fn)
194	mousedown(fn)
195	mouseenter(fn)
197	mouseleave(fn)
199	mousemove(fn)
200	mouseout(fn)
203	mouseover(fn)
203	mouseup(fn)
204	resize(fn)
204	scroll(fn)
206	select()
206	select(fn)
207	submit()
207	submit(fn)
209	unload(fn)
210	Effects Functions التوابع التأثيرات الحركية والشكلية
210	Basics Functions التوابع الأساسية
210	show()
210	show(speed, [callback])
211	hide()
212	hide(speed, [callback])
213	toggle()
214	toggle(switch)
215	toggle(speed, [callback])
217	Sliding Functions توابع الانزلاق
217	slideDown(speed, [callback])
218	slideUp(speed, [callback])
220	slideToggle(speed, [callback])
221	Fading Functions توابع التلاشي
221	fadeIn(speed, [callback])
222	fadeOut(speed, [callback])
225	fadeTo(speed, opacity, [callback])
229	Animate Functions توابع الحركة
229	animate(params, [duration], [easing], [callback])

233	animate(params, options)
235	stop(clearQueue, gotoEnd)
236	jQuery.fx.off
238	توابع العمليات الغير متزامنة AJAX Functions
238	الطلبات الغير متزامنة AJAX Requests
238	jQuery.ajax(options)
240	dataType
241	error
241	global
241	ifModified
241	jsonp
242	password
242	processData
242	scriptCharset
242	success
242	timeout
242	type
242	url
242	username
243	xhr
244	load(url, [data], [callback])
246	jQuery.get(url, [data], [callback], [type])
246	jQuerygetJSON(url, [data], [callback])
247	jQuery.getScript(url, [callback])
248	jQuery.post(url, [data], [callback], [type])
249	أحداث العمليات الغير متزامنة Ajax Events
249	ajaxComplete(callback)
249	ajaxError(callback)
250	ajaxSend(callback)
250	ajaxStart(callback)
250	ajaxStop(callback)
251	ajaxSuccess(callback)
251	متنوعات
251	jQuery.ajaxSetup(options)

251	serialize()
254	srializeArray()
255	أدوات
255	Browser and Feature Detection توابع التفاعل مع المتصفح وفحص الميزات
255	jQuery.support
257	jQuery.browser
258	jQuery.browser.version
260	jQuery.boxModel

أساسيات jQuery (jQuery Core)

التابع jQuery أو اختصاراً \$(...)

jQuery(expression, [context])

يقبل هذا التابع سلسلة نصية من الممكن أن تحتوي على محدد CSS حيث يمكن استخدامه لمطابقة مجموعة من العناصر.

الوظائف الأساسية لمكتبة jQuery تعتمد على هذا التابع حيث أن كل شيء فيها يعتمد على استخدام هذا التابع, والاستخدام الأساسي لهذا التابع يكون في تمرير سلسلة نصية قد تحتوي على محدد CSS عندها يعيد هذا التابع جميع عناصر الـ HTML التي تتطابق مع السلسلة النصية التي تم تمريرها بشكل افتراضي إذا لم يتم تحديد سياق context للتابع فإن التابع (\$) يبحث عن جميع العناصر التي تطابق شروط البحث ضمن جميع عناصر HTML الموجودة في الوثيقة الحالية.

أمثلة

المثال التالي يطابق جميع عناصر الإدخال من النوع radio في النموذج form الأول ضمن الوثيقة

الحالية:

```
jQuery("input:radio", document.forms[0]);
```

المثال التالي يطابق جميع العناصر div ضمن بيانات من النوع XML هي عبارة عن بيانات ناتجة عن

استدعاء توابع AJAX:

```
jQuery("div", xml.responseXML);
```

المثال التالي يضع إطار حول جميع العناصر p الموجودة ضمن العنصر div:

```
jQuery("div > p").css("border", "1px solid gray");
```

one
two
three

حيث أن شيفرة الـ HTML للمثال السابق هي:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>

  <script>
    $(document).ready(function(){
      $("div > p").css("border", "1px solid gray");
    });
  </script>

</head>
<body>
  <p>one</p> <div><p>two</p></div> <p>three</p>
</body>
</html>
```

jQuery(html, [ownerDocument])

يستخدم هذا التابع لبناء صيغة HTML مباشرة باستخدام سلسلة نصية تحتوي على العناصر المشكلة لعناصر الـ HTML. حيث يمكن بناء عناصر بسيطة بدون أي حواص إضافية مثل "<div />" حيث يستخدم التابع document.createElement. أما في حالة العناصر المعقدة فيتم معالجتها بإسناد السلسلة النصية إلى الخاصية innerHTML لعنصر div. لذلك لا يمكن أن تحتوي السلسلة النصية html على عناصر لا يمكن أن تُضمن ضمن عنصر div مثل html, head, body, title.

أمثلة

إنشاء عنصر div وجميع عناصره ديناميكياً وإلحاقها بمحتوى العنصر body, وما يتم تقنياً هو أنه يتم إنشاء عنصر div وإسناد القيمة html (قيمة السلسلة النصية) إلى خاصيته innerHTML وهذا ما يوفر مرونة عالية وبعض المحدودية:

```
$("#<div><p>Hello</p></div>").appendTo("body")
```

jQuery(elements)

هذا التابع يتعامل مع عنصر أو مصفوفة عناصر كما يمكن تمرير عنصر من النوع XML وخاصة عند التعامل مع توابع AJAX التي تعيد نتائج على شكل XML.

المثال التالي يعين اللون الأسود مخلفية للصفحة:

```
$(document.body).css( "background", "black" );
```

أما المثال التالي فيخفي جميع العناصر ضمن نموذج إدخال البيانات:

```
$(myForm.elements).hide()
```

jQuery(callback)

هذا التابع هو اختصار للعبارة `$(document).ready()` ويسمح التابع بتنفيذ تابع أو أكثر عندما يصبح من الممكن تنفيذ تعليمات JavaScript في الوثيقة. ويمكن استدعاء هذا التابع أكثر من مرة في نفس الوثيقة، وكمثال على ذلك:

```
$(function(){
  // Document is ready
});
```

أمثلة على التوابع الأساسية في jQuery

كود الـ HTML الذي سيتم تنفيذ التوابع عليه

```
<div id="box">
  <p id="par">The core functionality of jQuery centers around this function.</p>
  <a href="http://jquery.com">jQuery</a>
</div>
```

أما الهدف من شيفرة الـ JavaScript فهو وضع إطار حول العنصر `div` ذو الاسم `box` ومن ثم إخفاء العنصر `p` ذو الاسم `par`، وعند الضغط على الرابط `a` سيتم إنشاء نص ديناميكي وإدراجه ضمن العنصر `div`. والشيفرة التالية تنفذ كل ماورد أعلاه، حيث نلاحظ وجود تعليق قبل كل نوع من أنواع توابع jQuery، السطر التالي يعبر عن تنفيذ تابع من النوع `jQuery(callback)`:

```
/* jQuery(callback) */
$(function() {
```

حيث وكما ذكرنا أعلاه بأن الشيفرة التي ضمن هذا التابع يتم تنفيذها في حال أصبح من الممكن تنفيذ تعليمات DOM ضمن الوثيقة.

أما السطر التالي فيعبر عن تنفيذ التابع `jQuery(expression)`:

```
/* jQuery(expression) */
$('#div').css('border', 'solid 1px #000000');
```

حيث يعبر العنصر `div` عن قيمة الـ `expression` للتابع `jQuery` وتفيد بأن كل عناصر الـ `div` الموجودة في الوثيقة سيتم تحديدها بإطار أسود.

أما السطر التالي فيعبر عن تنفيذ التابع من الشكل `jQuery(html)`:

```
/* jQuery(html) */
$('#<p>Hello, I am jQuery sample</p>').appendTo('div#box');
```

حيث يتم إنشاء عنصر `p` بشكل ديناميكي ومن ثم يتم إدراجه تحت العنصر `div` ذو الاسم `box`.

وأخيراً فإن السطر التالي يعبر عن تنفيذ التابع من الشكل `jQuery(elements)`:

```
/* jQuery(elements) */
$(par).hide();
```

أما الشيفرة بشكل كامل فهي:

```
/* jQuery(callback) */
$(function() {

    /* jQuery(expression) */
    $('#div').css('border', 'solid 1px #000000');

    $('#div#box a').click(function() {

        /* jQuery(html) */
        $('#<p>Hello, I am jQuery sample</p>').appendTo('div#box');

        /* jQuery(elements) */
        $(par).hide();

        return false;
    });
});
```

المحددات في jQuery

المحددات الأساسية Basics

#id

يستخدم هذا المحدد للبحث عن عنصر واحد فقط باستخدام السمة id للعنصر, وفي حال أن قيمة الـ id تحتوي على محارف غير الأحرف مثل النقطة أو الرمز [فيجب معالجتها باستخدام الرمز \, ويتم استخدام هذا المحدد كما في المثال التالي:

```
$("#myID\\.entry\\[1\\]").css("border","3px solid red");
```

id="myID.entry[0]"

id="myID.entry[1]"

id="myID.entry[2]"

المثال السابق يبحث عن عنصر قيمة السمة id له هي myID.entry[1], والفقرة التالية تحتوي على كامل الشيفرة, حيث يتم البحث عن العنصر الذي سمته هي myID.entry[1] ويضع له إطار سمكه 3 بيكسل ولونه أحمر. ويجب الانتباه إلى استخدام رمزي \\ لتجاوز المحارف الخاصة وهي النقطة (.) والرمز [والرمز الآخر]:

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>

  <script>
  $(document).ready(function(){
    $("#myID\\.entry\\[1\\]").css("border","3px solid red");
  });
  </script>
  <style>
  div {
    width: 300px;
    float:left;
    padding: 2px;
    margin: 3px;
    background-color: #EEEEEE;
  }
  </style>
</head>
<body>
  <div id="myID.entry[0]">id="myID.entry[0]"</div>
  <div id="myID.entry[1]">id="myID.entry[1]"</div>
  <div id="myID.entry[2]">id="myID.entry[2]"</div>
</body>
</html>

```

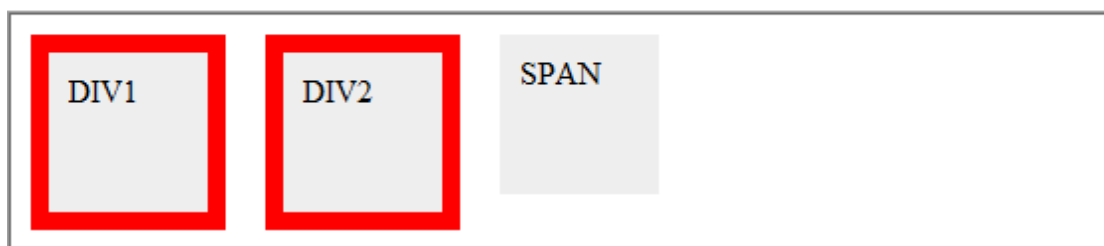
المحدد element

يستخدم هذا المحدد للبحث عن جميع العناصر التي لها نفس الاسم مثلاً div أو a وغيرها من عناصر الـ HTML. كما في المثال التالي حيث يتم البحث عن جميع العناصر div ضمن الوثيقة وإحاطة كل عنصر div بإطار سمكه 9 بيكسل ولونه أحمر:

```

$("div").css("border","9px solid red");

```



أما المصدر الكامل للشفيرة السابقة فهي:


```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>

  <script>
    $(document).ready(function(){
      $("#div").css("border","9px solid red");
    });
  </script>
  <style>
  div,span {
    width: 60px;
    height: 60px;
    float:left;
    padding: 10px;
    margin: 10px;
    background-color: #EEEEEE;
  }
  </style>
</head>
<body>
  <div>DIV1</div>
  <div>DIV2</div>
  <span>SPAN</span>
</body>
</html>

```

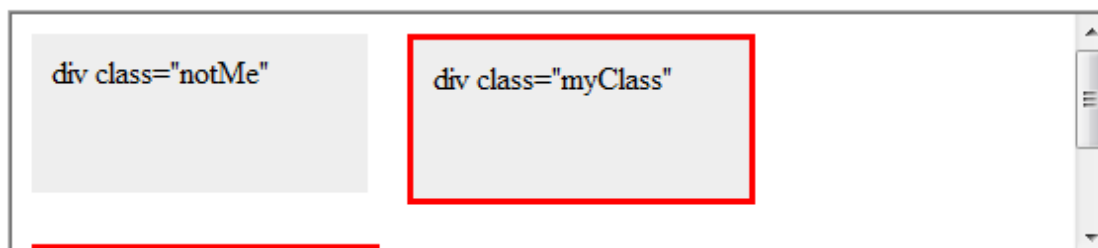
المحدد class.

يستخدم المحدد class. للبحث عن جميع العناصر التي يطبق هذا المحدد من عناصر الـ HTML, ففي المثال التالي يتم مطابقة جميع العناصر التي يطبق عليها المحدد myClass ومن ثم تحاط بإطار عرضه 3 بيكسل ولونه أحمر, وفي حال أن أحد العناصر طبق عليه مجموعة كلاسات (CSS Class) فيجب أن يتطابق أحدها على الأقل مع المحدد class. ليظهر مع نتائج المطابقة:

```

$("#myClass").css("border","3px solid red");

```



أما الشيفرة الكاملة للمثال السابق فهي:

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>

  <script>
  $(document).ready(function(){
    $(".myClass").css("border","3px solid red");
  });
  </script>
  <style>
  div,span {
    width: 150px;
    height: 60px;
    float:left;
    padding: 10px;
    margin: 10px;
    background-color: #EEEEEE;
  }
  </style>
</head>
<body>
  <div class="notMe">div class="notMe"</div>
  <div class="myClass">div class="myClass"</div>
  <span class="myClass">span class="myClass"</span>
</body>
</html>

```

* المحدد

يستخدم هذا المحدد لمطابقة جميع العناصر في الوثيقة بغض النظر عن أي عملية أو محدد آخر، والمثال التالي يحيط جميع عناصر الوثيقة بإطار عرضه 3 بيكسل ولونه أحمر:

```

$(".*").css("border","3px solid red");

```



والشيفرة الكاملة للمثال السابق هي:

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>

  <script>
$(document).ready(function() {
  $("*").css("border", "3px solid red");
});
</script>
<style>
div, span, p {
  width: 150px;
  height: 60px;
  float: left;
  padding: 10px;
  margin: 10px;
  background-color: #EEEEEE;
}
</style>
</head>
<body>
  <div>DIV</div>
  <span>SPAN</span>
  <p>P <button>Button</button></p>
</body>
</html>

```

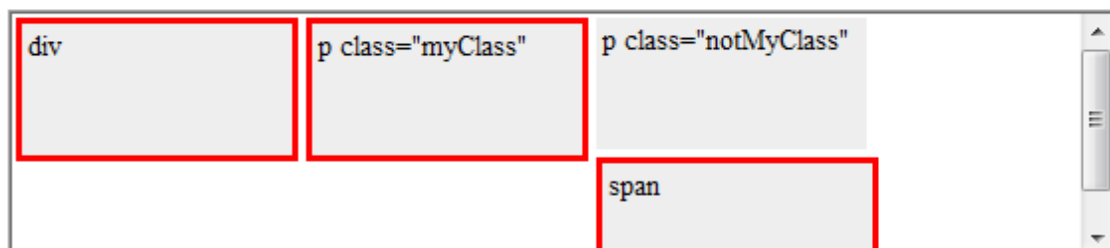
المحدد selector1, selector2, selectorN

يستخدم هذا المحدد أو إن صح القول مجموعة المحددات لمطابقة جميع العناصر التي تتطابق مع أحد المحددات, حيث يتم الفصل بين كل محدد والذي يليه بفاصلة (,), والمثال التالي يحدد العناصر div, span والعنصر p المطبق عليه الكلاس myClass ويضع إطار عرضه 3 بيكسل ولونه أحمر حول هذه العناصر:

```

$("div, span, p.myClass").css("border", "3px solid red");

```



أما الشيفرة الكاملة للمثال السابق فهي:

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>

  <script>
$(document).ready(function() {
  $("div,span,p.myClass").css("border","3px solid red");
});
</script>
<style>
div,span,p {
  width: 130px;
  height: 60px;
  float:left;
  padding: 3px;
  margin: 2px;
  background-color: #EEEEEE;
  font-size:14px;
}
</style>
</head>
<body>
  <div>div</div>
  <p class="myClass">p class="myClass"</p>
  <p class="notMyClass">p class="notMyClass"</p>
  <span>span</span>
</body>
</html>

```

محددات التسلسل الهرمي Hierarchy

المحدد ancestor descendant

يستخدم هذا المحدد لمطابقة جميع العناصر الأحفاد (descendant) للعنصر الأب (ancestor), ففي المثال التالي يتم مطابقة جميع العناصر input الموجودة داخل أي عنصر form في الوثيقة ويتم إحاطة كل عنصر input بإطار منقط عرضه 2 بيكسل ولونه أزرق, وقيمة ancestor ممكن أن تكون أي محدد صحيح من المحددات المذكورة سابقاً أو لاحقاً ضمن محددات jQuery:

```
$("#form input").css("border", "2px dotted blue");
```

Form is surrounded by the green outline

Child:

Grandchild:

Sibling to form:

أما الشيفرة الكاملة للمثال السابق فهي:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>

  <script>
    $(document).ready(function(){
      $("#form input").css("border", "2px dotted blue");
    });
  </script>
  <style>
    body { font-size:14px; }
    form { border:2px green solid; padding:2px; margin:0;
          background:#efe; }
    div { color:red; }
    fieldset { margin:1px; padding:3px; }
  </style>
</head>
<body>
  <form>
    <div>Form is surrounded by the green outline</div>
    <label>Child:</label>
    <input name="name" />
    <fieldset>
      <label>Grandchild:</label>
      <input name="newsletter" />
    </fieldset>
  </form>
  Sibling to form: <input name="none" />
</body>
</html>
```

المحدد parent > child

يستخدم المحدد لمطابقة العنصر child فقط التابع للعنصر parent ويمكن تحديد أكثر من مستوى للوصول إلى العنصر child, كما في المثال التالي حيث يتم تحديد القائمة ul الثانية ضمن العنصر li فقط وتحاط بإطار عرضه 1 بيكسل ولونه أحمر

```
<script language="javascript">
  $(function() {
    $('div > ul > li > ul').css('border', 'solid 1px #ff0000');
  });
</script>
```

أما شيفرة الـ HTML فهي كما يلي:

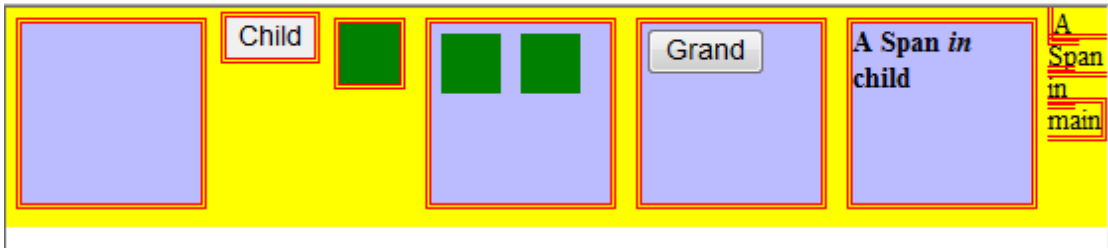
```
<div>
  <ul id="first">
    <li>
      <ul>
        <li>1 - 1</li>
        <li>2 - 1</li>
        <li>3 - 1</li>
      </ul>
    </li>
    <li>2</li>
    <li>3</li>
  </ul>
</div>
```

وتكون النتيجة كما يلي:

- 1 - 1
- 2 - 1
- 3 - 1
- 2
- 3

أما المثال التالي فيتم إحاطة جميع العناصر التي تدرج تحت العنصر الذي قيمة السمة id له هي main بإطار أحمر مضاعف بغض النظر عن العنصر عرضه 4 بيكسل, طبعاً المحدد * في هذا المثال يرمز إلى العناصر التي تدرج مباشرة تحت العنصر الذي قيمة السمة id تساوي main أي الأحفاد في المستوى الأول للعنصر:

```
$("#main > *").css("border", "3px double red");
```



أما الشيفرة الكاملة للمثال السابق فهي:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>

  <script>
    $(document).ready(function(){
      $("#main > *").css("border", "3px double red");
    });
  </script>
  <style>
    body { font-size:14px; }
    span#main { display:block; background:yellow; height:110px; }
    button { display:block; float:left; margin:2px;
            font-size:14px; }
    div { width:90px; height:90px; margin:5px; float:left;
          background:#bbf; font-weight:bold; }
    div.mini { width:30px; height:30px; background:green; }
  </style>
</head>
<body>
  <span id="main">
    <div></div>
    <button>Child</button>
    <div class="mini"></div>
    <div>
      <div class="mini"></div>
      <div class="mini"></div>
    </div>
    <div><button>Grand</button></div>
    <div><span>A Span <em>in</em> child</span></div>
  </span>A Span in main</span>
```

المحدد prev + next

يستخدم هذا المحدد لمطابقة جميع العناصر المحددة بالقيمة next والتي تأتي مباشرة خلف المحدد prev في

شيفرة الـ HTML, ففي المثال التالي تتم مطابقة جميع العناصر input التي تأتي مباشرة خلف العنصر label

ويتم تحديد لون الخط في عناصر الـ input إلى الأزرق مع إعطاء عناصر الـ input قيمة افتراضية هي , Labeled!, طبعاً نلاحظ وجود عنصر input لم يتأثر وذلك لأنه غير مسبوق بعنصر label:

```
$("#label + input").css("color", "blue").val("Labeled!");
```

Name:	<input type="text" value="Labeled!"/>
Newsletter:	<input type="text" value="Labeled!"/>
	<input type="text"/>

أما الشيفرة الكاملة للمثال السابق فهي:

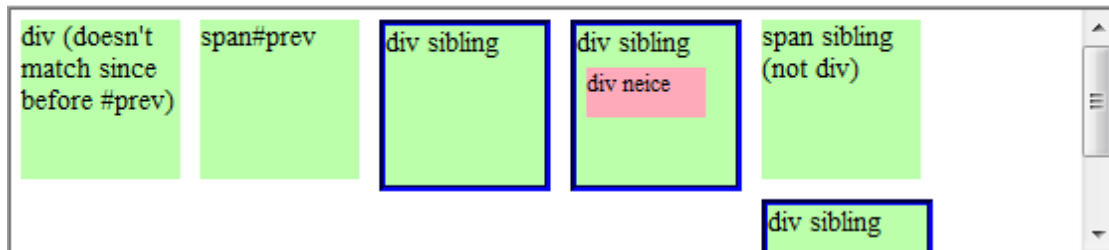
```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>

  <script>
    $(document).ready(function(){
      $("#label + input").css("color", "blue").val("Labeled!");
    });
  </script>
</head>
<body>
  <form>
    <label>Name:</label>
    <input name="name" />
    <fieldset>
      <label>Newsletter:</label>
      <input name="newsletter" />
    </fieldset>
  </form>
  <input name="none" />
</body>
</html>
```

المحدد prev ~ siblings

يستخدم هذا المحدد لمطابقة جميع العناصر التي تطابق المحدد في siblings والتي تظهر بعد العنصر المطابق في المحدد prev ويجب أن تكون العناصر siblings خارج العنصر المطابق بالمحدد prev, كما في المثال التالي, حيث يتم تحديد جميع العناصر div التي تظهر بعد العنصر الذي قيمة السمة id هي prev:


```
$("#prev ~ div").css("border", "3px groove blue");
```



أما الشيفرة الكاملة للمثال السابق فهي:

```
<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>

  <script>
    $(document).ready(function() {
      $("#prev ~ div").css("border", "3px groove blue");
    });
  </script>
  <style>
    div,span {
      display:block;
      width:80px;
      height:80px;
      margin:5px;
      background:#bbffaa;
      float:left;
      font-size:14px;
    }
    div#small {
      width:60px;
      height:25px;
      font-size:12px;
      background:#fab;
    }
  </style>
</head>
<body>
  <div>div (doesn't match since before #prev)</div>
  <span id="prev">span#prev</span>
  <div>div sibling</div>
  <div>div sibling <div id="small">div neice</div></div>
  <span>span sibling (not div)</span>
```

المرشحات الأساسية Basic Filters

المرشح :first

يستخدم هذا المرشح لمطابقة أول عنصر من العناصر التي تم تحديدها باستخدام أحد المحددات التي تدعمها مكتبة jQuery, ففي المثال التالي يتم مطابقة أول عنصر tr من العناصر tr التي تظهر في الوثيقة الحالية ويجول الخط الذي يُكتب داخل هذا السطر مائل:

```
$("#tr:first").css("font-style", "italic");
```

Row 1

Row 2

Row 3

أما الشيفرة الكاملة للمثال السابق فهي:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>

  <script>
    $(document).ready(function() {
      $("#tr:first").css("font-style", "italic");
    });
  </script>
  <style>
    td { color:blue; font-weight:bold; }
  </style>
</head>
<body>
  <table>
    <tr><td>Row 1</td></tr>
    <tr><td>Row 2</td></tr>
    <tr><td>Row 3</td></tr>
  </table>
</body>
</html>
```

المرشح `:last`:

يستخدم هذا المرشح لمطابقة آخر عنصر من العناصر التي تم تحديدها باستخدام أحد المحددات التي تدعمها مكتبة jQuery. ففي المثال التالي يتم مطابقة آخر عنصر `tr` من العناصر `tr` التي تظهر في الوثيقة الحالية ويلون الخلفية باللون الأصفر ويصبح الخط الذي يكتب فيه داخل السطر غامق `bold`:

```
$("#tr:last").css({backgroundColor: 'yellow', fontWeight: 'bolder'});
```

First Row
Middle Row
Last Row

أما الشيفرة الكاملة للمثال السابق فهي:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>

  <script>
    $(document).ready(function() {
      $("#tr:last").css({backgroundColor: 'yellow', fontWeight: 'bolder'});
    });
  </script>
</head>
<body>
  <table>
    <tr><td>First Row</td></tr>
    <tr><td>Middle Row</td></tr>
    <tr><td>Last Row</td></tr>
  </table>
</body>
</html>
```

المرشح `:not(selector)`:

يستخدم هذا المرشح لاستثناء جميع العناصر التي تتطابق مع صيغة `selector` من نتيجة المحدد الذي يسبق استخدام المرشح `:not(selector)`، ويدعم هذا المرشح المحددات من النوع `selector1, selectorN` وحتى يدعم المحددات المعقدة مثل استخدام المحدد `#id` وغيره من المحددات الأخرى، والمثال التالي يسند للون

الأصفر إلى خلفية جميع العناصر من النوع checkbox → input الغير محددة ويلغى تفعيل جميع العناصر من النوع input بدون استثناء:

```
$("#input:not(:checked) + span").css("background-color", "yellow");
$("#input").attr("disabled", "disabled");
```

- Mary
- lcm
- Peter

أما الشيفرة الكاملة للمثال السابق فهي:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>

  <script>
    $(document).ready(function(){

      $("#input:not(:checked) + span").css("background-color", "yellow");
      $("#input").attr("disabled", "disabled");

    });
  </script>
</head>
<body>
  <div>
    <input type="checkbox" name="a" />
    <span>Mary</span>
  </div>
  <div>
    <input type="checkbox" name="b" />
    <span>lcm</span>
  </div>
  <div>
    <input type="checkbox" name="c" checked="checked" />
    <span>Peter</span>
  </div>
</body>
</html>
```

المرشح even:

يُستخدم هذا المرشح لمطابقة العناصر ذات الترتيب الزوجي للعناصر التي تمت مطابقتها بالمحدد المستخدم قبل المرشح even: مع العلم أن العد يبدأ بالعدد صفر، والمثال التالي يُسند لون مختلف لخلفية لجميع العناصر tr ذات الترتيب الزوجي فقط:

```
$( "tr:even" ).css( "background-color", "#bbbbff" );
```

Row with Index #0
Row with Index #1
Row with Index #2
Row with Index #3

أما الشيفرة الكاملة للمثال السابق فهي:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>

  <script>
    $(document).ready(function(){
      $( "tr:even" ).css( "background-color", "#bbbbff" );
    });
  </script>
  <style>
    table {
      background:#eeeeee;
    }
  </style>
</head>
<body>
  <table border="1">
    <tr><td>Row with Index #0</td></tr>
    <tr><td>Row with Index #1</td></tr>
    <tr><td>Row with Index #2</td></tr>
    <tr><td>Row with Index #3</td></tr>
  </table>
</body>
</html>
```

المرشح odd:

يُستخدم هذا المرشح لمطابقة العناصر ذات الترتيب الفردي للعناصر التي تمت مطابقتها بالمحدد المستخدم قبل المرشح odd: مع العلم أن العد يبدأ بالعدد صفر، والمثال التالي يُسند لون مختلف لخلفية لجميع العناصر tr ذات الترتيب الفردي فقط:

```
$("#tr:odd").css("background-color", "#bbbbff");
```

Row with Index #0
Row with Index #1
Row with Index #2
Row with Index #3

أما الشيفرة الكاملة للمثال السابق فهي:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>

  <script>
    $(document).ready(function() {
      $("#tr:odd").css("background-color", "#bbbbff");
    });
  </script>
  <style>
    table {
      background:#f3f7f5;
    }
  </style>
</head>
<body>
  <table border="1">
    <tr><td>Row with Index #0</td></tr>
    <tr><td>Row with Index #1</td></tr>
    <tr><td>Row with Index #2</td></tr>
    <tr><td>Row with Index #3</td></tr>
  </table>
</body>
</html>
```

المرشح `eq(index)`:

يستخدم هذا المرشح لمطابقة عنصر واحد فقط بتحديد ترتيبه ضمن العناصر التي تمت مطابقتها بالمحدد المستخدم قبل المرشح `eq(index)`, والمثال التالي يلون الخط في الخلية الثالثة (ترتيبها هو 2 لأن العد يبدأ من الرقم 0), باللون الأحمر:

```
$("#td:eq(2)").css("color", "red");
```

TD #0	TD #1	TD #2
TD #3	TD #4	TD #5
TD #6	TD #7	TD #8

أما الشيفرة الكاملة للمثال السابق فهي:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>

  <script>
    $(document).ready(function() {
      $("#td:eq(2)").css("color", "red");
    });
  </script>
</head>
<body>
  <table border="1">
    <tr><td>TD #0</td><td>TD #1</td><td>TD #2</td></tr>
    <tr><td>TD #3</td><td>TD #4</td><td>TD #5</td></tr>
    <tr><td>TD #6</td><td>TD #7</td><td>TD #8</td></tr>
  </table>
</body>
</html>
```

المرشح `gt(index)`:

يستخدم هذا المرشح لمطابقة جميع العناصر التي ترتيبها أكبر تماماً من الترتيب المحدد `index` للمرشح `gt` مع العلم أن العد يبدأ من العدد 0, والمثال التالي يغير مواصفات خط جميع الخلايا التي ترتيبها أكبر من الترتيب 4 أي يبدأ التعديل من بعد الخلية الخامسة:

```
$("#td:gt(4)").css("text-decoration", "line-through");
```

TD #0	TD #1	TD #2
TD #3	TD #4	TD #5
TD #6	TD #7	TD #8

أما الشيفرة الكاملة للمثال السابق فهي:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>

  <script>
    $(document).ready(function() {
      $("#td:gt(4)").css("text-decoration", "line-through");
    });
  </script>
</head>
<body>
  <table border="1">
    <tr><td>TD #0</td><td>TD #1</td><td>TD #2</td></tr>
    <tr><td>TD #3</td><td>TD #4</td><td>TD #5</td></tr>
    <tr><td>TD #6</td><td>TD #7</td><td>TD #8</td></tr>
  </table>
</body>
</html>
```

المرشح `:gt(index)`

يستخدم هذا المرشح لمطابقة جميع العناصر التي ترتيبها أصغر تماماً من الترتيب المحدد `index` للمرشح `It` مع العلم أن العد يبدأ من العدد 0, والمثال التالي يغير مواصفات خط جميع الخلايا التي ترتيبها أصغر من الترتيب 4 أي يتم التعديل على الخلايا الأربعة الأولى فقط:

```
$("#td:lt(4)").css("color", "red");
```

TD #0	TD #1	TD #2
TD #3	TD #4	TD #5
TD #6	TD #7	TD #8

أما الشيفرة الكاملة للمثال السابق فيه:

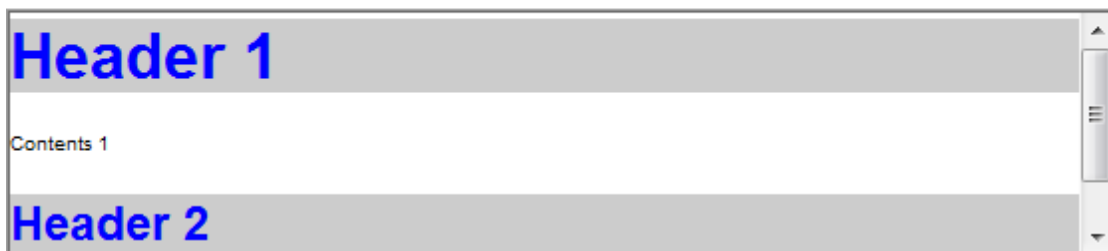
```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>

  <script>
    $(document).ready(function(){
      $("td:lt(4)").css("color", "red");
    });
  </script>
</head>
<body>
  <table border="1">
    <tr><td>TD #0</td><td>TD #1</td><td>TD #2</td></tr>
    <tr><td>TD #3</td><td>TD #4</td><td>TD #5</td></tr>
    <tr><td>TD #6</td><td>TD #7</td><td>TD #8</td></tr>
  </table>
</body>
</html>
```

المرشح header:

يُستخدم هذا المرشح لمطابقة العناصر h1, h2, h3, h4, h5, h6, وفي المثال التالي يتم تغيير لون
خط وخلفية العناصر التي تنتمي إلى المجموعة السابقة:

```
$("#:header").css({ background:'#CCC', color:'blue' });
```



أما الشيفرة الكاملة للمثال السابق فهي:

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>

  <script>
$(document).ready(function(){
  $(".header").css({ background:'#CCC', color:'blue' });
});
</script>
<style>
body { font-size: 10px; font-family: Arial; }
h1, h2 { margin: 3px 0; }
</style>
</head>
<body>
  <h1>Header 1</h1>
  <p>Contents 1</p>
  <h2>Header 2</h2>
  <p>Contents 2</p>
</body>
</html>

```

المرشح `animated`:

يُستخدم هذا المرشح لمطابقة العناصر التي يتم تحريكها أو تغيير أبعادها أو تم التأثير عليها بأحد التوابع التي تجعل العنصر يبدو وكأنه يتحرك، والمثال التالي يعين الكلاس `colored` إلى العنصر `div` الأوسط الذي يتحرك أو يحدفه منه:

```

$("#run").click(function(){
  $(".div:animated").toggleClass("colored");
});
function animateIt() {
  $(".#mover").slideToggle("slow", animateIt);
}
animateIt();

```



أما الشيفرة الكاملة للمثال السابق فهي:

```

"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<script src="http://code.jquery.com/jquery-latest.js"></script>

<script>
$(document).ready(function() {

    $("#run").click(function() {
        $("div:animated").toggleClass("colored");
    });
    function animateIt() {
        $("#mover").slideToggle("slow", animateIt);
    }
    animateIt();

});
</script>
<style>
div { background:yellow; border:1px solid #AAA; width:80px; height:80px; margin:5px; }
div.colored { background:green; }
</style>
</head>
<body>
<button id="run">Run</button>
<div></div>
<div id="mover"></div>
<div></div>
</body>
</html>

```

مرشحات المحتوى Content Filters

المرشح :contains(text)

يُستخدم هذا المرشح لمطابقة جميع العناصر التي تحتوي على النص الممثل بالقيمة text, والمثال التالي يبحث عن جميع عناصر الـ div التي تحتوي على كلمة John يُوغِير خصائص الخط لها:

```

$("div:contains('John')").css("text-decoration", "underline");

```

John Resig
George Martin
Malcom John Sinclair
J. Ohn

أما الشيفرة الكاملة للمثال السابق فهي:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>

  <script>
    $(document).ready(function() {
      $("div:contains('John')").css("text-decoration", "underline");
    });
  </script>
</head>
<body>
  <div>John Resig</div>
  <div>George Martin</div>
  <div>Malcom John Sinclair</div>
  <div>J. Ohn
</body>
</html>
```

المرشح empty:

يُستخدم هذا المرشح لمطابقة جميع العناصر التي ليس لها عناصر أبناء أي التي لا تحتوي عناصر أخرى داخلها, وهذه العناصر يتم مطابقتها من العناصر التي تنتج عن استخدام المحدد الذي يسبق هذا المرشح أو جميع عناصر الصفحة, ففي المثال التالي يتم مطابقة جميع عناصر td التي لا تحتوي على عناصر أبناء أو حتى نصوص:

```
$("td:empty").text("Was empty!").css('background', 'rgb(255,220,200)');
```

TD #0	Was empty!
TD #2	Was empty!
Was empty!	TD#5

أما الشيفرة الكاملة للمثال السابق فهي:

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>

  <script>
    $(document).ready(function() {
      $("td:empty").text("Was empty!").css('background', 'rgb(255,220,200)');
    });
  </script>
  <style>
    td { text-align:center; }
  </style>
</head>
<body>
  <table border="1">
    <tr><td>ID #0</td><td></td></tr>
    <tr><td>ID #2</td><td></td></tr>
    <tr><td></td><td>ID#5</td></tr>
  </table>
</body>
</html>

```

المرشح (selector):has

يُستخدم هذا المرشح لمطابقة جميع العناصر التي تحتوي على عنصر واحد على الأقل يتطابق مع المحدد selector الممرر كوسيط للمرشح has. والمثال التالي يطابق جميع العناصر div التي تحتوي على عنصر p واحد على الأقل ويضيف إليها الكلاس test:

```

$("div:has(p)").addClass("test");

```

Hello in a paragraph

Hello again! (with no paragraph)

أما الشيفرة الكاملة للمثال السابق فهي:

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>

  <script>
    $(document).ready(function(){
      $("div:has(p)").addClass("test");
    });
  </script>
  <style>
    .test{ border: 3px inset red; }
  </style>
</head>
<body>
  <div><p>Hello in a paragraph</p></div>
  <div>Hello again! (with no paragraph)</div>
</body>
</html>

