

عنوان: JQuery Plugins #1

نویسنده: مجتبی کاویانی

تاریخ: ۱۶:۲۵ ۱۳۹۱/۱۲/۰۷

آدرس: [www.dotnettips.info](http://www.dotnettips.info)

برچسب‌ها: JQuery, JQuery-Tips, Plugin

جی کوئری به عنوان مهم‌ترین و پرکاربردترین کتابخانه جاوا اسکریپتی، حالا در اکثر سایت‌های اینترنتی استفاده می‌شود و هر روز به قابلیت‌ها و امکانات آن اضافه می‌گردد. اما بیش از خود این کتابخانه، پلاگین‌های آن است که تحول عظیمی را در طراحی وب سایت‌ها ایجاد نموده است. از انواع اسلایدها، تصاویر، منوها، Tooltip ها، نمودارها، انیمیشن، جداول و هزاران پلاگین دیگر، همه و همه کدهای جاوا اسکریپتی است که با استفاده از جی کوئری به صورت پلاگین نوشته شده است و امکان استفاده مجدد را به ما می‌دهد.

## از کجا شروع کنیم

برای نوشتن پلاگین یک تابع با نام خاصیتی جدید را به `jQuery.fn` اضافه می‌نماییم.

```
jQuery.fn.myPlugin = function() {  
  //محتویات پلاگین را اینجا می‌نویسیم  
};
```

اما، برای اینکه بتوانیم از میانبر `$` در پلاگین استفاده نماییم و تداخلی با سایر کتابخانه‌ها نداشته باشد، از الگوی ( IIFE Immediately Invoked Function Expression ) به صورت زیر استفاده می‌نماییم:

```
(function( $ ) {  
  $.fn.myPlugin = function() {  
    //محتویات پلاگین را اینجا می‌نویسیم  
  };  
})( jQuery );
```

## محتوای پلاگین

حال می‌توانیم در تابع، کدهای پلاگین خود را بنویسیم. برای دسترسی به شیء پاس داده شده به پلاگین، از کلمه کلیدی `this` استفاده کرده و لازم نیست از `$(this)` استفاده نماییم. در زیر یک پلاگین ساده تهیه شده است که با رفتن ماوس بر روی یک متن، خطی زیر آن می‌کشد:

```
(function($){  
  $.fn.underline= function() {  
    this.hover(function(){  
      $(this).css( { text-decoration : underline } )  
    }, function(){  
      $(this).css( { text-decoration : none } )  
    });  
  };  
})(jQuery);  
$("p").underline();
```

پلاگین بالا مقدار یا شیءایی را بر نمی‌گرداند؛ اما اگر بخواهیم مقداری را برگردانیم از `return` استفاده می‌نماییم:

```
(function( $ ){  
  $.fn.maxHeight = function() {  
    var max = 0;  
    this.each(function() {  
      max = Math.max( max, $(this).height() );  
    });  
    return max;  
  };  
})(jQuery);
```

```
});
})( jQuery );
```

```
var tallest = $('div').maxHeight(); // بیشترین ارتفاع عنصر را برمی گرداند
```

### حفظ خاصیت زنجیره‌ای پلاگین ها

در مثال بالا یک مقدار عددی برگردانده شده است؛ اما برای اینکه بتوانیم بصورت زنجیر وار خروجی پلاگین را به تابع یا هر پلاگین دیگری پاس دهیم از تابع each بصورت زیر استفاده می‌نماییم:

```
(function( $ ){
    $.fn.lockDimensions = function( type ) {
        return this.each(function() {
            var $this = $(this);
            if ( !type || type == 'width' ) {
                $this.width( $this.width() );
            }
            if ( !type || type == 'height' ) {
                $this.height( $this.height() );
            }
        });
    };
})( jQuery );
```

```
$('div').lockDimensions('width').css('color', 'red');
```

در پلاگین بالا با از تابع each برای روی this و برگرداندن آن با return برای حفظ خاصیت زنجیره‌ای پلاگین استفاده می‌نماییم. در تابع each می‌بایست از \$(this) برای انجام عملیات بر روی شیء پاس داده شده استفاده کنیم. بدین صورت بعد از صدا زدن پلاگین، دوباره می‌توانیم از هر پلاگین یا تابع جی کوئری دیگری بر روی خروجی استفاده نماییم.

### پیش فرض‌ها و تنظیمات

در پلاگین‌های پیشرفته‌تر می‌توانیم تنظیمات پیش فرضی را برای پلاگین در نظر بگیریم و این تنظیمات را به عنوان پارامتر ورودی از کاربر دریافت نماییم. جی کوئری دارای تابعی به نام extend است که امکان گسترش و ترکیب دو شیء را امکان پذیر می‌سازد به مثال زیر توجه نمایید:

```
(function( $ ){
    $.fn.tooltip = function( options ) {
        var settings = $.extend( {
            'location' : 'top',
            'background-color' : 'blue'
        }, options);
        return this.each(function() {
            // Tooltip plugin code here
        });
    };
})( jQuery );
```

```
$('div').tooltip({
    'location' : 'left'
});
```

در این مثال، شیء settings با دو خاصیت location و background-color تعریف شده که با شیء options که از ورودی پلاگین دریافت نموده‌ایم با استفاده از تابع extend ترکیب شده است. خاصیت‌های که تعیین نشده باشند با مقادیر پیش فرض آنها تکمیل می‌گردد.  
ادامه دارد...

## نظرات خوانندگان

نویسنده: امیر

تاریخ: ۱۳:۵۴ ۱۳۹۱/۱۲/۰۹

مرسی عالی بود .استفاده کردم.فقط زمانی که سی اس اس میدی نباید اونطوری نوشته بشه

```
$(this).css( "text-decoration" ,"none" )
```

نویسنده: مجتبی کاویانی

تاریخ: ۱۶:۲۳ ۱۳۹۱/۱۲/۰۹

ممنون. در جاوااسکریپ هر دو [صحیح](#) است

نویسنده: محسن

تاریخ: ۱۶:۵۳ ۱۳۹۱/۱۲/۰۹

```
$(this).css( { text-decoration : underline })
```

فکر کنم منظور ایشون («اونطوری» که نوشته) وجود [dash](#) در نام متغیر بوده.

نویسنده: مجتبی کاویانی

تاریخ: ۱۸:۲۸ ۱۳۹۱/۱۲/۰۹

منظورشون نحوه ست کردن [css](#) که هر دو روش استفاده می شود ولی در [jquery 1.9 css](#) به بعد کمی تغییر کرده است