

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 می‌شود ولی در [jquery 1.9 css](#) به بعد کمی تغییر کرده است

در قسمت اول آموزش [jQuery Plugin #1](#) با نحوه ساخت اولیه پلاگین در جی کوئری آشنا شدید. در ادامه به موارد دیگری خواهیم پرداخت.

### فضای نام

در پلاگین شما، فضای نام، بخش مهمی از توسعه پلاگین می‌باشد. فضای نام در واقع تضمین می‌کند که پلاگین شما توسط دیگر پلاگین‌ها باز نویسی نشود یا با کدهای موجود در صفحه تداخل نداشته باشد. همچنین کمک می‌کند که توابع، رویدادها و داده‌های پلاگین خود را بهتر مدیر کنید.

### توابع پلاگین

تحت هیچ شرایطی نباید یک پلاگین، در چندین فضای نام، به شیء JQuery.fn اضافه گردند. به مثال زیر توجه نمایید:

```
(function( $ ){
    $.fn.tooltip = function( options ) {
        // این
    };
    $.fn.tooltipShow = function( ) {
        // تعریف
    };
    $.fn.tooltipHide = function( ) {
        // بد است
    };
    $.fn.tooltipUpdate = function( content ) {
        // !
    };
})( jQuery );
```

همین طور که در مثال بالا مشاهده می‌کنید، پلاگین به شکل بدی تعریف شده و هر تابع در یک فضای نام جداگانه تعریف گردیده‌است. برای این کار شما باید تمام توابع را در یک متغیر تعریف و آن را به پلاگین خود پاس دهید و توابع را با نام رشته ای صدا بزنید.

```
(function( $ ){
    var methods = {
        init : function( options ) {
            // این
        },
        show : function( ) {
            // تعریف
        },
        hide : function( ) {
            // خوب است
        },
        update : function( content ) {
            // !
        }
    };
    $.fn.tooltip = function( method ) {
        // منطق تابع را از اینجا صدا زده ایم
        if ( methods[method] ) {
            return methods[method].apply( this, Array.prototype.slice.call( arguments, 1 ));
        } else if ( typeof method === 'object' || ! method ) {
            return methods.init.apply( this, arguments );
        } else {
            $.error( 'Method ' + method + ' does not exist on jQuery.tooltip' );
        }
    };
})
```

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

**توضیح:** متغیر method اگر در متغیر methods توابع موجود باشد، تابع هم نام آن و در صورت داشتن پارامتر ورودی، به آن تابع پاس داده شده و برگردانده می‌شود (در واقع صدا زده می‌شود). در غیر اینصورت اگر نوع مقدار ورودی، object بود تابع init آن صدا زده می‌شود وگرنه پیام خطا ارسال می‌گردد. برای استفاده از پلاگین بصورت زیر عمل می‌کنیم:

```
// صدا زده می‌شود init تابع
$('div').tooltip();
```

9

```
// با پارامتر صدا زده می‌شود update تابع
$('div').tooltip('update', 'This is the new tooltip content!');
```

این معماری به شما امکان کپسوله کردن توابع در پلاگین را می‌دهد.

## رویدادها

یکی از روش‌های کمتر شناخته شده انقیاد توابع در فضای نام، امکان انقیاد رویدادها است. اگر پلاگین شما یک رویداد را انقیاد نماید، این یک عمل و تمرین خوب استفاده از فضای نام می‌باشد. بدین ترتیب اگر لازم باشد که انقیاد یک رویداد را حذف نمایید، بدون تداخل با دیگر رویدادها و بدون اینکه در یک شی دیگر از این پلاگین، اختلالی ایجاد نماید می‌توان آن را حذف نمود. به مثال زیر توجه نمایید.

```
(function( $ ){
    var methods = {
        init : function( options ) {
            return this.each(function(){
                $(window).bind('resize.tooltip', methods.reposition);
            });
        },
        destroy : function( ) {
            return this.each(function(){
                $(window).unbind('.tooltip');
            })
        },
        reposition : function( ) {
            // ...
        },
        show : function( ) {
            // ...
        },
        hide : function( ) {
            // ...
        },
        update : function( content ) {
            // ...
        }
    };

    $.fn.tooltip = function( method ) {

        if ( methods[method] ) {
            return methods[method].apply( this, Array.prototype.slice.call( arguments, 1 ));
        } else if ( typeof method === 'object' || ! method ) {
            return methods.init.apply( this, arguments );
        } else {

```

```
    $.error( 'Method ' + method + ' does not exist on jQuery.tooltip' );  
  }  
};  
})( jQuery );
```

این همان مثال قبل است که وقتی پلاگین با تابع Init مقدار دهی اولیه می‌شود، تابع reposition به رویداد resize پنجره در فضای نام پلاگین tooltip انقیاد می‌شود. پس از آن اگر توسعه دهنده نیاز داشت تا tooltip را از بین ببرد، با صدا زدن تابع destroy می‌تواند بصورت امن انقیاد ایجاد شده را حذف نماید.

```
$('#fun').tooltip();  
// مدتی بعد...  
$('#fun').tooltip('destroy');
```

ادامه دارد...

عنوان: آشنایی با پلاگین TickTack برای Mask ورودی کاربر

نویسنده: محسن موسوی

تاریخ: ۱۷:۵ ۱۳۹۲/۰۲/۱۴

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

برچسب‌ها: jQuery, Plugin, TimePicker

همانطور که می‌دانیم پلاگین‌های جی‌کوئری، نقش مهمی را در محیط وب ایفا می‌کنند. در اینجا با یکی از این پلاگین‌ها و چگونگی استفاده از آن آشنا می‌شویم.