```

المرشح parent:

يُستخدم هذا المرشح لمطابقة جميع العناصر التي تعتبر آباء لعناصر أخرى أي يطابق العناصر التي لها أبناء داخلها أو نصوص, والمثال التالي يطابق جميع الخلايا td التي تحتوي على أبناء وفي هذا المثال الأبناء هي عبارة عن نصوص:

```

$("td:parent").fadeOut(1500, 0.3);

```

Value 1	
Value 2	

أما الشيفرة الكاملة للمثال السابق فهي:

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>

  <script>
    $(document).ready(function() {
      $("td:parent").fadeTo(1500, 0.3);
    });
  </script>
  <style>
td { width:40px; background:green; }
  </style>
</head>
<body>
  <table border="1">
    <tr><td>Value 1</td><td></td></tr>
    <tr><td>Value 2</td><td></td></tr>
  </table>
</body>
</html>

```

مرشحات الرؤية أو الظهور Visibility Filters

المرشح hidden:

يُستخدم هذا المرشح لمطابقة جميع العناصر المخفية وهذا المرشح يعتبر أن العنصر مخفي في حال أن الخاصية display للـ CSS التابع للعنصر تساوي none أو إن كان عرضه وارتفاعه مساويان للصفر، والمثال التالي يُظهر جميع العناصر div المخفية مع إظهار عدد العناصر التي مازالت مخفية:

```

// in some browsers :hidden includes head, title, script, etc... so limit to
$("#span:first").text("Found " + $(":hidden", document.body).length +
    " hidden elements total.");
$("#div:hidden").show(3000);
$("#span:last").text("Found " + $("input:hidden").length + " hidden inputs.");

```

Found 8 hidden elements total.



Found 3 hidden inputs.

أما الشيفرة الكاملة للمثال السابق فهي:

```

<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>

  <script>
    $(document).ready(function() {

      // in some browsers :hidden includes head, title, script, etc... so limit
      $("span:first").text("Found " + $(":hidden", document.body).length +
        " hidden elements total.");

      $("div:hidden").show(3000);
      $("span:last").text("Found " + $("input:hidden").length + " hidden input:");

    });
  </script>
  <style>
    div { width:70px; height:40px; background:#ee77ff; margin:5px; float:left; }
    span { display:block; clear:left; color:red; }
    .starthidden { display:none; }
  </style>
</head>
<body>
  <span></span>
  <div></div>
  <div style="display:none;">Hider!</div>
  <div></div>
  <div class="starthidden">Hider!</div>
  <div></div>
  <form>
    <input type="hidden" />
    <input type="hidden" />
    <input type="hidden" />
  </form>
  <span>
</span>
</body>

```

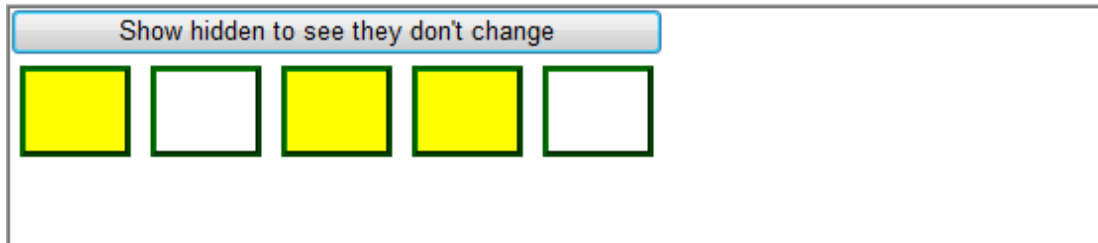
المرشح visible:

يُستخدم هذا المرشح لمطابقة جميع العناصر الظاهرة على الشاشة وهذا المرشح، والمثال التالي يحول لون خلفية العناصر div إلى الأصفر في حال النقر عليها. بمؤشر الفأرة وعند الضغط على الزر تظهر العناصر المخفية مع ملاحظة أنها غير متأثرة بعملية النقر بالفأرة على العناصر div:


```

$("div:visible").click(function () {
    $(this).css("background", "yellow");
});
$("button").click(function () {
    $("div:hidden").show("fast");
});

```



أما الشيفرة الكاملة للمثال السابق فهي:

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
    <script src="http://code.jquery.com/jquery-latest.js"></script>

    <script>
        $(document).ready(function(){

            $("div:visible").click(function () {
                $(this).css("background", "yellow");
            });
            $("button").click(function () {
                $("div:hidden").show("fast");
            });

        });
    </script>
    <style>
        div { width:50px; height:40px; margin:5px; border:3px outset green; float:
        .starthidden { display:none; }
    </style>
</head>
<body>
    <button>Show hidden to see they don't change</button>
    <div></div>
    <div class="starthidden"></div>
    <div></div>
    <div></div>
    <div style="display:none;"></div>
</body>

```

مرشحات السمات Attribute Filters

المرشح [attribute]

يُستخدم هذا المرشح لمطابقة جميع العناصر التي لديها السمة [attribute] مثل السمة id أو value، والمثال التالي يُظهر عند النقر على أحد العناصر div قيمة السمة id بجانب العناصر من النوع div في حال أن العنصر div لديه السمة id:

```
$("#div[id]").one("click", function(){
    var idString = $(this).text() + " = " + $(this).attr("id");
    $(this).text(idString);
});
```

```
no id
with id = hey
has an id = there
nope
```

أما الشيفرة الكاملة للمثال السابق فهي:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
    <script src="http://code.jquery.com/jquery-latest.js"></script>

    <script>
        $(document).ready(function(){

            $("#div[id]").one("click", function(){
                var idString = $(this).text() + " = " + $(this).attr("id");
                $(this).text(idString);
            });

        });
    </script>

</head>
<body>
    <div>no id</div>
    <div id="hey">with id</div>
    <div id="there">has an id</div>
    <div>nope</div>
</body>
</html>
```

المرشح [attribute=value]

يُستخدم هذا المرشح لمطابقة جميع العناصر التي لديها السمة المحددة مع قيمة محددة أيضاً، ففي المثال التالي يتم مطابقة جميع العناصر `input` التي لديها السمة `id` وقيمتها `newsletter` ويتم تغيير نصها إلى `is newsletter`:

```
$("#input[name='newsletter']").next().text(" is newsletter");
```

- is newsletter
- name?
- name?

أما الشيفرة الكاملة للمثال السابق فهي:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>

  <script>
    $(document).ready(function() {
      $("#input[name='newsletter']").next().text(" is newsletter");
    });
  </script>
</head>
<body>
  <div>
    <input type="radio" name="newsletter" value="Hot Fuzz" />
    <span>name?</span>
  </div>
  <div>
    <input type="radio" name="newsletters" value="Cold Fusion" />
    <span>name?</span>
  </div>
  <div>
    <input type="radio" name="accept" value="Evil Plans" />
    <span>name?</span>
  </div>
</body>
</html>
```

المرشح [attribute!=value]

يُستخدم هذا المرشح لمطابقة جميع العناصر التي لديها السمة attribute وقيمتها لا تساوي القيمة value أو ليس لديها السمة attribute بالأصل, والمثال التالي يضيف كلمة not newsletter إلى جميع العناصر input التي قيمة السمة id لها لا تساوي newsletter أو ليس لها السمة id أصلاً:

```
$("#input[name!=newsletter]").next().append("<b>; not newsletter</b>");
```

- name is newsletter
- no name; not newsletter
- name is accept; not newsletter

أما الشيفرة الكاملة للمثال السابق فهي:

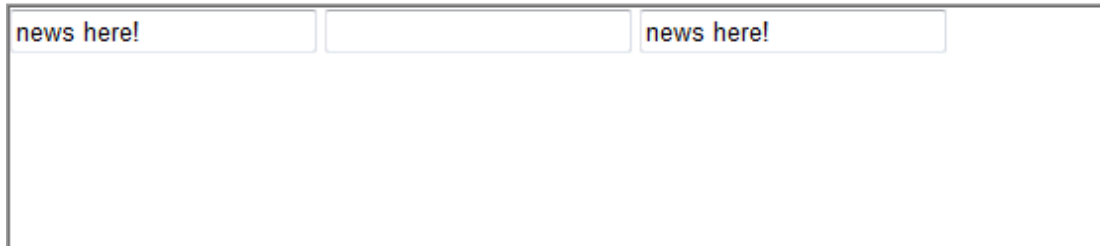
```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>

  <script>
    $(document).ready(function() {
      $("#input[name!=newsletter]").next().append("<b>; not newsletter</b>");
    });
  </script>
</head>
<body>
  <div>
    <input type="radio" name="newsletter" value="Hot Fuzz" />
    <span>name is newsletter</span>
  </div>
  <div>
    <input type="radio" value="Cold Fusion" />
    <span>no name</span>
  </div>
  <div>
    <input type="radio" name="accept" value="Evil Plans" />
    <span>name is accept</span>
  </div>
</body>
</html>
```

المرشح [attribute^=value]

يُستخدم هذا المرشح لمطابقة جميع العناصر التي لها السمة attribute وقيمة هذه السمة تبدأ بالقيمة value, والمثال التالي يكتب القيمة news here! في العناصر input التي لها السمة name وقيمتها تبدأ بالقيمة news:

```
$("#input[name^='news']").val("news here!");
```



أما الشيفرة الكاملة للمثال السابق فهي:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>

  <script>
    $(document).ready(function() {
      $("#input[name^='news']").val("news here!");
    });
  </script>
</head>
<body>
  <input name="newsletter" />
  <input name="milkman" />
  <input name="newsboy" />
</body>
</html>
```

المرشح [attribute\$=value]

يُستخدم هذا المرشح لمطابقة جميع العناصر التي لها السمة attribute وقيمة هذه السمة تنتهي بالقيمة value, والمثال التالي يكتب القيمة a letter في العناصر input التي لها السمة name وقيمتها تنتهي بالقيمة letter:

```
$("#input[name$='letter']").val("a letter");
```

a letter		a letter
----------	--	----------

أما الشيفرة الكاملة للمثال السابق فهي:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>

  <script>
    $(document).ready(function() {
      $("#input[name$='letter']").val("a letter");
    });
  </script>
</head>
<body>
  <input name="newsletter" />
  <input name="milkman" />
  <input name="jobletter" />
</body>
</html>
```

المرشح [attribute*=value]

يُستخدم هذا المرشح لمطابقة جميع العناصر التي لها السمة attribute وقيمة هذه السمة تحتوي على القيمة value, والمثال التالي يكتب القيمة has man in it! في العناصر input التي لها السمة name وقيمتها تنتهي بالقيمة man:

```
$("#input[name*='man']").val("has man in it!");
```

has man in it!	has man in it!	has man in it!

أما الشيفرة الكاملة للمثال السابق فهي:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>

  <script>
    $(document).ready(function() {
      $("input[name*='man']").val("has man in it!");
    });
  </script>
</head>
<body>
  <input name="man-news" />
  <input name="milkman" />
  <input name="letterman2" />
  <input name="newmilk" />
</body>
</html>
```

المرشح [attributeFilter1][attributeFilter2][attributeFilterN]

يُستخدم هذا المرشح لمطابقة جميع مرشحات السمات المحددة، والمثال التالي يكتب القيمة only thin

one في العنصر input الذي يمتلك السمة id والسمة name وتنتهي السمة name بالقيمة man:

```
$("#input[id][name$='man']").val("only this one");
```

<input type="text"/>	<input type="text"/>	<input type="text" value="only this one"/>
<input type="text"/>		

أما الشيفرة الكاملة للمثال السابق فهي:

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>

  <script>
    $(document).ready(function(){
      $("input[id][name$='man']").val("only this one");
    });
  </script>

</head>
<body>
  <input id="man-news" name="man-news" />
  <input name="milkman" />
  <input id="letterman" name="new-letterman" />
  <input name="newmilk" />
</body>
</html>

```

مرشحات العناصر الفرعية Child Filters

المرشح nth-child(index/even/odd/equation)

يُستخدم هذا المرشح لمطابقة جميع العناصر ذات الترتيب المحدد بالقيمة `index`, أما في حال تحديد القيمة `even` فستتم مطابقة العناصر ذات الترتيب الزوجي والقيمة `odd` تطابق العناصر ذات الترتيب الفردي مع العلم بأن العد يبدأ من القيمة 1. أما عن المعادلة `equation` فيمكن التعبير عنها كما يلي: $an+b$ أي مطابقة العنصر `b` بعد تقسيم العناصر إلى مجموعات تتألف من `a` عنصر. وهذا المرشح يمكن أن يطابق أكثر من عنصر على عكس المرشح `eq(index)` الذي يُستخدم لمطابقة عنصر واحد فقط باستخدام القيمة `index` التي تعبر عن ترتيب العنصر وهي تبدأ من القيمة 0 بينما المرشح `nth-child` فإن قيمة الترتيب تبدأ من القيمة 1, والمثال التالي يضيف النص `2nd!` - إلى العنصر الثاني من عناصر كل القوائم غير المرتبة `ul`:

```

$("ul li:nth-child(2)").append("<span> - 2nd!</span>");

```

- | | | |
|---------------|-------|---------------|
| • John | • Sam | • Glen |
| • Karl - 2nd! | | • Tane - 2nd! |
| • Brandon | | • Ralph |
| | | • David |

أما الشيفرة الكاملة للمثال السابق فهي:


```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>

  <script>
    $(document).ready(function(){
      $("ul li:nth-child(2)").append("<span> - 2nd!</span>");
    });
  </script>
  <style>
    div { float:left; }
    span { color:blue; }
  </style>
</head>
<body>
  <div><ul>
    <li>John</li>
    <li>Karl</li>
    <li>Brandon</li>
  </ul></div>
  <div><ul>
    <li>Sam</li>
  </ul></div>
  <div><ul>
    <li>Glen</li>
    <li>Tane</li>
    <li>Ralph</li>
    <li>David</li>
  </ul></div>
</body>
</html>

```

المرشح `:first-child`

يُستخدم هذا المرشح لمطابقة العنصر الأول لكل عنصر أب أي يطابق الابن الأول لكل عنصر يحتوي على عناصر أبناء, ففي المثال التالي فهو يغير خصائص أول عنصر `span` موجود داخل عنصر `div`:

```
$( "div span:first-child" )
  .css( "text-decoration", "underline" )
  .hover( function () {
    $( this ).addClass( "sogreen" );
  }, function () {
    $( this ).removeClass( "sogreen" );
  } );
```

[John](#), [Karl](#), [Brandon](#)
[Glen](#), [Tane](#), [Ralph](#)

أما الشيفرة الكاملة للمثال السابق فهي:

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>

  <script>
    $(document).ready(function() {

      $("div span:first-child")
        .css("text-decoration", "underline")
        .hover(function () {
          $(this).addClass("sogreen");
        }, function () {
          $(this).removeClass("sogreen");
        });

    });
  </script>
  <style>
    span { color:#008; }
    span.sogreen { color:green; font-weight: bolder; }
  </style>
</head>
<body>
  <div>
    <span>John,</span>
    <span>Karl,</span>
    <span>Brandon</span>
  </div>
  <div>
    <span>Glen,</span>
    <span>Tane,</span>
    <span>Ralph</span>
  </div>
</body>
</html>

```

المرشح `:last-child`

يُستخدم هذا المرشح لمطابقة العنصر الأخير لكل عنصر أب أي يطابق الابن الأخير لكل عنصر يحتوي

على عناصر أبناء, ففي المثال التالي فهو يغير خصائص آخر عنصر `span` موجود داخل عنصر `div`:

```
$("#div span:last-child")
  .css({color:"red", fontSize:"80%"})
  .hover(function () {
    $(this).addClass("solast");
  }, function () {
    $(this).removeClass("solast");
  });
```

John, Karl, Brandon, Sam
Glen, Tane, Ralph, David

أما الشيفرة الكاملة للمثال السابق فهي:

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>

  <script>
    $(document).ready(function() {

      $("div span:last-child")
        .css({color:"red", fontSize:"80%"})
        .hover(function () {
          $(this).addClass("solast");
        }, function () {
          $(this).removeClass("solast");
        });

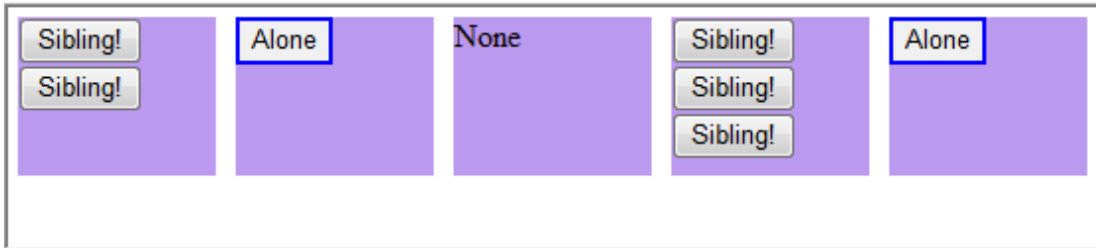
    });
  </script>
  <style>
    span.solast { text-decoration:line-through; }
  </style>
</head>
<body>
  <div>
    <span>John,</span>
    <span>Karl,</span>
    <span>Brandon,</span>
    <span>Sam</span>
  </div>
  <div>
    <span>Glen,</span>
    <span>Tane,</span>
    <span>Ralph,</span>
    <span>David</span>
  </div>
</body>

```

المرشح `:only-child`:

يُستخدم هذا المرشح لمطابقة العنصر الذي يعتبر الابن الوحيد للعنصر الأب، والمثال التالي يغير نص خصائص الإطار للزر الذي هو الابن الوحيد للعنصر `div` الثاني في الشيفرة:

```
$("#div button:only-child").text("Alone").css("border", "2px blue solid");
```



أما الشيفرة الكاملة للمثال السابق فهي:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>

  <script>
    $(document).ready(function() {
      $("#div button:only-child").text("Alone").css("border", "2px blue solid");
    });
  </script>
  <style>
    div { width:100px; height:80px; margin:5px; float:left; background:#b9e }
  </style>
</head>
<body>
  <div>
    <button>Sibling!</button>
    <button>Sibling!</button>
  </div>
  <div>
    <button>Sibling!</button>
  </div>
  <div>
    None
  </div>
  <div>
    <button>Sibling!</button>
    <button>Sibling!</button>
    <button>Sibling!</button>
  </div>
  <div>
    <button>Sibling!</button>
  </div>
</body>
</html>
```

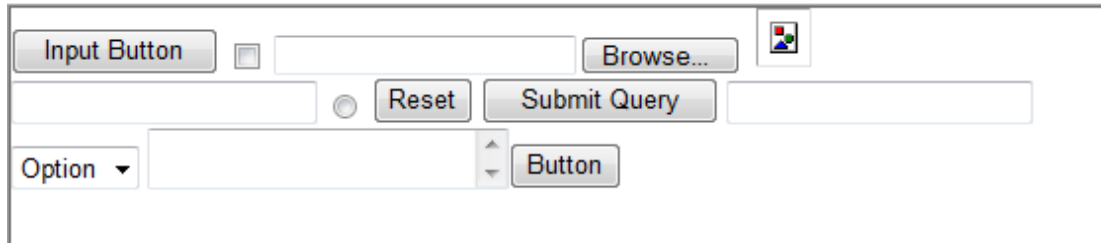
النماذج Forms

:input

يُستخدم لمطابقة جميع العناصر `input`, `textarea` (الحقل النصي الذي يقبل أكثر من سطر), `button` (الزر) و `select` (القائمة المنسدلة), والمثال التالي يظهر عدد العناصر `input` وعدد العناصر الأبناء في النموذج `form`:

```
var allInputs = $("input");
var formChildren = $("form > *");
$("#messages").text("Found " + allInputs.length + " inputs and the form has " +
    formChildren.length + " children.");

// so it won't submit
$("form").submit(function () { return false; });
```



Found 13 inputs and the form has 13 children.

أما المثال الكامل للشفيرة السابقة فهو:

```

<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>

  <script>
    $(document).ready(function() {

      var allInputs = $(":input");
      var formChildren = $("form > *");
      $("#messages").text("Found " + allInputs.length + " inputs and the form |
                          formChildren.length + " children.");

      // so it won't submit
      $("form").submit(function () { return false; });

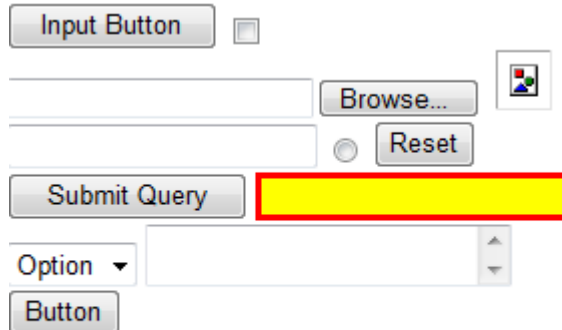
    });
  </script>
  <style>
    textarea { height:25px; }
  </style>
</head>
<body>
  <form>
    <input type="button" value="Input Button"/>
    <input type="checkbox" />
    <input type="file" />
    <input type="hidden" />
    <input type="image" />
    <input type="password" />
    <input type="radio" />
    <input type="reset" />
    <input type="submit" />
    <input type="text" />
    <select><option>Option</option></select>
    <textarea></textarea>
    <button>Button</button>
  </form>
  <div id="messages">
</div>

```

:text

يُستخدم لمطابقة جميع العناصر input التي هي من النوع text (حقل نصي بسطر واحد فقط), واستخدامه بالطريقة التالية \$(':text') يكافئ كتابة \$('*:text') وهذه الطريقة بطيئة في التنفيذ ويفضل استخدام الطريقة التالية \$('input:text'), والمثال التالي يغير خصائص الحقل النصي text:


```
var input = $("form input:text").css({background:"yellow", border:"3px re
$("div").text("For this type jQuery found " + input.length + ".")
.css("color", "red");
$("form").submit(function () { return false; }); // so it won't submit
```



The screenshot shows a web form with several elements: an "Input Button" with a checkbox, a text input field, a "Browse..." button with a file icon, a radio button, a "Reset" button, a "Submit Query" button, a yellow-highlighted text input field with a red border, a dropdown menu labeled "Option", and a "Button".

For this type jQuery found 1.

أما الشيفرة الكاملة للمثال السابق فهي:

```

<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>

  <script>
    $(document).ready(function() {

      var input = $("form input:text").css({background:"yellow", border:"3px r
      $("div").text("For this type jQuery found " + input.length + ".")
        .css("color", "red");
      $("form").submit(function () { return false; }); // so it won't submit

    });
  </script>
  <style>
    textarea { height:25px; }
  </style>
</head>
<body>
  <form>
    <input type="button" value="Input Button"/>
    <input type="checkbox" />
    <input type="file" />
    <input type="hidden" />
    <input type="image" />
    <input type="password" />
    <input type="radio" />
    <input type="reset" />
    <input type="submit" />
    <input type="text" />
    <select><option>Option</option></select>
    <textarea></textarea>
    <button>Button</button>
  </form>
  <div>
  </div>
</body>
</html>

```

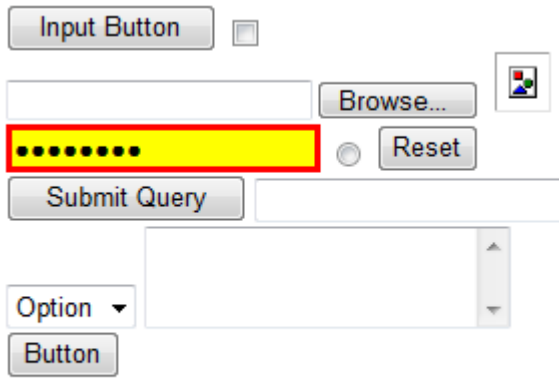
:password

يُستخدم لمطابقة جميع العناصر input التي هي من النوع text (حقل نصي بسطر واحد فقط), واستخدامه بالطريقة التالية (': password') \$(': password') يكافئ كتابة \$('*:password') وهذه الطريقة بطيئة في التنفيذ ويفضل استخدام الطريقة التالية \$('input: password'), والمثال التالي يغير خصائص الحقل النصي text:

```

var input = $("input:password").css({background:"yellow", border:"3px red so
$("div").text("For this type jQuery found " + input.length + ".")
  .css("color", "red");
$("form").submit(function () { return false; }); // so it won't submit

```



For this type jQuery found 1.

أما الشيفرة الكاملة للمثال السابق فهي:

```

<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>

  <script>
    $(document).ready(function(){

      var input = $("input:password").css({background:"yellow", border:"3px red
      $("div").text("For this type jQuery found " + input.length + ".")
        .css("color", "red");
      $("form").submit(function () { return false; }); // so it won't submit

    });
  </script>
  <style>
    textarea { height:45px; }
  </style>
</head>
<body>
  <form>
    <input type="button" value="Input Button"/>
    <input type="checkbox" />
    <input type="file" />
    <input type="hidden" />
    <input type="image" />
    <input type="password" />
    <input type="radio" />
    <input type="reset" />
    <input type="submit" />
    <input type="text" />
    <select><option>Option</option></select>
    <textarea></textarea>
    <button>Button</button>
  </form>
  <div>
  </div>
</body>
</html>

```

:radio

يطابق تماماً جميع العناصر الأخرى في قسم النماذج ولكن يتم استخدامه بكتابة: `$(input:radio')`

:checkbox

يطابق تماماً جميع العناصر أخرى في قسم النماذج ولكن يتم استخدامه بكتابة:

`$(input:checkbox')`

:submit

يطابق تماماً جميع العناصر أخرى في قسم النماذج ولكن يتم استخدامه بكتابة: `$(input:submit')`

:image

يطابق تماماً جميع العناصر أخرى في قسم النماذج ولكن يتم استخدامه بكتابة: `$('input:image')`

:reset

يطابق تماماً جميع العناصر أخرى في قسم النماذج ولكن يتم استخدامه بكتابة: `$('input:reset')`

:button

يطابق تماماً جميع العناصر أخرى في قسم النماذج ولكن يتم استخدامه بكتابة: `$('input:button')`

:file

يطابق تماماً جميع العناصر أخرى في قسم النماذج ولكن يتم استخدامه بكتابة: `$('input:file')`

:hidden

يطابق تماماً جميع العناصر أخرى في قسم النماذج ولكن يتم استخدامه بكتابة: `$('input:hidden')`

مرشحات النماذج Form Filters

المرشح :enabled

يُستخدم هذا المرشح لمطابقة جميع العناصر المفعلة (الممكنة), والمثال التالي يكتب النص `this is it` في

الحقل النصي المفعّل:

```
$( "input:enabled" ).val( "this is it" );
```

أما الشيفرة الكاملة للمثال السابق فهي:

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>

  <script>
    $(document).ready(function(){
      $("input:enabled").val("this is it");
    });
  </script>

</head>
<body>
  <form>
    <input name="email" disabled="disabled" />
    <input name="id" />
  </form>
</body>
</html>

```

المرشح disabled:

يُستخدم هذا المرشح لمطابقة جميع العناصر الغير المفعلة (الغير الممكنة), والمثال التالي يكتب النص this

is it في الحقل النصي الغير المفعّل:

```

$("input:disabled").val("this is it");

```

this is it

أما الشيفرة الكاملة للمثال السابق فهي:

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>

  <script>
    $(document).ready(function() {
      $("input:disabled").val("this is it");
    });
  </script>

</head>
<body>
  <form>
    <input name="email" disabled="disabled" />
    <input name="id" />
  </form>
</body>
</html>

```

المرشح checked:

يُستخدم هذا المرشح لمطابقة جميع العناصر المختارة وينطبق على العنصرين radio, checkbox، والمثال التالي يظهر عدد العناصر التي تم اختيارها، وعند تغيير اختيار أحد العناصر checkbox يتم تحديث العدد بشكل تلقائي:

```

function countChecked() {
  var n = $("input:checked").length;
  $("div").text(n + (n <= 1 ? " is" : " are") + " checked!");
}
countChecked();
$("input:checkbox").click(countChecked);

```

2 are checked!

أما الشيفرة الكاملة للمثال السابق فهي:

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>

  <script>
    $(document).ready(function(){

      function countChecked() {
        var n = $("input:checked").length;
        $("div").text(n + (n <= 1 ? " is" : " are") + " checked!");
      }
      countChecked();
      $(".checkbox").click(countChecked);

    });
  </script>
  <style>
  div { color:red; }
  </style>
</head>
<body>
  <form>
    <input type="checkbox" name="newsletter" checked="checked" value="Hourly" />
    <input type="checkbox" name="newsletter" value="Daily" />
    <input type="checkbox" name="newsletter" value="Weekly" />
    <input type="checkbox" name="newsletter" checked="checked" value="Monthly" />
    <input type="checkbox" name="newsletter" value="Yearly" />
  </form>
  <div></div>
</body>

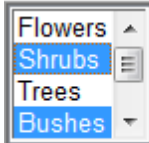
```

المرشح `:selected`

يُستخدم هذا المرشح لمطابقة جميع العناصر المحددة وينطبق على العنصر `select` (القائمة المنسدلة), والمثال

التالي يظهر العناصر التي تم تحديدها باللون الأحمر ويتغير بتغير التحديد بشكل تلقائي:


```
$("#select").change(function () {  
    var str = "";  
    $("#select option:selected").each(function () {  
        str += $(this).text() + " ";  
    });  
    $("#div").text(str);  
})  
-trigger('change');
```



Shrubs Bushes

أما الشيفرة الكاملة للمثال السابق فهي:

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>

  <script>
    $(document).ready(function(){

      $("select").change(function () {
        var str = "";
        $("select option:selected").each(function () {
          str += $(this).text() + " ";
        });
        $("div").text(str);
      })
      .trigger('change');

    });
  </script>
  <style>
  div { color:red; }
  </style>
</head>
<body>
  <select name="garden" multiple="multiple">
    <option>Flowers</option>
    <option selected="selected">Shrubs</option>
    <option>Trees</option>
    <option selected="selected">Bushes</option>
    <option>Grass</option>
    <option>Dirt</option>
  </select>
  <div></div>
</body>
</html>

```

ملاحظة: يمكن استخدام الرمزين \ لتجاوز الحارف الخاصة التي من الممكن أن تستخدم في أسماء العناصر.

السمات Attributes

attr(name)

يُستخدم للحصول على قيمة إحدى صفات أول عنصر تتم مطابقته باستخدام أحد محددات الـ jQuery, وفي حال أن العنصر لا يحتوي على الصفة المحددة بالقيمة name عندها يتم إرجاع القيمة undefined, ومن هذه الصفات title, alt, src, href, width, style وغيرها أخرى كما يمكن

الحصول على قيمة صفة غير موجودة في الـ HTML مثلاً يمكن الحصول على قيمة الصفة NewsId مع العلم أن هذه الصفة غير موجودة في الصفات القياسية لأي واسمة HTML, والمثال التالي يعرض قيمة الصفة title للواسمة (tag) التي اسمها em ويعرضه باللون الأحمر:

```
var title = $("em").attr("title");
$("#div").text(title);
```

Once there was a *large* dinosaur...

The title of the emphasis is:

huge, gigantic

أما الشيفرة الكاملة للمثال السابق:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>

  <script>
    $(document).ready(function(){

      var title = $("em").attr("title");
      $("#div").text(title);

    });
  </script>
  <style>
    em { color:blue; font-weight:bold; }
    div { color:red; }
  </style>
</head>
<body>
  <p>
    Once there was a <em title="huge, gigantic">large</em> dinosaur...
  </p>
  The title of the emphasis is:<div></div>
</body>
</html>
```

attr(properties)

يُستخدم لإسناد قيم لصفات جميع العناصر التي تمت مطابقتها باستخدام المحدد الذي يسبق استخدام التابع

attr حيث يُعتمد طريقة مفتاح/قيمة لتحديد الصفة وقيمتها حيث تأخذ الشكل التالي:

{attr1: value1, attr2: value2, attrN: valueN}

مع ملاحظة أنه لتحديد قيمة الكلاس CSS للعناصر المطابقة فيجب استخدام الصفة `className` كمفتاح أو استخدام التابع `addClass(class)` أو التابع `removeClass(class)`. مع الانتباه إلى أنه في حال استخدام تابع مكان القيمة `value` عندها سيتم تنفيذ التابع ولن يتم إسناده كقيمة للمفتاح (الصفة), والمثال التالي يحدد بعض صفات جميع الواسمات `img` في الصفحة:

```
$("#img").attr({
  src: "/images/hat.gif",
  title: "jQuery",
  alt: "jQuery Logo"
});
$("#div").text($("#img").attr("alt"));
```



أما الشيفرة الكاملة للمثال السابق فهي:

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>

  <script>
    $(document).ready(function(){

      $("img").attr({
        src: "/images/hat.gif",
        title: "jQuery",
        alt: "jQuery Logo"
      });
      $("div").text($("img").attr("alt"));

    });
  </script>
  <style>
    img { padding:10px; }
    div { color:red; font-size:24px; }
  </style>
</head>
<body>
  <img />
  <img />
  <img />
  <div><B>Attribute of Ajax</B></div>
</body>
</html>

```

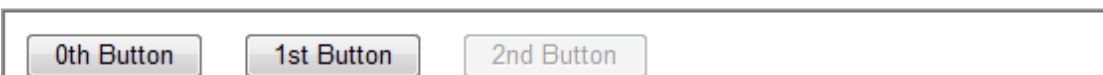
attr(key, value)

يُستخدم لتحديد قيمة لخاصية واحدة فقط لجميع العناصر التي تمت مطابقتها مع المحدد الذي تم استدعاؤه قبل التابع attr, والمثال التالي يعطي الخاصية disabled القيمة disabled لجميع الأزرار التي ترتيبها أكبر من 1 وفي مثالنا يوجد زر واحد مطابق لهذا الشرط:

```

$("button:gt(1)").attr("disabled", "disabled");

```



أما الشيفرة الكاملة للمثال السابق فهي:

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>

  <script>
  $(document).ready(function(){
    $("button:eq(1)").attr("disabled","disabled");
  });
  </script>
  <style>
  button { margin:10px; }
  </style>
</head>
<body>
  <button>0th Button</button>
  <button>1st Button</button>
  <button>2nd Button</button>
</body>
</html>

```

attr(key, fn)

يُستخدم لتحديد قيمة محسوبة لواصفة واحدة فقط لجميع العناصر التي تمت مطابقتها مع المحدد الذي تم استدعاؤه قبل التابع attr, وهنا يمكن استخدام تابع مكان الوسيط fn لإتاحة إمكانية حساب القيمة, والمثال التالي تغيير نص الواسمات span الموجودة ضمن الواسمات div قيمة الصفة id لكل عنصر div:

```

$("div").attr("id", function (arr) {
  return "div-id" + arr;
})
.each(function () {
  $("span", this).html("(ID = '<b>' + this.id + '</b>')");
});

```

```

Zero-th (ID = 'div-id0')
First (ID = 'div-id1')
Second (ID = 'div-id2')

```

أما الشيفرة الكاملة للمثال السابق فهي:

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>

  <script>
    $(document).ready(function() {

      $("#div").attr("id", function (arr) {
        return "div-id" + arr;
      })
      .each(function () {
        $("#span", this).html("(ID = '<b>' + this.id + '</b>')");
      });

    });
  </script>
  <style>
  div { color:blue; }
  span { color:red; }
  b { font-weight:bolder; }
  </style>
</head>
<body>
  <div>Zero-th <span></span></div>
  <div>First <span></span></div>
  <div>Second <span></span></div>
</body>
</html>

```

removeAttr(name)

يُستخدم لإزالة صفة من صفات جميع العناصر التي تمت مطابقتها مع المحدد الذي تم استدعاؤه قبل التابع `removeAttr` وتحدد الصفة التي ستتم إزالتها بالقيمة `name`, وفي المثال التالي يتم إزالة الصفة `disabled` من الحقل النصي عند الضغط على الزر `Enable` وبعدها يتم تغيير نص الحقل النصي إلى `editable now`:

```

$("#button").click(function () {
  $(this).next().removeAttr("disabled")
    .focus()
    .val("editable now");
});

```

أما الشيفرة الكاملة للمثال السابق فهي:

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>

  <script>
    $(document).ready(function(){

      $("button").click(function () {
        $(this).next().removeAttr("disabled")
          .focus()
          .val("editable now");
      });

    });
  </script>

</head>
<body>
  <button>Enable</button>
  <input type="text" disabled="disabled" value="can't edit this" />
</body>
</html>

```

الصفوف (كلاسات) Class

addClass(class)

يُستخدم هذا التابع لإضافة كلاس CSS Class إلى جميع العناصر التي تمت مطابقتها مع المحدد الذي تم استدعاؤه قبل هذا التابع، ويمكن تمرير أكثر من اسم كلاس وذلك بوضع فراغ بين كل كلاس، والمثال التالي يضيف الكلاس selected إلى آخر عنصر p من العناصر الموجودة في الصفحة، حيث يعطيه اللون الأزرق:

```

$("p:last").addClass("selected");

```

Hello
and
Goodbye

أما الشيفرة الكاملة للمثال السابق فهي:


```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>

  <script>
    $(document).ready(function() {
      $("p:last").addClass("selected");
    });
  </script>
  <style>
    p { margin: 8px; font-size:16px; }
    .selected { color:blue; }
    .highlight { background:yellow; }
  </style>
</head>
<body>
  <p>Hello</p>
  <p>and</p>
  <p>Goodbye</p>
</body>
</html>

```

hasClass(class)

يعيد هذا التابع القيمة true في حال أن عنصر واحد من العناصر التي تمت مطابقتها مع المحدد الذي تم استدعاؤه قبل هذا التابع يحتوي على الكلاس التي تم تحديد اسمه بالقيمة class وإلا فإن التابع يعيد false والقيمة class تقبل اسم كلاس واحد فقط, والمثال التالي يضيف القيمة true أو false إلى العنصر :div

```

$("#div#result1").append($("p:first").hasClass("selected").toString());
$("#div#result2").append($("p:last").hasClass("selected").toString());
$("#div#result3").append($("p").hasClass("selected").toString());

```

```

Hello
Goodbye
First paragraph has selected class: false
Last paragraph has selected class: true
Some paragraph has selected class: true

```

أما الشيفرة الكاملة للمثال السابق فهي:

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>

  <script>
    $(document).ready(function(){
      $("#div#result1").append($("#p:first").hasClass("selected").toString());
      $("#div#result2").append($("#p:last").hasClass("selected").toString());
      $("#div#result3").append($("#p").hasClass("selected").toString());
    });
  </script>
  <style>
    p { margin: 8px; font-size:16px; }
    .selected { color:red; }
    .highlight { background:yellow; }
  </style>
</head>
<body>
  <p>Hello</p>
  <p class="selected">Goodbye</p>
  <div id="result1">First paragraph has selected class: </div>
  <div id="result2">Last paragraph has selected class: </div>
  <div id="result3">Some paragraph has selected class: </div>
</body>
</html>

```

removeClass(class)

يُستخدم هذا التابع لإزالة جميع الكلاسات المحددة بالقيمة class من جميع العناصر التي تمت مطابقتها مع المحدد الذي تم استدعاؤه قبل هذا التابع حيث يمكن تمرير اسم أكثر من كلاس وذلك بوضع فراغ بينها، والمثال التالي يزيل الكلاس blue من جميع العناصر p ذات الترتيب الزوجي التي تم تحديدها باستخدام المحدد even:

```
$("#p:even").removeClass("blue");
```

Hello
and
then
Goodbye

أما الشيفرة الكاملة للمثال السابق فهي:

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>

  <script>
$(document).ready(function() {
  $("p:even").removeClass("blue");
});
</script>
<style>
p { margin: 4px; font-size:16px; font-weight:bolder; }
.blue { color:blue; }
.under { text-decoration:underline; }
.highlight { background:yellow; }
</style>
</head>
<body>
  <p class="blue under">Hello</p>
  <p class="blue under highlight">and</p>
  <p class="blue under">then</p>
  <p class="blue under">Goodbye</p>
</body>
</html>

```

toggleClass(class)

يُستخدم هذا التابع لإضافة الكلاس المحدد بالقيمة class إلى العناصر التي تمت مطابقتها مع المحدد الذي تم استدعاؤه قبل هذا التابع إن كان غير موجود أو يحدفه في حال أنه كان موجود، والمثال التالي يضيف أو يحدف الكلاس highlight عند الضغط على أي عنصر p في الصفحة:

```

$("p").click(function () {
  $(this).toggleClass("highlight");
});

```

Click to toggle

highlight

on these

paragraphs

أما الشيفرة الكاملة للمثال السابق فهي:

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>

  <script>
    $(document).ready(function() {

      $("p").click(function () {
        $(this).toggleClass("highlight");
      });

    });
  </script>
  <style>
    p { margin: 4px; font-size:16px; font-weight:bold;
      cursor:pointer; }
    .blue { color:blue; }
    .highlight { background:yellow; }
  </style>
</head>
<body>
  <p class="blue">Click to toggle</p>
  <p class="blue highlight">highlight</p>
  <p class="blue">on these</p>
  <p class="blue">paragraphs</p>
</body>
</html>

```

toggleClass(class, switch)

يُستخدم هذا التابع لإضافة الكلاس المحدد بالقيمة class إلى جميع العناصر التي تمت مطابقتها مع المحدد الذي تم استدعاؤه قبل هذا التابع في حال أن قيمة switch كانت مساوية إلى true أو يزيله في حال كانت قيمة switch مساوية إلى false والمثال التالي يطبق هذا التابع على جميع العناصر p الموجودة في الصفحة عند النقر على أحدها:

```

var count = 0;
$("p").click(function() {
  $(this).toggleClass("highlight", count++ % 3 == 0);
});

```

Click to toggle

highlight

on these

paragraphs

أما الشيفرة الكاملة للمثال السابق فهي:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
      "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>

  <script>
    $(document).ready(function() {

      var count = 0;
      $("p").click(function() {
        $(this).toggleClass("highlight", count++ % 3 == 0);
      });

    });
  </script>
  <style>
    p { margin: 4px; font-size:16px; font-weight:bold;
      cursor:pointer; }
    .blue { color:blue; }
    .highlight { background:yellow; }
  </style>
</head>
<body>
  <p class="blue">Click to toggle</p>
  <p class="blue highlight">highlight</p>
  <p class="blue">on these</p>
  <p class="blue">paragraphs</p>
</body>
</html>
```

توابع الـ HTML

html()

يُستخدم هذا التابع لإعادة شيفرة الـ html لأول عنصر تمت مطابقته ولا يعمل هذا التابع على وثيقة

XML بالرغم من أنه يعمل على وثيقة من النوع XHTML، والمثال التالي يعرض شيفرة الـ HTML

لجميع العناصر p الموجودة في الصفحة وذلك عند الضغط عليها:

```

$("p").click(function () {
    var htmlStr = $(this).html();
    $(this).text(htmlStr);
});

```

[Click](#) to change the html
to a text node.

