

# شروع کار با jQuery

## دستکاری المان‌ها

سید کاوه احمدی

## پیمایش المان‌ها (Traversing)

متد/خصوصیت	عملکرد
size(), length	تعداد المان‌های موجود را مجموعه‌ی نتیجه‌ی jQuery را بازمی‌گرداند.
get()	مجموعه انتخابی را به شکل مجموعه المان‌های DOM بازمی‌گرداند. برای زمانی که نیاز دارید المان‌های DOM را به بجای اشیای jQuery مدیریت کنید مفید است.
get(index)	دسترسی به یک المان DOM که توسط index مشخص شده است.
find(expression)	به دنبال اولادی می‌گردد که با عبارت مشخص شده جور باشد
each(fn)	تابع fn را روی تمامی المان‌های درون مجموعه اعمال می‌کند.

- `$( "p" ).length;`
- `$( "p" ).size();`
  
- `$( "li" ).get();` //a DOM array  
– `$( "li" ).get().length`
  
- `$( "li" ).get(0);`
  
- `$( "ul" ).find( "li.b" )`

```
var border = 3;
$( "p" ).each(function() {
    $(this).css("border", border + "px solid red");
    border += 2;
});
```

پیمایش با استفاده از یک تابع بدون نام

به المان در حال پیمایش اشاره دارد

```
$(selector).fn1().fn2().fn3()
```

یک زنجیره از اظهارات

## تغییر محتوای صفحه

▪ jQuery توابعی برای ایجاد، کپی کردن، حذف و جابجایی محتوای صفحه دارد

عملکرد	تابع
محتوای html اولین المان جور شده را باز می‌گرداند.	<code>html()</code>
محتوای html المان‌های جور شده را به <code>newcontent</code> تغییر می‌دهد.	<code>html(newcontent)</code>
محتوای متنی اولین المان جور شده را باز می‌گرداند.	<code>text()</code>
محتوای متنی المان‌های جور شده را به <code>newtext</code> تغییر می‌دهد.	<code>text(newtext)</code>

## امتحان کنید:

- `alert($("#list1").html());`
- `$("#list1").html("<li>New Item</li>");`
- `$("#p:last").text("Last P");`

## دستکاری خصوصیات

عملکرد	تابع
دسترسی به خصوصیت name اولین المان جور شده. اگر المان چنین خصوصیتی نداشته باشد، undefined باز می‌گرداند.	attr(name)
تنظیم یک مجموعه از خصوصیات روی تمامی المان‌های جور شده با استفاده از نحو object. راهی مناسب برای تنظیم تعداد زیادی خصوصیت:	attr(properties)
<pre>\$( 'img' ).attr( {   src: '#',   title: 'jQuery',   alt: '...' } );</pre>	
تنظیم مقدار خصوصیت key با مقدار value برای المان‌های جور شده.	attr(key, value)
مقدار مورد نظر برای خصوصیت می‌تواند به جای یک رشته، خروجی یک تابع باشد.	attr(key, fn)
حذف یک خصوصیت برای المان‌های جور شده	removeAttr(name)

## امتحان کنید:

- `$( 'a' ).attr( 'target', '_blank' );`
- `$( 'a' ).removeAttr( 'target' );`

عملکرد	تابع
مقدار خصوصیت CSS مشخص شده را برای اولین المان جور شده باز می‌گرداند.	css (name)
تنظیم مقدار یک خصوصیت CSS برای تمامی المان‌های جور شده. اگر برای value از اعداد استفاده کنیم، به شکل خودکار به عنوان پیکسل در نظر گرفته می‌شوند به جز برای خصوصیات zoom, opacity, font-weight, z-index و line-height	css(property, value)
تنظیم چندین خصوصیت CSS با استفاده از نحو شی	css(properties)

## کار با CSS

```
var cssObj = {
    'background-color': '#fff',
    'font-size': 20,
    'color': 'rgb(0, 100, 255)'
}
$('p').css(cssObj);
```

■ مشابه

```
$('p').css({
    'background-color': '#fff',
    'font-size': 20,
    'color': 'rgb(0, 100, 255)'
});
```