برای آشنایی با نوشتن Plugin در jQuery، می‌توان مباحث پیشین این سایت را دنبال کرد. ([jQuery Plugins #1](#) و [jQuery](#) ([Plugins #2](#)))

: [jQueryTickTack Plugin](#)

این Plugin برای ایجاد یک TextBox برای ورود زمان توسط کاربر استفاده می‌شود. با توجه به اینکه قبلاً چند Plugin برای این کار نوشته شده است ولی هر کدام از آنها معایب و مزایای خاص خود را داشتند، برای نمونه می‌توانید به [این سایت](#) مراجعه کنید.

ویژگی‌های این Plugin عبارتند از:

1- تنظیم زمان پیش فرض

2- کنترل حداقل و حداکثر زمان وارد شده

3- تغییر ساعت و دقیقه بوسیله کلیدهای جهتی بالا و پایین

4- تغییر انتخاب ساعت و دقیقه بوسیله کلیدهای جهتی چپ و راست

5- تغییر ساعت و دقیقه بوسیله فشردن اعداد روی صفحه کلید

چگونگی استفاده از این Plugin

ابتدا کتابخانه jQuery و [این پلاگین](#) را به صفحه خود اضافه نمایید و سپس کدهای زیر را برای استفاده از این Plugin اضافه نمایید:

```
jQuery(document).ready(function () {
    $("#TextBox1").TickTack();
    $("#TextBox2").TickTack({
        initialTime: '8:44',
        minHour: 8,
        minMinute: 0,
        maxHour: 22,
        maxMinute: 40
    });
});
```

در ادامه به بررسی تنظیمات انجام شده در این پلاگین می‌پردازیم:

initialTime: زمان اولیه جهت نمایش به کاربر (حتماً بایستی ساعت و دقیقه بوسیله ' : ' از یکدیگر جدا شوند)

minHour: حداقل ساعت ورودی



minMinute : حداقل دقیقه ورودی

maxHour : حداکثر ساعت ورودی

maxMinute : حداکثر دقیقه ورودی

پس از انجام این تنظیمات و اجرا کردن برنامه، شما به صورت زیر نمایش داده می‌شود:

please input time : 13:44

پس از انتخاب TextBox ، قسمت ساعت به صورت پیش فرض انتخاب می‌شود و کاربر باید ساعت مد نظر را وارد کند؛ در اینجا، عدد اول ساعت، مد نظر است.

please input time : 13:44

برای نمونه در اینجا عدد 2 توسط کاربر وارد می‌شود؛ پس از ورود عدد و با توجه به تنظیمات انجام شده، ساعت به صورت اتوماتیک به حداکثر مقداری که می‌تواند بپذیرد تغییر می‌کند (در این مثال چون کاربر عدد 2 را وارد کرده و در تنظیمات انجام شده حداکثر ساعت دریافتی 22 و حداکثر دقیقه 40 تعریف شده است، ساعت به صورت پیش‌فرض به 22:40 تغییر می‌یابد)

please input time : 22:40

و پس از وارد کردن عدد دوم ساعت توسط کاربر مکان نما به قسمت دقیقه منتقل می‌شود که در این جا عدد اول دقیقه مد نظر است

please input time : 21:40

وارد کردن عدد 3 برای دقیقه

please input time : 21:30

وارد کردن عدد دوم دقیقه

please input time : 21:35

پس از وارد کردن کامل دقیقه مکان نما دوباره به قسمت ساعت باز می‌گردد.

در ادامه دوستان علاقمند لطفا جهت بهبود کیفیت کار، باگ و یا مشکلات کدنویسی را اطلاع دهند.

با تشکر

## نظرات خوانندگان

نویسنده: بهروز  
تاریخ: ۱۶:۱۸ ۱۳۹۲/۰۶/۲۵

با سلام خدمت جناب آقای محسن موسوی

یه مشکلی داشتم تو این پلاگین که وقتی از تو کد به تکست باکس مقداری میدیم اون مقدارو نمیگیره و فقط مقدار پیشفرض خودشو نشون میده... میخوام در صورت امکان راهنمایی بفرمایید.

با تشکر فراوان

و من الله توفیق...

نویسنده: محسن موسوی  
تاریخ: ۱۷:۵۴ ۱۳۹۲/۰۶/۲۵

سلام

از طریق سرور پلاگین را صدا بزنید یا اینکه از طریق سرور یک مقدار به متغیر جاوااسکریپتی بدهید.

نویسنده: سعید حر  
تاریخ: ۱۱:۳۷ ۱۳۹۲/۰۷/۰۶

سلام

در آدرس <https://ticktack.codeplex.com> هیچ فایلی برای دانلود وجود ندارد.

نویسنده: وحید نصیری  
تاریخ: ۱۱:۴۵ ۱۳۹۲/۰۷/۰۶

به [برگه سورس پروژه](#) مراجعه کنید.

نویسنده: محسن موسوی  
تاریخ: ۱۲:۲۹ ۱۳۹۲/۰۷/۰۶

با تشکر از آقای نصیری

هنوز بازخورد خاصی از پروژه نگرفتم. در هفته‌ی آتی احتمالا release پروژه را ارائه میدهم.