

کار با تایمرها یا Timers در جاوااسکریپت | سایت آموزشی فری لرن

کار با تایمرها یا Timer در جاوااسکریپت

سلام خدمت تمامی دوستان و همراهان محترم سایت آموزشی **فری لرن** ، امیدوارم که حال همگیتون خوب باشه. لطفا در ادامه با آموزش کار با تایمرها یا Timers در جاوااسکریپت با من همراه باشید.



Free-Learn

آنچه در این جلسه میخوانید : [مخفی کن]

1 : تایمرها یا Timers در جاوااسکریپت

2 : انواع زمانبندی در جاوااسکریپت

3 : متد setTimeout در جاوااسکریپت

4 : متد setInterval در جاوااسکریپت

5 : نحوه متوقف کردن setTimeout

6 : نحوه متوقف کردن setInterval

7 : مثال های بیشتر

تایمرها یا Timers در جاوااسکریپت

یه چیزی در جاوااسکریپت داریم به اسم Timing Events (در لغت به فارسی میشه زمانبندی رویدادها) ولی خب در کل منظور از این زمانبندی رویدادها، یعنی اجرای یسری دستورات مشخص در یک بازه ی زمانی مشخص.

بفرض مثال ما میخوایم بعد از ۱۰ ثانیه در صفحه یک باکس پیغام نمایش داده بشه یا حالا هر کار دیگه، یا شاید بخوایم کاری کنیم که یه دستور در بازه زمانی مشخص (مثلا هر ۵ ثانیه یکبار) اجرا بشه، پس در این مواقع میتونیم از این زمانبندی ها استفاده کنیم.

Free-Learn

انواع زمانبندی در جاوااسکریپت

ما ۲ نمونه زمانبندی (یا روش کار با بازه زمانی) داریم که این روش ها به شرح زیر می باشند :

- اجرای یک کد پس از بازه زمانی مشخص « setTimeout »
- اجرای یک کد بطور مداوم در یک بازه زمانی مشخص « setInterval »

یعنی اگه خواستیم بگیم مثلا پس از ۱۰ ثانیه فلان کد یا حالا هرچیزی اجرا بشه باید از `setTimeout` استفاده نماییم و اگه بخوایم بطور مداوم مثلا هر ۵ ثانیه یکبار کدی اجرا بشه باید از `setInterval` استفاده نماییم.

Free-Learn

متد setTimeout در جاوااسکریپت

خب گفتیم با استفاده از این متد میتونیم کاری رو بعد از مدت زمان مشخصی اجرا کنیم، پس دیگه بدونید این متد برای اجرای یک تابع پس از یک مدت زمان مشخص می باشد.

- شکل کلی متد `setTimeout` :

```
1 | setTimeout( Function_Name , Time_Wit_Millisecond )
```

- Function_Name : اون کد یا تابعی که قراره اجرا بشه
- Time_Wit_Millisecond : مدت زمان به میلی ثانیه که باید سپری شود تا تابع اجرا شود

هر ۱۰۰۰ میلی ثانیه ای ۱ ثانیه می باشد.

مثال شماره ۱ : بعد از ۳ ثانیه یه پیغام در صفحه ظاهر میشه

```
1 | <script>
2 |
3 | // تابعی که قراره اجرا بشه
4 | function MyMessage(){
5 |     alert("سلام خوبی؟");
6 | }
7 |
8 | // تنظیم زمانبندی با 3 ثانیه
9 | setTimeout(MyMessage, 3000);
10 |
11 | </script>
```

امتحان کنید

مثال شماره ۲ : تغییر مسیر صفحه بعد از ۲ ثانیه به صفحه دیگر

```
1 | <script>
2 |
3 | function MySite(){
4 |     window.location.href = "https://free-learn.ir/";
5 | }
6 |
7 | setTimeout(MySite, 2000);
8 |
9 | </script>
```

امتحان کنید

Free-Learn

متد `setInterval` در جاوااسکریپت

با استفاده از این متد میتونیم یک تابع (تکه کد یا برنامه) رو در مدت زمان مشخصی بطور مداوم اجرا کنیم، مثلا به تابع رو هر ۵ثانیه یکبار در صفحه اجرا کنیم.

- شکل کلی متد `setInterval` :

```
1 | setInterval( Function_Name , Time_Wit_Millisecond )
```

- Function_Name : اون کد یا برنامه یا تابعی که قراره اجرا بشه
- Time_Wit_Millisecond : فاصله زمانی (به میلی ثانیه) بین اجرا شدن هربار تابع

هر ۱۰۰۰ میلی ثانیه ای ۱ ثانیه می باشد.

مثال شماره ۱ : هر ۲ ثانیه یکبار کلمه " فری لرن " در صفحه چاپ میشود

```
1 | <script>
2 |
3 | var Result = document.getElementById("Result");
```

```

4
5 function MyMessage(){
6   Result.innerHTML = Result.innerHTML + "فری لرن" + "<br>";
7 }
8
9 setInterval(MyMessage, 2000);
10
11 </script>

```

امتحان کنید

مثال شماره ۲: ری لود (رفرش شدن) صفحه در هر ۵ ثانیه یکبار بصورت خودکار

```

1 <script>
2
3 function MyReload(){
4   location.reload();
5 }
6
7 setInterval(MyReload, 5000);
8
9 </script>

```

امتحان کنید

مثال شماره ۳: نمایش ساعت ، دقیقه ، ثانیه و میلی ثانیه با استفاده از متدهای تاریخ و ساعت ، و قرار دادن در یک تایمر که ساعت رو در لحظه و زنده بهمون نشون بده

```

1 <script>
2
3 function ShowTime(){
4   ....
5 }
6
7 setInterval(ShowTime, 10);
8
9 </script>

```

امتحان کنید

Free-Learn

نحوه متوقف کردن setTimeout

خب فرض میکنیم ما زمانبندی رو با استفاده از `setTimeout` ایجاد کردیم، حال ممکنه بعدش بخوایم این زمانبندی رو از بین ببریم یا حذفش کنیم، باید چکار کرد؟! به راحتی و با استفاده از متد `clearTimeout` میتونیم تایمر رو Stop یا متوقف کنیم.