This does nothing.

أما الشيفرة الكاملة للمثال السابق فهي:

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
    <script src="http://code.jquery.com/jquery-latest.js"></script>

    <script>
    $(document).ready(function(){

        $("p").click(function () {
            var htmlStr = $(this).html();
            $(this).text(htmlStr);
        });

    });
    </script>
    <style>
    p { margin:8px; font-size:20px; color:blue;
        cursor:pointer; }
    b { text-decoration:underline; }

    </style>
</head>
<body>
    <p>
        <b>Click</b> to change the <span id="tag">html</span>
    </p>
    <p>
        to a <span id="text">text</span> node.
    </p>
    <p>
        This <button name="nada">button</button> does nothing.
    </p>
</body>
</html>

```

html(val)

يُستخدم هذا التابع لتغيير شيفرة الـ HTML لجميع العناصر التي تمت مطابقتها مع المحدد الذي تم استدعاؤه قبل هذا التابع، والمثال التالي يغير شيفرة الـ HTML لجميع عناصر div الموجودة في الصفحة:

```
$("#div").html("<span class='red'>Hello <b>Again</b></span>");
```

Hello
Hello Again
Hello Again
Hello Again

أما الشيفرة الكاملة للمثال السابق فهي:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>

  <script>
    $(document).ready(function() {
      $("#div").html("<span class='red'>Hello <b>Again</b></span>");
    });
  </script>
  <style>
    .red { color:red; }
  </style>
</head>
<body>
  <span>Hello</span>
<div></div>
<div></div>
<div></div>
</body>
</html>
```

التوابع النصية Text

text()

يُستخدم لإعادة القيم النصية لجميع العناصر التي تمت مطابقتها مع المحدد الذي تم استدعاؤه قبل هذا التابع حيث يتم دمج كل القيم النصية في سلسلة نصية واحدة، والمثال التالي يعيد القيمة النصية لأول عنصر p موجود في الصفحة ويُسنده إلى شيفرة الـ HTML لآخر عنصر p موجود في الصفحة:

```
var str = $("p:first").text();
$("p:last").html(str);
```

Test Paragraph.

Test Paragraph.

أما الشيفرة الكاملة للمثال السابق فهي:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>

  <script>
    $(document).ready(function() {

      var str = $("p:first").text();
      $("p:last").html(str);

    });
  </script>
  <style>
    p { color:blue; margin:8px; }
    b { color:red; }
  </style>
</head>

<p><b>Test</b> Paragraph.</p>
<p></p>
</body>
</html>
```

text(val)

يُستخدم هذا التابع لتغيير القيمة النصية لجميع العناصر التي تمت مطابقتها مع المحدد الذي تم استدعاؤه قبل هذا التابع, والمثال التالي يغير القيمة النصية لجميع العناصر p الموجودة في الصفحة:

```
$("p").text("<b>Some</b> new text.");
```

Some new text.

أما الشيفرة الكاملة للمثال السابق فهي:


```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>

  <script>
    $(document).ready(function(){
      $("p").text("<b>Some</b> new text.");
    });
  </script>
  <style>
    p { color:blue; margin:8px; }
  </style>
</head>
<body>
  <p>Test Paragraph.</p>
</body>
</html>

```

توابع القيمة Value

val()

يُستخدم هذا التابع للحصول على قيمة الصفة value لأول عنصر تتم مطابقته, ويمكن استخدامه مع جميع العناصر حتى العنصر checkbox (زر الاختيار) والعنصر select (القائمة المنسدلة أو القائمة التي تسمح باختيار أكثر من عنصر), فمن أجل العنصر checkbox يمكن استخدام المحدد checked: لتحديد العنصر المحدد كما في المثال التالي:

```
$('#input:checkbox:checked').val();
```

أما من أجل العنصر select فيجب استخدام المحدد selected: كما في المثال التالي:

```
$('#select#foo option:selected').val();
```

أما من أجل العنصر radio فيجب أيضاً استخدام المحدد checked: كما في المثال التالي:

```
$('#input:radio[name=bar]:checked').val();
```

أما المثال التالي فيعيد قيمة واحدة فقط من العنصر select الذي يسمح بتحديد عنصر واحد فقط ومصفوفة قيم من العنصر select الذي يسمح باختيارات متعددة:

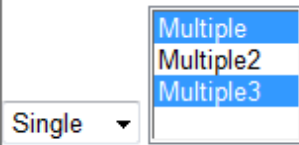
```

function displayVals() {
    var singleValues = $("#single").val();
    var multipleValues = $("#multiple").val() || [];
    $("#p").html("<b>Single:</b> " +
        singleValues +
        " <b>Multiple:</b> " +
        multipleValues.join(", "));
}

$("#select").change(displayVals);
displayVals();

```

Single: Single Multiple: Multiple, Multiple3



The screenshot shows a web form with two input elements. On the left is a dropdown menu with the text 'Single' and a downward arrow. To its right is a list box containing three items: 'Multiple', 'Multiple2', and 'Multiple3'. The 'Multiple' item is currently selected and highlighted in blue.

أما الشيفرة الكاملة للمثال السابق فهي:

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>

  <script>
    $(document).ready(function() {

      function displayVals() {
        var singleValues = $("#single").val();
        var multipleValues = $("#multiple").val() || [];
        $("p").html("<b>Single:</b> " +
          singleValues +
          " <b>Multiple:</b> " +
          multipleValues.join(", "));
      }

      $("select").change(displayVals);
      displayVals();

    });
  </script>
  <style>
    p { color:red; margin:4px; }
    b { color:blue; }
  </style>
</head>
<body>
  <p></p>
  <select id="single">
    <option>Single</option>
    <option>Single2</option>
  </select>
  <select id="multiple" multiple="multiple">
    <option selected="selected">Multiple</option>
    <option>Multiple2</option>
    <option selected="selected">Multiple3</option>
  </select>
</body>
</html>

```

val(val)

يُستخدم لتغيير قيمة الصفة value لجميع العناصر التي تمت مطابقتها مع المحدد الذي تم استدعاؤه قبل هذا التابع، والمثال التالي يغير قيمة الصفة value للحقل النصي أي يغير النص الذي يظهر في الحقل النصي وذلك عند الضغط على أي زر من الأزرار الموجودة في الصفحة:

```

$("button").click(function () {
    var text = $(this).text();
    $("input").val(text);
});

```



أما الشيفرة الكاملة للمثال السابق فهي:

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
    <script src="http://code.jquery.com/jquery-latest.js"></script>

    <script>
    $(document).ready(function() {

        $("button").click(function () {
            var text = $(this).text();
            $("input").val(text);
        });

    });
    </script>
<style>
    button { margin:4px; cursor:pointer; }
    input { margin:4px; color:blue; }
</style>
</head>
<body>
    <div>
        <button>Feed</button>
        <button>the</button>
        <button>Input</button>
    </div>
    <input type="text" value="click a button" />
</body>
</html>

```

val(val)

يختلف هذا التابع عن التابع السابق في أنه يُستخدم للعناصر select, checkbox, radio حيث يمكن استخدامه لتحديد قيمة واحدة للعنصر select أو أكثر من قيمة, واختيار عنصر checkbox واحد أو أكثر, كما يمكن استخدامه لتحديد عنصر radio كما في المثال التالي:

```

$("#single").val("Single2");
$("#multiple").val(["Multiple2", "Multiple3"]);
$("input").val(["check1", "check2", "radio1" ]);

```

أما الشيفرة الكاملة للمثال السابق فهي:

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>

  <script>
    $(document).ready(function(){

      $("#single").val("Single2");
      $("#multiple").val(["Multiple2", "Multiple3"]);
      $("input").val(["check1", "check2", "radio1" ]);

    });
  </script>
  <style>
    body { color:blue; }
  </style>
</head>
<body>
  <select id="single">
    <option>Single</option>
    <option>Single2</option>
  </select>
  <select id="multiple" multiple="multiple">
    <option selected="selected">Multiple</option>
    <option>Multiple2</option>
    <option selected="selected">Multiple3</option>
  </select><br/>
  <input type="checkbox" name="checkboxname" value="check1"/> check1
  <input type="checkbox" name="checkboxname" value="check2"/> check2
  <input type="radio" name="r" value="radio1"/> radio1
  <input type="radio" name="r" value="radio2"/> radio2
</body>
</html>

```

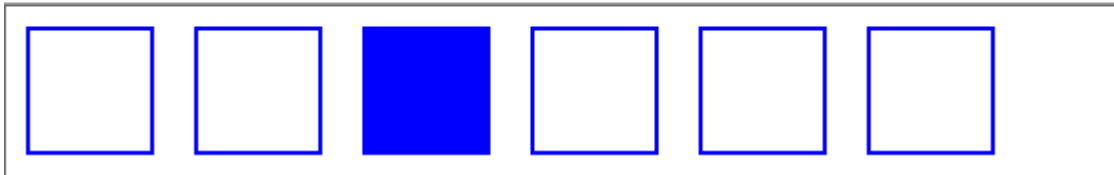
توابع العبور

توابع الترشيح

eq(index)

يُستخدم هذا التابع لمطابقة العنصر ذو الترتيب index من بين كل العناصر التي تمت مطابقتها مع المحدد الذي تم استدعاؤه قبل هذا التابع مع العلم بأن قيمة الوسيط index تبدأ بالقيمة 0 أي أن العنصر الخامس تكون قيمة index الموافقة له هي 4 وفي حال تمرير للوسيط index لا توافق وجود أي عنصر عندها يعيد هذا التابع مصفوفة لا يوجد فيها أي عنصر بدلاً من القيمة null, والمثال التالي يضيف الكلاس blue إلى العنصر ذو الترتيب 2 أي العنصر الثالث من عناصر div الموجودة في الصفحة:

```
$("#div").eq(2).addClass("blue");
```



أما الشيفرة الكاملة للمثال السابق فهي:

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>

  <script>
    $(document).ready(function() {

      $("#div").eq(2).addClass("blue");

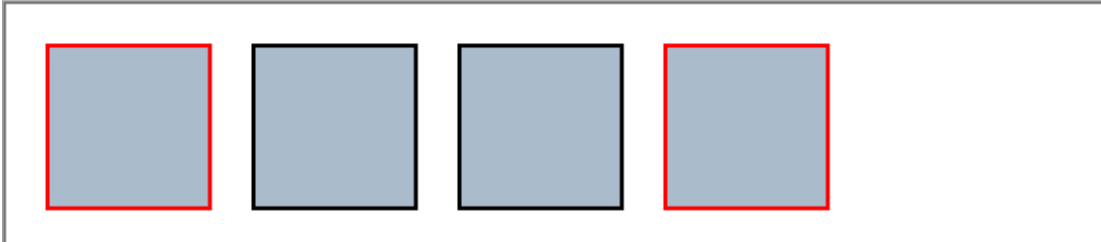
    });
  </script>
  <style>
    div { width:60px; height:60px; margin:10px; float:left;
          border:2px solid blue; }
    .blue { background:blue; }
  </style>
</head>
<body>
  <div></div>
  <div></div>
  <div></div>
  <div></div>
  <div></div>
  <div></div>
</body>
</html>

```

hasClass(class)

يعيد هذا التابع القيمة true في حال أن عنصر واحد من العناصر التي تمت مطابقتها مع المحدد الذي تم استدعاؤه قبل هذا التابع يحتوي على الكلاس التي تم تحديده باسمه بالقيمة class وإلا فإن التابع يعيد false والقيمة class تقبل اسم كلاس واحد فقط, والمثال التالي يتم تحريك العنصر div الذي يحتوي على الكلاس :protected

```
$("#div").click(function(){
  if ( $(this).hasClass("protected") )
    $(this).animate({ left: -10 }, 75)
      .animate({ left: 10 }, 75)
      .animate({ left: -10 }, 75)
      .animate({ left: 10 }, 75)
      .animate({ left: 0 }, 75);
});
```



أما الشيفرة الكاملة للمثال السابق فهي:


```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>

  <script>
    $(document).ready(function(){

      $("div").click(function(){
        if ( $(this).hasClass("protected") )
          $(this).animate({ left: -10 }, 75)
            .animate({ left: 10 }, 75)
            .animate({ left: -10 }, 75)
            .animate({ left: 10 }, 75)
            .animate({ left: 0 }, 75);

      });

    });
  </script>
  <style>
  div { width: 80px; height: 80px; background: #abc;
    position: relative; border: 2px solid black;
    margin: 20px 0px; float: left; left:0 }
  div.protected { border-color: red; }
  span { display:block; float:left; width:20px;
    height:20px; }
  </style>
</head>
<body>
  <span></span><div class="protected"></div>
  <span></span><div></div>
  <span></span><div></div>
  <span></span><div class="protected"></div>
</body>
</html>

```

filter(expr)

يُستخدم هذا التابع لاستبعاد جميع العناصر من العناصر التي تمت مطابقتها مع المحدد الذي تم استدعاؤه قبل هذا التابع والتي لا تحقق التعبير أو التعابير المحددة بالوسيط expr حيث يمكن ترميز أكثر من تعبير يفصل بين كل منها المحرف فاصلة (,)، والمثال التالي يضع إطار أحمر حول جميع العناصر div التي تحتوي على الكلاس :middle

```

$("div").css("background", "#c8ebcc")
        .filter(".middle")
        .css("border-color", "red");

```



أما الشيفرة الكاملة للمثال السابق فهي:

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
        "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>

  <script>
    $(document).ready(function() {

      $("div").css("background", "#c8ebcc")
                .filter(".middle")
                .css("border-color", "red");

    });
  </script>
  <style>
    div { width:60px; height:60px; margin:5px; float:left;
          border:2px white solid;}
  </style>
</head>
<body>
  <div></div>
  <div class="middle"></div>
  <div class="middle"></div>
  <div class="middle"></div>
  <div class="middle"></div>
  <div></div>
</body>
</html>

```

filter(fn)

يُستخدم هذا التابع لاستبعاد جميع العناصر من العناصر التي تمت مطابقتها مع المحدد الذي تم استدعاؤه قبل هذا التابع والتي لا تحقق التابع fn حيث يتم استبعاد العنصر في حال أن عملية تقييم التابع fn أعادة false, والمثال التالي يضع إطار أحمر حول العنصر div الثاني الموجود في الصفحة أي ترتيبه يساوي القيمة 1 أو العنصر div الذي قيمة الصفة id له تساوي fourth:

```

$("div").css("background", "#b4b0da")
    .filter(function (index) {
        return index == 1 || $(this).attr("id") == "fourth";
    })
    .css("border", "3px double red");

```



أما الشيفرة الكاملة للمثال السابق فهي:

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
    <script src="http://code.jquery.com/jquery-latest.js"></script>

    <script>
        $(document).ready(function(){

            $("div").css("background", "#b4b0da")
                .filter(function (index) {
                    return index == 1 || $(this).attr("id") == "fourth";
                })
                .css("border", "3px double red");

        });
    </script>
    <style>
        div { width:60px; height:60px; margin:5px; float:left;
            border:3px white solid; }
    </style>
</head>
<body>
    <div id="first"></div>
    <div id="second"></div>
    <div id="third"></div>
    <div id="fourth"></div>
    <div id="fifth"></div>
    <div id="sixth"></div>
</body>
</html>

```

is(expr)

يُستخدم هذا التابع لفحص التعبير الممرر كقيمة للوسيط `expr` ففي حال أنه يتطابق مع أحد العناصر التي تمت مطابقتها مع المحدد الذي تم استدعاؤه قبل هذا التابع فإنه يعيد القيمة `true` وإلا فإنه يعيد القيمة `false`, ويمكن تمرير أكثر من تعبير للقيمة `expr` يُفصل بينها بفاصلة, والمثال التالي يوضح طريقة استخدام هذا التابع, باستخدام المحددات التالية:

- المحدد `first-child`: يفحص هل العنصر `div` الذي تم الضغط عليه بالفأرة هو أول عنصر في الصفحة بعد الواسمة `body` أم لا.
- المحدد `blue`, `red`: هذين المحددين هما من المحدد `class`. حيث يتم فحص هل العنصر `div` يحتوي على أحد هذين الكلاسين `CSS classes` أم لا.
- المحدد `contains('Peter')`: يفحص هل العنصر `div` يحتوي على القيمة النصية `Peter` أم لا.

```
$("#div").one('click', function () {
  if ($(this).is(":first-child")) {
    $("#p").text("It's the first div.");
  } else if ($(this).is(".blue, .red")) {
    $("#p").text("It's a blue or red div.");
  } else if ($(this).is(":contains('Peter')")) {
    $("#p").text("It's Peter!");
  } else {
    $("#p").html("It's nothing <em>special</em>.");
  }
  $("#p").hide().slideDown("slow");
  $(this).css({"border-style": "inset", "cursor": "default"});
});
```



أما الشيفرة الكاملة للمثال فهي:

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>

  <script>
    $(document).ready(function() {

      $(".div").one('click', function () {
        if ($(this).is(":first-child")) {
          $("p").text("It's the first div.");
        } else if ($(this).is(".blue, .red")) {
          $("p").text("It's a blue or red div.");
        } else if ($(this).is(":contains('Peter')")) {
          $("p").text("It's Peter!");
        } else {
          $("p").html("It's nothing <em>special</em>.");
        }
        $("p").hide().slideDown("slow");
        $(this).css({"border-style": "inset", cursor:"default"});
      });

    });
  </script>
  <style>
  div { width:60px; height:60px; margin:5px; float:left;
        border:4px outset; background:green; text-align:center;
        font-weight:bolder; cursor:pointer; }
  .blue { background:blue; }
  .red { background:red; }
  span { color:white; font-size:16px; }
  p { color:red; font-weight:bolder; background:yellow;
      margin:3px; clear:left; display:none; }
  </style>
</head>

<body>
  <div></div>
  <div class="blue"></div>
  <div></div>
  <div class="red"></div>
  <div><br/><span>Peter</span></div>
  <div class="blue"></div>
  <p> </p>
</body>
</html>

```

map(callback)

يُستخدم هذا التابع لتحويل مجموعة من العناصر في كائن jQuery إلى مجموعة أخرى من القيم في مصفوفة jQuery وهذه المصفوفة قد تحتوي أو لا تحتوي على قيم والوسيط callback هو تابع يتم تنفيذه على كل عنصر من العناصر التي تمت مطابقتها. وفي المثال التالي يتم بناء قائمة بكل القيم الموجودة في النموذج الموجود في الصفحة:

```
$( "p" ).append( $( "input" ).map(function(){
    return $(this).val();
}).get().join(", " ) );
```

Values: John, password, http://ejohn.org/

John password http://ejohn.org/

أما الشيفرة الكاملة للمثال السابق فهي:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
    <script src="http://code.jquery.com/jquery-latest.js"></script>
    <script>
    $(document).ready(function(){
        $( "p" ).append( $( "input" ).map(function(){
            return $(this).val();
        }).get().join(", " ) );
    });
    </script>
    <style>
    p { color:red; }
    </style>
</head>
<body>
    <p><b>Values: </b></p>
    <form>
        <input type="text" name="name" value="John"/>
        <input type="text" name="password" value="password"/>
        <input type="text" name="url" value="http://ejohn.org/">
    </form>
</body>
</html>
```

not(expr)

يُستخدم هذا التابع لإزالة جميع العناصر من العناصر التي تمت مطابقتها مع المحدد الذي تم استدعاؤه قبل هذا التابع والتي تحقق المحددات التي يعبر عنها بالوسيط `expr` ويمكن تمرير أكثر من محدد مفصول بينها بفاصلة, وكأمثلة على هذا التابع:

- استثناء العنصر `p` الذي قيمة الوصفة `id` له تساوي `:selected`:

```
$( "p" ).not( "#selected" )
```

- استثناء جميع العناصر `p` الموجودة داخل عنصر `div` ويحتوي على الكلاس `:selected`:

```
$( "p" ).not( $ ( "div p.selected" ) )
```

ففي المثال التالي يتم وضع إطار أحمر حول جميع العناصر `div` الموجودة في الصفحة ماعدا العناصر `div` التي تحتوي على الكلاس `green` أو التي تكون قيمة الوصفة `id` لها مساوي القيمة `blueone`:

```
$( "div" ).not( ".green, #blueone" )  
  .css( "border-color", "red" );
```



أما الشيفرة الكاملة للمثال السابق فهي:

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>

  <script>
$(document).ready(function() {

  $("div").not(".green, #blueone")
    .css("border-color", "red");

});
</script>
<style>
div { width:60px; height:60px; margin:10px; float:left;
      background:yellow; border:2px solid white; }
.green { background:#8f8; }
.gray { background:#ccc; }
#blueone { background:#99f; }
</style>
</head>
<body>
  <div></div>
  <div id="blueone"></div>
  <div></div>
  <div class="green"></div>
  <div class="green"></div>
  <div class="gray"></div>
  <div></div>
</body>
</html>

```

slice(start, [end])

يُستخدم هذا التابع لاختيار مجموعة فرعية من مجموعة العناصر التي تمت مطابقتها باستخدام المحدد الذي تم استدعاؤه قبل التابع، والوسيط start هو قيمة عددية صحيحة وأقل قيمة ممكنة لها هي القيمة 0، أما الوسيط end فهو اختياري أي لا يجب تمريره عندها يتم اختيار جميع العناصر بدءاً من الترتيب start، ففي المثال التالي يتم اختيار عدد عشوائي من العناصر div الموجودة في الصفحة ويتم تلوينها باللون الأصفر:

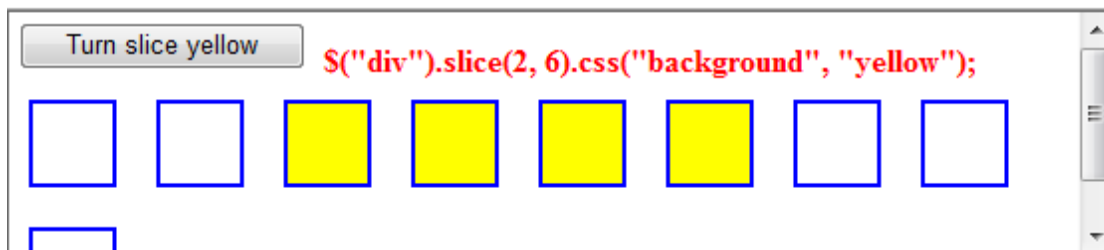

```

function colorEm() {
  var $div = $("div");
  var start = Math.floor(Math.random() *
    $div.length);
  var end = Math.floor(Math.random() *
    ($div.length - start)) +
    start + 1;
  if (end == $div.length) end = undefined;
  $div.css("background", "");
  if (end)
    $div.slice(start, end).css("background", "yellow");
  else
    $div.slice(start).css("background", "yellow");

  $("span").text('$("div").slice(' + start +
    (end ? ', ' + end : '') +
    ').css("background", "yellow");');
}

$("button").click(colorEm);

```



أما الشيفرة الكاملة للمثال السابق فهي:

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>

  <script>
    $(document).ready(function(){

      function colorEm() {
        var $div = $("div");
        var start = Math.floor(Math.random() *
            $div.length);
        var end = Math.floor(Math.random() *
            ($div.length - start)) +
            start + 1;
        if (end == $div.length) end = undefined;
        $div.css("background", "");
        if (end)
          $div.slice(start, end).css("background", "yellow");
        else
          $div.slice(start).css("background", "yellow");

        $("span").text('$("div").slice(' + start +
            (end ? ', ' + end : '') +
            ').css("background", "yellow");');
      }

      $("button").click(colorEm);

    });
  </script>
  <style>
  div { width:40px; height:40px; margin:10px; float:left;
        border:2px solid blue; }
  span { color:red; font-weight:bold; }
  button { margin:5px; }
  </style>

```

```

</head>
<body>
  <button>Turn slice yellow</button>
  <span>Click the button!</span>
  <div></div>
  <div></div>
  <div></div>
  <div></div>
  <div></div>
  <div></div>
  <div></div>
  <div></div>
  <div></div>
  <div></div>
  <div></div>
  <div></div>
</body>
</html>

```

توابع البحث

add(expr)

يُستخدم هذا التابع لإضافة عناصر أخرى إلى مجموعة العناصر التي تمت مطابقتها باستخدام المحدد الذي تم استدعاؤه قبل التابع ولكن فقط إضافة عناصر تتطابق مع الشروط المحددة في التعبير `expr`, ففي المثال التالي يتم إضافة جميع العناصر `p` الموجودة في الصفحة إلى جميع العناصر `div` بعد تغيير خصائص الإطار لعناصر `div` وبعدها يتم تغيير لون الخلفية للعناصر `p`:

```

$("div").css("border", "2px solid red")
.add("p")
.css("background", "yellow");

```



Added this... (notice no border)

أما الشيفرة الكاملة للمثال السابق فهي:

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>

  <script>
    $(document).ready(function(){

      $("#div").css("border", "2px solid red")
        .add("p")
        .css("background", "yellow");

    });
  </script>
  <style>
    div { width:60px; height:60px; margin:10px; float:left; }
    p { clear:left; font-weight:bold; font-size:16px;
      color:blue; margin:0 10px; padding:2px; }
  </style>
</head>
<body>
  <div></div>
  <div></div>
  <div></div>
  <div></div>
  <div></div>
  <div></div>
  <p>Added this... (notice no border)</p>
</body>
</html>

```

children([expr])

يُستخدم للحصول على مجموعة من الأبناء المباشرين لجميع العناصر التي تمت مطابقتها باستخدام المحدد الذي تم استدعاؤه قبل التابع، ويمكن استخدام الوسيط الافتراضي `expr` لترشيح العناصر الأبناء التي يجب أن تظهر حيث فقط يتم إظهار الأبناء التي تتطابق مع الشرط في التعبير `expr`، والمثال التالي يظهر عدد العناصر الأبناء في العنصر الذي يتم الضغط عليه مع اسم العنصر:

```

$("#container").click(function (e) {
    $(".").removeClass("hilite");
    var $kids = $(e.target).children();
    var len = $kids.addClass("hilite").length;

    $("#results span:first").text(len);
    $("#results span:last").text(e.target.tagName);

    e.preventDefault();
    return false;
});

```

**This is the way
we write *the*
demo,**

**write the demo,
write the demo,**

**This the way we
write the demo so

in**

the morning.

Found 0 children in TAG.

أما الشيفرة الكاملة للمثال السابق فهي:

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>

  <script>
    $(document).ready(function(){

      $("#container").click(function (e) {
        $("*").removeClass("hilite");
        var $kids = $(e.target).children();
        var len = $kids.addClass("hilite").length;

        $("#results span:first").text(len);
        $("#results span:last").text(e.target.tagName);

        e.preventDefault();
        return false;
      });

    });
  </script>
  <style>
  body { font-size:16px; font-weight:bolder; }
  div { width:130px; height:82px; margin:10px; float:left;
        border:1px solid blue; padding:4px; }
  #container { width:auto; height:105px; margin:0; float:none;
              border:none; }
  .hilite { border-color:red; }
  #results { display:block; color:red; }
  p { margin:10px; border:1px solid transparent; }
  span { color:blue; border:1px solid transparent; }
  input { width:100px; }
  em { border:1px solid transparent; }
  a { border:1px solid transparent; }
  b { border:1px solid transparent; }
  button { border:1px solid transparent; }
  </style>

```

```

</style>
</head>
<body>
  <div id="container">
    <div>
      <p>This <span>is the <em>way</em> we</span>
        write <em>the</em> demo,</p>
    </div>
    <div>
      <a href="#"><b>w</b>rit<b>e</b></a> the <span>demo,</span> <button>wri
        the</button> demo,
    </div>
    <div>
      This <span>the way we <em>write</em> the <em>demo</em> so</span>
      <input type="text" value="early" /> in
    </div>
    <p>
      <span>t</span>he <span>m</span>orning.
      <span id="results">Found <span>0</span> children in <span>TAG</span>.<
    </p>
  </div>
</body>
</html>

```

closest([expr])

هذا التابع جديد في الإصدار 1.3 ويُستخدم لمطابقة جميع العناصر التي لها عناصر أبناء تطابق المحدد المستخدم بالتعبير `expr` مع العلم أن التعبير `expr` هو وسيط اختياري. وهذا التابع يبحث أولاً بالعنصر الحالي فإن تطابق مع التعبير المحدد سيتم إعادة العنصر نفسه، وفي حال أنه لا يتطابق عندها يتم البحث بالعناصر الأعلى فالأعلى حتى يتم إيجاد كل العناصر التي تتطابق مع التعبير المحدد. والمثال التالي يوضح عمل هذا التابع:

```

$(document).bind("click", function (e) {
  $(e.target).closest("li").toggleClass("highlight");
});

```

- **Click me!**
- **You can also Click me!**

أما الشيفرة الكاملة للمثال السابق فهي:

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>

  <script>
    $(document).ready(function(){

      $(document).bind("click", function (e) {
        $(e.target).closest("li").toggleClass("highlight");
      });

    });
  </script>
  <style>
    li { margin: 3px; padding: 3px; background: #EEEEEE; }
    li.highlight { background: yellow; }
  </style>
</head>
<body>
  <ul>
<li><b>Click me!</b></li>
<li>You can also <b>Click me!</b></li>
</ul>
</body>
</html>

```

find(expr)

يُستخدم للبحث عن جميع العناصر التي تحقق الشروط في التعبير `expr` ويتم البحث ضمن العناصر التي تمت مطابقتها باستخدام المحدد الذي تم استدعاؤه قبل التابع، وفي المثال التالي يتم البحث عن جميع العناصر `span` داخل جميع العناصر `p` الموجودة في الصفحة:

```

$("p").find("span").css('color', 'red');

```

Hello, how are you?

Me? I'm good.

