

بسم الله الرحمن الرحيم

آموزشگاه تحلیل داده

تخصصی ترین مرکز برنامه نویسی و دیتابیس در ایران

متدهای bind() و unbind()

مدرس : مهندس افشین رفوآ

[دوره آموزشی JQuery](#)

متدهای bind() و unbind()

متد bind() و unbind()، همان طور که از نام آن ها پیدا است، یک سری کد را به یک یا چند رخداد از المان (ها) متصل کرده یا از آن حذف می کنند. مثال:

```
<a href="javascript:void(0);">Test 1</a>
<a href="javascript:void(0);">Test 2</a>
<script type="text/javascript">
$(function () {
    $("a").bind("click", function () {
        alert($(this).text());
    });
});
</script>
```

این مثال تمامی المان های <a> را انتخاب می کند و سپس تابع بی نام (anonymous) را با استفاده از متد bind() به رخداد کلیک متصل می نماید. یک اتفاق جالب که رخ می دهد این است که جی کوئری به صورت خودکار المانی که بر روی آن کلیک می شود را به کلیدواژه ی "this" داخل تابع بی نام تخصیص می دهد. حتی اگر یک کد را به چندین المان تخصیص دهید، این قابلیت باعث می شود رخداد به المان فعال نسبت داده شود.

زمانی که JQuery متد شما را صدا می زند، در صورتی که یک یا چند پارامتر به آن پاس داده باشید، اطلاعات مربوط به رخداد را به عنوان اولین پارامتر به تابع پاس می دهد. به عنوان مثال، شیء event که به عنوان پارامتر به تابع بی نام ارسال شده دربردارنده ی اطلاعاتی در خصوص مکان قرارگیری اشاره گر موس، نوع آن رخداد، اینکه کدام یک از دکمه های صفحه کلید یا موس فشار داده شده و غیره ... می باشد.

مثال:

```
<div id="divArea" style="background-color: silver; width: 100px; height: 100px;">
</div>
<script type="text/javascript">
    $("#divArea").bind("mousemove", function (event) {
        $(this).text(event.pageX + "," + event.pageY);
    });
</script>
```

در این مثال یک المان **div** با اندازه و رنگ پس زمینه ی معین ایجاد می کنیم. برای این **div**، رخداد **mousemove** را تعریف می کنیم، یک تابع بی نام که پارامتر **event** به آن ارسال شده نیز برای آن مشخص می کنیم. شی **this** به ما اجازه ی دسترسی به **property** های **pageX** و **pageY** (مختصات مکان نمای موس بر روی المان **div** نسبت به گوشه ی بالا سمت چپ) را می دهد. اگر این مثال را امتحان کنید، می بینید که مختصات موس بر روی المان **div** به نمایش گذاشته می شود و همین که مکان نمای موس حرکت می کند، مختصات آن نیز بروز رسانی می گردد.

متد **bind()** همچنین به شما اجازه می دهد اطلاعات خود را پاس داده و سپس از طریق شی **event** به اطلاعات آن دسترسی پیدا کنید. همچنین به شما این امکان را می دهد مقادیری را در زمان **bind** کردن رخداد تنظیم کرده و آن را در زمان فراخوانی رخداد بخوانید، حتی اگر مقدار اصلی تغییر کرده باشد. مثال:

```
<a href="javascript:void(0);">Test 1</a>
<a href="javascript:void(0);">Test 2</a>
<script type="text/javascript">
    var msg = "Hello, world!";
    $(function () {
        $("a").bind("click", { message: msg }, function (event) {
            msg = "Changed msg";
            alert(event.data.message);
        });
    });
</script>
```

پارامتر دوم متد **bind()** را به صورت یک سری جفت کلید و مقدار به آن ارسال می کنیم. می توان چندین مقدار را به عنوان پارامتر ارسال کرد که در آن صورت بایستی مقادیر را توسط ویرگول از هم تفکیک نمود. برای دسترسی به مقدار ریخته شده در **event handler**، از **data property** شی **event** استفاده می کنیم. این **property**، به ازای هر یک از مقادیری که ارسال کرده اید، دربردارنده ی **subproperty** هایی می باشد. بدین

معنی که شما می توانید به وسیله ی `event.data.message` به مقدار پارامتر `message` دسترسی داشته باشید.

با اینکه مقدار متغیر "`msg`" را داخل `event handler` تغییر می دهیم، بازهم پیغامی که با هر بار کلیک بر روی المان لینک نمایش داده می شود، رشته ی "`Hello, world!`" خواهد بود. علت عدم تغییر پیغامی که نمایش داده می شود، این است که مقدار آن به محض `bind` شدن `event handler` ارزیابی می شود که زمانی است که صفحه به طور کامل بارگذاری می گردد.

