

بسم الله الرحمن الرحيم

آموزشگاه تحلیل داده

تخصصی ترین مرکز برنامه نویسی و دیتابیس در ایران

## DOM HTML DOM EVENTS – رخدادهای مدل

مدرس : مهندس افشین رفوآ

[دوره آموزش JQuery](#)

[دوره آموزش JavaScript](#)

## DOM HTML DOM EVENTS – رخدادهای مدل

مدل مبتنی بر شی سند HTML (مدل DOM) به جاوا اسکریپت این امکان را می دهد که به رخدادهای HTML واکنش نشان دهد.

واکنش نشان دادن به رخدادها

کد جاوا اسکریپت می تواند زمانی که یک رویداد (event) رخ می دهد، اجرا شود، به عنوان مثال به زمانی که کاربر روی یک المان مشخص HTML کلیک می کند، اشاره کرد.

برای اینکه یک کد پس از کلیک کاربر روی المان مشخص اجرا شود، کافی است کد جاوا اسکریپت را به خصیصه ی رخداد یا همان event attribute عنصر مربوطه اضافه کرد:

`onclick=JavaScript`

مثالی از رخدادهای HTML:

1. زمانی که کاربر موس را کلیک می کند
2. هنگامی که یک صفحه به طور کامل بارگذاری می شود
3. زمانی که یک تصویر بارگذاری می شود
4. هنگامی که مکان نمای موس روی یک المان قرار می گیرد
5. زمانی که فیلد ورودی (input field) تغییر می کند
6. هنگامی که فرم HTML ارسال می گردد
7. وقتی که کاربر یکی از دکمه های صفحه کلید را فشار می دهد

در این مثال، هنگامی که کاربر روی المان **<h1>** کلیک می کند، محتویات المان نام برده نیز تغییر می کند:

```
<!DOCTYPE html>
<html>
<body>
  <h1 onclick="this.innerHTML='Oops!'">Click on this text!</h1>
</body>
</html>
```

در این مثال، یک تابع از **event handler** فراخوانی می شود:

```
<!DOCTYPE html>
<html>
<body>
  <h1 onclick="changeText(this)">Click on this text!</h1>
  <script>
    function changeText(id) {
      id.innerHTML = "Oops!";
    }
  </script>
</body>
</html>
```

## خصیصه های event، HTML

به منظور تخصیص رخداد (**event**) به عنصرهای HTML، می توان از خصیصه ی رخداد (**event attribute**)

بهره جست:

در مثال پیش رو یک رخداد **onClick** به المان **button** اختصاص می دهیم:

```
<!DOCTYPE html>
```

```

<html>
<body>
  <p>Click the button to display the date.</p>
  <button onclick="displayDate()">The time is?</button>
  <script>
    function displayDate() {
      document.getElementById("demo").innerHTML = Date();
    }
  </script>
  <p id="demo"></p>
</body>
</html>

```

در مثال فوق، زمانی که روی دکمه (عنصر **button**) کلیک می شود، تابعی به نام *displayDate* فراخوانده شده و اجرا می گردد.

## تخصیص رخداد به وسیله ی مدل DOM

مدل **DOM** به شما امکان می دهد با استفاده از جاوا اسکریپت، به المان های **HTML** رویداد تخصیص دهید.

در مثال زیر یک رخداد **onClick** به المان **button** انتساب می دهیم:

```

<!DOCTYPE html>
<html>
<body>
  <p>Click "Try it" to execute the displayDate() function.</p>
  <button id="myBtn">Try it</button>
  <p id="demo"></p>
  <script>
    document.getElementById("myBtn").onclick = displayDate;
    function displayDate() {
      document.getElementById("demo").innerHTML = Date();
    }
  </script>
</body>
</html>

```

در این مثال تابعی به نام *displayDate*، به المانی که شناسه ی آن "myBtn" است تخصیص می دهیم. تابع مزبور به محض کلیک روی دکمه، اجرا می شود.