أما الشيفرة الكاملة للمثال السابق فهي:


```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>

  <script>
    $(document).ready(function(){
      $("p").find("span").css('color','red');
    });
  </script>

</head>
<body>
  <p><span>Hello</span>, how are you?</p>
  <p>Me? I'm <span>good</span>.</p>
</body>
</html>

```

next([expr])

يُستخدم هذا التابع للحصول على العناصر القريبة للعنصر الحالي ويمكن أن تطبق عليها الشروط المحددة باستخدام التعبير الاختياري `expr`, ففي المثال التالي يتم إضافة النص `this button is disabled` إلى أقرب عنصر من الزر الغير مفعل وهو هنا في هذه الحالة العنصر `span` أي ليس من الضروري أن يطابق هذا التابع عناصر من نفس نوع العنصر الحالي ولكن يمكن تحديد نوع العناصر التي تريد من التابع مطابقتها عندها يمكن استخدام التعبير `expr` لفعل ذلك:

```

$("button[disabled]").next().text("this button is disabled");

```



أما الشيفرة الكاملة للمثال السابق فهي:

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">

<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>

  <script>
$(document).ready(function(){
  $("button[disabled]").next().text("this button is disabled");
});
</script>
<style>
span { color:blue; font-weight:bold; }
button { width:100px; }
</style>
</head>
<body>
  <div><button disabled="disabled">First</button> - <span></span></div>
  <div><button>Second</button> - <span></span></div>
  <div><button disabled="disabled">Third</button> - <span></span></div>
</body>
</html>

```

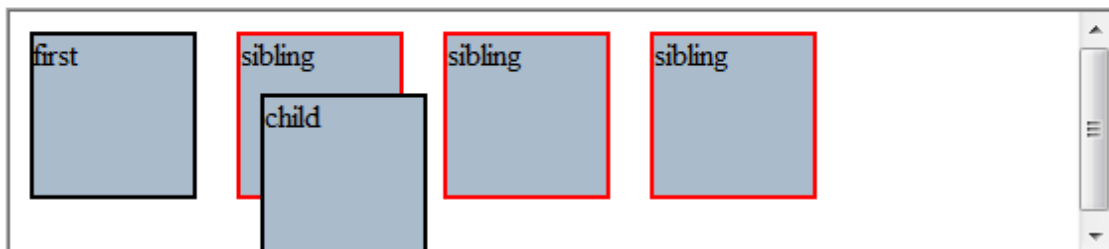
nextAll([expr])

يُستخدم هذا التابع لمطابقة جميع العناصر القريبة (التي على نفس المستوى) للعنصر الحالي ويمكن تحديد شروط مطابقة هذه العناصر باستخدام التعبير الاختياري `expr`, والمثال التالي يطابق جميع العناصر الأخرى التي تأتي بعد العنصر الأول `div` في الصفحة بغض النظر عن نوعها مع ملاحظة أنه يتم مطابقة العناصر التي على نفس المستوى فقط:

```

$("div:first").nextAll().addClass("after");

```



أما الشيفرة الكاملة للمثال السابق فهي:

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>

  <script>
    $(document).ready(function(){
      $("div:first").nextAll().addClass("after");
    });
  </script>
  <style>
    div { width: 80px; height: 80px; background: #abc;
      border: 2px solid black; margin: 10px; float: left; }
    div.after { border-color: red; }
  </style>
</head>
<body>
  <div>first</div>
  <div>sibling<div>child</div></div>
  <div>sibling</div>
  <div>sibling</div>
</body>
</html>

```

offsetParent()

-

parent([expr])

يُستخدم هذا التابع لمطابقة مجموعة من العناصر تحتوي على الآباء المباشرين لجميع العناصر التي تمت مطابقتها باستخدام المحدد الذي تم استدعاؤه قبل التابع، ويمكن استخدام التعبير الاختياري `expr` لترشيح العناصر الآباء التي يمكن مطابقتها، وفي المثال التالي يتم عرض اسم العنصر الأب بجانب كل عنصر من عناصر الصفحة:

```

$("*", document.body).each(function () {
  var parentTag = $(this).parent().get(0).tagName;
  $(this).prepend(document.createTextNode(parentTag + " > "));
});

```

BODY > div, DIV > span, DIV > b

BODY > p, P > span, SPAN > em

BODY > div, DIV > strong, STRONG > span, STRONG > em, EM > b, DIV > b

أما الشيفرة الكاملة للمثال السابق فهي:

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>

  <script>
    $(document).ready(function(){

      $("*", document.body).each(function () {
        var parentTag = $(this).parent().get(0).tagName;
        $(this).prepend(document.createTextNode(parentTag + " > "));
      });

    });
  </script>
  <style>
    div,p { margin:10px; }
  </style>
</head>
<body>
  <div>div,
    <span>span, </span>
    <b>b </b>
  </div>
  <p>p,
    <span>span,
    <em>em </em>
  </span>
</p>
  <div>div,
    <strong>strong,
    <span>span, </span>
    <em>em,
    <b>b, </b>
  </em>
</strong>
<b>b </b>

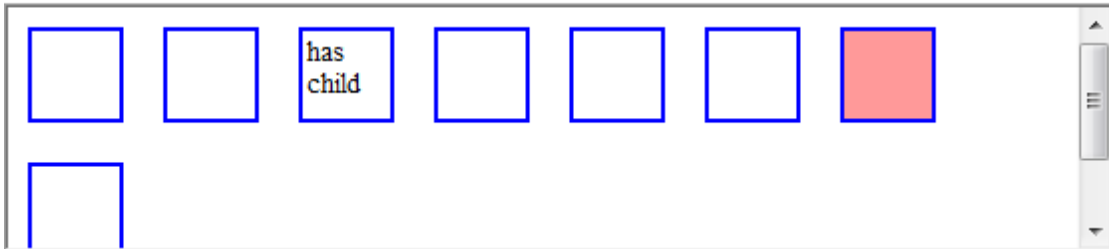
  </div>
</body>
</html>

```

prev([expr])

يُستخدم هذا التابع للحصول على العناصر القريبة السابقة للعنصر الحالي ويمكن أن تطبق عليها الشروط المحددة باستخدام التعبير الاختياري `expr`, ففي المثال التالي يتم تغيير بعض خصائص العنصر القريب السابق للعنصر الحالي عند النقر على الزر:

```
var $curr = $("#start");
$curr.css("background", "#f99");
$("#button").click(function () {
    $curr = $curr.prev();
    $("#div").css("background", "");
    $curr.css("background", "#f99");
});
```



أما الشيفرة الكاملة للمثال السابق فهي:

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>

  <script>
    $(document).ready(function() {

      var $curr = $("#start");
      $curr.css("background", "#f99");
      $("button").click(function () {
        $curr = $curr.prev();
        $("div").css("background", "");
        $curr.css("background", "#f99");
      });

    });
  </script>
  <style>
  div { width:40px; height:40px; margin:10px;
        float:left; border:2px blue solid;
        padding:2px; }
  span { font-size:14px; }
  p { clear:left; margin:10px; }
  </style>
</head>
<body>
  <div></div>
  <div></div>
  <div><span>has child</span></div>
  <div></div>
  <div></div>
  <div></div>
  <div id="start"></div>

  <div></div>
  <p><button>Go to Prev</button></p>
</body>
</html>

```

prevAll([expr])

يُستخدم هذا التابع لمطابقة جميع العناصر القريبة السابقة (التي على نفس المستوى) للعنصر الحالي ويمكن تحديد شروط مطابقة هذه العناصر باستخدام التعبير الاختياري `expr`, والمثال التالي يغير خصائص جميع العناصر التي تسبق آخر عنصر `div` موجود في الصفحة:

```
$("#div:last").prevAll().addClass("before");
```



أما الشيفرة الكاملة للمثال السابق فهي:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>

  <script>
    $(document).ready(function() {
      $("#div:last").prevAll().addClass("before");
    });
  </script>
  <style>
    div { width:70px; height:70px; background:#abc;
          border:2px solid black; margin:10px; float:left; }
    div.before { border-color: red; }
  </style>
</head>
<body>
  <div></div>
  <div></div>
  <div></div>
  <div></div>
</body>
</html>
```

siblings([expr])

يُستخدم هذا التابع لمطابقة جميع العناصر التي على نفس المستوى السابقة أو اللاحقة للعنصر الحالي ويمكن تحديد شروط المطابقة باستخدام التعبير الاختياري `expr`, والمثال التالي يغير لون خط كل العناصر التي تأتي قبل وبعد العناصر المحددة بالكلاس `hilite`:

```

var len = $(".hilite").siblings()
    .css("color", "red")
    .length;

$("b").text(len);

```

One **Five** **Eight**
Two **Six** **Nine** Unique siblings: 7
Three **Seven** **Ten**
Four **Eleven**

أما الشيفرة الكاملة للمثال السابق فهي:

```

<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>

  <script>
    $(document).ready(function() {

      var len = $(".hilite").siblings()
          .css("color", "red")
          .length;

      $("b").text(len);

    });
  </script>
  <style>
    ul { float:left; margin:5px; font-size:16px; font-weight:bold; }
    p { color:blue; margin:10px 20px; font-size:16px; padding:5px;
      font-weight:bolder; }
    .hilite { background:yellow; }
  </style>
</head>
<body>
  <ul>
    <li>One</li>
    <li>Two</li>
    <li class="hilite">Three</li>
    <li>Four</li>
  </ul>
  <ul>
    <li>Five</li>
    <li>Six</li>
    <li>Seven</li>
  </ul>
  <ul>
    <li>Eight</li>

```



```

<li class="hilite">Nine</li>
<li>Ten</li>
<li class="hilite">Eleven</li>
</ul>
<p>Unique siblings: <b></b></p>
</body>
</html>

```

توابع التسلسلية Chaining Functions

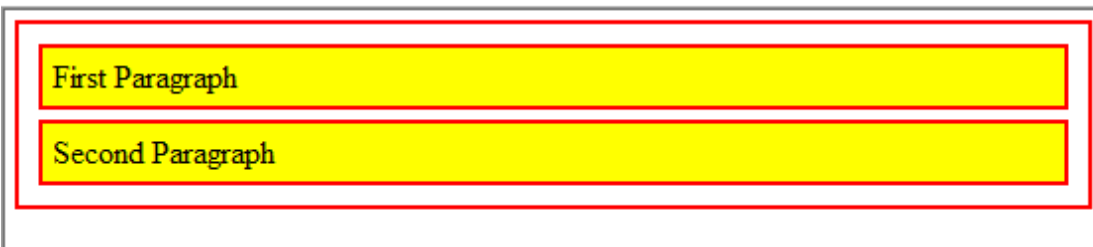
andSelf()

يُستخدم هذا التابع إضافة العناصر المحددة المختارة مسبقاً إلى العناصر المختارة حالياً ويفيد هذا التابع في العبور عبر العناصر وإضافة العناصر التي تمت مطابقتها قبل آخر استخدام لتابع عبور، ففي المثال التالي يتم البحث عن جميع العناصر div وبعدها البحث عن جميع العناصر p التي داخل العناصر div ويتم إضافة نتيجة البحث عن العناصر div إلى مجموعة العناصر p وأخيراً إضافة الكلاس border إلى العناصر div و p أما السطر الذي يضيف خلفية صفراء إلى جميع العناصر p الموجودة داخل العناصر div بدون تحديد خلفية صفراء للعناصر div وذلك بسبب عدم استخدام التابع adnSelf:

```

$ ("div").find("p").andSelf().addClass("border");
$ ("div").find("p").addClass("background");

```



أما الشيفرة الكاملة للمثال السابق فهي:

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>

  <script>
    $(document).ready(function() {

      $("div").find("p").andSelf().addClass("border");
      $("div").find("p").addClass("background");

    });
  </script>
  <style>
    p, div { margin:5px; padding:5px; }
    .border { border: 2px solid red; }
    .background { background:yellow; }
  </style>
</head>
<body>
  <div>
    <p>First Paragraph</p>
    <p>Second Paragraph</p>
  </div>
</body>
</html>

```

end()

يُستخدم هذا التابع لإعادة العناصر التي تمت مطابقتها في آخر عملية استدعاء لتابع عبور Traversing Function يعيد قيمة من النوع jQuery وقد تم شرح هذه التوابع في قسم توابع العبور إضافة إلى بعض التوابع الأخرى مثل clone, appendTo, prependTo, insertBefore, insertAfter, replaceAll, والمثال التالي يطابق جميع العناصر p الموجودة في الصفحة ومن ثم يتم تعديل خصائص العناصر span الموجودة داخلها وبعد ذلك يتم استدعاء التابع end للعودة إلى مجموعة العناصر p لتنفيذ عملية أخرى عليها:

```

jQuery.fn.showTags = function (n) {
  var tags = this.map(function () {
    return this.tagName;
  })
    .get().join(", ");
  $("b:eq(" + n + ")").text(tags);
  return this;
};

$("p").showTags(0)
  .find("span")
  .showTags(1)
  .css("background", "yellow")
  .end()
  .showTags(2)
  .css("font-style", "italic");

```

*Hi there **how** are you **doing**?*

*This **span** is one of several **spans** in this **sentence**.*

Tags in jQuery object initially: P, P

Tags in jQuery object after find: SPAN, SPAN, SPAN, SPAN, SPAN

Tags in jQuery object after end: P, P

أما الشيفرة الكاملة للمثال السابق فهي:

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>

  <script>
$(document).ready(function(){

  jQuery.fn.showTags = function (n) {
    var tags = this.map(function () {
      return this.tagName;
    })
      .get().join(", ");
    $("b:eq(" + n + ")").text(tags);
    return this;
  };

  $("p").showTags(0)
    .find("span")
    .showTags(1)
    .css("background", "yellow")
    .end()
    .showTags(2)
    .css("font-style", "italic");

});
</script>
<style>
p, div { margin:1px; padding:1px; font-weight:bold;
        font-size:16px; }
div { color:blue; }
b { color:red; }
</style>
</head>
<body>
  <p>
    Hi there <span>how</span> are you <span>doing</span>?
  </p>
  <p>
    This <span>span</span> is one of
    several <span>spans</span> in this

```

```

</p>
<div>
  Tags in jQuery object initially: <b></b>
</div>
<div>
  Tags in jQuery object after find: <b></b>
</div>
<div>
  Tags in jQuery object after end: <b></b>
</div>
</body>
</html>

```

توابع التعامل مع المحتوى Manipulation Functions

توابع تغيير المحتوى Changing Contents

html()

يُستخدم هذا التابع لإعادة شيفرة الـ html لأول عنصر تمت مطابقته ولا يعمل هذا التابع على وثيقة XML بالرغم من أنه يعمل على وثيقة من النوع XHTML, وفي المثال التالي يتم عرض شيفرة الـ HTML للعنصر p عن الضغط على أي عنصر p أو أي عنصر داخل أي عنصر من النوع p:

```

$("p").click(function () {
  var htmlStr = $(this).html();
  $(this).text(htmlStr);
});

```

[Click to change the html](#)

to a text node.

This does nothing.

أما الشيفرة الكاملة للمثال السابق فهي:

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>

  <script>
    $(document).ready(function(){

      $("p").click(function () {
        var htmlStr = $(this).html();
        $(this).text(htmlStr);
      });

    });
  </script>
  <style>
    p { margin:8px; font-size:20px; color:blue;
      cursor:pointer; }
    b { text-decoration:underline; }
      { cursor:pointer; }
  </style>
</head>
<body>
  <p>
    <b>Click</b> to change the <span id="tag">html</span>
  </p>
  <p>
    to a <span id="text">text</span> node.
  </p>
  <p>
    This <button name="nada">button</button> does nothing.
  </p>
</body>
</html>

```

html(val)

يُستخدم هذا التابع لتغيير شيفرة الـ HTML لجميع العناصر التي تمت مطابقتها مع المحدد الذي تم استدعاؤه قبل هذا التابع، والمثال التالي يغير شيفرة الـ HTML لجميع العناصر `div`:

```

$("div").html("<span class='red'>Hello <b>Again</b></span>");

```

Hello
Hello Again
Hello Again
Hello Again

أما الشيفرة الكاملة للمثال السابق فهي:

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>

  <script>
  $(document).ready(function() {
    $("div").html("<span class='red'>Hello <b>Again</b></span>");
  });
  </script>
  <style>
  .red { color:red; }
  </style>
</head>
<body>
  <span>Hello</span>
  <div></div>
  <div></div>
  <div></div>
</body>
</html>

```

text()

يُستخدم لإعادة القيم النصية لجميع العناصر التي تمت مطابقتها مع المحدد الذي تم استدعاؤه قبل هذا التابع حيث يتم دمج كل القيم النصية في سلسلة نصية واحدة، والمثال التالي يعيد القيمة النصية لأول عنصر p موجود في الصفحة ويُسنده إلى شيفرة الـ HTML لآخر عنصر p موجود في الصفحة:

```

var str = $("p:first").text();
$("p:last").html(str);

```

Test Paragraph.

Test Paragraph.

أما الشيفرة الكاملة للمثال السابق فهي:

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>

  <script>
    $(document).ready(function(){

      var str = $("p:first").text();
      $("p:last").html(str);

    });
  </script>
  <style>
    p { color:blue; margin:8px; }
    b { color:red; }
  </style>
</head>

  <p><b>test</b> Paragraph.</p>
  <p></p>
</body>
</html>

```

text(val)

يُستخدم هذا التابع لتغيير القيمة النصية لجميع العناصر التي تمت مطابقتها مع المحدد الذي تم استدعاؤه قبل هذا التابع, والمثال التالي يغير القيمة النصية لجميع العناصر p الموجودة في الصفحة:

```

$("p").text("<b>Some</b> new text.");

```

Some new text.

أما الشيفرة الكاملة للمثال السابق فهي:


```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>

  <script>
    $(document).ready(function(){
      $("p").text("<b>Some</b> new text.");
    });
  </script>
  <style>
    p { color:blue; margin:8px; }
  </style>
</head>
<body>
  <p>Test Paragraph.</p>
</body>
</html>

```

توابع إدراج المعلومات في داخل العناصر Inserting Inside

append(content)

يُستخدم هذا التابع لإلحاق معلومات إلى جميع العناصر التي تمت مطابقتها باستخدام المحدد الذي تم استدعاؤه قبل التابع, وفي المثال التالي يلحق شيفرة HTML إلى جميع العناصر p الموجودة في الصفحة:

```

$("p").append("<strong>Hello</strong>");

```

I would like to say: **Hello**

أما الشيفرة الكاملة للمثال السابق فهي:

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>

  <script>
    $(document).ready(function(){
      $("p").append("<strong>Hello</strong>");
    });
  </script>
  <style>p { background:yellow; }</style>
</head>
<body>
  <p>I would like to say: </p>
</body>
</html>

```

appendTo(selector)

يُستخدم لإلحاق جميع العناصر التي تمت مطابقتها باستخدام المحدد الذي تم استدعاؤه قبل التابع بالعناصر التي تمت مطابقتها باستخدام المحدد selector, والمثال التالي يلحق جميع بيانات العناصر span بالعنصر div الذي قيمة الواصفة id له تساوي foo:

```

$("span").appendTo("#foo"); // check append\(\) examples

```

FOO! I have nothing more to say...

أما الشيفرة الكاملة للمثال السابق فهي:

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>

  <script>
$(document).ready(function(){
  $("span").appendTo("#foo"); // check append\(\) examples
});
</script>
  <style>#foo { background:yellow; }</style>
</head>
<body>
  <span>I have nothing more to say... </span>
  <div id="foo">FOO! </div>
</body>
</html>

```

prepend(content)

يُستخدم هذا التابع لإدراج بيانات في بداية جميع العناصر التي تمت مطابقتها باستخدام المحدد الذي تم استدعاؤه قبل التابع، والمثال التالي يضيف القيمة Hello إلى بداية جميع العناصر p الموجودة في الصفحة:

```

$("p").prepend("<b>Hello </b>");

```

Hello there friend!

Hello amigo!

أما الشيفرة الكاملة للمثال السابق فهي:

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>

  <script>
    $(document).ready(function(){
      $("p").prepend("<b>Hello </b>");
    });
  </script>
  <style>p { background:yellow; }</style>
</head>
<body>
  <p>there friend!</p>
  <p>amigo!</p>
</body>
</html>

```

prependTo(selector)

يُستخدم هذا التابع لإدراج جميع العناصر التي تمت مطابقتها باستخدام المحدد الذي تم استدعاؤه قبل التابع إلى جميع العناصر التي تتم مطابقتها باستخدام المحدد selector, والمثال التالي يدرج قيمة جميع العناصر span الموجودة في الصفحة إلى في بداية العنصر div الذي قيمة الواصفة id له تساوي foo:

```

$("span").prependTo("#foo"); // check prepend\(\) examples

```

I have something to say... FOO!

أما الشيفرة الكاملة للمثال السابق فهي:

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">

<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>

  <script>
    $(document).ready(function(){
      $("span").prependTo("#foo"); // check prepend\(\) examples
    });
  </script>
  <style>div { background:yellow; }</style>
</head>
<body>
  <div id="foo">FOO!</div>
  <span>I have something to say... </span>
</body>
</html>

```

توابع إدراج المعلومات خارج العناصر Inserting Outside

after(content)

يُستخدم هذا التابع لإدراج بيانات بعد جميع العناصر التي تمت مطابقتها باستخدام المحدد الذي تم استدعاؤه قبل التابع والوسيط content ممكن أن يكون شيفرة HTML أو نص عادي وهذا التابع يعيد جميع العناصر التي تمت مطابقتها باستخدام المحدد الذي تم استدعاؤه قبل التابع. وفي المثال التالي يُدرج الكلمة Hello بعد كل عنصر من العناصر p الموجودة في الصفحة ويعيد التابع after في هذا المثال مجموعة العناصر p الموجودة في الصفحة:

```

$("p").after("<b>Hello</b>");

```

I would like to say:

Hello

أما الشيفرة الكاملة للمثال السابق فهي:

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>

  <script>
    $(document).ready(function() {
      $("p").after("<b>Hello</b>");
    });
  </script>
  <style>p { background:yellow; }</style>
</head>
<body>
  <p>I would like to say: </p>
</body>
</html>

```

before(content)

يُستخدم هذا التابع لإدراج بيانات قبل جميع العناصر التي تمت مطابقتها باستخدام المحدد الذي تم استدعاؤه قبل التابع والوسيط content ممكن أن يكون شيفرة HTML أو نص عادي وهذا التابع يعيد جميع العناصر التي تمت مطابقتها باستخدام المحدد الذي تم استدعاؤه قبل التابع. وفي المثال التالي يُدرج الكلمة Hello قبل كل عنصر من العناصر p الموجودة في الصفحة ويعيد التابع before في هذا المثال مجموعة العناصر الموجودة في الصفحة:

```

$("p").before("<b>Hello</b>");

```

Hello

is what I said...

أما الشيفرة الكاملة للمثال السابق فهي:

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>

  <script>
    $(document).ready(function() {
      $("p").before("<b>Hello</b>");
    });
  </script>
  <style>p { background:yellow; }</style>
</head>
<body>
  <p> is what I said...</p>
</body>
</html>

```

insertAfter(selector)

يُستخدم هذا التابع لإدراج جميع العناصر التي تمت مطابقتها باستخدام المحدد الذي تم استدعاؤه قبل التابع بعد كل عنصر من العناصر التي تتم مطابقتها باستخدام المحدد selector, والمثال التالي يدرج جميع العناصر p مع محتواها بعد العنصر div الذي قيمة الواصفة id له تساوي foo:

```

$("p").insertAfter("#foo"); // check after\(\) examples

```

FOO!

is what I said...

أما الشيفرة الكاملة للمثال السابق فهي:

ملاحظة: الشرح الموجود في الصفحة موقع jQuery الرسمي الخاص بهذا التابع غير دقيق, والرابط التالي هو الشرح حول insertAfter:

<http://docs.jquery.com/Manipulation/insertAfter#content>

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">

<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>

  <script>
    $(document).ready(function(){
      $("p").insertAfter("#foo"); // check after\(\) examples
    });
  </script>
  <style>#foo { background:yellow; }</style>
</head>
<body>
  <p> is what I said... </p><div id="foo">FOO!</div>
</body>
</html>

```

insertBefore(selector)

يُستخدم هذا التابع لإدراج جميع العناصر التي تمت مطابقتها باستخدام المحدد الذي تم استدعاؤه قبل التابع قبل كل عنصر من العناصر التي تتم مطابقتها باستخدام المحدد selector, والمثال التالي يدرج جميع العناصر p مع محتواها قبل العنصر div الذي قيمة الواصفة id له تساوي foo:

```

$("p").insertBefore("#foo"); // check before\(\) examples

```

I would like to say:

FOO!

أما الشيفرة الكاملة للمثال السابق فهي:

ملاحظة: الشرح الموجود في الصفحة موقع jQuery الرسمي الخاص بهذا التابع غير دقيق, والرابط التالي هو الشرح حول insertBefore:

<http://docs.jquery.com/Manipulation/insertBefore#content>


```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>

  <script>
    $(document).ready(function(){
      $("p").insertBefore("#foo"); // check before\(\) examples
    });
  </script>
  <style>#foo { background:yellow; }</style>
</head>
<body>
  <div id="foo">FOO!</div><p>I would like to say: </p>
</body>
</html>

```

توابع الإحاطة (التطويق) Inserting Around

wrap(html)

يُستخدم هذا التابع لإحاطة جميع العناصر التي تمت مطابقتها باستخدام المحدد الذي تم استدعاؤه قبل التابع بشيفرة الوسيط html التي يمكن أن تكون أي صيغة html صحيحة, ففي المثال التالي يتم إحاطة كل عنصر من العناصر p الموجودة في الصفحة بالعنصر div:

```

$("p").wrap("<div></div>");

```



أما الشيفرة الكاملة للمثال السابق فهي:

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>

  <script>
    $(document).ready(function(){
      $("p").wrap("<div></div>");
    });
  </script>
  <style>
    div { border: 2px solid blue; }
    p { background:yellow; margin:4px; }
  </style>
</head>
<body>
  <p>Hello</p>
  <p>cruel</p>
  <p>World</p>
</body>
</html>

```

wrap(elem)

يُستخدم هذا التابع لإحاطة جميع العناصر التي تمت مطابقتها باستخدام المحدد الذي تم استدعاؤه قبل التابع بالعنصر المحدد بالوسيط elem, ففي المثال التالي يتم إحاطة كل عنصر من العناصر p الموجودة في الصفحة بالعنصر div:

```

$("p").wrap(document.createElement("div"));

```



أما الشيفرة الكاملة للمثال السابق فهي:

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>

  <script>
    $(document).ready(function() {
      $("p").wrap(document.createElement("div"));
    });
  </script>
  <style>
    div { border: 2px solid blue; }
    p { background:yellow; margin:4px; }
  </style>
</head>
<body>
  <p>Hello</p>
  <p>cruel</p>
  <p>World</p>
</body>
</html>

```

wrapAll(html)

يُستخدم هذا التابع لإحاطة جميع العناصر التي تمت مطابقتها باستخدام المحدد الذي تم استدعاؤه قبل التابع بشيفرة HTML المحددة بالوسيط html وفي هذا التابع يتم إحاطة جميع العناصر بعنصر واحد فقط، ففي المثال التالي يتم إحاطة جميع العناصر p الموجودة في الصفحة بعنصر div واحد فقط:

```

$("p").wrapAll("<div></div>");

```



أما الشيفرة الكاملة للمثال السابق فهي:

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>

  <script>
    $(document).ready(function(){
      $("p").wrapAll("<div></div>");
    });
  </script>
  <style>
    div { border: 2px solid blue; }
    p { background:yellow; margin:4px; }
  </style>
</head>
<body>
  <p>Hello</p>
  <p>cruel</p>
  <p>World</p>
</body>
</html>

```

wrapAll(elem)

يُستخدم هذا التابع لإحاطة جميع العناصر التي تمت مطابقتها باستخدام المحدد الذي تم استدعاؤه قبل التابع بالعنصر المحدد بالوسيط div, ففي المثال التالي يتم إحاطة جميع العناصر p الموجودة في الصفحة بالعنصر elem:

```

$("p").wrapAll(document.createElement("div"));

```



أما الشيفرة الكاملة للمثال السابق فهي:

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>

  <script>
    $(document).ready(function(){
      $("p").wrapAll(document.createElement("div"));
    });
  </script>
  <style>
    div { border: 2px solid blue; }
    p { background:yellow; margin:4px; }
  </style>
</head>
<body>
  <p>Hello</p>
  <p>cruel</p>
  <p>World</p>
</body>
</html>

```

wrapInner(html)

يُستخدم هذا التابع لإحاطة العناصر الأبناء لجميع العناصر التي تمت مطابقتها باستخدام المحدد الذي تم استدعاؤه قبل التابع بصيغة الـ HTML التي تم تحديدها بالوسيط html, وفي المثال التالي يتم إحاطة جميع أبناء العناصر p الموجودة في الصفحة بالعنصر :

```

$("p").wrapInner("<b></b>");

```

Hello

cruel

World

أما الشيفرة الكاملة للمثال السابق فهي:

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>

  <script>
    $(document).ready(function() {
      $("p").wrapInner("<b></b>");
    });
  </script>
  <style>p { background:#bbf; }</style>
</head>
<body>
  <p>Hello</p>
  <p>cruel</p>
  <p>World</p>
</body>
</html>

```

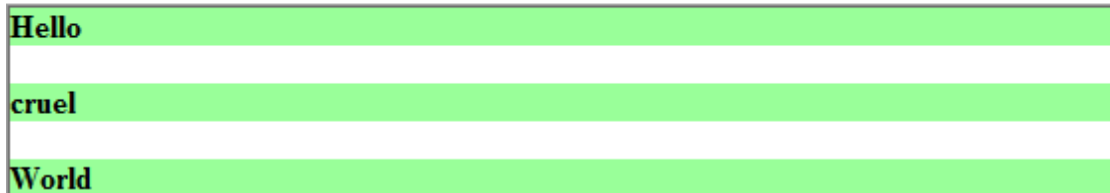
wrapInner(elem)

يُستخدم هذا التابع لإحاطة العناصر الأبناء لجميع العناصر التي تمت مطابقتها باستخدام المحدد الذي تم استدعاؤه قبل التابع بالعنصر المحدد بالوسيط elem, وفي المثال التالي يتم إحاطة جميع أبناء العناصر p الموجودة في الصفحة بالعنصر ****:

```

$("p").wrapInner(document.createElement("b"));

```



أما الشيفرة الكاملة للمثال السابق فهي:

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>

  <script>
    $(document).ready(function() {
      $("p").wrapInner(document.createElement("b"));
    });
  </script>
  <style>p { background:#9f9; }</style>
</head>
<body>
  <p>Hello</p>
  <p>cruel</p>
  <p>World</p>
</body>
</html>

```

توابع الاستبدال

replaceWith(content)

يُستخدم هذا التابع لاستبدال جميع العناصر التي تمت مطابقتها باستخدام المحدد الذي تم استدعاؤه قبل التابع بالمحتوى المحدد بالوسيط content سواء كان شيفرة HTML أو عناصر DOM, ويعيد التابع العناصر التي تم استبدالها على صيغة عنصر jQuery, وفي المثال التالي يتم استبدال أي زر يتم الضغط عليه بعنصر div:

```

$("button").click(function () {
  $(this).replaceWith("<div>" + $(this).text() + "</div>");
});

```



أما الشيفرة الكاملة للمثال السابق فهي:

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>

  <script>
    $(document).ready(function(){

      $("button").click(function () {
        $(this).replaceWith("<div>" + $(this).text() + "</div>");
      });

    });
  </script>
  <style>
    button { display:block; margin:3px; color:red; width:200px; }
    div { color:red; border:2px solid blue; width:200px;
      margin:3px; text-align:center; }
  </style>
</head>
<body>
  <button>First</button>
  <button>Second</button>
  <button>Third</button>
</body>
</html>

```

replaceAll(selector)

يستخدم هذا التابع لاستبدال كل عنصر من العناصر التي تمت مطابقتها باستخدام الوسيط selector بجميع العناصر التي تمت مطابقتها باستخدام المحدد الذي تم استدعاؤه قبل التابع وطبعاً يعيد هذا التابع جميع العناصر التي تم إدراجها بدلاً من العناصر القديمة، وفي المثال التالي يتم استبدال جميع العناصر p وما تحتويه بالقيمة :Paragraph

```

$("<b>Paragraph. </b>").replaceAll("p"); // check replaceWith\(\) examples

```

Paragraph. Paragraph. Paragraph.

أما الشيفرة الكاملة للمثال السابق فهي:


```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>

  <script>
    $(document).ready(function(){
      $("<b>Paragraph. </b>").replaceAll("p"); // check replaceWith\(\) examples
    });
  </script>

</head>
<body>
  <p>Hello</p>
  <p>cruel</p>
  <p>World</p>
</body>
</html>

```

توابع إدالة العناصر Removing

empty()

يُستخدم هذا التابع لإزالة جميع العناصر الأبناء من جميع العناصر التي تمت مطابقتها باستخدام المحدد الذي تم استدعاؤه قبل التابع, وفي المثال التالي يتم إزالة جميع العناصر الأبناء لكل عنصر p موجود في الصفحة عند الضغط على الزر:

```

$("button").click(function () {
  $("p").empty();
});

```

Hello, Person [and person](#)

Call empty() on above paragraph

أما الشيفرة الكاملة للمثال السابق فهي:

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>

  <script>
    $(document).ready(function(){

      $("#button").click(function () {
        $("#p").empty();
      });

    });
  </script>
  <style>
  p { background:yellow; }
  </style>
</head>
<body>
  <p>
    Hello, <span>Person</span> <a href="javascript:;>">and person</a>
  </p>
  <button>Call empty() on above paragraph</button>
</body>
</html>

```

remove([expr])

يُستخدم هذا التابع يمحذف جميع العناصر التي تمت مطابقتها باستخدام المحدد الذي تم استدعاؤه قبل التابع ويمكن استخدام الوسيط `expr` الاختياري لترشيح العناصر التي تريد حذفها، مع العلم بأن هذا التابع لا يمحذف العناصر من الكائن jQuery ولكن يمحذف جميع الأحداث المتصلة بهذه العناصر لذلك فإن:

```
$("#foo").remove().appendTo("#bar");
```

يجب كتابته بالشكل التالي لتفادي عدم خسارة الأحداث:

```
$("#foo").appendTo("#bar");
```

أما في المثال التالي فيتم حذف جميع العناصر `p` الموجودة في الصفحة عند الضغط على الزر:

```
$("#button").click(function () {
    $("#p").remove();
});
```

Hello

how are

you?

Call remove() on paragraphs

أما الشيفرة الكاملة للمثال السابق فهي:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
    <script src="http://code.jquery.com/jquery-latest.js"></script>

    <script>
        $(document).ready(function() {

            $("#button").click(function () {
                $("#p").remove();
            });

        });
    </script>
    <style>p { background:yellow; margin:6px 0; }</style>
</head>
<body>
    <p>Hello</p>
    how are
    <p>you?</p>
    <button>Call remove() on paragraphs</button>
</body>
</html>
```

توابع النسخ Copying

clone()

يُستخدم هذا التابع لنسخ جميع العناصر التي تمت مطابقتها باستخدام المحدد الذي تم استدعاؤه قبل التابع

ويعيد هذا التابع العناصر الجديدة، وفي المثال التالي يتم نسخ العنصر **b** مع محتواه وإدراجه قبل العنصر **p**:

```
$("#b").clone().prependTo("p");
```

Hello

Hello, how are you?

أما الشيفرة الكاملة للمثال السابق فهي:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>

  <script>
    $(document).ready(function() {
      $("#b").clone().prependTo("p");
    });
  </script>

</head>
<body>
  <b>Hello</b><p>, how are you?</p>
</body>
```

clone(bool)

يُستخدم هذا التابع لنسخ جميع العناصر مع الأحداث المتصلة بها التي تمت مطابقتها باستخدام المحدد الذي تم استدعاؤه قبل التابع ويعيد هذا التابع العناصر الجديدة والوسيط bool يأخذ القيمة true للسماح بنسخ الأحداث أيضاً أو false لعدم نسخ الأحداث, وفي المثال التالي يتم نسخ الزر وإدراجه بعد الزر الحالي:

```
$("#button").click(function() {
  $(this).clone(true).insertAfter(this);
});
```

Clone Me!

Clone Me!

أما الشيفرة الكاملة للمثال السابق فهي:

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>

  <script>
    $(document).ready(function() {

      $("button").click(function() {
        $(this).clone(true).insertAfter(this);
      });

    });
  </script>
</head>
<body>
  <button>Clone Me!</button>
</body>
</html>

```

توابع الأنماط المتعاقبة CSS Functions

CSS

css(name)

يُستخدم هذا التابع للحصول على القيمة النصية الخاصة بنمط style property من أول عنصر من العناصر التي تمت مطابقتها باستخدام المحدد الذي تم استدعاؤه قبل التابع وذلك بتحديد اسم الخاصية بالوسيط name, وفي المثال التالي يتم الحصول على قيمة الخاصية background-color لأي عنصر div موجود في الصفحة وذلك عند الضغط عليه بالفأرة:

```

$("div").click(function () {
  var color = $(this).css("background-color");
  $("#result").html("That div is <span style='color:" +
    color + "';>" + color + "</span>.");
});

```

That div is #f11.



أما الشيفرة الكاملة للمثال السابق فهي:

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>

  <script>
    $(document).ready(function() {

      $("div").click(function () {
        var color = $(this).css("background-color");
        $("#result").html("That div is <span style='color:" +
          color + ";>" + color + "</span>.");
      });

    });
  </script>
  <style>
    div { width:60px; height:60px; margin:5px; float:left; }
  </style>
</head>
<body>
  <span id="result"> </span>
  <div style="background-color:blue;"></div>
  <div style="background-color:rgb(15,99,30);"></div>
  <div style="background-color:#123456;"></div>
  <div style="background-color:#f11;"></div>
</body>
</html>

```

css(properties)

يُستخدم هذا التابع لتعيين قيم خصائص النمط لجميع العناصر التي تمت مطابقتها باستخدام المحدد الذي تم استدعاؤه قبل التابع وذلك بتحديد أزواج من المفاتيح والقيم من الشكل التالي:

{key1 : value1, key2 : value2, keyN : valueN}

وفي المثال التالي يتم تغيير لون الخلفية وسماكة الخط لجميع العناصر p الموجودة في الصفحة عند وضع مؤشر الفأرة فوقها وتغييره مرة أخرى عند إزالة مؤشر الفأرة من فوقه:

```
$("#p").hover(function () {
    $(this).css({'background-color' : 'yellow', 'font-weight' : 'bolder'});
}, function () {
    var cssObj = {
        'background-color' : '#ddd',
        'font-weight' : '',
        'color' : 'rgb(0,40,244)'
    }
    $(this).css(cssObj);
});
```

[Move the mouse over a paragraph.](#)

[Like this one or the one above.](#)

أما الشيفرة الكاملة للمثال السابق فهي:

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>

  <script>
    $(document).ready(function(){

      $("p").hover(function () {
        $(this).css({'background-color' : 'yellow', 'font-weight' : 'bolder'});
      }, function () {
        var cssObj = {
          'background-color' : '#ddd',
          'font-weight' : '',
          'color' : 'rgb(0,40,244)'
        }
        $(this).css(cssObj);
      });

    });
  </script>
  <style>
  p { color:green; }
  </style>
</head>
<body>
  <p>
    Move the mouse over a paragraph.
  </p>
  <p>
    Like this one or the one above.
  </p>
</body>
</html>

```

css(name, value)

يُستخدم هذا التابع لتعيين قيمة نمط واحدة فقط لجميع العناصر التي تمت مطابقتها باستخدام المحدد الذي تم استدعاؤه قبل التابع وذلك بتحديد اسمها بالوسيط name وقيمتها بالوسيط value وفي المثال التالي يتم تغيير لون خط جميع العناصر p الموجودة في الصفحة عند مادم مؤشر الفأرة فوقها:


```
$("#p").mouseover(function () {
    $(this).css("color", "red");
});
```

Just roll the mouse over me.

Or me to see a color change.

