

بسم الله الرحمن الرحيم

آموزشگاه تحلیل داده

تخصصی ترین مرکز برنامه نویسی و دیتابیس در ایران

زمان بندی رخدادها (timing events) در جاوا اسکریپت

مدرس : مهندس افشین رفوآ

[دوره آموزش JavaScript](#)

[دوره آموزش JQuery](#)

زمان بندی رخدادها (timing events) در جاوا اسکریپت

کدهای جاوا اسکریپت را می توان در فواصل زمانی معین اجرا کرد.

به این قابلیت (اجرای تابع در بازه های زمانی مشخص) زمان بندی رویدادها گفته می شود.

در جاوا اسکریپت این امکان وجود دارد که یک تابع (قطعه کد) را در فواصل زمانی مشخص اجرا کرد. به عبارتی

دیگر به وسیله توابع خاصی در جاوا اسکریپت، می توانید کاری کنید که کدهای شما در دوره های زمانی معین

(تکرار) اجرا شوند و یا یک کد و تابع با گذشت زمان مشخصی از زمان بارگذاری اجرا گردد.

به چنین توابعی در زبان جاوا اسکریپت، تابع های زمانی می گویند.

زمان بندی رخدادها در جاوا اسکریپت بسیار ساده است. دو متد اصلی که برای زمان بندی رخدادها بکار می

روند، عبارتند از:

1. `setInterval()` – یک تابع را در فواصل زمانی مشخص بارها و بارها اجرا (تکرار) می کند.

2. `setTimeout()` – یک تابع را پس از گذشت زمان معینی (تعداد هزارم ثانیه ی مشخصی) یکبار اجرا

می کند.

هر دو تابع ذکر شده متدهای شی `window` مدل `HTML DOM` می باشند.

آدرس آموزشگاه : تهران – خیابان شریعتی – بالاتراز خیابان ملک – جنب بانک صادرات – پلاک – 561 طبقه دوم واحد 7

88146330-88446780-88146332

تابع () setInterval

متد () **setInterval** به صورت متناوب و در فواصل زمانی معین، پس از گذشت مدت زمان مشخص شده (هزارم ثانیه) برای آن، یک تابع یا کد را اجرا می کند و مادام اینکه به وسیله برنامه متوقف نشده، این رویه را ادامه خواهد داد.

دستور نگارش این تابع به نحو زیر می باشد:

```
window.setInterval("javascript function", milliseconds);
```

این تابع را می توان بدون پیشوند **window** نیز بکاربرد.

اولین پارامتر () **setInterval**، نام تابعی است که این متد آن را به طور متناوب اجرا خواهد کرد.

دومین پارامتر ورودی، مدت یا طول زمان بین هر بار اجرا را بر حسب میلی ثانیه تعیین می کند.

مثال:

این مثال هر سه ثانیه یکبار رشته ی "hello" را نمایش می دهد:

```
setInterval(function () {alert("Hello")}, 3000);
```

زیر مثالی را مشاهده می کنید که تابع را پس از گذشت هر ثانیه، یکبار اجرا می کند، درست مانند یک ساعت

دیجیتال:

```
var myVar=setInterval(function () {myTimer()}, 1000);
```

```
function myTimer() {  
    var d = new Date();  
    document.getElementById("demo").innerHTML = d.toLocaleTimeString();  
}
```

چگونه می توان اجرای تابع را متوقف کرد؟

به وسیله ی متد () **clearInterval** می توان اجرای تابع مشخص شده در () **setInterval** را متوقف کرد.

نحوه ی تعریف آن:

آدرس آموزشگاه : تهران – خیابان شریعتی – بالاتراز خیابان ملک – جنب بانک صادرات – پلاک – 561 طبقه دوم واحد7
88146330-88446780-88146332

```
window.clearInterval(intervalVariable)
```

متد `clearInterval()` را می توان بدون پیشوند `window` نیز نوشت.

به منظور استفاده از متد `clearInterval()`، باید زمان ایجاد متد `setInterval()`، از یک متغیر سراسری استفاده نمود:

```
myVar=setInterval("javascript function", milliseconds);
```

سپس می توانید با فراخوانی متد `clearInterval()`، اجرا تابع مورد نظر را متوقف کنید:

مثال زیر همان مثال قبل است که یک دکمه ی `"Stop time"` به آن اضافه شده:

```
<p id="demo"></p>
```

```
<button onclick="clearInterval(myVar)">Stop time</button>
```

```
<script>
```

```
var myVar = setInterval(function () {myTimer()}, 1000);  
function myTimer() {  
  var d = new Date();  
  document.getElementById("demo").innerHTML = d.toLocaleTimeString();  
}
```

```
</script>
```

تابع `setTimeout()`

متد `setTimeout()` پس از گذشت مدت زمان مشخص شده برای آن پس از اتمام بارگذاری صفحه، تنها یکبار کد تابع مربوط به خود را اجرا می کند.

ساختار نگارشی این متد به شرح زیر می باشد:

```
window.setTimeout("javascript function", milliseconds);
```

این متد را بدون پیشوند `window` نیز می توان نوشت.

اولین پارامتر ورودی آن باید حتما یک تابع باشد.

پارامتر دوم مشخص می کند پس از گذشت چند میلی ثانیه (از زمان جاری)، پارامتر اولی باید اجرا شود.

مثال:

پس از گذشت 3 ثانیه پیغام "Hello" ظاهر می شود:

```
<button onclick="setTimeout(function(){alert('Hello')},3000)">Try it</button>
```

چگونه می توان اجرای این تابع را متوقف ساخت؟

برای متوقف سازی اجرای تابع مشخص شده در `setTimeout()`، از تابع `clearTimeout()` کمک می گیریم.

نحوی نگارش تابع:

```
window.clearTimeout(timeoutVariable)
```

این متد را می توان بدون واژه `window` نوشت.

جهت استفاده از متد `clearTimeout()`، باید به هنگام ایجاد متد `setTimeout()`، از یک متغیر سراسری استفاده کرد:

```
myVar=setTimeout("javascript function", milliseconds);
```

سپس چنانچه تابع از پیش اجرا نشده باشد، قادر خواهید بود برای توقف اجرای تابع، از متد `setTimeout()` استفاده کنید.

نمونه ی زیر همان مثال بخش قبلی است که یک دکمه ی "Stop" به آن اضافه شده است:

```
<button onclick="myVar=setTimeout(function(){alert('Hello')},3000)">Try it</button>
```

```
<button onclick="clearTimeout(myVar)">Try it</button>
```