

بسم الله الرحمن الرحيم

آموزشگاه تحلیل داده

تخصصی ترین مرکز برنامه نویسی و دیتابیس در ایران

## Event ها - رخدادها در jQuery

مدرس : مهندس افشین رفوآ

### Event ها - رخدادها در jQuery

jQuery برای واکنش نشان دادن به رخدادهای که در یک سند HTML روی می دهند، بهینه سازی شده اند.

رویدادهای jQuery ، متدهایی هستند که به مجرد اجرای یک اتفاق در HTML مثل کلیک بر روی یک عنصر یا قرار گرفتن موس بر روی آن ، فراخوانی شده و اجرا می گردد.

### اساسا رخداد (Event) به چه گفته می شود؟

تمام اتفاق هایی که در یک صفحه ی وب روی می دهد و می توانیم به آنها واکنش نشان دهیم، رخداد (Event) نامیده می شود.

نمونه

قرار گرفتن یا گذشتن نشانگر موس از روی یک عنصر

انتخاب یک radio button

کلیک بر روی یک المان

اصطلاح " اتفاق افتادن " اغلب همراه با واژه ی " رخداد " بکار می رود. به عنوان مثال، رخداد **keypress** هنگامی اتفاق می افتد که یکی از دکمه های صفحه کلید را فشار دهید.

در جدول زیر، برخی از رویدادهای پرکاربرد در مدل **DOM** (مدل شی گرای سند) ارائه گردیده است:

رخداد های صفحه/پنجره	رخداد های مربوط به فرم	رخداد های ویژه ی صفحه کلید	رخداد های موس
load	submit	keypress	click
resize	change	keydown	dblclick
scroll	focus	keyup	mouseenter
unload	blur		mouseleave

## چگونگی استفاده از رخدادها در jQuery

اغلب رخداد های مدل شی گرای سند (**DOM**)، یک تابع همتا و برابر در **jQuery** دارند.

برای تخصیص رخداد **onclick** به تمام المان های **<p>** پاراگراف های صفحه، می توانید به صورت زیر اقدام نمایید:

```
$("#p").click();
```

در قدم بعد بایستی تعریف کنید که به محض فعال (اجرا) رخداد، چه اتفاقی باید روی دهد. باید یک تابع را به عنوان آرگومان به رخداد مربوطه پاس دهید:

```
$("#p").click(function(){  
  // action goes here!!  
});
```

## توابع رخداد رایج در jQuery

تابع **\$(document).ready()**:

تابع **\$(document).ready()** به ما این امکان را می دهد که به محض اتمام بارگذاری صفحه، تابع مورد نظر را اجرا کنیم.

**click()**

یک تابع را برای پاسخ دادن به رویداد **onclick** یک عنصر، به آن اختصاص می دهد؛ به عبارتی دیگر این متد یک تابع مدیریت کننده ی رخداد (**event handler function**) به المان **HTML** الحاق می کند.

به مجرد اینکه کاربر بر روی یک المان **HTML** کلیک کند، رخداد **onclick** اتفاق می افتد (اجرا یا فعال می شود).

همان طور که در نمونه ی زیر مشاهده می کنید، کاربر با کلیک بر روی هر کدام از تگ های **<p>** در صفحه ، پاراگراف مورد نظر را مخفی می کند:

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"></script>
<script>
$(document).ready(function () {
    $("p").click(function () {
        $(this).hide();
    });
});
</script>
</head>
<body>

<p>If you click on me, I will disappear.</p>
<p>Click me away!</p>
<p>Click me too!</p>

</body>
</html>
```

**dblclick()**

رخداد **dblclick** زمانی اتفاق می افتد که کاربر بر روی عنصر مورد نظر دوبار کلیک نماید . یکی از کاربردهای معمول این تابع: می توان کدی طراحی کرد که با دوبار کلیک کاربر بر روی یک عنصر رنگ پس زمینه آن تغییر یافته و یا عنصر مخفی شود.

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"></script>
<script>
$(document).ready(function () {
```

```

        $("p").dblclick(function () {
            $(this).hide();
        });
    });
</script>
</head>
<body>
<p>If you double-click on me, I will disappear.</p>
<p>Click me away!</p>
<p>Click me too!</p>
</body>
</html>

```

mouseenter()

تابع زمانی اجرا می شود که اشاره گر موس وارد المان HTML می شود:

```

<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"></script>
<script>
$(document).ready(function () {
    $("#p1").mouseenter(function () {
        alert("You entered p1!");
    });
});
</script>
</head>
<body>
<p id="p1">Enter this paragraph.</p>

</body>
</html>