أما الشيفرة الكاملة للمثال السابق فهي:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
    <script src="http://code.jquery.com/jquery-latest.js"></script>

    <script>
    $(document).ready(function(){

        $("#p").mouseover(function () {
            $(this).css("color", "red");
        });

    });
</script>
<style>
p { color:blue; width:200px; font-size:14px; }
</style>
</head>
<body>
    <p>
        Just roll the mouse over me.
    </p>
    <p>
        Or me to see a color change.
    </p>
</body>
</html>
```

توابع التموضع Positioning Functions

offset()

يُستخدم هذا التابع للحصول على إحداثيات العنصر الذي تمت مطابقته باستخدام أحد توابع التحديد وهذه الإحداثيات تكون بالبيكسل ويعيد هذا التابع كائن يحتوي على قيمتين `left`, `top`, ففي المثال التالي يتم عرض قيمة `left` و `top` لآخر عنصر `p` موجود في الصفحة:

```
var p = $("#p:last");
var offset = p.offset();
p.html( "left: " + offset.left + ", top: " + offset.top );
```

Hello

left: 10, top: 38

أما الشيفرة الكاملة للمثال السابق فهي:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>

  <script>
    $(document).ready(function(){
      var p = $("#p:last");
      var offset = p.offset();
      p.html( "left: " + offset.left + ", top: " + offset.top );
    });
  </script>
  <style>
    p { margin-left:10px; }
  </style>
</head>
<body>
  <p>Hello</p><p>2nd Paragraph</p>
</body>
</html>
```

position()

يُستخدم هذا التابع للحصول على قيمة top و left للعنصر المحدد نسبة إلى إحداثيات العنصر الأب للعنصر، القيمة التي يعيدها هذا التابع هي كائن يحتوي على قيمتين عدديين صحيحين، ويُستخدم هذا التابع مع العناصر التي تظهر على الصفحة فقط، ففي المثال التالي يتم عرض قيم موقع أول عنصر p في الصفحة:

```
var p = $("#p:first");
var position = p.position();
$("#p:last").text( "left: " + position.left + ", top: " + position.top );
```

Hello left: 15, top: 34

أما الشيفرة الكاملة للمثال السابق فهي:

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">

<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>

  <script>
    $(document).ready(function(){
      var p = $("p:first");
      var position = p.position();
      $("p:last").text( "left: " + position.left + ", top: " + position.top );
    });
  </script>
  <style>
    div { padding: 15px;}
    p { margin-left:10px; }
  </style>
</head>
<body>
  <div><p>Hello</p></div><p></p>
</body>
</html>

```

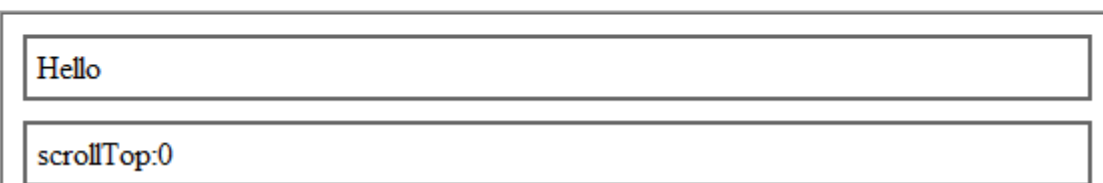
scrollTop()

يُستخدم هذا التابع لإعادة قيمة الإحداثية top شريط التنقل الجانبي لأول عنصر يتم تحديده وتكون قيمة عددية صحيحة. يمكن استخدام هذا التابع مع العناصر الظاهرة والمخفية في الصفحة، وفي المثال التالي يتم إظهار قيمة الإحداثية top لشريط التمرير لأول عنصر p في الصفحة:

```

var p = $("p:first");
$("p:last").text( "scrollTop:" + p.scrollTop() );

```



أما الشيفرة الكاملة للمثال السابق فهي:

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>

  <script>
    $(document).ready(function() {
      var p = $("p:first");
      $("p:last").text( "scrollTop:" + p.scrollTop() );
    });
  </script>
  <style>
    p { margin:10px;padding:5px;border:2px solid #666; }
  </style>
</head>
<body>
  <p></p><p></p>
</body>
</html>

```

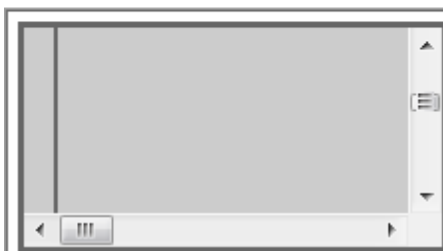
scrollTop(val)

يُستخدم هذا التابع لتعيين قيمة الإحداثية top لشريط التمرير العناصر التي تمت مطابقتها باستخدام المحدد الذي تم استدعاؤه قبل التابع، ويمكن استخدام هذا التابع لجميع العناصر الظاهرة والمخفية في الصفحة، ففي المثال السابق يتم تعيين قيمة الإحداثية top لجميع العناصر div التي تطبق الكلاس demo إلى القيمة 300:

```

$("div.demo").scrollTop(300);

```



أما الشيفرة الكاملة للمثال السابق فهي:

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>

  <script>
    $(document).ready(function() {
      $("#div.demo").scrollTop(300);

    });
  </script>
  <style>
div.demo {
background:#CCCCCC none repeat scroll 0 0;
border:3px solid #666666;
margin:5px;
padding:5px;
position:relative;
width:200px;
height:100px;
overflow:auto;
}
  p { margin:10px;padding:5px;border:2px solid #666;width:1000px;height:1000;
  </style>
</head>
<body>
  <div class="demo"><p>Hello</p></div>
</body>
</html>

```

scrollTop()

يُستخدم هذا التابع لإعادة قيمة الإحداثية left شريط التنقل الجانبي لأول عنصر يتم تحديده وتكون قيمة عددية صحيحة. يمكن استخدام هذا التابع مع العناصر الظاهرة والمخفية في الصفحة، وفي المثال التالي يتم إظهار قيمة الإحداثية left لشريط التمرير لأول عنصر p في الصفحة:

```

var p = $("#p:first");
$("#p:last").text( "scrollTop:" + p.scrollTop() );

```

Hello

scrollTop:0

أما الشيفرة الكاملة للمثال السابق فهي:

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>

  <script>
    $(document).ready(function() {
      var p = $("p:first");
      $("p:last").text( "scrollLeft:" + p.scrollLeft() );
    });
  </script>
  <style>
    p { margin:10px;padding:5px;border:2px solid #666; }
  </style>
</head>
<body>
  <p></p><p></p>
</body>
</html>

```

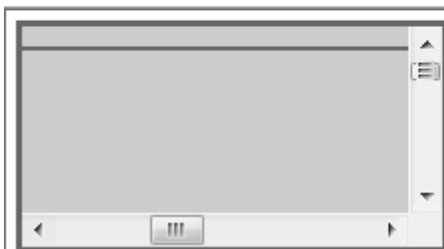
scrollLeft(val)

يُستخدم هذا التابع لتعيين قيمة الإحداثية left لشريط التمرير العناصر التي تمت مطابقتها باستخدام المحدد الذي تم استدعاؤه قبل التابع، ويمكن استخدام هذا التابع لجميع العناصر الظاهرة والمخفية في الصفحة، ففي المثال السابق يتم تعيين قيمة الإحداثية left لجميع العناصر div التي تطبق الكلاس demo إلى القيمة 300:

```

$("div.demo").scrollLeft(300);

```



أما الشيفرة الكاملة للمثال السابق فهي:

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>

  <script>
    $(document).ready(function() {
      $("#div.demo").scrollLeft(300);

    });
  </script>
  <style>
div.demo {
background:#CCCCCC none repeat scroll 0 0;
border:3px solid #666666;
margin:5px;
padding:5px;
position:relative;
width:200px;
height:100px;
overflow:auto;
}
  p { margin:10px;padding:5px;border:2px solid #666;width:1000px;height:1000;
  </style>
</head>
<body>
  <div class="demo"><p>Hello</p></div>
</body>
</html>

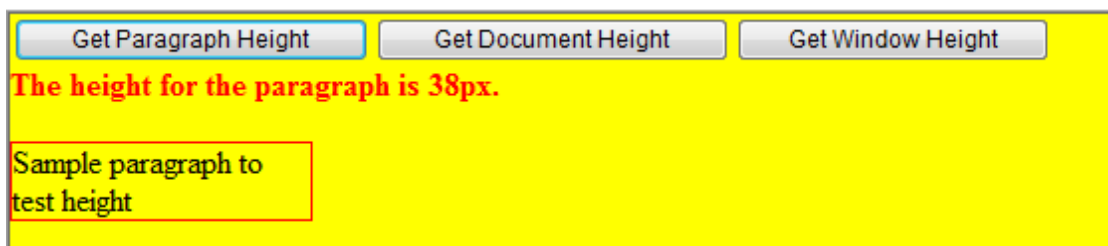
```

توابع العرض والارتفاع Height and Width Functions

height()

يُعيد هذا التابع ارتفاع أول عنصر يتم تحديده بأحد توابع التحديد وهذه القيمة تعبر عن عدد صحيح ووحدة القياس المستخدمة هي البيكسل, وفي المثال التالي يوجد ثلاثة أزار لطباعة ارتفاع العنصر p والوثيقة document والنافذة:

```
function showHeight(ele, h) {
    $("#div").text("The height for the " + ele +
        " is " + h + "px.");
}
$("#getp").click(function () {
    showHeight("paragraph", $("#p").height());
});
$("#getd").click(function () {
    showHeight("document", $(document).height());
});
$("#getw").click(function () {
    showHeight("window", $(window).height());
});
```



The screenshot shows a yellow background with three buttons at the top: "Get Paragraph Height" (highlighted in blue), "Get Document Height", and "Get Window Height". Below the buttons, the text "The height for the paragraph is 38px." is displayed in red. At the bottom, a red-bordered box contains the text "Sample paragraph to test height".

أما الشيفرة الكاملة للمثال السابق فهي:


```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>

  <script>
    $(document).ready(function() {

      function showHeight(ele, h) {
        $("#div").text("The height for the " + ele +
          " is " + h + "px.");
      }

      $("#getp").click(function () {
        showHeight("paragraph", $("#p").height());
      });
      $("#getd").click(function () {
        showHeight("document", $(document).height());
      });
      $("#getw").click(function () {
        showHeight("window", $(window).height());
      });

    });
  </script>
  <style>
  body { background:yellow; }
  button { font-size:12px; margin:2px; }
  p { width:150px; border:1px red solid; }
  div { color:red; font-weight:bold; }
</style>
</head>
<body>
  <button id="getp">Get Paragraph Height</button>

```

```

  <button id="getd">Get Document Height</button>
  <button id="getw">Get Window Height</button>
  <div> </div>
  <p>
    Sample paragraph to test height
  </p>
</body>
</html>

```

height(val)

يُستخدم هذا التابع لتعيين الارتفاع لجميع العناصر التي تمت مطابقتها باستخدام المحدد الذي تم استدعاؤه قبل التابع باستخدام الوسيط val ووحدة القياس الافتراضية المستخدمة هي البيكسل إلا إن تم تحديد وحدة قياس مختلفة مثل % أو em, وفي المثال التالي يتم تغيير ارتفاع أي عنصر div يتم الضغط عليه بالفأرة لمرة واحدة فقط:

```

$("div").one('click', function () {
    $(this).height(30)
        .css({cursor:"auto", backgroundColor:"green"});
});

```



أما الشيفرة الكاملة للمثال السابق فهي:

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
    <script src="http://code.jquery.com/jquery-latest.js"></script>

    <script>
        $(document).ready(function() {

            $("div").one('click', function () {
                $(this).height(30)
                    .css({cursor:"auto", backgroundColor:"green"});
            });

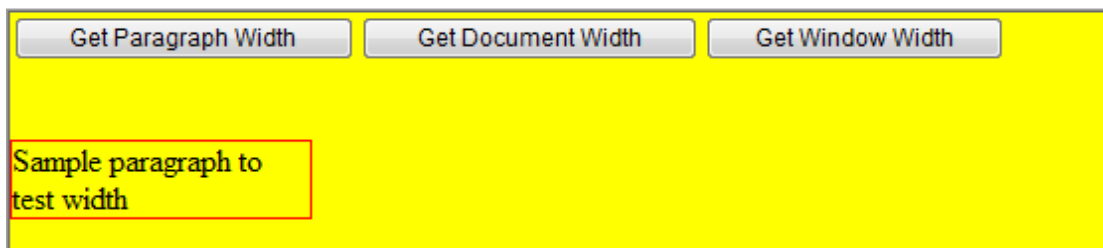
        });
    </script>
<style>
    div { width:50px; height:70px; float:left; margin:5px;
        background:rgb(255,140,0); cursor:pointer; }
</style>
</head>
<body>
    <div></div>
    <div></div>
    <div></div>
    <div></div>
    <div></div>
</body>
</html>

```

width()

يُعيد هذا التابع عرض أول عنصر يتم تحديده بأحد توابع التحديد وهذه القيمة تعبر عن عدد صحيح ووحدة القياس المستخدمة هي البيكسل، وفي المثال التالي يوجد ثلاثة أزرار لطباعة عرض العنصر p والوثيقة document والنافذة:

```
function showWidth(ele, w) {
    $("#div").text("The width for the " + ele +
        " is " + w + "px.");
}
$("#getp").click(function () {
    showWidth("paragraph", $("#p").width());
});
$("#getd").click(function () {
    showWidth("document", $(document).width());
});
$("#getw").click(function () {
    showWidth("window", $(window).width());
});
```



أما الشيفرة الكاملة للمثال السابق فهي:

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>

  <script>
    $(document).ready(function(){

      function showWidth(ele, w) {
        $("#div").text("The width for the " + ele +
          " is " + w + "px.");
      }
      $("#getp").click(function () {
        showWidth("paragraph", $("#p").width());
      });
      $("#getd").click(function () {
        showWidth("document", $(document).width());
      });
      $("#getw").click(function () {
        showWidth("window", $(window).width());
      });

    });
  </script>
  <style>
    body { background:yellow; }
    button { font-size:12px; margin:2px; }
    p { width:150px; border:1px red solid; }
    div { color:red; font-weight:bold; }
  </style>
</head>
<body>
  <button id="getp">Get Paragraph Width</button>

```

Overview

```

</body>
  <button id="getd">Get Document Width</button>
  <button id="getw">Get Window Width</button>
</div>
<p>
  Sample paragraph to test width
</p>
</body>
</html>

```

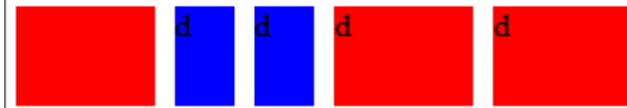
width(val)

يُستخدم هذا التابع لتعيين العرض لجميع العناصر التي تمت مطابقتها باستخدام المحدد الذي تم استدعاؤه قبل التابع باستخدام الوسيط val ووحدة القياس الافتراضية المستخدمة هي البيكسل إلا إن تم تحديد وحدة قياس مختلفة مثل % أو em, وفي المثال التالي يتم تغيير عرض أي عنصر div يتم الضغط عليه بالفأرة لمرة واحدة فقط:

```

$("div").one('click', function () {
    $(this).width(30)
        .css({cursor:"auto", "background-color":"blue"});
});

```



أما الشيفرة الكاملة للمثال السابق فهي:

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
    <script src="http://code.jquery.com/jquery-latest.js"></script>

    <script>
        $(document).ready(function(){

            $("div").one('click', function () {
                $(this).width(30)
                    .css({cursor:"auto", "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>

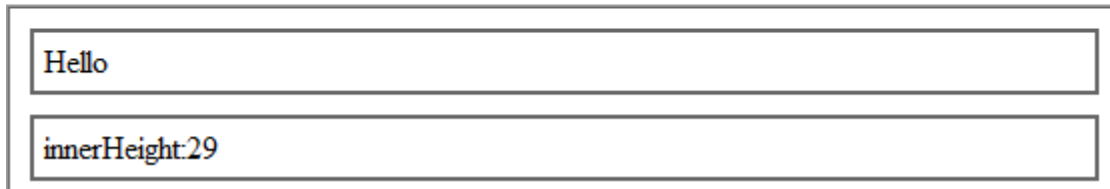
```

innerHeight()

يُعيد هذا التابع عدد صحيح يعبر عن قيمة الارتفاع الداخلي لأول عنصر يتم تحديده باستخدام أحد توابع التحديد ويستثني هذا التابع الإطار Border عند حساب الارتفاع الداخلي للعنصر, ويعمل هذا التابع مع

العناصر الظاهرة والعناصر المخفية في الصفحة, وفي المثال التالي يتم عرض قيمة الارتفاع الداخلي لأول عنصر p موجود في الصفحة:

```
var p = $("#p:first");
$("#p:last").text( "innerHeight:" + p.innerHeight() );
```



أما الشيفرة الكاملة للمثال السابق فهي:

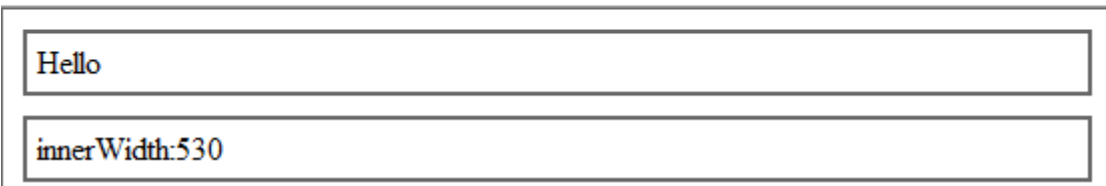
```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>

  <script>
    $(document).ready(function(){
      var p = $("#p:first");
      $("#p:last").text( "innerHeight:" + p.innerHeight() );
    });
  </script>
  <style>
    p { margin:10px;padding:5px;border:2px solid #666; }
  </style>
</head>
<body>
  <p>Hello</p><p></p>
</body>
</html>
```

innerWidth()

يُعيد هذا التابع عدد صحيح يعبر عن قيمة العرض الداخلي لأول عنصر يتم تحديده باستخدام أحد توابع التحديد ويستثني هذا التابع الإطار Border عند حساب العرض الداخلي للعنصر, ويعمل هذا التابع مع العناصر الظاهرة والعناصر المخفية في الصفحة, وفي المثال التالي يتم عرض قيمة العرض الداخلي لأول عنصر p موجود في الصفحة:

```
var p = $("#p:first");
$("#p:last").text( "innerWidth:" + p.innerWidth() );
```



أما الشيفرة الكاملة للمثال السابق فهي:

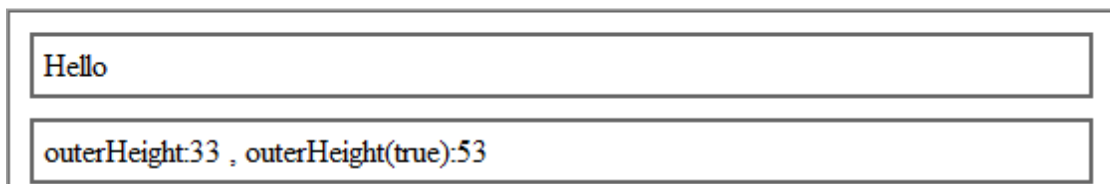
```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>

  <script>
    $(document).ready(function() {
      var p = $("#p:first");
      $("#p:last").text( "innerWidth:" + p.innerWidth() );
    });
  </script>
  <style>
    p { margin:10px;padding:5px;border:2px solid #666; }
  </style>
</head>
<body>
  <p>Hello</p><p></p>
</body>
</html>
```

outerHeight(margin)

يُعيد هذا التابع عدد صحيح يعبر عن قيمة الارتفاع الخارجي لأول عنصر يتم تحديده باستخدام أحد توابع التحديد ويتمضن هذا التابع الإطار Border عند حساب الارتفاع الخارجي للعنصر, ويعمل هذا التابع مع العناصر الظاهرة والعناصر المخفية في الصفحة, أما الوسيط الاختياري margin فعند تحديد قيمة true عندها يتم إدخال هوامش العنصر في عملية حساب الارتفاع الخارجي وفي المثال التالي يتم عرض قيمة العرض الخارجي لأول عنصر p موجود في الصفحة:

```
var p = $("p:first");
$("p:last").text( "outerHeight:" + p.outerHeight() + " , outerHeight(true):"
```



أما الشيفرة الكاملة للمثال السابق فهي:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>

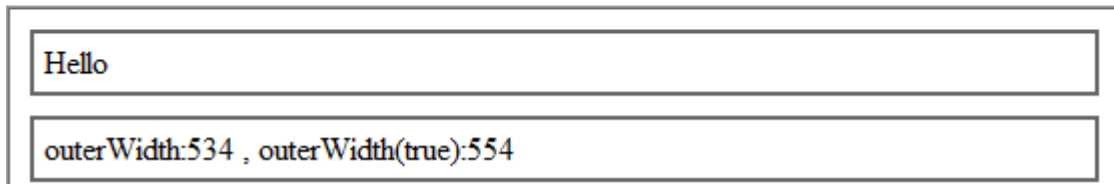
  <script>
    $(document).ready(function(){
      var p = $("p:first");
      $("p:last").text( "outerHeight:" + p.outerHeight() + " , outerHeight(true):"

    });
  </script>
  <style>
    p { margin:10px;padding:5px;border:2px solid #666; }
  </style>
</head>
<body>
  <p>Hello</p><p></p>
</body>
</html>
```

outerWidth(margin)

يُعيد هذا التابع عدد صحيح يعبر عن قيمة العرض الخارجي لأول عنصر يتم تحديده باستخدام أحد توابع التحديد ويتمضن هذا التابع الإطار Border عند حساب العرض الخارجي للعنصر, ويعمل هذا التابع مع العناصر الظاهرة والعناصر المخفية في الصفحة, أما الوسيط الاختياري margin فعند تحديد قيمة true عندها يتم إدخال هوامش العنصر في عملية حساب العرض الخارجي وفي المثال التالي يتم عرض قيمة العرض الخارجي لأول عنصر p موجود في الصفحة:


```
var p = $("#p:first");
$("#p:last").text( "outerWidth:" + p.outerWidth()+ " , outerWidth(true):" + p
```



أما الشيفرة الكاملة للمثال السابق فهي:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>

  <script>
    $(document).ready(function(){
      var p = $("#p:first");
      $("#p:last").text( "outerWidth:" + p.outerWidth()+ " , outerWidth(true):" + p
    });
  </script>
  <style>
    p { margin:10px;padding:5px;border:2px solid #666; }
  </style>
</head>
<body>
  <p>Hello</p><p></p>
</body>
</html>
```

توابع الأحداث Events Functions

توابع تحميل الصفحة Page Load Functions

ready(fn)

يُستخدم هذا التابع لتهيئة التابع الذي سيتم تنفيذه عندما تصبح عناصر الصفحة جاهزة لتنفيذ تعليمات الـ JavaScript عليها، ويعتبر هذا التابع من أهم التوابع التي يجب أن تستخدمها حيث تحسن من أداء تطبيقات الويب التي تستخدم مكتبة الـ jQuery. ويعتبر هذا التابع بديلاً عن الحدث window.onload وربط التوابع معه، وباستخدام التابع ready سيتم تنفيذ التابع الذي تحدده بالوسيط fn مباشرة عندما يصبح بالإمكان قراءة عناصر الوثيقة و 99.99% من شيفرات الـ JavaScript تبدأ عملها بعد تحميل الصفحة مباشرة.

ويمكن استدعاء التابع ready باستخدام الصيغة \$(document).ready أكثر من مرة داخل صفحتك عندها سيتم تنفيذ التوابع المتصلة معها بالترتيب الذي ذكرت فيه.

ملاحظة: يجب التأكد من أن جميع عناصر الـ CSS يتم تضمينها في الصفحة قبل استدعاء توابع الـ jQuery لتضمن بأن جميع خصائص العناصر تم تعريفها بشكل صحيح قبل البدء بتنفيذ شيفرة jQuery وإلا ستظهر مشاكل متعددة نتيجة لذلك, والمثال التالي يطبع رسالة عندما تصبح عناصر الوثيقة جاهزة للتعامل معها:

```
$(document).ready(function () {  
    $("p").text("The DOM is now loaded and can be manipulated.");  
});
```

The DOM is now loaded and can be manipulated.

أما الشيفرة الكاملة للمثال السابق فهي:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"  
    "http://www.w3.org/TR/html4/loose.dtd">  
<html>  
<head>  
    <script src="http://code.jquery.com/jquery-latest.js"></script>  
  
    <script>  
        $(document).ready(function() {  
            $(document).ready(function () {  
                $("p").text("The DOM is now loaded and can be manipulated.");  
            });  
        });  
    </script>  
    <style>p { color:red; }</style>  
</head>  
<body>  
    <p>  
    </p>  
</body>  
</html>
```

توابع التعامل مع الحدث Event Handling Functions

bind(type, data, fn)

يُستخدم هذا التابع لربط تابع للتعامل مع حدث أو أكثر لجميع العناصر التي تمت مطابقتها باستخدام المحدد الذي تم استدعاؤه قبل التابع, كما يمكن استخدام التابع bind لربط تابع مع أحداث مخصصة, والقيم الممكنة للوسيط type الذي يعبر عن حدث معين هي:

blur, focus, load, resize, scroll, unload, beforeunload, click, dblclick, mousedown, mouseup, mousemove, mouseover, mouseout, mouseenter, mouseleave, change, select, submit, keydown, keypress, keyup, error

والتابع الممثل بالوسيط fn له وسيط e يحتوي معلومات عن الحدث والعنصر الذي أطلق الحدث, ويمكن استخدام هذا الكائن لإلغاء الوظيفة الافتراضية للحدث ومنع أيضاً عملية توريث الحدث أو ما يسمى بـ Event Bubbling, وطبعاً لتنفيذ ذلك فإن التابع fn يجب أن يعيد القيمة false, وطبعاً هذا سيمنع التوابع الأخرى المتصلة بنفس الحدث في العناصر الآباء للعنصر الحالي من العمل ولكن لا يؤثر على التوابع الأخرى في نفس العنصر. وفي معظم الحالات تستطيع تعريف التابع fn على أنه تابع مجهول أو ما يسمى anonymous function وفي حال عدم مقدرتك على ذلك يمكن تمرير معلومات إضافية تتمثل بالوسيط الاختياري الثاني data.

المثال التالي يُظهر طريقة ربط الحدث click لجميع العناصر p بتابع يُظهر رسالة تحتوي على نص العنصر p:

```
$("#p").bind("click", function(){
    alert( $(this).text() );
});
```

أما المثال الثاني فهو يُظهر طريقة تمرير معلومات إضافية للتابع fn:

```
function handler(event) {
    alert(event.data.foo);
}
$("#p").bind("click", {foo: "bar"}, handler)
```

المثال الثالث يُظهر طريقة منع الحدث submit في النموذج form من العمل ومنع عملية توريث الحدث إلى العناصر الآباء وذلك بإعادة القيمة false:

```
$("#form").bind("submit", function() { return false; })
```

أما المثال الرابع فيظهر كيف يمكن منع عمل الحدث الافتراضي فقط باستخدام التابع preventDefault():

```
$("#form").bind("submit", function(event){
    event.preventDefault();
});
```

أما المثال الأخير فيظهر كيفية منع توريث الحدث إلى العناصر الآباء للعنصر الحالي باستخدام التابع stopPropagation:

```
$("#form").bind("submit", function(event){
    event.stopPropagation();
});
```

والمثال التالي يُظهر كيفية ربط التابع fn بحدث مخصص غير موجود اسمه myCustomEvent لينطلق هذا الحدث عند الضغط على الزر:

```
$("#p").bind("myCustomEvent", function(e, myName, myValue){
    $(this).text(myName + ", hi there!");
    $("#span").stop().css("opacity", 1)
        .text("myName = " + myName)
        .fadeIn(30).fadeOut(1000);
});
$("#button").click(function () {
    $("#p").trigger("myCustomEvent", [ "John" ]);
});
```

Has an attached custom event.

Trigger custom event

أما الشيفرة الكاملة للمثال السابق فهي:

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>

  <script>
    $(document).ready(function() {

      $("p").bind("myCustomEvent", function(e, myName, myValue) {
        $(this).text(myName + ", hi there!");
        $("span").stop().css("opacity", 1)
          .text("myName = " + myName)
          .fadeIn(30).fadeOut(1000);
      });
      $("button").click(function () {
        $("p").trigger("myCustomEvent", [ "John" ]);
      });

    });
  </script>
  <style>
    p { color:red; }
    span { color:blue; }
  </style>
</head>
<body>
  <p>Has an attached custom event.</p>
  <button>Trigger custom event</button>
  <span style="display:none;"></span>
</body>
</html>

```

one(type, [data], fn)

يُستخدم هذا التابع لربط التابع المحدد بالوسيط `fn` مع الحدث المحدد بالوسيط `type` لجميع العناصر التي تمت مطابقتها باستخدام المحدد الذي تم استدعاؤه قبل التابع، وهذا التابع سيتم تنفيذه مرة واحدة فقط لكل عنصر استجابة للحدث `type`.

والتابع الممثل بالوسيط `fn` له وسيط `e` يحتوي معلومات عن الحدث والعنصر الذي أطلق الحدث، ويمكن استخدام هذا الكائن لإلغاء الوظيفة الافتراضية للحدث ومنع أيضاً عملية توريث الحدث أو ما يسمى بـ `Event Bubbling`، وطبعاً لتنفيذ ذلك فإن التابع `fn` يجب أن يعيد القيمة `false`، وطبعاً هذا سيمنع التوابع الأخرى المتصلة بنفس الحدث في العناصر الآباء للعنصر الحالي من العمل ولكن لا يؤثر على التوابع الأخرى في نفس العنصر.

وفي معظم الحالات تستطيع تعريف التابع fn على أنه تابع مجهول أو ما يسمى anonymous function وفي حال عدم مقدرتك على ذلك يمكن تمرير معلومات إضافية تتمثل بالوسيط الاختياري الثاني .data.

والقيم الممكنة للوسيط type هي:

blur, focus, load, resize, scroll, unload, beforeunload, click, dblclick, mousedown, mouseup, mousemove, mouseover, mouseout, mouseenter, mouseleave, change, select, submit, keydown, keypress, keyup, error

وفي المثال التالي يتم ربط تابع مع الحدث click لجميع العناصر div الموجودة في الصفحة:

```
var n = 0;
$("#div").one("click", function(){
    var index = $("#div").index(this);
    $(this).css({ borderStyle:"inset",
        cursor:"auto" });
    $("#p").text("Div at index #" + index + " clicked." +
        " That's " + ++n + " total clicks.");
});
```



أما الشيفرة الكاملة للمثال السابق فهي:

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>

  <script>
    $(document).ready(function(){

      var n = 0;
      $("div").one("click", function(){
        var index = $("div").index(this);
        $(this).css({ borderStyle:"inset",
          cursor:"auto" });
        $("p").text("Div at index #" + index + " clicked." +
          " That's " + ++n + " total clicks.");
      });

    });
  </script>
  <style>
  div { width:60px; height:60px; margin:5px; float:left;
    background:green; border:10px outset;
    cursor:pointer; }
  p { color:red; margin:0; clear:left; }
  </style>
</head>
<body>
  <div></div>
  <div></div>
  <div></div>
  <div></div>
  <div></div>
  <p>Click a green square...</p>
</body>

```

trigger(event, [data])

يُستخدم هذا التابع لفتح حدث من أحدث جميع العناصر التي تمت مطابقتها باستخدام المحدد الذي تم استدعاؤه قبل التابع، مما يؤدي إلى أن الوظيفة الافتراضية في حال أهما موجودة سيتم تنفيذها من قبل المتصفح. على سبيل المثال تمرير القيمة submit إلى التابع trigger سيؤدي إلى إرسال معلومات النموذج form أو ما يطلق عليه submit ويمكن منع هذا الحدث من تنفيذ وظيفته الافتراضية بإعادة القيمة false من أحد التوابع المرتبطة بالحدث باستخدام أحد توابع الربط مثل bind و one. كما يمكن للتابع trigger فتح أحداث خاصة غير موجودة بشكل افتراضي ولكن هذا الحدث يجب أن يكون مسجلاً باستخدام التابع bind.

التابع المتمثل بالوسيط `fn` له وسيط `e` يحتوي على معلومات مفيدة مثل العنصر الذي قدح الحدث. والجديد في jQuery 1.3 هو أن جميع الأحداث التي يتم قدحها باستخدام التابع `trigger` تورث الحدث المقدوح إلى جميع العناصر الآباء للعنصر الحالي. فعلى سبيل المثال في حال قدح حدث على مستوى العنصر `p` فإن هذا الحدث سيتم قدحه للعنصر الأب للعنصر `p` وهكذا حتى يصل الحدث إلى مستوى الكائن `document`. والكائن `e` الذي هو وسيط `fn` سيحتوي على الخاصية `target` الذي يمثل العنصر الأول على قُدح الحدث له. ويمكن إيقاف عملية تورث الحدث باستدعاء التابع `stopPropagation` أو بإعادة القيمة `false` من التابع `fn`.

المثال التالي يرسل معلومات أول نموذج في الصفحة إلى الخادم بدون الضغط على زر `submit`:

```
$("#form:first").trigger("submit")
```

أما المثال التالي فيُظهر كيفية إرسال معلومات النموذج الأول في الصفحة إلى الخادم بتمرير كائن من النوع `jQuery.Event` بدلاً من القيمة النصية `submit`:

```
var event = jQuery.Event("submit");
$("#form:first").trigger(event);
if ( event.isDefaultPrevented() ) {
    // Perform an action...
}
```

أما المثال التالي فيُظهر كيفية تمرير بيانات عند قدح الحدث `click` لأي عنصر `p` موجود في الصفحة:

```
$("#p").click( function (event, a, b) {
    // when a normal click fires, a and b are undefined
    // for a trigger like below a refers too "foo" and b refers to "bar"
} ).trigger("click", ["foo", "bar"]);
```

والمثال التالي يُظهر كيفية تمرير بيانات عند قدح الحدث `logged` للعنصر `body` ولكن باستخدام الكائن `jQuery.Event`:

```
var event = jQuery.Event("logged");
event.user = "foo";
event.pass = "bar";
$("#body").trigger(event);
```

وهذه طريقة أخرى لتمرير بيانات عند قدح الحدث `logged` للعنصر `body` باستخدام الكائن `jQuery.Event`:


```
$("#body").trigger({
  type:"logged",
  user:"foo",
  pass:"bar"
});
```

أما المثال التالي ففيه يتم قذح الحدث click للزر 1 button من خلال الضغط على الزر 2 button:

```
$("#button:first").click(function () {
  update($("#span:first"));
});
$("#button:last").click(function () {
  $("#button:first").trigger('click');

  update($("#span:last"));
});

function update(j) {
  var n = parseInt(j.text(), 10);
  j.text(n + 1);
}
```

Button #1Button #2

8 button #1 clicks.
6 button #2 clicks.

أما الشيفرة الكاملة للمثال السابق فهي:

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>

  <script>
    $(document).ready(function() {

      $("button:first").click(function () {
        update($("#span:first"));
      });
      $("button:last").click(function () {
        $("#button:first").trigger('click');

        update($("#span:last"));
      });

      function update(j) {
        var n = parseInt(j.text(), 10);
        j.text(n + 1);
      }

    });
  </script>
  <style>
  button { margin:10px; }
  div { color:blue; font-weight:bold; }
  span { color:red; }
  </style>
</head>
<body>
  <button>Button #1</button>
  <button>Button #2</button>

```

```

<div><span>0</span> button #1 clicks.</div>
<div><span>0</span> button #2 clicks.</div>
</body>
</html>

```

triggerHandler(event, [data])

يُستخدم هذا التابع لفتح جميع التوابع المرتبطة بأحد أحداث العنصر الحالي, وهذا التابع يشابه تماماً التابع

trigger بوجود فارقين فقط:

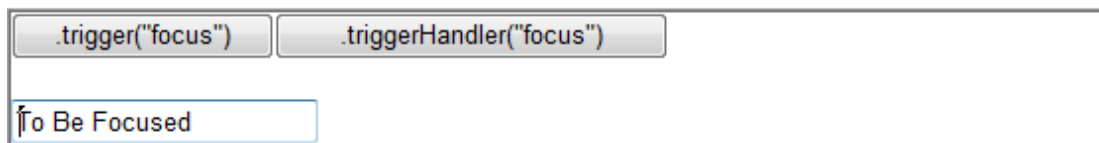
- الأول هو أن الوظيفة الافتراضية للحدث المدعومة من قبل المتصفح لا تعمل ولا يتم توريث الحدث إلى العناصر الآباء والأحداث Live Events لا يتم قدها أيضاً.

- يتم قرح الحدث فقط لأول عنصر من مجموعة العناصر jQuery, ويعيد هذا التابع نفس القيمة التي يعيدها التابع المرتبط بالحدث event للعنصر الأول وفي حال أن مجموعة العناصر jQuery لا تحتوي على عناصر فإن هذا التابع يعيد القيمة undefined, والوسيط event يقبل قيمة نصية أو مثل عن الكائن jQuery.Event. ففي المثال التالي يتم عرض تابعي القرح trigger و triggerHandler ولا حظ عند الضغط على الزر الثاني بأنه سيتم تشغيل التابع المرتبط بالحدث focus بدون أن يتم تشغيل الوظيفة الافتراضية للحدث focus مع العلم أن الوظيفة الافتراضية للحدث focus هي أن يصبح مؤشر الكتابة داخل الحقل النصي:

```

$("#old").click(function() {
    $("input").trigger("focus");
});
$("#new").click(function() {
    $("input").triggerHandler("focus");
});
$("input").focus(function() {
    $("<span>Focused!</span>").appendTo("body").fadeOut(1000);
});

```



أما الشيفرة الكاملة للمثال السابق فهي:

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">

<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>

  <script>
    $(document).ready(function() {

      $("#old").click(function() {
        $("input").trigger("focus");
      });
      $("#new").click(function() {
        $("input").triggerHandler("focus");
      });
      $("input").focus(function() {
        $("<span>Focused!</span>").appendTo("body").fadeOut(1000);
      });

    });
  </script>

</head>
<body>
  <button id="old">.trigger("focus")</button>
  <button id="new">.triggerHandler("focus")</button><br/><br/>
  <input type="text" value="To Be Focused"/>
</body>
</html>

```

unbind([type], [fn])

وظيفة هذا التابع هي عكس وظيفة التابع bind حيث يتم فك ارتباط أحداث جميع العناصر التي تمت مطابقتها باستخدام المحدد الذي تم استدعاؤه قبل التابع بالتتابع التي ترتبط معها، وعند عدم تمرير أي وسيط فعندها سيطبق التابع على جميع الأحداث، أما في حال تمرير قيمة للوسيط الاختياري type عندها سيتم فك ارتباط جميع الأحداث التي توافقت قيمة الوسيط type، أما في حالة تمرير قيمة الوسيط fn أيضاً عندها سيتم فك ارتباط التتابع التي تتوافق مع قيمة الوسيط fn، كما يمكن استخدام هذا التابع لفك ارتباط الأحداث المخصصة، وفي المثال التالي يتم فك ارتباط التابع aClick من الزر الأصفر عند الضغط على الزر Unbind Click:

```
function aClick() {
    $("#div").show().fadeOut("slow");
}
$("#bind").click(function () {
    // could use .bind('click', aClick) instead but for variety...
    $("#theone").click(aClick)
        .text("Can Click!");
});
$("#unbind").click(function () {
    $("#theone").unbind('click', aClick)
        .text("Does nothing...");
});
```

Does nothing...

Bind Click

Unbind Click

أما الشيفرة الكاملة للمثال السابق فهي:

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>

  <script>
    $(document).ready(function(){

      function aClick() {
        $("#div").show().fadeOut("slow");
      }
      $("#bind").click(function () {
        // could use .bind('click', aClick) instead but for variety...
        $("#theone").click(aClick)
          .text("Can Click!");
      });
      $("#unbind").click(function () {
        $("#theone").unbind('click', aClick)
          .text("Does nothing...");
      });

    });
  </script>
  <style>
    button { margin:5px; }
    button#theone { color:red; background:yellow; }
  </style>
</head>
<body>
  <button id="theone">Does nothing...</button>
  <button id="bind">Bind Click</button>
  <button id="unbind">Unbind Click</button>
  <div style="display:none;">Click!</div>
</body>
</html>

```

توابع الأحداث المباشرة Live Events Functions

live(type, fn)

تم إضافة هذا التابع في jQuery 1.3 ويُستخدم لربط الحدث المتمثل بالوسيط type الذي يقبل قيمة نصية تعبر عن الحدث بالتابع المتمثل بالوسيط fn وتتم عملية الربط على جميع العناصر التي تمت مطابقتها باستخدام الحدد الذي تم استدعاؤه قبل التابع الحالية والعناصر المماثلة التي سيتم إضافتها مستقبلاً، والقيم الممكنة للوسيط type: click, dblclick, mousedown, mouseup, mousemove, mouseover, mouseout, keydown, keypress, keyup

على عكس التابع bind فإنه يمكن استخدام التابع live لربط حدث واحد type مع التابع fn في كل مرة يتم استدعاء التابع live. والتابع live يعمل بشكل مشابه للتابع bind مع بعض الفروق الهامة:

- عندما تستخدم التابع live لربط حدث عندها سيتم تطبيق عملية الربط على جميع العناصر الحالية والعناصر الجديدة في الصفحة, على سبيل المثال عند ربط الحدث click لجميع العناصر li الحالية في الصفحة فإن عملية ربط الحدث click ستتأثر بها العناصر li التي يتم إضافتها إلى الصفحة, وهذه الحالة غير موجودة عند استخدام التابع bind.
- الأحداث المباشرة لا يتم توريثها للعناصر الآباء بالطريقة التقليدية ولا يمكن إيقاف هذه الأحداث باستخدام التابع stopPropagation أو التابع stopImmediatePropagation, فعلى سبيل المثال عند ربط حدثي click واحد للعنصر li والآخر للعنصر li a فعند الضغط على العنصر li a بمؤشر الفأرة فإن الحدثين على li a والعنصر li سيتم قدهما لأنه في حال استخدام التابع bind لربط الأحداث فإن الحدث الذي يحدث ضمن عنصر داخلي يؤدي إلى توريث الحدث إلى العنصر الأب كما حدث في مثالنا السابق.
- التابع live يعمل فقط بعد استخدام توابع التحديد, فمثلاً \$(li a).live(...) سيعمل أما استخدام التابع \$(a, someElem).live(...) لن يعمل أيضاً المثال التالي لن يعمل \$(a).parent().live(...).