## رخدادهای onload و onunload

**Event** های **onLoad** و **onunLoad** زمانی اجرا می شوند که کاربر دکمه ای کلیک کرده یا صفحه را ترک کند.

**onload** رخدادی است که با وارد شدن کاربر به صفحه اجرا می شود و **onunload** رویدادی است با خروج کاربر از صفحه راه اندازی می شود. **onload** معمولا برای چک کردن نوع و ورژن مرورگر بازدیدکننده بکار می رود و با توجه به اطلاعات بدست آمده یک نسخه ی مناسب از صفحه وب را بارگذاری یا **load** می کند.

رخدادهای **unload** و **onunload** را می توان برای کار با کوکی ها مورد استفاده قرار داد. دو رویداد ذکر شده اغلب برای تنظیم **cookie** ها، زمانی که کاربر وارد صفحه می شود و یا از صفحه خارج می شود، نیز بکار گرفته می شوند.

```
<!DOCTYPE html>
<html>
<body onload="checkCookies()">
  <p id="demo"></p>
  <script>
    function checkCookies() {
      var text = "";
      if (navigator.cookieEnabled == true) {
        text = "Cookies are enabled.";
      } else {
        text = "Cookies are not enabled.";
      }
      document.getElementById("demo").innerHTML = text;
    }
  </script>
</body>
</html>
```

## رخداد onchange

این رخداد اغلب به صورت ترکیبی و به همراه اعتبارسنجی فیلدهای ورودی بکار گرفته می شود. **onchange** رخدادی است که به هنگام ایجاد تغییر در اجزای فرم، اجرا می گردد.

زیر مثالی از کاربرد **onchange** را مشاهده می کنید. به محض اینکه کاربر محتوای فیلد ورودی را تغییر می دهد، تابع **upperCase()** فراخوانده می شود.

```
<!DOCTYPE html>
<html>
<head>
  <script>
```

```

function myFunction() {
    var x = document.getElementById("fname");
    x.value = x.value.toUpperCase();
}
</script>
</head>
<body>
    Enter your name: <input type="text" id="fname" onchange="myFunction()">
    <p>When you leave the input field, a function is triggered which transforms the input text to upper case.</p>
</body>
</html>

```

## رخدادهای onmouseover و onmouseout

به وسیله ی دو رخداد مذکور می توان به مجرد اینکه کاربر نشانه گر موس را روی المان قرار داد و یا آن را از روی المان یاد شده برداشت، یک تابع معین را فراخواند.

## رخدادهای onclick و onmouseup، onmousedown

سه رخداد نام برده همگی بخشی از **mouse-click** هستند. **onmousedown** - این رخداد زمانی که دکمه موس، فشار داده می شود اجرا می گردد. **Onmouseup** این رویداد هنگامی که دکمه ی موس، رها می شود رخ می دهد.

```

<!DOCTYPE html>
<html>
<body>
    <div onmouseover="mOver(this)" onmouseout="mOut(this)"
        style="background-color:#D94A38;width:120px;height:20px;padding:40px;">
        Mouse Over Me
    </div>
    <script>
        function mOver(obj) {
            obj.innerHTML = "Thank You"
        }

        function mOut(obj) {
            obj.innerHTML = "Mouse Over Me"
        }
    </script>
</body>
</html>

```

**onclick** - زمانی که عملیات کلیک کردن موس، کامل می شود، روی می دهد.

```
<!DOCTYPE html>
<html>
<body>
  <div onmousedown="mDown(this)" onmouseup="mUp(this)"
    style="background-color:#D94A38;width:90px;height:20px;padding:40px;">
    Click Me
  </div>
  <script>
    function mDown(obj) {
      obj.style.backgroundColor = "#1ec5e5";
      obj.innerHTML = "Release Me";
    }
    function mUp(obj) {
      obj.style.backgroundColor = "#D94A38";
      obj.innerHTML = "Thank You";
    }
  </script>
</body>
</html>
```