```

mouseleave()

تابع به محض اینکه اشاره گر موس المان HTML را ترک می کند، اجرا می شود:

```

<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"></script>
<script>
$(document).ready(function () {
    $("#p1").mouseleave(function () {
        alert("Bye! You now leave p1!");
    });
});
</script>

```

```
</head>
<body>
<p id="p1">This is a paragraph.</p>
</body>
</html>
```

mousedown()

رویداد **mousedown** هنگامی فعال می شود که دکمه سمت چپ موس را بر روی عنصر مورد نظر فشار دهید :

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"></script>
<script>
$(document).ready(function () {
    $("#p1").mousedown(function () {
        alert("Mouse down over p1!");
    });
});
</script>
</head>
<body>
<p id="p1">This is a paragraph.</p>
</body>
</html>
```

mouseup()

رخداد **mouseup** زمانی اجرا می شود که دکمه فشرده شده موس بر روی المان مربوطه را رها کنید :

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"></script>
<script>
$(document).ready(function () {
    $("#p1").mouseup(function () {
        alert("Mouse up over p1!");
    });
});
</script>
</head>
<body>
```

```
<p id="p1">This is a paragraph.</p>
```

```
</body>
```

```
</html>
```

## hover()

رویداد **hover()** دو تابع را مشخص می کند که با گذر اشاره گر موس از روی المان مورد نظر ، اجرا می شوند.

با قرار گرفتن اشاره گر موس کاربر روی عنصر **HTML** ، متد اول اجرا می شود و زمانی که موس از روی آن کنار می رود، متد دوم اجرا خواهد شد. **Hover()** باعث می شود هر دو رویداد **mouseenter** و **mouseleave** اجرا شوند.

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"></script>
<script>
$(document).ready(function () {
    $("#p1").hover(function () {
        alert("You entered p1!");
    },
    function () {
        alert("Bye! You now leave p1!");
    });
});
</script>
</head>
<body>
<p id="p1">This is a paragraph.</p>
</body>
</html>
```

## focus()

زمانی که علامت چشمک زن اشاره گر موس روی یکی از اعضای فرم، مانند کادر متن قرار گیرد، این رخداد فعال می شود؛ به عبارتی دیگر رخداد **focus** زمانی روی می دهد که کنترل یا عنصر مورد نظر تمرکز **(focus)** برنامه را کسب کند.

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"></script>
```

```

<script>
$(document).ready(function () {
    $("input").focus(function () {
        $(this).css("background-color", "#cccccc");
    });
    $("input").blur(function () {
        $(this).css("background-color", "#ffffff");
    });
});
</script>
</head>
<body>
Name: <input type="text" name="fullname"><br>
Email: <input type="text" name="email">
</body>
</html>

```

blur()

به مجرد اینکه تمرکز (focus) از روی یکی از اعضای فرم خارج شود، رویداد blur() فعال و اجرا می شود.

```

<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"></script>
<script>
$(document).ready(function () {
    $("input").focus(function () {
        $(this).css("background-color", "#cccccc");
    });
    $("input").blur(function () {
        $(this).css("background-color", "#ffffff");
    });
});
</script>
</head>
<body>
Name: <input type="text" name="fullname"><br>
Email: <input type="text" name="email">
</body>
</html>

```

on()

این متد یک یا چندین event handler به المان های انتخابی پیوست می کند.

در این مثال یک رخداد کلیک به المان <p> ضمیمه می شود:

```

<!DOCTYPE html>
<html>

```

```

<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"></script>
<script>
$(document).ready(function () {
$("p").on("click", function () {
$(this).hide();
});
});
</script>
</head>
<body>

<p>If you click on me, I will disappear.</p>
<p>Click me away!</p>
<p>Click me too!</p>

</body>
</html>

```

در نمونه ی زیر چندین مدیریت کننده ی رخداد (event handler) به یک المان <p> اضافه می گردد:

```

<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"></script>
<script>
$(document).ready(function () {
$("p").on({
mouseenter: function () {
$(this).css("background-color", "lightgray");
},
mouseleave: function () {
$(this).css("background-color", "lightblue");
},
click: function () {
$(this).css("background-color", "yellow");
}
});
});
</script>
</head>
<body>
<p>Click or move the mouse pointer over this paragraph.</p>
</body>
</html>

```