ولإزالة الربط مع الأحداث المباشرة يجب استخدام التابع die, والمثال التالي يوضح كيفية عمل التابع live, حيث أن العنصر الجديد p الذي تتم إضافته عند الضغط على العنصر p الأول فإن الحدث click سيكون فعالاً عند الضغط عليه:

```
$( "p" ).live( "click", function () {
    $( this ).after( "<p>Another paragraph!</p>" );
} );
```

Click me!

أما الشيفرة الكاملة للمثال السابق فهي:

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>

  <script>
    $(document).ready(function(){

      $("p").live("click", function(){
        $(this).after("<p>Another paragraph!</p>");
      });

    });
  </script>
  <style>
    p { background:yellow; font-weight:bold; cursor:pointer;
      padding:5px; }
    p.over { background: #ccc; }
    span { color:red; }
  </style>
</head>
<body>
  <p>Click me!</p>
  <span></span>
</body>
</html>

```

die(type, fn)

يُستخدم هذا التابع لإزالة جميع الأحداث المرتبطة بتوابع باستخدام التابع live وبدون تحديد أي قيمة للوسيطين type و fn عندها يتم إزالة جميع الأحداث المرتبطة، أما في حال تحديد قيمة الوسيط type عندها سيتم إزالة الأحداث التي تطابق قيمة الوسيط type، وبتحديد قيمة الوسيط fn سيتم إزالة جميع الأحداث المرتبطة التي تتطابق مع قيمة الوسيط type وقيمة الوسيط fn.


```
function aClick() {
    $("#div").show().fadeOut("slow");
}
$("#bind").click(function () {
    $("#theone").live("click", aClick)
                .text("Can Click!");
});
$("#unbind").click(function () {
    $("#theone").die("click", aClick)
                .text("Does nothing...");
});
```

Does nothing...

Bind Click

Unbind Click

أما الشيفرة الكاملة للمثال السابق فهي:

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>

  <script>
    $(document).ready(function() {

      function aClick() {
        $("#div").show().fadeOut("slow");
      }
      $("#bind").click(function () {
        $("#theone").live("click", aClick)
          .text("Can Click!");
      });
      $("#unbind").click(function () {
        $("#theone").die("click", aClick)
          .text("Does nothing...");
      });

    });
  </script>
  <style>
  button { margin:5px; }
  button#theone { color:red; background:yellow; }
  </style>
</head>
<body>
  <button id="theone">Does nothing...</button>
  <button id="bind">Bind Click</button>
  <button id="unbind">Unbind Click</button>
  <div style="display:none;">Click!</div>
</body>
</html>

```

توابع التفاعل مع العناصر Interaction Helpers Functions

hover(over, out)

يحاكي هذا التابع حركة مؤشر الفأرة فوق العنصر الحالي, حيث يتم تنفيذ التابع المحدد بالوسيط OVER عند وضع مؤشر الفأرة على العنصر بينما يتم تنفيذ التابع المحدد بالوسيط OUT عند خروج مؤشر الفأرة من فوق العنصر الحالي ويتم تنفيذ التابع OVER مرة واحدة فقط عند وضع مؤشر الفأرة على العنصر.

في المثال التالي يتم إضافة الكلاس hover إلى أي عنصر td يتم وضع مؤشر الفأرة فوقها, وعند إخراج مؤشر الفأرة من العنصر td يتم إزالة الكلاس hover من هذا العنصر:

```

$("td").hover(
  function () {
    $(this).addClass("hover");
  },
  function () {
    $(this).removeClass("hover");
  }
);

```

أما الشيفرة التالية فتمنع تنفيذ التابع hover عند وضع مؤشر الفأرة فوق العنصر td وذلك بفك ارتباط الحدثين mouseenter والحدث mouseleave من جميع العناصر td باستخدام تابع فك الارتباط unbind:

```

$("td").unbind('mouseenter mouseleave');

```

أما في المثال التالي فيتم تغيير بعض خصائص العناصر li عند وضع مؤشر الفأرة فوقها وإعادةتها إلى حالتها السابقة بعد خروج مؤشر الفأرة:

```

$("li").hover(
  function () {
    $(this).append($("<span> ***</span>"));
  },
  function () {
    $(this).find("span:last").remove();
  }
);

//li with fade class
$("li.fade").hover(function() {$(this).fadeOut(100);$(this).fadeIn(500);});

```

- Milk
- Bread
- Chips
- Socks

أما الشيفرة الكاملة للمثال السابق فهي:

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>

  <script>
    $(document).ready(function() {

      $("li").hover(
        function () {
          $(this).append("<span> ***</span>");
        },
        function () {
          $(this).find("span:last").remove();
        }
      );

      //li with fade class
      $("li.fade").hover(function() {$(this).fadeOut(100);$(this).fadeIn(500);});

    });
  </script>
  <style>
  ul { margin-left:20px; color:blue; }
  li { cursor:default; }
  span { color:red; }
  </style>
</head>
<body>
  <ul>
    <li>Milk</li>

```

```

    <li>Bread</li>
    <li class='fade'>Chips</li>
    <li class='fade'>Socks</li>
  </ul>
</body>
</html>

```

toggle(fn1, fn2, [fn3], [fn4], ...)

يُستخدم هذا التابع لتنفيذ تابع واحد في كل مرة يتم الضغط على العنصر المحدد، حيث يتم تنفيذ التابع fn1 عند الضغط لأول مرة على العنصر بمؤشر الفأرة ومن ثم يتم تنفيذ التابع fn2 عند الضغط للمرة الثانية بمؤشر الفأرة على العنصر وهكذا، وفي المثال التالي يتم تغيير الكلاس لكل خلية يتم الضغط عليها بمؤشر الفأرة، حيث يتم في المرة الأولى إضافة الكلاس selected وفي المرة الثانية يتم إزالة الكلاس selected:

```

$("td").toggle(
  function () {
    $(this).addClass("selected");
  },
  function () {
    $(this).removeClass("selected");
  }
);

```

أما الشيفرة الكاملة للمثال السابق فهي:

```

<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <title></title>
  <script src="jquery-1.3.2.min.js" type="text/javascript"></script>
  <style type="text/css">
    .selected
    {
      background-color: Yellow;
    }
  </style>
  <script type="text/javascript">
    $(function() {
      $("td").toggle(
        function() {
          $(this).addClass("selected");
        },
        function() {
          $(this).removeClass("selected");
        }
      );

      // $("td").unbind('mouseenter mouseleave');
    });
  </script>
</head>
<body>
  <table><tr><td>Hello</td></tr></table>
</body>

```

توابع الأحداث Event Helpers Functions

blur()

يُستخدم هذا التابع لفتح الحدث blur لجميع العناصر التي تمت مطابقتها باستخدام الحدد الذي تم استدعاؤه قبل التابع, عند استدعاء هذا التابع يتم تنفيذ جميع التوابع المرتبطة بالحدث blur للعنصر كما يتم تنفيذ الوظيفة الافتراضية للحدث blur كما يمكن منع تنفيذ الوظيفة الافتراضية للحدث blur بإعادة القيمة false من أحد التوابع المرتبطة بالحدث blur. ومن المعلوم أن الحدث blur يتم قدهه عندما يفقد العنصر التركيز إما

باستخدام الزر Tab أو باستخدام مؤشر الفأرة. وفي المثال التالي يتم قده الحدث blur لجميع العناصر p الموجودة في الصفحة:

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

blur(fn)

يُستخدم هذا التابع لربط التابع المحدد بالوسيط fn بالحدث blur لجميع العناصر التي تمت مطابقتها باستخدام المحدد الذي تم استدعاؤه قبل التابع, وفي المثال التالي يتم ربط الحدث blur لكل عناصر الـ input وهنا تتمثل بالحقلين النصيين:

```
$("#input").blur(function () {
    $(this).next("span").css('display','inline').fadeOut(1000);
});
```

blur fire

أما الشيفرة الكاملة للمثال السابق فهي:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<script src="http://code.jquery.com/jquery-latest.js"></script>
<script>
$(document).ready(function() {
    $("#input").blur(function () {
        $(this).next("span").css('display','inline').fadeOut(1000);
    });
});
</script>
<style>span {display:none;}</style>
</head>
<body>
<p><input type="text" /> <span>blur fire</span></p>
<p><input type="password" /> <span>blur fire</span></p>
</body>
</html>
```

change()

يُستخدم هذا التابع لفتح الحدث `change` لجميع العناصر التي تمت مطابقتها باستخدام المحدد الذي تم استدعاؤه قبل التابع, عند استدعاء هذا التابع يتم تنفيذ جميع التوابيع المرتبطة بالحدث `change` للعنصر كما يتم تنفيذ الوظيفة الافتراضية للحدث `change` ويمكن منع تنفيذ الوظيفة الافتراضية للحدث `change` بإعادة القيمة `false` من أحد التوابيع المرتبطة بالحدث `change`. ومن المعلوم أن الحدث `change` يتم قده عندما يفقد العنصر التركيز إما باستخدام الزر `Tab` أو باستخدام مؤشر الفأرة وقد تم تغيير قيمته.

change(fn)

يُستخدم هذا التابع لربط التابع المحدد بالوسيط `fn` بالحدث `change` لجميع العناصر التي تمت مطابقتها باستخدام المحدد الذي تم استدعاؤه قبل التابع, وفي المثال التالي يتم ربط الحدث `change` لكل عنصر `select` موجود في الصفحة مع التابع الذي يعرض القيم التي تم تحديدها في العنصر:

```
$("#select").change(function () {
    var str = "";
    $("#select option:selected").each(function () {
        str += $(this).text() + " ";
    });
    $("#div").text(str);
})
.change();
```



Chocolate Caramel

أما الشيفرة الكاملة للمثال السابق فهي:

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>

  <script>
    $(document).ready(function() {

      $("#select").change(function () {
        var str = "";
        $("#select option:selected").each(function () {
          str += $(this).text() + " ";
        });
        $("#div").text(str);
      })
      .change();

    });
  </script>
  <style>
  div { color:red; }
  </style>
</head>
<body>
  <select name="sweets" multiple="multiple">
    <option>Chocolate</option>
    <option selected="selected">Candy</option>
    <option>Taffy</option>
    <option selected="selected">Caramel</option>
    <option>Fudge</option>
    <option>Cookie</option>
  </select>
  <div></div>
</body>
</html>

```

click()

يُستخدم هذا التابع لقدح الحدث click لجميع العناصر التي تمت مطابقتها باستخدام المحدد الذي تم استدعاؤه قبل التابع, عند استدعاء هذا التابع يتم تنفيذ جميع التوابيع المرتبطة بالحدث click للعنصر. وفي المثال التالي يتم قدح الحدث click لجميع العناصر p الموجودة في الصفحة:

```

$("#p").click();

```


click(fn)

يُستخدم هذا التابع لربط الحدث click لجميع العناصر التي تمت مطابقتها باستخدام المحدد الذي تم استدعاؤه قبل التابع, ويتم قرح الحدث click عندما يتم بالضغط على مؤشر الفأرة فوق عنصر ما, ويتم تعريف عملية النقر click بأها الحدثين mousedown و mouseup في نفس المكان من الشاشة وتسلسل الأحداث كما يلي:

- mousedown
- mouseup
- click

وفي المثال التالي يتم ربط الحدث click لجميع العناصر p الموجودة في الصفحة بتابع يؤدي إلى إخفاء العنصر p عند الضغط عليه:

```
$("#p").click(function () {
    $(this).slideUp();
});
$("#p").hover(function () {
    $(this).addClass("hilite");
}, function () {
    $(this).removeClass("hilite");
});
```

First Paragraph

Second Paragraph

Yet one more Paragraph

أما الشيفرة الكاملة للمثال السابق فهي:

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>

  <script>
    $(document).ready(function() {

      $("p").click(function () {
        $(this).slideUp();
      });
      $("p").hover(function () {
        $(this).addClass("hilite");
      }, function () {
        $(this).removeClass("hilite");
      });

    });
  </script>
  <style>
  p { color:red; margin:5px; cursor:pointer; }
  p.hilite { background:yellow; }
  </style>
</head>
<body>
  <p>First Paragraph</p>
  <p>Second Paragraph</p>
  <p>Yet one more Paragraph</p>
</body>
</html>

```

dblclick()

يُستخدم هذا التابع لفتح الحدث `dblclick` لجميع العناصر التي تمت مطابقتها باستخدام المحدد الذي تم استدعاؤه قبل التابع، عند استدعاء هذا التابع يتم تنفيذ جميع التوابع المرتبطة بالحدث `dblclick` للعنصر كما يتم تنفيذ الوظيفة الافتراضية للحدث `dblclick`، ويمكن منع تنفيذ الوظيفة الافتراضية للحدث `dblclick` بإعادة القيمة `false` من أحد التوابع المرتبطة بالحدث `dblclick`. ومن المعلوم أن الحدث `dblclick` يتم قدهه عندما عندما يتم بالضغط على مؤشر الفأرة فوق عنصر ما مرتين متتاليتين.

dblclick(fn)

يُستخدم هذا التابع لربط الحدث `click` لجميع العناصر التي تمت مطابقتها باستخدام المحدد الذي تم استدعاؤه قبل التابع، ويتم فتح الحدث `click` عندما يتم بالضغط على مؤشر الفأرة فوق عنصر ما مرتين متتاليتين، وفي المثال التالي يتم ربط الحدث `dblclick` لأول عنصر `div` موجود في الصفحة بتابع يغير بعض خصائص العنصر:

```
var divdbl = $("div:first");
divdbl.dblclick(function () {
    divdbl.toggleClass('dbl');
});
```



Double click the block

أما الشيفرة الكاملة للمثال السابق فهي:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
    <script src="http://code.jquery.com/jquery-latest.js"></script>

    <script>
    $(document).ready(function(){

        var divdbl = $("div:first");
        divdbl.dblclick(function () {
            divdbl.toggleClass('dbl');
        });

    });
    </script>
    <style>
    div { background:blue;
        color:white;
        height:100px;
        width:150px;
    }
    div.dbl { background:yellow;color:black; }
    </style>
</head>
<body>
    <div></div><span>Double click the block</span>
</body>
</html>
```

error()

يُستخدم هذا التابع لفتح الحدث `error` لجميع العناصر التي تمت مطابقتها باستخدام المحدد الذي تم استدعاؤه قبل التابع, عند استدعاء هذا التابع يتم تنفيذ جميع التوابيع المرتبطة بالحدث `error` للعنصر كما يتم تنفيذ الوظيفة الافتراضية للحدث `error` ويمكن منع تنفيذ الوظيفة الافتراضية للحدث `error` بإعادة القيمة `false` من أحد التوابيع المرتبطة بالحدث `error`.

error(fn)

يُستخدم هذا التابع لربط الحدث `error` لجميع العناصر التي تمت مطابقتها باستخدام المحدد الذي تم استدعاؤه قبل التابع, بشكل عام لا يوجد مقياس عالمي للحدث `error`, ولكن في معظم المتصفحات يتم فتح الحدث `error` عندما يحدث خطأ في JavaScript في الصفحة مثل إسناد مسار صورة غير موجودة للصفة `src` للعنصر `img`.

إن تم فتح الحدث `error` من قبل الكائن `window` عندها سيتم تمرير ثلاثة قيم كوسائط للتابع المرتبط بالحدث `error`:

- رسالة تشرح الخطأ (مثل: المتحول غير معرف, يوجد معامل مفقود في التعبير ...).
- المسار الكامل للصفحة التي حدث فيها الخطأ.
- رقم السطر الذي حدث فيه الخطأ.

focus()

يُستخدم هذا التابع لفتح الحدث `focus` لجميع العناصر التي تمت مطابقتها باستخدام المحدد الذي تم استدعاؤه قبل التابع, عند استدعاء هذا التابع يتم تنفيذ جميع التوابيع المرتبطة بالحدث `focus` للعنصر, وفي المثال التالي يتم إعطاء التركيز للحقل النصي الذي قيمة الواصفة `id` له تساوي `login`:

```
$(document).ready(function(){
    $('#login').focus();
});
```

focus(fn)

يُستخدم هذا التابع لربط الحدث `focus` لجميع العناصر التي تمت مطابقتها باستخدام المحدد الذي تم استدعاؤه قبل التابع, ويتم فتح الحدث `focus` عندما يستقبل العنصر التركيز إما باستخدام التنقل عبر الـ `Tab` أو بنقل التركيز باستخدام مؤشر الفأرة, وفي المثال التالي يتم منع المستخدم من الكتابة في أي حقل نصي وذلك بفتح الحدث `blur` للحقل النصي عندما يصبح التركيز عليه:

```

$("input[@type=text]").focus(function(){
    $(this).blur();
});

```

وفي المثال التالي يتم عرض نص بجانب الحقل النصي الذي يستقبل التركيز:

```

$("input").focus(function () {
    $(this).next("span").css('display','inline').fadeOut(1000);
});

```



أما الشيفرة الكاملة للمثال السابق فهي:

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
    <script src="http://code.jquery.com/jquery-latest.js"></script>

    <script>
    $(document).ready(function(){

        $("input").focus(function () {
            $(this).next("span").css('display','inline').fadeOut(1000);
        });

    });
    </script>
    <style>span {display:none;}</style>
</head>
<body>
    <p><input type="text" /> <span>focus fire</span></p>
    <p><input type="password" /> <span>focus fire</span></p>
</body>
</html>

```

keydown()

يُستخدم هذا التابع لفتح الحدث `keydown` لجميع العناصر التي تمت مطابقتها باستخدام المحدد الذي تم استدعائه قبل التابع، عند استدعاء هذا التابع يتم تنفيذ جميع التوابع المرتبطة بالحدث `keydown` للعنصر، كما يتم تنفيذ الوظيفة الافتراضية للحدث `keydown` ويمكن منع تنفيذ الوظيفة الافتراضية للحدث `keydown`

بإعادة القيمة `false` من أحد التوابع المرتبطة بالحدث `keydown`. ومن المعلوم أن الحدث `keydown` يتم قده عندما يتم الضغط على زر في لوحة المفاتيح.

keydown(fn)

يُستخدم هذا التابع لربط الحدث `keydown` لجميع العناصر التي تمت مطابقتها باستخدام المحدد الذي تم استدعاؤه قبل التابع، ومن المعلوم أن الحدث `keydown` يتم قده عندما يتم الضغط على زر في لوحة المفاتيح. وفي المثال التالي يتم ربط الحدث `keydown` للكائن `window` بتابع:

```
$(window).keydown(function(event){
  switch (event.keyCode) {
    // ...
    // different keys do different things
    // Different browsers provide different codes
    // see here for details: http://unixpapa.com/js/key.html
    // ...
  }
});
```

keypress()

يُستخدم هذا التابع لفتح الحدث `keypress` لجميع العناصر التي تمت مطابقتها باستخدام المحدد الذي تم استدعاؤه قبل التابع، عند استدعاء هذا التابع يتم تنفيذ جميع التوابع المرتبطة بالحدث `keypress` للعنصر، كما يتم تنفيذ الوظيفة الافتراضية للحدث `keypress` ويمكن منع تنفيذ الوظيفة الافتراضية للحدث `keypress` بإعادة القيمة `false` من أحد التوابع المرتبطة بالحدث `keypress`. ومن المعلوم أن الحدث `keypress` يتم قده عندما يتم الضغط على زر في لوحة المفاتيح.

keypress(fn)

يُستخدم هذا التابع لربط الحدث `keypress` لجميع العناصر التي تمت مطابقتها باستخدام المحدد الذي تم استدعاؤه قبل التابع، ومن المعلوم أن الحدث `keypress` يتم قده عندما يتم الانتهاء من الضغط على زر في لوحة المفاتيح، ويعرف الحدث `keypress` على أنه الحدثين `keydown` و `keypress` على نفس المفتاح، وتسلسل هذه الأحداث هو كما يلي:

- `keydown`
- `keypress`
- `keyup`

وفي المثال التالي يتم إظهار الأحرف والفراغات التي تتم طباعتها في الحقل النصي:

```

$("input").keypress(function (e) {
    if (e.which == 32 || (65 <= e.which && e.which <= 65 + 25)
        || (97 <= e.which && e.which <= 97 + 25)) {
        var c = String.fromCharCode(e.which);
        $("p").append($("<span/>"))
            .children(":last")
            .append(document.createTextNode(c));
    } else if (e.which == 8) {
        // backspace in IE only be on keydown
        $("p").children(":last").remove();
    }
    $("div").text(e.which);
});

```

Hello

Add text - Hello

111

أما الشيفرة الكاملة للمثال السابق فهي:

```

<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>

  <script>
    $(document).ready(function() {

      $("input").keypress(function (e) {
        if (e.which == 32 || (65 <= e.which && e.which <= 65 + 25)
            || (97 <= e.which && e.which <= 97 + 25)) {
          var c = String.fromCharCode(e.which);
          $("p").append($("<span/>"))
            .children(":last")
            .append(document.createTextNode(c));
        } else if (e.which == 8) {
          // backspace in IE only be on keydown
          $("p").children(":last").remove();
        }
        $("div").text(e.which);
      });
    });
  </script>
  <style>
  input { margin:10px; }
  p { color:blue; margin:10px; font-size:18px; }
  p.hilite { background:yellow; }
  div { color:red; }
  </style>
</head>
<body>
  <input type="text" />
  <p>Add text - </p>
  <div></div>
</body>
</html>

```

keyup()

يُستخدم هذا التابع لقدر الحدث `keyup` لجميع العناصر التي تمت مطابقتها باستخدام المحدد الذي تم استدعاؤه قبل التابع، عند استدعاء هذا التابع يتم تنفيذ جميع التوابع المرتبطة بالحدث `keyup` للعنصر، كما يتم تنفيذ الوظيفة الافتراضية للحدث `keyup` ويمكن منع تنفيذ الوظيفة الافتراضية للحدث `keyup` بإعادة القيمة `false` من أحد التوابع المرتبطة بالحدث `keyup`. ومن المعلوم أن الحدث `keydown` يتم فدحه عندما يتم تحرير الزر في لوحة المفاتيح بعد الضغط عليه.

keyup(fn)

يُستخدم هذا التابع لربط الحدث keyup لجميع العناصر التي تمت مطابقتها باستخدام المحدد الذي تم استدعاؤه قبل التابع، ومن المعلوم أن الحدث keydown يتم قده عندما يتم الضغط على زر في لوحة المفاتيح. وفي المثال التالي يتم ربط الحدث keyup بتابع يستجيب فقط عند الضغط على زر الهروب escape key:

```
$(document).keyup(function(event){
    if (event.keyCode == 27) {
        alert('escaped!');
    }
});
```

load(fn)

يُستخدم لربط الحدث load لجميع العناصر التي تمت مطابقتها باستخدام المحدد الذي تم استدعاؤه قبل التابع بتابع يتم تنفيذه عند الانتهاء من تحميل العنصر المحدد. وعند استخدام هذا التابع للربط مع الحدث load للكائن window فإن هذا الحدث سيتم قده عند الانتهاء من تحميل كامل محتويات الوثيقة بما فيها الإطارات Frames والصور وعناصر الصفحة بشكل كامل. أما في حال استخدام التابع للربط مع الحدث load لعنصر ما فإن هذا الحدث يتم قده عندما يتم الانتهاء من تحميل كامل محتويات العنصر والعنصر نفسه.

سيعمل الحدث load في حال ربطه مع تابع ما قبل الانتهاء من تحميل العنصر وإلا فإن شيئاً لن يحدث، طبعاً هذه الحالة لن تحدث عند استخدام الصيغة \$(document).ready() لأنها تحل هذه المشكلة بشكل تلقائي. وفي المثال التالي يتم تنفيذ شيفرة ما عند تحميل كامل الصفحة مع كامل الصور وعناصر الصفحة:

```
$(window).load(function () {
    // run code
});
```

أما في المثال التالي يتم إضافة الكلاس bigImg إلى كل صورة ارتفاعها أكبر من 100 بعد الانتهاء من تحميلها مباشرة:

```
$('#img.userIcon').load(function(){
    if($('#this').height() > 100) {
        $('#this').addClass('bigImg');
    }
});
```

mousedown(fn)

يُستخدم لربط الحدث mousedown لجميع العناصر التي تمت مطابقتها باستخدام المحدد الذي تم استدعاؤه قبل التابع، ويتم قذح الحدث mousedown عند الضغط بمؤشر الفأرة على عنصر ما، وفي المثال التالي يتم ربط الحدث mousedown لجميع العناصر p الموجودة في الصفحة:

```
$( "p" ).mouseup(function() {
    $(this).append('<span style="color:#F00;">Mouse up.</span>');
}).mousedown(function() {
    $(this).append('<span style="color:#00F;">Mouse down.</span>');
});
```

Press mouse and release here. **Mouse down**. **Mouse up**. **Mouse down**. **Mouse up**. **Mouse down**. **Mouse up**.

أما الشيفرة الكاملة للمثال السابق فهي:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
    <script src="http://code.jquery.com/jquery-latest.js"></script>

    <script>
    $(document).ready(function() {

        $( "p" ).mouseup(function() {
            $(this).append('<span style="color:#F00;">Mouse up.</span>');
        }).mousedown(function() {
            $(this).append('<span style="color:#00F;">Mouse down.</span>');
        });

    });
</script>

</head>
<body>
    <p>Press mouse and release here.</p>

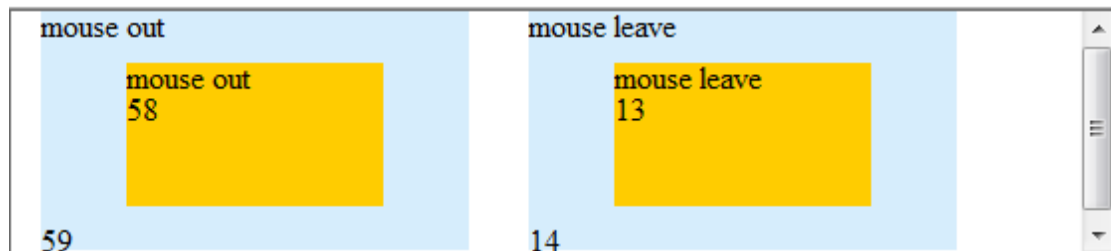
</body>
</html>
```

mouseenter(fn)

يُستخدم لربط الحدث mouseenter لجميع العناصر التي تمت مطابقتها باستخدام المحدد الذي تم استدعاؤه قبل التابع, ويتم قذح الحدث mouseenter عند تحريك مؤشر الفأرة فوق عنصر ما, والمثال التالي يوضح الفرق بين الحدثين mouseenter و mouseover:

```
var i = 0;
$("#div.overout").mouseover(function(){
    $("#p:first",this).text("mouse over");
    $("#p:last",this).text(++i);
}).mouseout(function(){
    $("#p:first",this).text("mouse out");
});

var n = 0;
$("#div.enterleave").mouseenter(function(){
    $("#p:first",this).text("mouse enter");
    $("#p:last",this).text(++n);
}).mouseleave(function(){
    $("#p:first",this).text("mouse leave");
});
```



أما الشيفرة الكاملة للمثال السابق فهي:

```
<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>


  <script>
    $(document).ready(function() {

      var i = 0;
      $("div.overout").mouseover(function() {
        $("p:first",this).text("mouse over");
        $("p:last",this).text(++i);
      }).mouseout(function() {
        $("p:first",this).text("mouse out");
      });

      var n = 0;
      $("div.enterleave").mouseenter(function() {
        $("p:first",this).text("mouse enter");
        $("p:last",this).text(++n);
      }).mouseleave(function() {
        $("p:first",this).text("mouse leave");
      });

    });
  </script>
  <style>
div.out {
width:40%;
height:120px;
margin:0 15px;
background-color:#D6EDFC;
float:left;
}
div.in {
width:60%;
height:60%;

```



```

background-color:#FFCC00;
margin:10px auto;
}
p {
line-height:1em;
margin:0;
padding:0;
}
</style>
</head>
<body>

<div class="out overout"><p>move your mouse</p><div class="in overout"><p>mo
<div class="out enterleave"><p>move your mouse</p><div class="in enterleave":

</body>
</html>

```

mouseleave(fn)

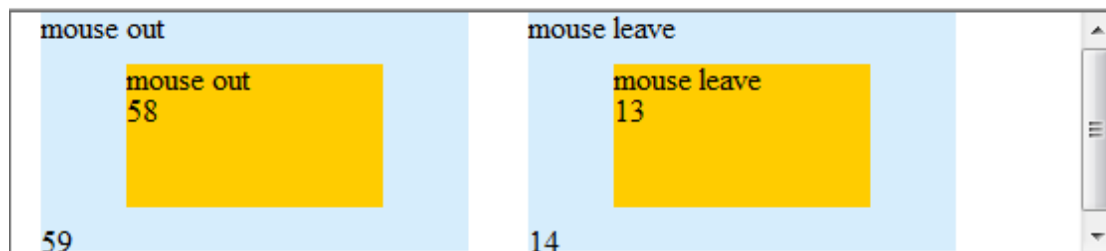
يُستخدم لربط الحدث `mouseleave` لجميع العناصر التي تمت مطابقتها باستخدام المحدد الذي تم استدعاؤه قبل التابع، ويتم قده الحدث `mouseleave` مرة واحدة فقط عند تحريك مؤشر الفأرة خارج العنصر الحالي، والمثال التالي يوضح الفرق بين الحدثين `mouseover` و `mouseenter`:

```

var i = 0;
$("#div.overout").mouseover(function(){
    $("#p:first",this).text("mouse over");
    $("#p:last",this).text(++i);
}).mouseout(function(){
    $("#p:first",this).text("mouse out");
});

var n = 0;
$("#div.enterleave").mouseenter(function(){
    $("#p:first",this).text("mouse enter");
    $("#p:last",this).text(++n);
}).mouseleave(function(){
    $("#p:first",this).text("mouse leave");
});

```



أما الشيفرة الكاملة للمثال السابق فهي:

```
<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>

  <script>
    $(document).ready(function(){

      var i = 0;
      $("div.overout").mouseover(function(){
        $("p:first",this).text("mouse over");
        $("p:last",this).text(++i);
      }).mouseout(function(){
        $("p:first",this).text("mouse out");
      });

      var n = 0;
      $("div.enterleave").mouseenter(function(){
        $("p:first",this).text("mouse enter");
        $("p:last",this).text(++n);
      }).mouseleave(function(){
        $("p:first",this).text("mouse leave");
      });

    });
  </script>
  <style>
div.out {
width:40%;
height:120px;
margin:0 15px;
background-color:#D6EDFC;
float:left;
}
div.in {
width:60%;
height:60%;

```

```

background-color:#FFCC00;
margin:10px auto;
}
p {
line-height:1em;
margin:0;
padding:0;
}
</style>
</head>
<body>

<div class="out overout"><p>move your mouse</p><div class="in overout"><p>mo
<div class="out enterleave"><p>move your mouse</p><div class="in enterleave":

</body>
</html>

```

mousemove(fn)

يُستخدم لربط الحدث `mousemove` لجميع العناصر التي تمت مطابقتها باستخدام المحدد الذي تم استدعاؤه قبل التابع، ويتم قذح الحدث `mousemove` عند تحريك مؤشر الفأرة فوق طالما أنه فوق العنصر، والتابع المرتبط بالحدث له وسيط `e` فيه خاصيتين هما `.clientX` و `.clientY`. يعبران عن إحداثيات مؤشر الفأرة على الصفحة. وفي المثال التالي يتم عرض إحداثيات مؤشر الفأرة عند تحريكه فوق العنصر `div` الأصفر:

```

$("div").mousemove(function(e) {
    var pageCoords = "( " + e.pageX + ", " + e.pageY + " )";
    var clientCoords = "( " + e.clientX + ", " + e.clientY + " )";
    $("#span:first").text(" ( e.pageX, e.pageY ) - " + pageCoords);
    $("#span:last").text(" ( e.clientX, e.clientY ) - " + clientCoords);
});

```

Try scrolling too.

(e.pageX, e.pageY) - (424, 190)

(e.clientX, e.clientY) - (426, 192)



أما الشيفرة الكاملة للمثال السابق فهي:

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>

  <script>
    $(document).ready(function(){

      $("#div").mousemove(function(e){
        var pageCoords = "( " + e.pageX + ", " + e.pageY + " )";
        var clientCoords = "( " + e.clientX + ", " + e.clientY + " )";
        $("#span:first").text("( e.pageX, e.pageY ) - " + pageCoords);
        $("#span:last").text("( e.clientX, e.clientY ) - " + clientCoords);
      });

    });
  </script>
  <style>
  div { width:250px; height:170px; margin:10px; margin-right:50px;
    background:yellow; border:2px groove; float:right; }
  p { margin:0; margin-left:10px; color:red; width:250px;
    height:120px; padding-top:70px;
    float:left; font-size:14px; }
  span { display:block; }
  </style>
</head>
<body>
  <p>
    Try scrolling too.
    <span>Move the mouse over the div.</span>
    <span> </span>
  </p>
  <div></div>
</body>
</html>

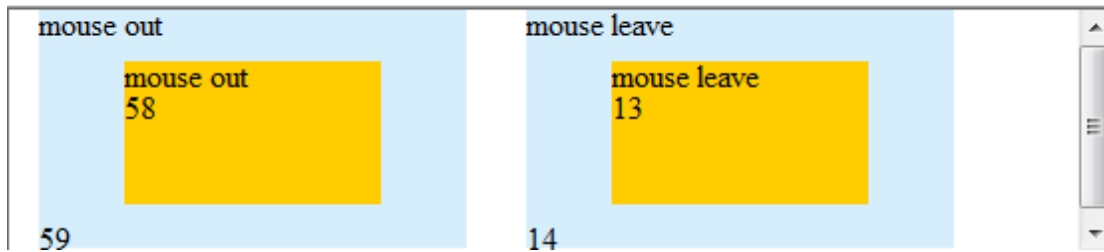
```

mouseout(fn)

يُستخدم لربط الحدث mouseout لجميع العناصر التي تمت مطابقتها باستخدام المحدد الذي تم استدعاؤه قبل التابع، ويتم قذح الحدث mouseout عند تحريك مؤشر الفأرة خارج العنصر الحالي، والمثال التالي يوضح الفرق بين الحدثين mouseenter و mouseover:


```
var i = 0;
$("#div.overout").mouseover(function(){
    $("#p:first",this).text("mouse over");
    $("#p:last",this).text(++i);
}).mouseout(function(){
    $("#p:first",this).text("mouse out");
});

var n = 0;
$("#div.enterleave").mouseenter(function(){
    $("#p:first",this).text("mouse enter");
    $("#p:last",this).text(++n);
}).mouseleave(function(){
    $("#p:first",this).text("mouse leave");
});
```



أما الشيفرة الكاملة للمثال السابق فهي:

```
<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>


  <script>
    $(document).ready(function() {

      var i = 0;
      $("div.overout").mouseover(function() {
        $("p:first",this).text("mouse over");
        $("p:last",this).text(++i);
      }).mouseout(function() {
        $("p:first",this).text("mouse out");
      });

      var n = 0;
      $("div.enterleave").mouseenter(function() {
        $("p:first",this).text("mouse enter");
        $("p:last",this).text(++n);
      }).mouseleave(function() {
        $("p:first",this).text("mouse leave");
      });

    });
  </script>
  <style>
div.out {
width:40%;
height:120px;
margin:0 15px;
background-color:#D6EDFC;
float:left;
}
div.in {
width:60%;
height:60%;

```



```

background-color:#FFCC00;
margin:10px auto;
}
p {
line-height:1em;
margin:0;
padding:0;
}
</style>
</head>
<body>

<div class="out overout"><p>move your mouse</p><div class="in overout"><p>mo
<div class="out enterleave"><p>move your mouse</p><div class="in enterleave":

</body>
</html>

```

mouseover(fn)

يُستخدم لربط الحدث `mouseover` لجميع العناصر التي تمت مطابقتها باستخدام المحدد الذي تم استدعاؤه قبل التابع، ويتم قذح الحدث `mouseover` عند تحريك مؤشر الفأرة داخل العنصر الحالي.

mouseup(fn)

يُستخدم لربط الحدث `mouseup` لجميع العناصر التي تمت مطابقتها باستخدام المحدد الذي تم استدعاؤه قبل التابع، ويتم قذح الحدث `mouseup` عند تحرير مؤشر الفأرة بعد الضغط على العنصر الحالي:

```

$("p").mouseup(function(){
    $(this).append('<span style="color:#F00;">Mouse up.</span>');
}).mousedown(function(){
    $(this).append('<span style="color:#00F;">Mouse down.</span>');
});

```

Press mouse and release here.

