

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...