تابع	مقدار
<code>addClass(class)</code>	اضافه کردن کلاس(های) مشخص شده به مجموعه المان‌های جور شده.
<code>hasClass(class)</code>	در صورتی که دست کم یکی از المان‌های جور شده دارای کلاس مشخص شده باشند، <code>true</code> باز می‌گرداند.
<code>removeClass(class)</code>	حذف کردن کلاس(های) مشخص شده از مجموعه المان‌های جور شده.
<code>toggleClass(class)</code>	اضافه کردن کلاس مشخص شده‌ها المان‌های جور شده اگر المان آنرا ندارد و حذف آن اگر المان آنرا دارد.
<code>toggleClass(class, switch)</code>	کلاس را اضافه می‌کند اگر <code>switch</code> مقدار <code>true</code> دارد و آنرا حذف می‌کند از <code>switch</code> مقدار <code>false</code> دارد.

## دستکاری عرض و ارتفاع المان‌ها

تابع	عملکرد
<code>height()</code>	ارتفاع محاسبه شده برای اولین المان جور شده را برحسب پیکسل باز می‌گرداند.
<code>height(val)</code>	تنظیم ارتفاع تمامی المان‌های جور شده.
<code>width()</code>	عرض محاسبه شده برای اولین المان جور شده را برحسب پیکسل باز می‌گرداند.
<code>width(val)</code>	تنظیم عرض تمامی المان‌های جور شده.
<code>innerHeight()</code>	ارتفاع داخلی را برای اولین المان جور شده باز می‌گرداند ( <code>padding</code> در نظر گرفته می‌شود ولی <code>border</code> نه).
<code>outerHeight(margin)</code>	ارتفاع خارجی را برای اولین المان جور شده باز می‌گرداند ( <code>padding</code> و <code>border</code> به طور پیش‌فرض در نظر گرفته می‌شود. اگر <code>margin</code> برابر <code>true</code> باشد، مقدار <code>margin</code> نیز در آن محاسبه می‌شود).
<code>innerWidth()</code>	عرض داخلی را برای اولین المان جور شده باز می‌گرداند ( <code>padding</code> در نظر گرفته می‌شود ولی <code>border</code> نه).
<code>outerWidth(margin)</code>	عرض خارجی را برای اولین المان جور شده باز می‌گرداند ( <code>padding</code> و <code>border</code> به طور پیش‌فرض در نظر گرفته می‌شود. اگر <code>margin</code> برابر <code>true</code> باشد، مقدار <code>margin</code> نیز در آن محاسبه می‌شود).

```
var newHeader = $( "<h1>New Header</h1>" );
```

یا

- ```
var newHeader = $( "<h1 />" );
$( "para2" ).text( 'New Header' );
```
- ```
var new = $( "<p>new P</p>" );
$( "para2" ).html( new.html( ) );
```

## درج

عملکرد	تابع
content را به داخل تمامی المان‌های مجموعه اضافه می‌کند (در انتها).	append( content )
المان‌های مجموعه را به داخل تمامی المان‌هایی که توسط انتخابگر selector انتخاب می‌شوند اضافه می‌کند (در انتها).	appenderTo( selector )
content را به داخل تمامی المان‌های مجموعه اضافه می‌کند (در ابتدا).	prepend( content )
المان‌های مجموعه را به داخل تمامی المان‌هایی که توسط انتخابگر selector انتخاب می‌شوند اضافه می‌کند (در ابتدا).	prependTo( selector )
content را پس از تمامی المان‌های مجموعه اضافه می‌کند.	after( content )
content را پیش از تمامی المان‌های مجموعه اضافه می‌کند.	before( content )
المان‌های مجموعه را پس از تمامی المان‌هایی که توسط انتخابگر selector انتخاب می‌شوند اضافه می‌کند.	insertAfter( selector )
المان‌های مجموعه را پیش از تمامی المان‌هایی که توسط انتخابگر selector انتخاب می‌شوند اضافه می‌کند.	insertBefore( selector )



- `$( "p" ).append( " this is appended" );`
- `$( "p" ).prepend( ":D" );`
  
- `$( "p:last" ).appendTo( "p:first" );`
- `$( "p:last" ).prependTo( "p:first" );`

## تمرین

- اضافه کردن یک آیکن خاص به لینک‌هایی که به یک فایل pdf زده شده است.