أما الشيفرة الكاملة للمثال السابق فهي:

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>

  <script>
    $(document).ready(function() {

      $("p").mouseup(function() {
        $(this).append('<span style="color:#F00;">Mouse up.</span>');
      }).mousedown(function() {
        $(this).append('<span style="color:#00F;">Mouse down.</span>');
      });

    });
  </script>

</head>
<body>
  <p>Press mouse and release here.</p>

</body>
</html>

```

resize(fn)

يُستخدم لربط الحدث `resize` لجميع العناصر التي تمت مطابقتها باستخدام المحدد الذي تم استدعاؤه قبل التابع، ويتم قذح الحدث `resize` عند تغيير أبعاد العنصر:

```

$(window).resize(function() {
  alert("Stop it!");
});

```

scroll(fn)

يُستخدم لربط الحدث `scroll` لجميع العناصر التي تمت مطابقتها باستخدام المحدد الذي تم استدعاؤه قبل التابع، ويتم قذح الحدث `scroll` عند تغيير حالة شريط التمرير، وفي المثال التالي يتم الاستجابة لتغيير مكان شريط التمرير في الصفحة:

```

$( "p" ).clone().appendTo(document.body);
$( "p" ).clone().appendTo(document.body);
$( "p" ).clone().appendTo(document.body);
$(window).scroll(function () {
    $( "span" ).css("display", "inline").fadeOut("slow");
});

```

Try scrolling the iframe.

Paragraph -

Paragraph -

أما الشيفرة الكاملة للمثال السابق فهي:

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
    <script src="http://code.jquery.com/jquery-latest.js"></script>

    <script>
        $(document).ready(function(){

            $( "p" ).clone().appendTo(document.body);
            $( "p" ).clone().appendTo(document.body);
            $( "p" ).clone().appendTo(document.body);
            $(window).scroll(function () {
                $( "span" ).css("display", "inline").fadeOut("slow");
            });

        });
    </script>
    <style>
        div { color:blue; }
        p { color:green; }
        span { color:red; display:none; }
    </style>
</head>
<body>
    <div>Try scrolling the iframe.</div>
    <p>Paragraph - <span>Scroll happened!</span></p>
</body>
</html>

```

select()

يُستخدم هذا التابع لفتح الحدث select لجميع العناصر التي تمت مطابقتها باستخدام المحدد الذي تم استدعاؤه قبل التابع, عند استدعاء هذا التابع يتم تنفيذ جميع التوابع المرتبطة بالحدث select للعنصر كما يتم تنفيذ الوظيفة الافتراضية للحدث select ويمكن منع تنفيذ الوظيفة الافتراضية للحدث select بإعادة القيمة false من أحد التوابع المرتبطة بالحدث select, وفي المثال التالي يتم فتح الحدث select لجميع العناصر input في الصفحة:

```
$("#input").select();
```

select(fn)

يُستخدم لربط الحدث select لجميع العناصر التي تمت مطابقتها باستخدام المحدد الذي تم استدعاؤه قبل التابع, ويتم فتح الحدث select عند تحديد نص في الحقل النصي أو حتى في الحقل النصي المتعدد الأسطر:

```
$("#:input").select( function () {  
    $("#div").text("Something was selected").show().fadeOut(1000);  
});
```

Click and drag the mouse to select text in the inputs.

Something was selected

أما الشيفرة الكاملة للمثال السابق فهي:

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>

  <script>
    $(document).ready(function(){

      $(".input").select( function () {
        $(".div").text("Something was selected").show().fadeOut(1000);
      });

    });
  </script>
  <style>
    p { color:blue; }
    div { color:red; }
  </style>
</head>
<body>
  <p>
    Click and drag the mouse to select text in the inputs.
  </p>
  <input type="text" value="Some text" />
  <input type="text" value="to test on" />
  <div></div>
</body>
</html>

```

submit()

يُستخدم هذا التابع لفتح الحدث submit لجميع العناصر التي تمت مطابقتها باستخدام المحدد الذي تم استدعاؤه قبل التابع, عند استدعاء هذا التابع يتم تنفيذ جميع التوابع المرتبطة بالحدث submit للعنصر كما يتم تنفيذ الوظيفة الافتراضية للحدث submit ويمكن منع تنفيذ الوظيفة الافتراضية للحدث submit بإعادة القيمة false من أحد التوابع المرتبطة بالحدث submit, وفي المثال التالي يتم قدها الحدث submit لأول نموذج form موجود في الصفحة:

```
$(".form:first").submit();
```

submit(fn)

يُستخدم لربط الحدث submit لجميع العناصر التي تمت مطابقتها باستخدام المحدد الذي تم استدعاؤه قبل التابع, ويتم قدها الحدث submit عندما يتم إرسال معلومات النموذج form إلى الخادم, وفي المثال التالي يتم منع الحدث submit من التنفيذ ما لم يتم كتابة الكلمة correct في الحقل النصي:

```
$("#form").submit(function() {  
  if ($("#input:first").val() == "correct") {  
    $("#span").text("Validated...").show();  
    return true;  
  }  
  $("#span").text("Not valid!").show().fadeOut(1000);  
  return false;  
});
```

Type 'correct' to validate.

أما الشيفرة الكاملة للمثال السابق فهي:


```

<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>

  <script>
    $(document).ready(function(){

      $("form").submit(function() {
        if ($("#input:first").val() == "correct") {
          $("#span").text("Validated...").show();
          return true;
        }
        $("#span").text("Not valid!").show().fadeOut(1000);
        return false;
      });

    });
  </script>
  <style>
  p { margin:0; color:blue; }
  div,p { margin-left:10px; }
  span { color:red; }
  </style>
</head>
<body>
  <p>Type 'correct' to validate.</p>
  <form action="javascript:alert('success!');">
    <div>
      <input type="text" />
      <input type="submit" />
    </div>
  </form>
  <span></span>
</body>
</html>

```

unload(fn)

يُستخدم لربط الحدث unload لجميع العناصر التي تمت مطابقتها باستخدام المحدد الذي تم استدعاؤه

قبل التابع.

توابع التأثيرات الحركية والشكلية Effects Functions

التوابع الأساسية Basics Functions

show()

يُستخدم لإظهار جميع العناصر التي تمت مطابقتها باستخدام المحدد الذي تم استدعاؤه قبل التابع أو بعضها في حال أنها مخفية أي غير ظاهرة في الصفحة, ولا يهم كيف تم إخفاء العنصر سواء باستخدام التابع (`hide()`) أو باستخدام `display:none` في الـ CSS, وفي المثال التالي يتم إظهار جميع العناصر `p` المخفية في الصفحة:

```
$("#p").show()
```

Hello 1

أما الشيفرة الكاملة للمثال السابق فهي:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>

  <script>
    $(document).ready(function(){
      $("#p").show()
    });
  </script>
</head>
<body>
  <p style="display:none">Hello 1</p>
</body>
```

show(speed, [callback])

يُستخدم لإظهار جميع العناصر التي تمت مطابقتها باستخدام المحدد الذي تم استدعاؤه قبل التابع بشكل تدريجي وبعدها ينفذ تابع اختياري عند انتهاء عملية إظهار العنصر يتم تمريره كوسيط `callback`, ويتم تغيير عرض وارتفاع وشفافية العنصر بشكل ديناميكي بناء على قيمة الوسيط `speed`, والوسيط `speed` يقبل إحدى القيم التالية:

- `slow`, `normal`, `fast` كقيم نصية.
- أو رقم يعبر بالميللي ثانية يعبر عن سرعة إظهار العنصر.

وفي المثال التالي يتم إظهار جميع العناصر `p` المخفية بشكل بطيء بعد 600 ميلي ثانية:

```
$("#button").click(function () {
    $("#p").show("slow");
});
```

Show it

أما الشيفرة الكاملة للمثال السابق فهي:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
    <script src="http://code.jquery.com/jquery-latest.js"></script>

    <script>
    $(document).ready(function(){

        $("#button").click(function () {
            $("#p").show("slow");
        });

    });
    </script>
    <style>
    p { background:yellow; }
    </style>
</head>
<body>
    <button>Show it</button>
    <p style="display: none">Hello 2</p>
</body>
</html>
```

hide()

يُستخدم لإخفاء جميع العناصر التي تمت مطابقتها باستخدام المحدد الذي تم استدعاؤه قبل التابع أو بعضها في حال أنها ظاهرة في الصفحة، وفي المثال التالي يتم إخفاء جميع العناصر p الموجودة في الصفحة بعد تحميل الصفحة ويمكن إخفاء أي عنصر رابط a عند الضغط عليه:

```
$("#p").hide();
$("#a").click(function () {
    $(this).hide();
    return true;
});
```

[Click to hide me too](#)

أما الشيفرة الكاملة للمثال السابق فهي:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
    <script src="http://code.jquery.com/jquery-latest.js"></script>

    <script>
    $(document).ready(function() {

        $("#p").hide();
        $("#a").click(function () {
            $(this).hide();
            return true;
        });

    });
    </script>
</head>
<body>
    <p>Hello</p>

    <p>hide me too</a>
    <p>Here is another paragraph</p>
</body>
</html>
```

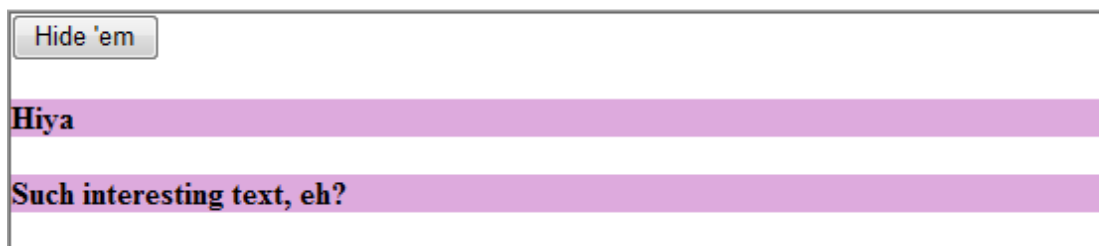
hide(speed, [callback])

يُستخدم لإخفاء جميع العناصر التي تمت مطابقتها باستخدام المحدد الذي تم استدعاؤه قبل التابع بشكل تدريجي وبعدها ينفذ تابع اختياري عند انتهاء عملية إخفاء العنصر يتم تمريره كوسيط callback, ويتم تغيير عرض وارتفاع وشفافية العنصر بشكل ديناميكي بناء على قيمة الوسيط speed, والوسيط speed يقبل إحدى القيم التالية:

- slow, normal, fast كقيم نصية.
- أو رقم يعبر بالملي ثانية يعبر عن سرعة إظهار العنصر.

وفي المثال التالي يتم إخفاء جميع العناصر p بشكل تدريجي في الصفحة بمجرد الضغط على الزر خلال 600 ميلي ثانية:

```
$("#button").click(function () {
    $("#p").hide("slow");
});
```



أما الشيفرة الكاملة للمثال السابق فهي:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>

  <script>
    $(document).ready(function(){

      $("#button").click(function () {
        $("#p").hide("slow");
      });

    });
  </script>
  <style>
    p { background:#dad; font-weight:bold; }
  </style>
</head>
<body>
  <button>Hide 'em</button>
  <p>Hiya</p>
  <p>Such interesting text, eh?</p>
</body>
</html>
```

toggle()

يُستخدم لإظهار أو إخفاء جميع العناصر التي تمت مطابقتها باستخدام المحدد الذي تم استدعاؤه قبل التابع بناء على حالة العنصر الحالية فإن كان ظاهراً عندها يتم إخفاؤه وإن كان مخفياً سيتم إظهاره:

```

$("button").click(function () {
    $("p").toggle();
});

```



أما الشيفرة الكاملة للمثال السابق فهي:

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
    <script src="http://code.jquery.com/jquery-latest.js"></script>

    <script>
    $(document).ready(function(){

        $("button").click(function () {
            $("p").toggle();
        });

    });
    </script>

</head>
<body>
    <button>Toggle</button>
    <p></p>
    <p style="display: none">Good Bye</p>
</body>
</html>

```

toggle(switch)

يُستخدم لإظهار أو إخفاء جميع العناصر التي تمت مطابقتها باستخدام المحدد الذي تم استدعاؤه قبل التابع بناء على قيمة الوسيط switch حيث يتم إظهار جميع العناصر عندما تكون قيمة switch تساوي false ويتم إخفاء جميع العناصر في حالة كانت قيمة switch تساوي true:

```
var flip = 0;
$("#button").click(function () {
    $("#p").toggle( flip++ % 2 == 0 );
});
```

Toggle

Hello

أما الشيفرة الكاملة للمثال السابق فهي:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
    <script src="http://code.jquery.com/jquery-latest.js"></script>

    <script>
        $(document).ready(function(){

            var flip = 0;
            $("#button").click(function () {
                $("#p").toggle( flip++ % 2 == 0 );
            });

        });
    </script>

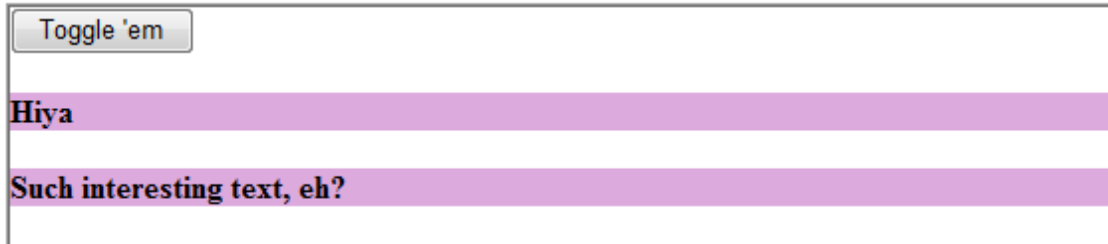
</head>
<body>
    <button>Toggle</button>
    <p>Hello</p>
    <p style="display: none">Good Bye</p>
</body>
</html>
```

toggle(speed, [callback])

يُستخدم للإظهار أو إخفاء جميع العناصر التي تمت مطابقتها باستخدام المحدد الذي تم استدعاؤه قبل التابع بناء على حالة العنصر الحالية فإن كان ظاهراً عندها يتم إخفاؤه وإن كان مخفياً سيتم إظهاره, وتتم العملية بشكل تدريجي بناء على قيمة speed وبعدها ينفذ تابع اختياري عند انتهاء عملية إخفاء العنصر يتم تمريره كوسيط callback, ويتم تغيير عرض وارتفاع وشفافية العنصر بشكل ديناميكي بناء على قيمة الوسيط speed, والوسيط speed يقبل إحدى القيم التالية:

- slow, normal, fast كقيم نصية.
- أو رقم يعبر بالميللي ثانية يعبر عن سرعة إظهار العنصر.

```
$("#button").click(function () {
    $("#p").toggle("slow");
});
```



أما الشيفرة الكاملة للمثال السابق فهي:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>

  <script>
    $(document).ready(function(){

      $("#button").click(function () {
        $("#p").toggle("slow");
      });

    });
  </script>
  <style>
    p { background:#dad;
    font-weight:bold;
    font-size:16px; }
  </style>
</head>
<body>
  <button>Toggle 'em</button>
  <p>Hiya</p>
  <p>Such interesting text, eh?</p>
</body>
</html>
```


توابع الانزلاق Sliding Functions

slideDown(speed, [callback])

يُستخدم هذا التابع لإظهار جميع العناصر التي تمت مطابقتها باستخدام المحدد الذي تم استدعاؤه قبل التابع بتغيير قيمة خاصية الارتفاع height للعنصر بشكل تدريجي بناء على قيمة الوسيط speed وبعدها ينفذ تابع اختياري عند انتهاء عملية إخفاء العنصر يتم تمريره كوسيط callback, ويتم تغيير ارتفاع العنصر بشكل ديناميكي بناء على قيمة الوسيط speed, والوسيط speed يقبل إحدى القيم التالية:

- slow, normal, fast كقيم نصية.
- أو رقم يعبر بالملي ثانية يعبر عن سرعة إظهار العنصر.

وفي المثال التالي سيتم إظهار جميع العناصر div بشكل تدريجي عند النقر للمرة الأولى على الكلمة Click me!, وإخفاؤها جميعها عند الضغط مرة ثانية على الكلمة Click me!:

```
$(document.body).click(function () {  
  if ($("#div:first").is(":hidden")) {  
    $("#div").slideDown("slow");  
  } else {  
    $("#div").hide();  
  }  
});
```

Click me!



أما الشيفرة الكاملة للمثال السابق فهي:

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>

  <script>
    $(document).ready(function() {

      $(document.body).click(function () {
        if ($("#div:first").is(":hidden")) {
          $("#div").slideDown("slow");
        } else {
          $("#div").hide();
        }
      });
    });
  </script>
  <style>
  div { background:#de9a44; margin:3px; width:80px;
        height:40px; display:none; float:left; }
  </style>
</head>
<body>
  Click me!
  <div></div>
  <div></div>
  <div></div>
</body>
</html>

```

slideUp(speed, [callback])

يُستخدم هذا التابع لإخفاء جميع العناصر التي تمت مطابقتها باستخدام المحدد الذي تم استدعاؤه قبل التابع بتغيير قيمة خاصية الارتفاع height للعنصر بشكل تدريجي بناء على قيمة الوسيط speed وبعدها ينفذ تابع اختياري عند انتهاء عملية إخفاء العنصر يتم تمريره كوسيط callback, ويتم تغيير ارتفاع العنصر بشكل ديناميكي بناء على قيمة الوسيط speed, والوسيط speed يقبل إحدى القيم التالية:

- slow, normal, fast كقيم نصية.
- أو رقم يعبر بالملي ثانية يعبر عن سرعة إظهار العنصر.

وفي المثال التالي يتم إخفاء جميع العناصر div عند الضغط بمؤشر الفأرة على Click me! ويتم إظهارها مرة أخرى عند النقر على Click me! مرة أخرى:

```

$(document.body).click(function () {
  if ($("#div:first").is(":hidden")) {
    $("#div").show("slow");
  } else {
    $("#div").slideUp();
  }
});

```

Click me!



أما الشيفرة الكاملة للمثال السابق فهي:

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>

  <script>
    $(document).ready(function(){

      $(document.body).click(function () {
        if ($("#div:first").is(":hidden")) {
          $("#div").show("slow");
        } else {
          $("#div").slideUp();
        }
      });

    });
  </script>
  <style>
    div { background:#3d9a44; margin:3px; width:80px;
          height:40px; float:left; }
  </style>
</head>
<body>
  Click me!
  <div></div>
  <div></div>
  <div></div>
  <div></div>
  <div></div>
</body>
</html>

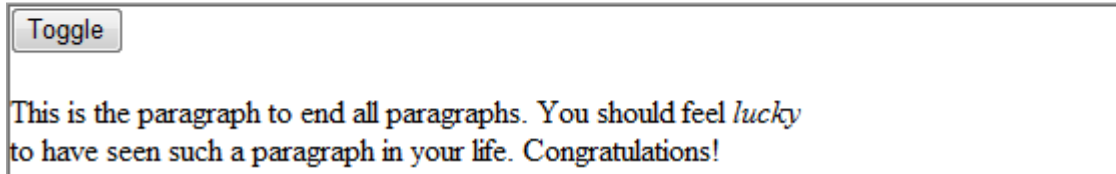
```

slideToggle(speed, [callback])

يُستخدم لإظهار أو إخفاء جميع العناصر التي تمت مطابقتها باستخدام المحدد الذي تم استدعاؤه قبل التابع بناء على حالة العنصر الحالية فإن كان ظاهراً عندها يتم إخفاؤه وإن كان مخفياً سيتم إظهاره, وتتم العملية بشكل تدريجي بناء على قيمة speed وبعدها ينفذ تابع اختياري عند انتهاء عملية إخفاء العنصر يتم تمريره كوسيط callback, ويتم تغيير عرض وارتفاع وشفافية العنصر بشكل ديناميكي بناء على قيمة الوسيط speed, والوسيط speed يقبل إحدى القيم التالية:

- slow, normal, fast كقيم نصية.
- أو رقم يعبر بالميللي ثانية يعبر عن سرعة إظهار العنصر.

```
$("#button").click(function () {  
    $("#p").slideToggle("slow");  
});
```



أما الشيفرة الكاملة للمثال السابق فهي:

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>

  <script>
    $(document).ready(function() {

      $("button").click(function () {
        $("p").slideToggle("slow");
      });

    });
  </script>
  <style>
  p { width:400px; }
  </style>
</head>
<body>
  <button>Toggle</button>
  <p>
    This is the paragraph to end all paragraphs. You
    should feel <em>lucky</em> to have seen such a paragraph in
    your life. Congratulations!
  </p>
</body>
</html>

```

توابع التلاشي Fading Functions

fadeIn(speed, [callback])

يُستخدم هذا التابع لإظهار جميع العناصر التي تمت مطابقتها باستخدام المحدد الذي تم استدعاؤه قبل التابع بتغيير قيمة خاصية الشفافية opacity للعنصر بشكل تدريجي بناء على قيمة الوسيط speed وبعدها ينفذ تابع اختياري عند انتهاء عملية إظهار العنصر يتم تمريره كوسيط callback, ويتم تغيير شفافية العنصر بشكل ديناميكي بناء على قيمة الوسيط speed, والوسيط speed يقبل إحدى القيم التالية:

- slow, normal, fast كقيم نصية.
- أو رقم يعبر بالملي ثانية يعبر عن سرعة إظهار العنصر.

وفي المثال التالي يتم إظهار أول عنصر div مخفي عند النقر على [Click here](#):

```
$(document.body).click(function () {
    $("div:hidden:first").fadeIn("slow");
});
```

Click here...



أما الشيفرة الكاملة للمثال السابق فهي:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
    <script src="http://code.jquery.com/jquery-latest.js"></script>

    <script>
    $(document).ready(function() {

        $(document.body).click(function () {
            $("div:hidden:first").fadeIn("slow");
        });

    });
    </script>
<style>
span { color:red; cursor:pointer; }
div { margin:3px; width:80px; display:none;
    height:80px; float:left; }
div#one { background:#f00; }
div#two { background:#0f0; }
div#three { background:#00f; }
</style>
</head>
<body>
    <span>Click here...</span>
    <div id="one"></div>
    <div id="two"></div>
    <div id="three"></div>
</body>
</html>
```

fadeOut(speed, [callback])

يُستخدم هذا التابع لإخفاء جميع العناصر التي تمت مطابقتها باستخدام المحدد الذي تم استدعاؤه قبل التابع بتغيير قيمة خاصية الشفافية opacity للعنصر بشكل تدريجي بناء على قيمة الوسيط speed وبعدها ينفذ تابع

اختياري عند انتهاء عملية إخفاء العنصر يتم تمريره كوسيط callback, ويتم تغيير شفافية العنصر بشكل ديناميكي بناء على قيمة الوسيط speed, والوسيط speed يقبل إحدى القيم التالية:

- slow, normal, fast كقيم نصية.
- أو رقم يعبر بالميللي ثانية يعبر عن سرعة إظهار العنصر.

وفي المثال التالي يتم إخفاء العنصر span عند الضغط عليه لمؤشر الفأرة:

```
$("#span").click(function () {
    $(this).fadeOut(1000, function () {
        $("#div").text("'" + $(this).text() + "' has faded!");
        $(this).remove();
    });
});
$("#span").hover(function () {
    $(this).addClass("hilite");
}, function () {
    $(this).removeClass("hilite");
});
```

Find the modifiers -

If you really want to go outside in the cold then make sure to wear your warm jacket given to you by your favorite teacher.

أما الشيفرة الكاملة للمثال السابق فهي:

```

<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>

  <script>
    $(document).ready(function() {

      $("span").click(function () {
        $(this).fadeOut(1000, function () {
          $("div").text("'" + $(this).text() + "' has faded!");
          $(this).remove();
        });
      });

      $("span").hover(function () {
        $(this).addClass("hilite");
      }, function () {
        $(this).removeClass("hilite");
      });

    });
  </script>
<style>
span { cursor:pointer; }
span.hilite { background:yellow; }
div { display:inline; color:red; }
</style>
</head>
<body>
  <h3>Find the modifiers - <div></div></h3>
  <p>
    If you <span>really</span> want to go outside
    <span>in the cold</span> then make sure to wear
    your <span>warm</span> jacket given to you by
    your <span>favorite</span> teacher.
  </p>
</body>
</html>

```

أما المثال التالي يتم إخفاء العنصر p عند الضغط عليه:

```

$("p").click(function () {
  $("p").fadeOut("slow");
});

```

If you click on this paragraph you'll see it just fade away.

أما الشيفرة الكاملة للمثال السابق فهي:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>

  <script>
    $(document).ready(function() {

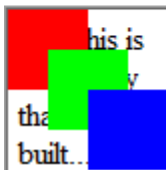
      $("p").click(function () {
        $("p").fadeOut("slow");
      });

    });
  </script>
  <style>
    p { font-size:150%; cursor:pointer; }
  </style>
</head>
<body>
  <p>
    If you click on this paragraph
    you'll see it just fade away.
  </p>
</body>
</html>
```

fadeTo(speed, opacity, [callback])

يُستخدم هذا التابع لتغيير قيمة الشفافية opacity لجميع العناصر التي تمت مطابقتها باستخدام المحدد الذي تم استدعاؤه قبل التابع, وفي المثال التالي يتم تغيير قيمة الخاصية opacity لأي عنصر div موجود في الصفحة يتم الضغط عليه بمؤشر الفأرة إلى قيمة عشوائية:

```
$("#div").click(function () {
  $(this).fadeTo("fast", Math.random());
});
```



أما الشيفرة الكاملة للمثال السابق فهي:

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>

  <script>
    $(document).ready(function(){

      $("#div").click(function () {
        $(this).fadeTo("fast", Math.random());
      });

    });
  </script>
  <style>
    p { width:80px; margin:0; padding:5px; }
    div { width:40px; height:40px; position:absolute; }
    div#one { top:0; left:0; background:#f00; }
    div#two { top:20px; left:20px; background:#0f0; }
    div#three { top:40px; left:40px; background:#00f; }
  </style>
</head>
<body>
  <p>And this is the library that John built...</p>
  <div id="one"></div>
  <div id="two"></div>
  <div id="three"></div>
</body>
</html>

```

أما في المثال التالي فيتم أيضاً تغيير قيمة الخاصية `opacity` عند الضغط على أي عنصر `div` موجود في الصفحة إلى القيمة `0.25`:

```

var getPos = function (n) {
    return (Math.floor(n) * 90) + "px";
};
$("p").each(function (n) {
    var r = Math.floor(Math.random() * 3);
    var tmp = $(this).text();
    $(this).text($("#p:eq(" + r + ")").text());
    $("#p:eq(" + r + ")").text(tmp);
    $(this).css("left", getPos(n));
});
$("div").each(function (n) {
    $(this).css("left", getPos(n));
})
.css("cursor", "pointer")
.click(function () {
    $(this).fadeTo(250, 0.25, function () {
        $(this).css("cursor", "")
        .prev().css({"font-weight": "bolder",
            "font-style": "italic"});
    });
});

```

Right!

أما الشيفرة الكاملة للمثال السابق فهي:

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>

  <script>
    $(document).ready(function() {

      var getPos = function (n) {
        return (Math.floor(n) * 90) + "px";
      };
      $("p").each(function (n) {
        var r = Math.floor(Math.random() * 3);
        var tmp = $(this).text();
        $(this).text($("#p:eq(" + r + ")").text());
        $("#p:eq(" + r + ")").text(tmp);
        $(this).css("left", getPos(n));
      });
      $("div").each(function (n) {
        $(this).css("left", getPos(n));
      })
      .css("cursor", "pointer")
      .click(function () {
        $(this).fadeTo(250, 0.25, function () {
          $(this).css("cursor", "")
            .prev().css({"font-weight": "bolder",
              "font-style": "italic"});
        });
      });
    });
  </script>
<style>
div, p { width:80px; height:40px; top:0; margin:0;
  position:absolute; padding-top:8px; }
p { background:#fcc; text-align:center; }

```

```

div { background:blue; }
</style>
</head>
<body>
  <p>Wrong</p>
  <div></div>
  <p>Wrong</p>
  <div></div>
  <p>Right!</p>
  <div></div>
</body>
</html>

```

توابع الحركة Animate Functions

animate(params, [duration], [easing], [callback])

يُستخدم هذا التابع لتنفيذ بعض التأثيرات الحركية على العناصر الموجودة في الصفحة، والمفهوم الرئيسي لعمل هذا التابع يتمحور حول قيم الوسيط params الذي هو عبارة عن مجموعة من المفاتيح والقيم حيث تعبر المفاتيح عن خصائص CSS أما القيم فهي القيم الجديدة لكل خاصية مثل height, top, opacity ... ملاحظة أن هذه الخصائص يجب كتابتها بالشكل camel case أي مثلاً يجب كتابة marginLeft بدلاً عن margin-left أي هي نفس الكلمة ولكن تم حذف المحرف "-" وأول حرف من الكلمة الثانية يكون كبيراً.

والقيم المرتبطة مع المفاتيح تعبر عن القيم التي ستنتهي إليها قيمة كل خاصية، فعندما يتم تمرير قيمة رقمية كقيمة جديدة للخاصية عندها فإن قيمة هذه الخاصية ستدرج من القيمة الحالية إلى القيمة الجديدة، أما إن تم تحديد قيم نصية مثل hid, show عندها سيتم تنفيذ الوظيفة الافتراضية لهذه القيم. أيضاً يمكن تغيير قيم الخصائص باستخدام وحدات قياس مختلفة %، em، في حال أنها مدعومة من قبل الخاصية نفسها، كما يمكن استخدام += أو -= قبل قيمة الخاصية لكي يتم تغيير قيمة الخاصية بشكل متزايد أم متناقص.

أما بالنسبة للوسيط duration ففي حال أن قيمته مساوية للقيمة 0 عندها يتم تغيير جميع قيم الخصائص المحددة بالوسيط params بدون أية تأخير.

والوسيط duration يقبل إحدى القيم التالية، التي تعبر عن سرعة تنفيذ التغيير:

- slow, normal, fast كقيم نصية.
- أو رقم يعبر بالملي ثانية يعبر عن سرعة إظهار العنصر.

أما بالنسبة للوسيط easing فيقبل القيمتين linear أو swing، ومن أجل استخدامه يجب استخدام plugin، أما الوسيط callback الاختياري فهو تابع يتم تنفيذه بعد الانتهاء من تنفيذ التابع animate.

في المثال التالي يتم تحريك العنصر div الموجود في الصفحة إلى اليمين أو إلى اليسار حسب الزر الذي يتم الضغط عليه، ولاحظ استخدام الرمز += و -= قبل القيم الجديدة:

```
$("#right").click(function(){
    $(".block").animate({"left": "+=50px"}, "slow");
});

$("#left").click(function(){
    $(".block").animate({"left": "-=50px"}, "slow");
});
```



أما الشيفرة الكاملة للمثال السابق فهي:

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>

  <script>
    $(document).ready(function() {

      $("#right").click(function() {
        $(".block").animate({"left": "+=50px"}, "slow");
      });

      $("#left").click(function() {
        $(".block").animate({"left": "-=50px"}, "slow");
      });

    });
  </script>
  <style>
  div {
    position: absolute;
    background-color: #abc;
    left: 50px;
    width: 90px;
    height: 90px;
    margin: 5px;
  }
  </style>
</head>
<body>
  <button id="left"><</button> <button id="right">>>/button>
  <div class="block"></div>

</body>
</html>

```

أما المثال التالي فيتم تنفيذ تأثير حركي على العنصر div بتغيير بعض خصائصه عند الضغط على الزر Run:

```
// Using multiple unit types within one animation.
$("#go").click(function(){
  $("#block").animate({
    width: "70%",
    opacity: 0.4,
    marginLeft: "0.6in",
    fontSize: "3em",
    borderWidth: "10px"
  }, 1500 );
});
```

» Run

Hello!

أما الشيفرة الكاملة للمثال السابق فهي:


```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">

<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>

  <script>
    $(document).ready(function() {

      // Using multiple unit types within one animation.
      $("#go").click(function() {
        $("#block").animate({
          width: "70%",
          opacity: 0.4,
          marginLeft: "0.6in",
          fontSize: "3em",
          borderWidth: "10px"
        }, 1500 );
      });

    });
  </script>
  <style>
  div {
    background-color:#bca;
    width:100px;
    border:1px solid green;
  }
  </style>
</head>
<body>
  <button id="go">» Run</button>
  <div id="block">Hello!</div>
</body>
</html>

```

animate(params, options)

يُستخدم هذا التابع لتنفيذ بعض التأثيرات الحركية على العناصر الموجودة في الصفحة، والمفهوم الرئيسي لعمل هذا التابع يتمحور حول قيم الوسيط params الذي هو عبارة عن مجموعة من المفاتيح والقيم حيث تعبر المفاتيح عن خصائص CSS أما القيم فهي القيم الجديدة لكل خاصية مثل height, top, opacity ... ملاحظة أن هذه الخصائص يجب كتابتها بالشكل camel case أي مثلاً يجب كتابة marginLeft بدلاً عن margin-left أي هي نفس الكلمة ولكن تم حذف الحرف "-" وأول حرف من الكلمة الثانية يكون كبيراً.

والقيم المرتبطة مع المفاتيح تعبر عن القيم التي ستنتهي إليها قيمة كل خاصية، فعندما يتم تمرير قيمة رقمية كقيمة جديدة للخاصية عندها فإن قيمة هذه الخاصية ستندرج من القيمة الحالية إلى القيمة الجديدة، أما إن تم تحديد قيم

نصية مثل `hid, show` عندها سيتم تنفيذ الوظيفة الافتراضية لهذه القيم. أيضاً يمكن تغيير قيم الخصائص باستخدام وحدات قياس مختلفة `%`, `em`, في حال أهما مدعومة من قبل الخاصية نفسها, كما يمكن استخدام `+=` أو `--` قبل قيمة الخاصية لكي يتم تغيير قيمة الخاصية بشكل متزايد أم متناقص.

أما الوسيط `options` فله القيم التالية:

- الوسيط `duration` يقبل إحدى القيم التالية, التي تعبر عن سرعة تنفيذ التغيير:
 - `slow, normal, fast` كقيم نصية.
 - أو رقم يعبر بالملي ثانية يعبر عن سرعة إظهار العنصر.
- الوسيط `easing` يقبل القيمتين `linear` أو `swing`, ومن أجل استخدامه يجب استخدام `plugin`.
- `complete`, تابع يتم تنفيذه عند الانتهاء من تنفيذ التابع `animate`.
- `queue`, يقبل هذا الوسيط قيمتين إما `true` أو `false` وفي حال تحديد القيمة `false` عندها سيتم تنفيذ التابع `animate` مباشرة بدون انتظار دوره.

في المثال التالي يتم تغيير قيمة الخاصيتين `height` و `opacity` لكل العناصر `p` في الصفحة باستخدام قيمة `slow` للوسيط `duration`:

```
$("#p").animate({
  "height": "toggle", "opacity": "toggle"
}, { duration: "slow" });
```

أما المثال التالي فيتم تحريك جميع العناصر `p` الموجودة في الصفحة إلى الياسر بمقدار 50 بيكسل ويتم تنفيذ التابع خلال 500 ميلي ثانية وقيمة الوسيط `queue` هي `false` لتنفيذ التابع مباشرة:

```
$("#p").animate({
  left: "50px", opacity: 1
}, { duration: 500, queue: false });
```

أما المثال التالي فيظهر كيفية استخدام الوسيط `easing` الذي لن يعمل إلا بوجود الـ `plugin` الذي يفعل هذه الخاصية:

```
$("#p").animate({
  "opacity": "show"
}, { "duration": "slow", "easing": "easein" });
```

stop(clearQueue, gotoEnd)

يُستخدم هذا التابع لإيقاف كل التأثيرات الحركية المطبقة على جميع العناصر التي تمت مطابقتها باستخدام المحدد الذي تم استدعاؤه قبل التابع, وبعد تنفيذ هذا التابع وفي حال وجود استدعاء لتابع animate ينتظر التنفيذ وكانت قيمة الوسيط الاختياري clearQueue مساوية false عندها يتم تنفيذ التابع التالي في ال- Queue, أما الوسيط gotoEnd فيقبل قيمة منطقية (true/false) وفي حال كانت قيمة الوسيط الاختياري gotoEnd تساوي true عندها يتم إكمال عملية تحريك العنصر, وفي المثال التالي يتم تحريك العنصر div عند الضغط على الزر Go وإذا أردت إيقاف الحركة اضغط على الزر STOP, وفي حال الضغط على الزر Go أكثر من مرة عندها سيتم تحريك العنصر div بعد الضغط على STOP وذلك لأن قيمة الوسيط clearQueue المستخدمة تساوي false:

```
// Start animation
$("#go").click(function(){
    $(".block").animate({left: '+=100px'}, 2000);
});

// Stop animation when button is clicked
$("#stop").click(function(){
    $(".block").stop();
});

// Start animation in the opposite direction
$("#back").click(function(){
    $(".block").animate({left: '-=100px'}, 2000);
});
```



أما الشيفرة الكاملة للمثال السابق فهي:

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>

  <script>
    $(document).ready(function(){

      // Start animation
      $("#go").click(function(){
        $(".block").animate({left: '+=100px'}, 2000);
      });

      // Stop animation when button is clicked
      $("#stop").click(function(){
        $(".block").stop();
      });

      // Start animation in the opposite direction
      $("#back").click(function(){
        $(".block").animate({left: '-=100px'}, 2000);
      });

    });
  </script>
  <style>div {
    position: absolute;
    background-color: #abc;
    left: 0px;
    top: 30px;
    width: 60px;
    height: 60px;
    margin: 5px;
  }

```

```

</style>
</head>
<body>
  <button id="go">Go</button>
  <button id="stop">STOP!</button>
  <button id="back">Back</button>
  <div class="block"></div>
</body>
</html>

```

jQuery.fx.off

متحول عام على مستوى الصفحة يمنع التأثيرات الحركية مثل تنفيذ التوابع Fading, Animation وغيرها من التي تعطي تأثيرات حركية على العناصر في حال تعيين القيمة إلى true, ففي المثال التالي يتم منع

التأثيرات الحركية حيث من المفترض أن يتم إخفاء العنصر `div` بشكل تدريجي ولكن بما أن قيمة المتحول `jQuery.fx.off` تساوي `true` عندها سيتم إخفاء العنصر مباشرة:

```
jQuery.fx.off = true;
$("input").click(function() {
    $("div").toggle("slow");
});
```

Run



أما المثال الكامل للشفرة السابقة فهي:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
    <script src="http://code.jquery.com/jquery-latest.js"></script>

    <script>
    $(document).ready(function() {

        jQuery.fx.off = true;
        $("input").click(function() {
            $("div").toggle("slow");
        });

    });
    </script>
    <style>
    body { cursor:pointer; }
    div { width:50px; height:30px; margin:5px; float:left;
        background:green; }
    span { color:red; }
    </style>
</head>
<body>
    <input type="button" value="Run"/><div></div>
</body>
</html>
```

توابع العمليات الغير متزامنة AJAX Functions

الطلبات الغير متزامنة AJAX Requests

jQuery.ajax(options)

يُستخدم هذا التابع لتنفيذ صفحة أخرى بتنفيذ HTTP Request, وهذا التابع هو تطبيق لتقنية الـ AJAX في بشكل مجرد في مكتبة jQuery, ويمكن الإطلاع على التابعين \$.get, \$.post لتنفيذ عمليات غير متزامنة AJAX بشكل أكثر سهولة في الفهم والاستخدام, ولكن لا تقدم وظائف إضافية مثل تنفيذ استدعاء عكسي عند حدوث خطأ error callbacks.