شکل کلی متد `clearTimeout()` :

```

1 | clearTimeout( Timer_Variable )

```

- نام اون متغیری که زمانبندی رو داخلش نگهداری میکنه : `Timer_Variable`

مثال : در مثال زیر بعد از ۵ ثانیه یه پیغام نمایش داده خواهد شد، ولی خب میشه با کلیک بروی دکمه از نمایش پیغام جلوگیری کرد (یعنی در واقع متوقف کردن تایمر)

```

1 <button onclick="clearTimeout(MyTimer);">کلیک کنید</button>
2
3 <script>
4
5 function MyMessage(){
6   alert("سلام خوبی؟");
7 }
8
9 var MyTimer = setTimeout( MyMessage, 5000 );
10

```

11 | `</script>`

امتحان کنید

Free-Learn

نحوه متوقف کردن setInterval

با استفاده از این متد نیز میتونیم اجرای متد `setInterval` رو Stop یا متوقف کنیم.

شکل کلی متد `clearInterval()` :

```
1 | clearInterval( Timer_Variable )
```

- نام اون متغیری که زمانبندی رو داخلش نگهداری میکنه : `Timer_Variable`

مثال : در مثال زیر قراره هر ۲ ثانیه یکبار کلمه فری لرن در صفحه چاپ بشه، ولی خب با کلیک با بروی دکمه میخوایم تایمر رو متوقف کنیم.

```
1 | <button onclick="clearInterval(MyTimer);">متوقفش کن</button>
2 |
3 | <script>
4 |
5 | var result = document.getElementById("Result");
6 |
7 | function MyMessage(){
8 |     result.innerHTML = result.innerHTML + " فری لرن " + "<br>";
9 | }
10 |
11 | var MyTimer = setInterval(MyMessage, 2000);
12 |
13 | </script>
```

امتحان کنید

Free-Learn

مثال های بیشتر

مثال شماره ۱ : ایجاد یک تایمر کم شونده (تایمر معکوس یا Countdown Timer)

```
1 | <script>
2 |
3 | var second = 1000;
4 | var minute = second * 60;
5 | var hour = minute * 60;
6 | var day = hour * 24;
7 |
8 | // اون تاریخی که قراره بهش برسیم
9 | let countDown = new Date('Sep 30, 2024 00:00:00').getTime();
10 |
11 | function MyRacing() {
12 |     let nowDate = new Date().getTime(),
13 |     distance = countDown - nowDate;
14 |     document.getElementById('days').innerText = Math.floor(distance / (day)),
15 |     document.getElementById('hours').innerText = Math.floor((distance % (day)) / (hour)),
16 |     document.getElementById('minutes').innerText = Math.floor((distance % (hour)) / (minute)),
17 |     document.getElementById('seconds').innerText = Math.floor((distance % (minute)) / second);
18 |
19 |     // وقتی تاریخ و زمان گذشته بود
20 |     if (distance < 0) {
21 |         clearInterval(MyTimer);
22 |         alert("مسابقه شروع شده است");
23 |     }
24 |
25 | }
26 |
```

```

27 | MyTimer = setInterval( MyRacing , 1000);
28 |
29 | </script>

```

امتحان کنید

مثال شماره ۲ : با کلیک بروی یک دکمه عدد اضافه میشه و با کلیک بروی دکمه دیگه کم میشه

```

1 | function CountUp(){
2 |   totalSeconds++;
3 |   if( totalSeconds < 10 ){
4 |     totalSeconds = "0" + totalSeconds;
5 |   }
6 |   if( totalSeconds > 0 ){
7 |     document.getElementById("CD").disabled = "";
8 |   }
9 |   secondsLabel.innerHTML = totalSeconds;
10 | }

```

امتحان کنید

مثال شماره ۳ : ایجاد یک تایمر اضافه شونده (ثانیه و دقیقه و ساعت)

```

1 | function countUpTimer() {
2 |   var Result = document.getElementById("count_up_timer");
3 |   totalSeconds++;
4 |   var hour = Math.floor(totalSeconds / 3600);
5 |   var minute = Math.floor((totalSeconds - hour * 3600) / 60);
6 |   var seconds = totalSeconds - (hour * 3600 + minute * 60);
7 |   if( hour < 10 ){
8 |     hour = "0" + hour;
9 |   }
10 |   if( minute < 10 ){
11 |     minute = "0" + minute;
12 |   }
13 |   if( seconds < 10 ){
14 |     seconds = "0" + seconds;
15 |   }
16 |   Result.innerHTML = hour + ":" + minute + ":" + seconds;
17 | }

```

امتحان کنید

Free-Learn

این فایل متعلق به وبسایت آموزشی **فری لرن** می باشد و استفاده از آن به هرنحوی برای عموم کاملاً رایگان و آزاد می باشد.