التابع \$.ajax يعيد كائن من النوع XMLHttpRequest, ولكن في معظم الحالات لا تحتاج استخدامه مباشرة ولكنه متوفر في حال وجود ضرورة لمتابعة عملية تنفيذ الطلب بشكل مباشر, ويقبل هذا التابع وسطاء على شكل مجموعة من المفاتيح والقيم التي تستخدم لتهيئة وتنفيذ ومتابعة عملية التنفيذ.

ملاحظة: في حال استخدام الخيار dataType في تنفيذ التابع فيجب التأكد من أن الخادم يرسل نوع صحيح من البيانات تتوافق مع قيمة الخيار dataType وفي حال أن نوع البيانات المرسل من الخادم لا تتوافق مع قيمة dataType هذا سيؤدي إلى أخطاء في عملية التنفيذ.

ملاحظة: أي عملية تنفيذ لصفحة خارجية (صفحة خارج النطاق الحالي) يجب تنفيذها باستخدام GET في حال أن قيمة الخيار dataType يساوي إحدى القيمتين script أو jsonp. وأيضاً فإن الخيارات التي تتطلب كائن XMLHttpRequest لا تكون ممكنة عند تنفيذ عملية على صفحة خارج النطاق الحالي, وأيضاً يتم تنفيذ التابعين complete و success عند الانتهاء من تنفيذ الطلب ولكن بدون الحصول على كائن من النوع XMLHttpRequest, أما التابعين beforeSend و dataFilter فلا يتم تنفيذهما.

التابع ajax في الإصدار 1.3 كما في الإصدار 1.2 يمكن أن يحمل بيانات JSON موجودة على نطاق آخر في حال تحديد الاستدعاء العكسي لـ JSONP, والذي يمكن تنفيذها بإحدى الطريقتين:

- كتابة عنوان الـ URL الذي تريد تنفيذه بالصيغة myurl?callback=? عندها فإن jQuery تستبدل الإشارة ? الثانية وبشكل تلقائي باسم التابع الصحيح لتنفيذه.
- أو في حال تحديد قيمة dataType إلى jsonp فعندها يتم بتنفيذ الاستدعاء العكسي المناسب بشكل تلقائي في عملية التنفيذ.

أما قيم الوسيط options فهي عبارة عن مجموعة من المفاتيح والقيم, وكلها عبارة عن قيمة اختيارية يمكن تعيين القيم الافتراضية لها باستخدام التابع \$.ajaxSetup().

async

يقبل هذا الخيار قيمة منطقية true أو false والقيمة الافتراضية تساوي true حيث أن كل الطلبات باستخدام التابع ajax يتم تنفيذها بشكل غير متزامن, وعند تعيين القيمة false للخيار async عندها سيتم تنفيذ الطلب بشكل متزامن, مع العلم أن عملية التنفيذ المتزامنة يمكن أن تؤدي إلى إيقاف استجابة المتصفح إلى حين الانتهاء من تنفيذ التابع ajax.

beforeSend

تابع استدعاء عكسي يتم تنفيذه قبل عملية إرسال الطلب يمكن من خلاله تعديل قيم الكائن XMLHttpRequest قبل إرساله, وفي حال إعادة القيمة false من هذا التابع عندها سيتم إيقاف تنفيذ التابع ajax, وللتابع BeforeSend الصيغة التالية:

```
function (XMLHttpRequest) {  
    this; // the options for this ajax request  
}
```

cache

يقبل هذا الخيار قيمة منطقية true أو false والقيمة الافتراضية تساوي true أما في حال أن قيمة dataType تساوي script أو jsonp فإن القيمة الافتراضية لهذا الخيار هي false, وفي حين أن قيمتها تساوي false فهذا سيمنع المتصفح من تخزين الصفحة التي يتم تنفيذها في الذاكرة المؤقتة للمتصفح.

complete

تابع استدعاء عكسي يتم تنفيذه بعد الانتهاء من تنفيذ الطلب بشكل كامل (بعد تنفيذ توابع الاستدعاء success وتوابع الاستدعاء الخاصة بالأخطاء error callbacks), ولهذا التابع وسيطين الأول هو كائن من النوع XMLHttpRequest والثاني قيمة نصية تحدد معلومات عن نجاح الطلب, ويتم تنفيذ هذا التابع دائماً سواء نجحت أو فشلت عملية التنفيذ وفي عمليات التنفيذ الغير متزامنة والمتزامنة:

```
function (XMLHttpRequest, textStatus) {  
    this; // the options for this ajax request  
}
```

contentType

قيمة نصية تحدد نوع المعلومات التي ترسل إلى الخادم والقيمة الافتراضية هي application/x-www-form-urlencoded وهذا القيمة مناسبة في معظم الحالات.

data

يتمضمّن هذا الخيار المعلومات التي سيتم إرسالها إلى الخادم, حيث يتم تحويلها إلى قيم ترسل بواسطة الـ query string أي تحول إلى قيمة نصية يتم إلحاقها بعنوان الصفحة المحدد, ويمكن استخدام الخيار

processData لمنع عملية المعالجة الافتراضية هذه. وقيم الخيار data يجب أن تكون على شكل أزواج من المفاتيح والقيم key/value, وفي حال أن القيمة عبارة عن مصفوفة عندها يتم تحويلها إلى مجموعة من القيم لها المفتاح نفسه (مثال: {foo:['bar1', 'bar2']} تصبح &foo=bar1&foo=bar2).

dataFilter

تابع استدعاء عكسي يتم استخدامه لمعالجة البيانات الأصلية المعادة من قبل الخادم في الكائن XMLHttpRequest, هو تابع ينفذ قبل عملية ترشيح البيانات لتجهيز البيانات بالشكل المطلوب لإعادة كقيمة مرتجعة من هذا التابو لهذا التابع وسيطين:

- البيانات الأصلية المعادة من قبل الخادم.
- قيمة الوسيط dataType.

```
function (data, type) {  
  // do something  
  // return the sanitized data  
  return data;  
}
```

dataType

هذا الوسيط يحدد نوع البيانات التي سيتم إرجاعها من الخادم, وفي حال لم يتم تحديد قيمة لهذا الوسيط فإن jQuery ستمرر إما responseXML أو.responseText إلى تابع الاستدعاء العكسي الذي يعبر عنه بالوسيط success اعتماداً على نوع البيانات التي أعادها الخادم والأنواع الممكنة هي:

- xml: وثيقة XML يمكن معالجتها باستخدام jQuery.
- html: يعيد شيفرة HTML بشكل قيمة نصية مع الواسمات scripts التي يتم تقييمها عند إدراجها في الصفحة.
- script: يتم تقييم النتيجة على أنها شيفرة JavaScript وإعادة على شكل قيمة نصية, وعند استخدام هذا النوع سيتم إلغاء تفعيل عملية التخزين المؤقت Cache إلا في حال استخدام قيمة مختلفة يتم تمريرها للوسيط cache, وفي حال استخدام هذا النوع عند التعامل مع عنوان نطاق آخر عندها سيتم تحويل العملية POST إلى العملية GET.
- json: سيتم تقييم القيمة المعادة على شكل JSON ويعيد كائن JavaScript.
- jsonp: سيتم تحميل القيمة المعادة من الخادم ضمن كتلة json باستخدام JSONP مع العلم أن سيتم إضافة "?callback=?" إلى نهاية الـ URL المطلوب لتحديد تابع الاستدعاء العكسي.
- text: القيمة المعادة من الخادم ستكون قيمة نصية فقط.

error

التابع الذي سيتم تنفيذه في حال فشل تنفيذ التابع ajax, ولهذا التابع ثلاثة وسطاء هي:

- كائن من النوع XMLHttpRequest.
- قيمة نصية تصف نوع الخطأ الذي حدث, والقيم المحتملة لهذا الوسيط هي:
 - null
 - timeout
 - error
 - notmodified
 - parsererror
- وكائن استثناء اختياري exception object.

وشكل التابع error هو:

```
function (XMLHttpRequest, textStatus, errorThrown) {  
  // typically only one of textStatus or errorThrown  
  // will have info  
  this; // the options for this ajax request  
}
```

global

قيمة هذا الوسيط الافتراضية تساوي true وتحدد هذه القيمة إن كان سيتم قرح أحداث العملية الغير متزامنة لتنفيذ التابع ajax مثل التابع ajaxStart و ajaxStop وغيرها من التوابع التي تدرج تحت Ajax Events.

ifModified

هذه القيمة تسمح للطلب بالنجاح فقط في حالة أن قيمة النتيجة مختلفة عن آخر نتيجة تم الحصول عليه وذلك بفحص قيمة Last-Modified في الترويسة الخاصة بالنتيجة والقيمة الافتراضية تساوي false أي تجاهل الترويسة Last-Modified.

jsonp

يستخدم هذا الوسيط لتجاوز اسم تابع الاستدعاء العكسي عند تنفيذ طلب jsonp, حيث سيتم استخدام هذه القيمة بدلاً من callback في الصيغة "callback=?" في الـ URL الذي سيتم تنفيذه سواء باستخدام GET أو POST والنتيجة هي أن {jsonp:'onJsonPLoad'} ستصبح "onJsonPLoad=?".

password

كلمة المرور التي يجب استخدامها عند وجود توثيق يعتمد على البروتوكول HTTP.

processData

بشكل افتراضي البيانات التي يتم إرسالها باستخدام الخيار data على شكل كائن (بشكل تقني أي شيء ماعدا القيمة النصية) ستتم معالجتها وتحويلها لكي ترسل عن طريق query string بشكل يتناسب مع نوع المحتوى الافتراضي "application/x-www-form-urlencoded", وفي حال وجود ضرورة لإرسال كائن DOMDocument أو أي بيانات غير معالجة عندها يجب تعيين قيمة false لهذا الخيار.

scriptCharset

يتم أخذ قيمة هذا الخيار بالاعتبار فقط عند تنفيذ طلب jsonp أو تكون قيمة الخيار dataType تساوي script ويتم تنفيذ الطلب باستخدام العملية GET, فإن قيمة هذا الخيار تحدد نوع الحروف التي سيتم تفسير الطلب على أساسها ويتم تحديدها فقط من أجل التوفيق بين الحروف في المحتوى البعيد والمحتوى المحلي.

success

اسم التابع الذي سيتم تنفيذه إذا تم تنفيذ الطلب بشكل ناجح, ولهذا التابع وسيطين هما:

- البيانات المعادة من الخادم والتي تمت إعادتها على حسب القيمة المحددة بالخيار dataType.
- قيمة نصية تعبر عن حالة الطلب.

```
function (data, textStatus) {  
    // data could be xmlDoc, jsonObj, html, text, etc...  
    this; // the options for this ajax request  
}
```

timeout

تحديد قيمة مدة قطع الاتصال قبل استجابة الخادم timeout للطلب الحالي, حيث أن تحديد قيمة الخيار timeout ستؤدي إلى تجاوز القيمة العامة المحددة باستخدام التابع \$.ajaxSetup.

type

يحدد هذا الخيار طريقة تنفيذ الطلب (POST, GET) والقيمة الافتراضية هي GET مع العلم أنه يمكن استخدام طرق أخرى مثل PUT, DELETE ولكنها غير مدعومة من قبل المتصفحات نفسها.

url

عنوان الصفحة التي سيتم طلبها.

username

اسم الدخول الذي سيتم استخدامه في حال وجود ضرورة للتوثيق باستخدام البروتوكول HTTP.

يحدد هذا الخيار اسم تابع يتم تنفيذه عند إنشاء كائن من النوع XMLHttpRequest والقيمة الافتراضية هي XMLHttpRequest في حال أن المتصفح هو IE وإلا يتم استخدام الكائن XMLHttpRequest, ويمكن استخدام هذا التابع لتجاوز الكائن XMLHttpRequest الذي تم إنشاؤه أو تغيير بعض خصائصه.

أمثلة

في المثال التالي يتم تحميل ملف JavaScript وتنفيذه مباشرة:

```
$.ajax({
  type: "GET",
  url: "test.js",
  dataType: "script"
});
```

أما في المثال التالي فيتم إرسال بعض البيانات إلى الخادم مع إعلام المستخدم في حال انتهاء العملية بنجاح:

```
$.ajax({
  type: "POST",
  url: "some.php",
  data: "name=John&location=Boston",
  success: function(msg){
    alert( "Data Saved: " + msg );
  }
});
```

في المثال التالي يتم إعادة آخر تحديث من محتوى الصفحة test.html:

```
$.ajax({
  url: "test.html",
  cache: false,
  success: function(html){
    $("#results").append(html);
  }
});
```

أما في المثال التالي فيتم تحميل بعض البيانات من الخادم بشكل متزامن مما يعني توقف المتصفح عن الاستجابة حتى الانتهاء من تنفيذ الطلب:

```
var html = $.ajax({
  url: "some.php",
  async: false
}).responseText;
```

أما في المثال التالي فيتم إرسال وثيقة XML إلى الخادم وذلك بتعيين القيمة false إلى الخيار processData لمنع عملية تحويل البيانات إلى قيمة نصية:

```
var xmlDocument = [create xml document];
$.ajax({
  url: "page.php",
  processData: false,
  data: xmlDocument,
  success: handleResponse
});
```

وأخيراً في هذا المثال يتم إرسال قيمة إلى الخادم وإعلام المستخدم في حال الانتهاء من تنفيذ العملية:

```
bodyContent = $.ajax({
  url: "script.php",
  global: false,
  type: "POST",
  data: ({id : this.getAttribute('id')}),
  dataType: "html",
  success: function(msg) {
    alert(msg);
  }
}).responseText;
```

load(url, [data], [callback])

يُستخدم هذا التابع لإعادة شيفرة HTML من الصفحة المحدد بالوسيط url, ويتم تنفيذ هذا التابع باستخدام العملية GET بشكل افتراضي إلا إن تم تمرير وسطاء إضافيين عندها سيتم تنفيذ التابع باستخدام العملية POST. وبدءاً من الإصدار 1.2 تستطيع تعيين محددات لترشيح شيفرة الـ HTML المعادة من قبل هذا التابع حيث تستطيع إرجاع جزء أو كل شيفرة الـ HTML من الـ rul المحدد. وصيغة تمرير المحدد تبدو كما في الشكل التالي: "url #some > selector", وبشكل افتراضي يتم تطبيق المحدد * > body. أما الوسيط data فهو وسيط اختياري يتألف من أزواج من المفاتيح والقيم {key:value}, أما الوسيط callback الاختياري فهو تابع يتم تنفيذه عند الانتهاء من تنفيذ التابع load بغض النظر هل نجح تنفيذ التابع أم لا.

في المثال التالي يتم تحميل محتوى الصفحة feeds.html داخل العنصر div الذي قيمة الوصفة id له تساوي feeds:

```
$("#feeds").load("feeds.html");
```

أما في المثال التالي فيتم تمرير مصفوفة قيم إلى الصفحة test.php:

```
$("#objectID").load("test.php", { 'choices[]': ["Jon", "Susan"] } );
```

أما في المثال الأخير فيتم تمرير القيمة limit تساوي 25 إلى الصفحة feeds.php باستخدام العملية POST وبعد انتهاء من تنفيذ التابع load يتم تنفيذ التابع callback:

```
$("#feeds").load("feeds.php", {limit: 25}, function(){
    alert("The last 25 entries in the feed have been loaded");
});
```

أما المثال التالي فيتم تحميل جميع العناصر اللائحة التي قيمة الوصفة id لها تساوي jq-p-Getting-Started ضمن لائحة في الصفحة الحالية قيمة الوصفة id لها تساوي links:

```
$("#links").load("/Main_Page #jq-p-Getting-Started li");
```



أما الشيفرة الكاملة للمثال السابق فهي:

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>

  <script>
    $(document).ready(function() {
      $("#links").load("/Main_Page #jq-p-Getting-Started li");
    });
  </script>
  <style>body{ font-size: 11px; font-family: Arial; }</style>
</head>
<body>
  <b>jQuery Links:</b>
  <ul id="links"></ul>
</body>
</html>

```

jQuery.get(url, [data], [callback], [type])

يُستخدم هذا التابع لتحميل محتويات صفحة محدد بالوسيط url باستخدام العملية GET. ويعتبر هذا التابع طريقة بسيطة إرسال طلب باستخدام العملية GET إلى الخادم بدون إمكانية استخدام التابع \$.ajax, والوسيط callback يحدد تابع يتم تنفيذه في حال الانتهاء وبنجاح من تنفيذ التابع get, أما الوسيط data فهو وسيط اختياري يتألف من أزواج من المفاتيح والقيم {key:value}, أما الوسيط الاختياري type فيقبل القيم التالية:

- .xml
- .html
- .script
- .json
- .jsonp
- .text

jQuery.getJSON(url, [data], [callback])

يُستخدم هذا التابع لتحميل بيانات بصيغة JSON باستخدام العملية GET, وبدءاً من الإصدار jQuery 1.2 يمكن تحميل بيانات بصيغة JSON موجودة على نطاق آخر بتحديد تابع الاستدعاء العكسي JSONP الذي يمكن تمريره كما يلي: "myurl?callback=?", حيث تستبدل مكتبة jQuery قيمة callback باسم التابع المناسب.

تحميل بيانات JSON من الملف test.js وعرض قيمة احقل name من هذه البيانات:

```
$.getJSON("test.js", function(json){
    alert("JSON Data: " + json.users[3].name);
});
```

أما المثال التالي فهو شبيه للمثال السابق مع وجود فارق وهو تمرير وسطاء عند استدعاء التابع:

```
$.getJSON("test.js", { name: "John", time: "2pm" }, function(json){
    alert("JSON Data: " + json.users[3].name);
});
```

وهذا التابع له ثلاثة وسطاء هي:

- url: عنوان الصفحة التي سيتم تنفيذها ومن الممكن أن تكون على نفس النطاق أو نطاق آخر.
- data: وسيط اختياري يتألف من مجموعة من أزواج المفاتيح والقيم.
- callback: تابع استدعاء عكسي سيتم تنفيذه عند الانتهاء من تنفيذ التابع getJSON بشكل ناجح, ولهذا التابع وسيطين:

○ data: وهي بيانات بصيغة JSON

○ textStatus: قيمة نصية لها إحدى القيم التالية:

- timeout
- error
- notmodified
- success
- parsererror

jQuery.getScript(url, [callback])

يُستخدم هذا التابع لتحميل وتنفيذ ملف JavaScript باستخدام العملية GET, وهذا التابع له

وسيطين هما:

- url: مسار ملف الـ JavaScript ويمكن أن يكون هذا المسار على نفس النطاق أو على نطاق آخر.
- callback: تابع استدعاء عكسي يتم تنفيذه عند تحميل الملف بشكل ناجح.

في المثال التالي يتم تحميل الملف test.js:

```
$.getScript("test.js");
```

أما المثال التالي فهو مشابه للمثال السابق مع وجود استدعاء للتابع العكسي:

```
$.getScript("test.js", function(){  
    alert("Script loaded and executed.");  
});
```

jQuery.post(url, [data], [callback], [type])

يُستخدم هذا التابع لتحميل محتويات صفحة محدد بالوسيط url باستخدام العملية POST. ويعتبر هذا التابع طريقة بسيطة إرسال طلب باستخدام العملية POST إلى الخادم بدون إمكانية استخدام التابع \$.ajax, والوسيط callback يحدد تابع يتم تنفيذه في حال الانتهاء وبنجاح من تنفيذ التابع post, أما الوسيط data فهو وسيط اختياري يتألف من أزواج من المفاتيح والقيم {key:value}, أما الوسيط الاختياري type فيقبل القيم التالية:

- .xml
- .html
- .script
- .json
- .jsonp
- .text

أمثلة

في المثال التالي يتم طلب الصفحة test.php مع تجاهل القيمة التي تعيدها الصفحة:

```
$.post("test.php");
```

أيضاً في المثال التالي يتم طلب الصفحة test.php لكن هذه المرة مع إرسال بيانات إلى الصفحة عن طريق الوسيط الاختياري data وأيضاً يتم تجاهل البيانات التي تعيدها الصفحة:

```
$.post("test.php", { name: "John", time: "2pm" });
```

وأيضاً المثال التالي:

```
$.post("test.php", { 'choices[]': ["Jon", "Susan"] });
```


أما في المثال التالي فيتم طلب الصفحة test.php مع تنفيذ تابع الاستدعاء العكسي الصرح عنه بالوسيط الاختياري callback:

```
$.post("test.php", function(data){
    alert("Data Loaded: " + data);
});
```

أيضاً المثال التالي يشبه المثال السابق مع وجود تمرير للبيانات باستخدام الوسيط الاختياري data:

```
$.post("test.php", { name: "John", time: "2pm" },
function(data){
    alert("Data Loaded: " + data);
});
```

أحداث العمليات الغير متزامنة Ajax Events

ajaxComplete(callback)

يُستخدم هذا الحدث لتنفيذ التابع callback عند الانتهاء من تنفيذ عملية غير متزامنة AJAX Request, ويقبل هذا التابع وسيطين هما كائن من النوع XMLHttpRequest وإعدادات تنفيذ العملية الغير متزامنة:

```
function (event, XMLHttpRequest, ajaxOptions) {
    this; // dom element listening
}
```

وفي المثال التالي يتم عرض رسالة عند الانتهاء من تنفيذ عملية غير متزامنة:

```
$("#msg").ajaxComplete(function(request, settings){
    $(this).append("<li>Request Complete.</li>");
});
```

ajaxError(callback)

يُستخدم هذا الحدث لتنفيذ التابع callback عند حدوث خطأ أثناء تنفيذ عملية غير متزامنة AJAX Request, ويقبل هذا التابع وسيطين هما كائن من النوع XMLHttpRequest وإعدادات تنفيذ العملية الغير متزامنة:

```
function (event, XMLHttpRequest, ajaxOptions, thrownError) {
    // thrownError only passed if an error was caught
    this; // dom element listening
}
```

وفي المثال التالي يتم عرض رسالة عند حدوث خطأ أثناء تنفيذ عملية غير متزامنة:

```
$("#msg").ajaxError(function(event, request, settings){
    $(this).append("<li>Error requesting page " + settings.url + "</li>");
});
```

ajaxSend(callback)

يُستخدم هذا الحدث لتنفيذ التابع callback قبل بدء تنفيذ العملية غير متزامنة AJAX Request، ويقبل هذا التابع وسيطين هما كائن من النوع XMLHttpRequest وإعدادات تنفيذ العملية الغير متزامنة:

```
function (event, XMLHttpRequest, ajaxOptions) {
    this; // dom element listening
}
```

وفي المثال التالي يتم عرض رسالة قبل بدء تنفيذ العملية الغير متزامنة:

```
$("#msg").ajaxSend(function(evt, request, settings){
    $(this).append("<li>Starting request at " + settings.url + "</li>");
});
```

ajaxStart(callback)

يُستخدم هذا الحدث لتنفيذ التابع callback عند البدء بتنفيذ العملية غير متزامنة AJAX Request، وليس لهذا التابع أي وسيط:

```
function () {
    this; // dom element listening
}
```

المثال التالي يظهر رسالة "جاري التحميل" عند بدء تنفيذ الطلب:

```
$("#loading").ajaxStart(function(){
    $(this).show();
});
```

ajaxStop(callback)

يُستخدم هذا الحدث لتنفيذ التابع callback عند الانتهاء من تنفيذ العملية غير متزامنة AJAX Request، وليس لهذا التابع أي وسيط:

```
function () {
    this; // dom element listening
}
```

في المثال التالي يتم إخفاء رسالة "جاري التحميل" عند بدء الانتهاء من تنفيذ العمليات الغير متزامنة:

```
$("#loading").ajaxStop(function(){
    $(this).hide();
});
```

ajaxSuccess(callback)

يُستخدم هذا الحدث لتنفيذ التابع callback عند الانتهاء بنجاح من تنفيذ عملية غير متزامنة AJAX Request, ويقبل هذا التابع وسيطين هما كائن من النوع XMLHttpRequest وإعدادات تنفيذ العملية الغير متزامنة:

```
function (event, XMLHttpRequest, ajaxOptions) {
    this; // dom element listening
}
```

في المثال التالي يتم إظهار رسالة عند الانتهاء من تنفيذ العملية بنجاح:

```
$("#msg").ajaxSuccess(function(evt, request, settings){
    $(this).append("<li>Successful Request!</li>");
});
```

متنوعات

jQuery.ajaxSetup(options)

يُستخدم هذا التابع لتعيين الإعدادات العامة للطلبات الغير متزامنة, والخيار options هو عبارة عن مجموعة من المفاتيح والقيم. وفي المثال التالي يتم تحديد قيمة الخيار url إلى القيمة "/xmlhttp/" مع إلغاء الأحداث العامة واستخدام العملية POST بدلاً من العملية GET لتنفيذ الطلبات:

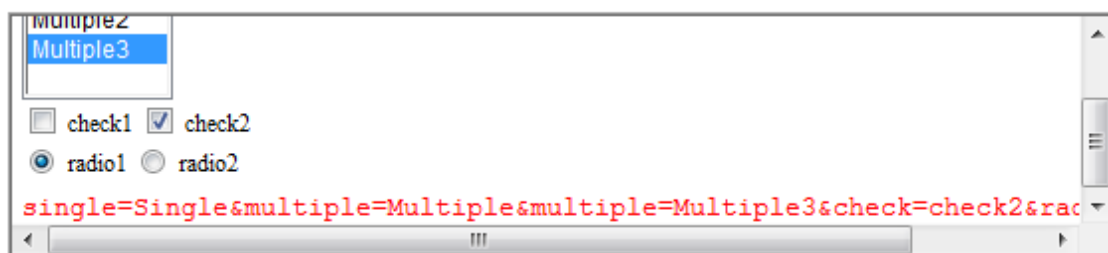
```
$.ajaxSetup({
    url: "/xmlhttp/",
    global: false,
    type: "POST"
});
$.ajax({ data: myData });
```

serialize()

يُستخدم هذا التابع لسلسلة مجموعة من القيم وتحويلها إلى سلسلة نصية, وهذه العملية تستخدم لتجهيز البيانات التي أدخلها المستخدم لإرسالها إلى الخادم, وهذه العملية هي عملية قياسية تتوافق مع جميع لغات البرمجة ومنصات العمل. ولكي يعمل هذا التابع بشكل صحيح يجب أن تحتوي حقول النموذج على قيمة للوصفة name وإلا فإن هذا التابع لن يعمل بشكل صحيح.

في المثال التالي يتم تحويل القيم المدخلة في النموذج إلى سلسلة نصية يمكن إرسالها عبر الـ URL لتنفيذ عمليات غير متزامنة:

```
function showValues() {  
    var str = $("form").serialize();  
    $("#results").text(str);  
}  
  
$(":checkbox, :radio").click(showValues);  
$("select").change(showValues);  
showValues();
```



أما والشيفرة الكاملة للمثال السابق فهي:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>

  <script>
    $(document).ready(function(){

      function showValues() {
        var str = $("form").serialize();
        $("#results").text(str);
      }

      $(".checkbox, :radio").click(showValues);
      $("select").change(showValues);
      showValues();

    });
  </script>
  <style>
    body, select { font-size:12px; }
    form { margin:5px; }
    p { color:red; margin:5px; font-size:14px; }
    b { color:blue; }
  </style>
</head>
<body>

<form>
  <select name="single">
    <option>Single</option>
    <option>Single2</option>
  </select>
<br />
  <select name="multiple" multiple="multiple">
```

```

    <option selected="selected">Multiple</option>
    <option>Multiple2</option>
    <option selected="selected">Multiple3</option>
  </select>
<br/>
  <input type="checkbox" name="check" value="check1" id="ch1"/>
  <label for="ch1">check1</label>

  <input type="checkbox" name="check" value="check2" checked="checked" id="ch2"/>
  <label for="ch2">check2</label>
<br />
  <input type="radio" name="radio" value="radio1" checked="checked" id="r1"/>
  <label for="r1">radio1</label>
  <input type="radio" name="radio" value="radio2" id="r2"/>
  <label for="r2">radio2</label>
</form>
<p><tt id="results"></tt></p>
</body>
</html>

```

serializeArray()

يُستخدم هذا التابع لسلسلة جميع حقول النموذج كما في التابع `serialize()` لكن هذا التابع يحول القيم إلى صيغة JSON، ونتيجة التابع ليست قيمة نصية وإنما كائن من نوع JSON. وشكل النتيجة يكون مشابه لما يلي:

```

[
  {name: 'firstname', value: 'Hello'},
  {name: 'lastname', value: 'World'},
  {name: 'alias'}, // this one was empty
]

```

```

function showValues() {
    var fields = $("input").serializeArray();
    $("#results").empty();
    jQuery.each(fields, function(i, field){
        $("#results").append(field.value + " ");
    });
}

$("checkbox, :radio").click(showValues);
$("select").change(showValues);
showValues();

```

Results: Single Multiple Multiple3 check2 radio1

أدوات

توابع التفاعل مع المتصفح وفحص الميزات Browser and Feature Detection

jQuery.support

مجموعة من الخصائص تمت إضافتها في الإصدار 1.3 يتم استخدامها لمعرفة بعض خصائص المتصفح والعيوب التي فيه. ومكتبة jQuery تحتوي على مجموعة من الخصائص الجاهزة للاستخدام ويمكن إضافة خصائص أخرى خاصة بك. طبعاً معظم هذه الخصائص ليس من الضرورة أن يتم استخدامها في المهام اليومية في عملك ولكنها تُستخدم أثناء تطوير البرمجيات الجاهزة Plugins وبنية مكتبة jQuery.

boxModel

تكون قيمة هذه الخاصية تساوي true إن تمت عملية رسم الصفحة أو المتصفح باستخدام W3C CSS Box Model وهذه القيمة تساوي false في IE6, IE7 عندما تعمل في النمط Quirks وتكون قيمة هذه الخاصية تساوي null حتى يتم تحميل كامل الوثيقة.

cssFloat

قيمة هذه الخاصية تساوي true عندما يتم استخدام style.cssFloat للوصول إلى القيمة العائمة الحالية في CSS, وهذه القيمة تساوي false في جميع إصدارات IE حيث يتم استخدام styleFloat بدلاً عن .cssFloat.

hrefNormalized

قيمة هذه الخاصية تساوي true إن كان المتصفح يترك نتائج استدعاء التابع `getAttribute('href')` بدون تعديل (القيمة في المتصفح IE تساوي false).

htmlSerialize

قيمة هذه الخاصية تساوي true إن كان المتصفح يسلسل عناصر الرابط عند استخدام الخاصية `innerHTML` (قيمة هذه الخاصية تساوي false في IE).

leadingWhitespace

قيمة هذه الخاصية تساوي true في حال أن المتصفح يحافظ على الفراغات عند استخدام الخاصية `innerHTML` (قيمة هذه الخاصية تساوي false في IE6 و IE8).

noCloneEvent

قيمة هذه الخاصية تساوي true إذا كان المتصفح ينسخ توابع الاستجابة للأحداث عند نسخ العنصر (قيمة هذه الخاصية تساوي false في المتصفح IE).

objectAll

قيمة هذه الخاصية تساوي true إن كان استدعاء التابع `getElementsByTagName('*')` يعيد كل العناصر الأحفاد للعنصر الحالي (قيمة هذه الخاصية تساوي false في IE7 و IE8).

opacity

قيمة هذه الخاصية تساوي true إن كان المتصفح يستطيع تفسير `opacity` css بشكل صحيح (قيمة هذه الخاصية تساوي false في المتصفح IE حيث يتم استخدام المرشح `filter` بدلاً عن `opacity`).

scriptEval

قيمة هذه الخاصية تساوي true إن كان استخدام التابع `appendChild` أو التابع `createTextNode` لإدراج نص يؤدي إلى تفسير قيمة هذا النص (قيمة هذه الخاصية تساوي false في المتصفح IE حيث يتم استخدام `text` بدلاً من ذلك).

style

قيمة هذه الخاصية تساوي true إن كان استدعاء التابع `getAttribute('style')` تعيد قيمة الوصفة `style` الخطية للعنصر الحالي (قيمة هذه الخاصية تساوي false في المتصفح IE ويمكن استخدام الخاصية `cssText` بدلاً عن `style`).

tbody

قيمة هذه الخاصية تساوي true إن كان المتصفح يسمح بالواسمة `tbody` بدون الواسمة `table` (قيمة هذه الخاصية في المتصفح IE تساوي false حيث يتم إدراج الواسمة `tbody` إن لم تكن موجودة).

jQuery.browser

تم إلغاء هذه الخاصية في الإصدار 1.3 (يمكن استخدام jQuery.support بدلاً عنها) وتحتوي على إشارة لكل متصفح، وقيمتها هي من navigator.userAgent والقيم المتوفرة هي:

- safari
- opera
- msie
- mozilla

وتكون قيمة هذه الخاصية جاهزة قبل تحميل كامل عناصر الوثيقة لذلك يمكن استخدام هذه الخاصية لإضافة أحداث جاهزة لبعض المتصفحات فقط.

في المثال التالي يتم عرض إصدار مع اسم المتصفح الحالي:

```
jQuery.each(jQuery.browser, function(i, val) {
    $("

" + i + " : <span>" + val + "</span>")
        .appendTo(document.body);
});


```

Browser info:

```
version : 7.0
safari : false
opera : false
msie : true
mozilla : false
```

أما الشيفرة الكاملة للمثال السابق فهي:

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>

  <script>
    $(document).ready(function() {

      jQuery.each(jQuery.browser, function(i, val) {
        $("<div>" + i + " : <span>" + val + "</span>")
          .appendTo(document.body);
      });

    });
  </script>
  <style>
    p { color:green; font-weight:bolder; margin:3px 0 0 10px; }
    div { color:blue; margin-left:20px; font-size:14px; }
    span { color:red; }
  </style>
</head>
<body>
  <p>Browser info:</p>
</body>
</html>

```

وفي المثال التالي يتم إظهار رسالة فقط إن كان المتصفح هو Firefox الإصدار رقم 3:

```

jQuery.each(jQuery.browser, function(i, val) {
  if(i=="mozilla" && jQuery.browser.version.substr(0,3)=="1.9")
    alert("Do stuff for firefox 3")
});

```

أما في المثال التالي يتم تحديد قيمة النمط CSS بناء على نوع المتصفح:

```

jQuery.each(jQuery.browser, function(i) {
  if($.browser.msie){
    $("#div ul li").css("display","inline");
  }else{
    $("#div ul li").css("display","inline-table");
  }
});

```

jQuery.browser.version

تم إلغاء هذه الخاصية في الإصدار 1.3 (يمكن استخدام jQuery.support بدلاً عنها) وتعيد هذه

الخاصية رقم الإصدار الحالي للمتصفح الحالي، ومايلي بعض القيم الخاصة:

- Internet Explorer: 6.0, 7.0 •
- Moziatl/Firefox/Flock/Camino: 1.7.12, 1.8.1.3, 1.9 •
- Opera: 9.20 •
- Safari/Webkit: 312.8, 418.9 •

في المثال التالي يتم عرض إصدار المتصفح الحالي:

```
$( "p" ).html( "The browser version is: <span>" +
    jQuery.browser.version + "</span>" );
```

The browser version is: 7.0

أما الشيفرة الكاملة للمثال السابق فهي:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>

  <script>
    $(document).ready(function(){

      $( "p" ).html( "The browser version is: <span>" +
        jQuery.browser.version + "</span>" );

    });
  </script>
  <style>
    p { color:blue; margin:20px; }
    span { color:red; }
  </style>
</head>
<body>
  <p>
  </p>
</body>
</html>
```

أما في المثال التالي فيتم عرض رسالة نصية للمتصفح IE:

```
if ( $.browser.msie )
  alert( $.browser.version );
```

أما المثال التالي فيتم عرض الإصدار رقم الإصدار الرئيسي للمتصفح IE:

```
if (jQuery.browser.msie) {  
    alert (parseInt (jQuery.browser.version));  
}
```

jQuery.boxModel

تم إلغاء هذه الخاصية في الإصدار 1.3 (يمكن استخدام jQuery.support بدلاً عنها) وتعيد هذه الخاصية true إن كان المتصفح الحالي يتم عرضه باستخدام W3C CSS Box Model:

```
$("#p").html("The box model for this iframe is: <span>" +  
    jQuery.boxModel + "</span>");
```

The box model for this iframe is: true