JavaScript Timing Events

- description : JavaScript Timing Events
- author : 오션
- email : shlim@repia.com
- lastupdate : 2021-04-12

Source of the article

- JavaScript Timing Events

JavaScript는 시간 간격을 두고 실행될 수 있습니다.

이를 타이밍 이벤트(Timing Events)라고 합니다.

Timing Events

window 오브젝트를 사용하면 지정된 시간 간격으로 코드를 실행할 수 있습니다.

이러한 시간 간격을 타이밍 이벤트라고 합니다.

JavaScript와 함께 사용하는 두 가지 주요 메서드는 다음과 같습니다.

- `setTimeout(function, milliseconds)`
  - 지정된 시간 (밀리 초)을 기다린 후 함수를 실행합니다.
- `setInterval(function, milliseconds)`
  - `setTimeout()`과 동일하지만 함수 실행을 계속 반복합니다.

`setTimeout()` 및 `setInterval()`은 모두 HTML DOM Window 오브젝트의 메서드입니다.

The setTimeout() Method

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

`window.setTimeout()` 메서드는 window prefix없이 작성할 수 있습니다.

첫 번째 매개 변수는 실행할 함수입니다.

두 번째 매개 변수는 실행 전 밀리 초 수를 나타냅니다.

예제

버튼을 클릭하십시오. 3 초 동안 기다리면 페이지에 “Hello”라는 window 경고가 표시됩니다.

Click "Try it". Wati 3 seconds, and the page will alert "Hello".

Try it

How to Stop the Execution?

`clearTimeout()` 메서드는 setTimeout()에 지정된 함수의 실행을 중지합니다.

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

`window.clearTimeout()` 메서드는 window prefix 없이 작성할 수 있습니다.

`clearTimeout()` 메서드는 `setTimeout()`에서 반환된 변수를 사용합니다.

```
myVar = setTimeout(function, milliseconds);
clearTimeout(myVar);
```

함수가 아직 실행되지 않은 경우, `clearTimeout()` 메서드를 호출하여 실행을 중지 할 수 있습니다.

예제

위와 같은 예제이지만, “Stop” 버튼이 추가되었습니다.

```
<!DOCTYPE html>
<html>
<body>
    <p>Click "Try it". Wati 3 seconds, and the page will alert "Hello".</p>
    <p>Click "Stop" to prevent the first function to execute.</p>
    <p>(You must click "Stop" before the 3 seconds are up.)</p>
    <button onclick="myVar = setTimeout(myFunction, 3000)">Try it</button>
    <button onclick="clearTimeout(myVar)">Stop it</button>
    <script>
        function myFunction() {
            alert("Hello");
        }
    </script>
</body>
</html>
```

The setInterval() Method

`setInterval()` 메서드는 주어진 시간 간격마다 주어진 함수를 반복합니다.

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

`window.setInterval()` 메서드는 window prefix 없이 작성할 수 있습니다.

첫 번째 매개 변수는 실행할 함수입니다.

두 번째 매개 변수는 각 실행 사이의 시간 간격 길이를 나타냅니다.

이 예제는 디지털 시계처럼 1 초에 한 번씩 “myTimer”라는 함수를 실행합니다.

예제

현재 시간을 표시합니다.

```
<!DOCTYPE html>
<html>
<body>
    <p>A script on this page starts this clock:</p>
    <p id="demo"></p>
    <script>
        var myVar = setInterval(myTimer, 1000);
        function myTimer() {
            var d = new Date();
            document.getElementById("demo").innerHTML =
d.toLocaleTimeString();
        }
    </script>
</body>
</html>
```

1 초에는 1000 밀리 초가 있습니다.

How to Stop the Execution?

`clearInterval()` 메서드는 setInterval() 메서드에서 지정된 함수의 실행을 중지합니다.

```
window.clearInterval(timerVariable)
```

window.clearInterval() 메서드는 window prefix없이 작성할 수 있습니다.

clearInterval() 메서드는 setInterval()에서 반환된 변수를 사용합니다.

```
myVar = setInterval(function, milliseconds);
clearInterval(myVar);
```

예제

위와 같은 예제이지만, “Stop time”버튼을 추가했습니다.

```
<!DOCTYPE html>
<html>
<body>
  <p>A script on this page starts this clock:</p>
  <p id="demo"></p>
  <button onclick="clearInterval(myVar)">Stop time</button>
  <script>
    var myVar = setInterval(myTimer, 1000);
    function myTimer() {
      var d = new Date();
      document.getElementById('demo').innerHTML = d.toLocaleTimeString();
    }
  </script>
</body>
</html>
```

More Examples

Another simple timing

```
<!DOCTYPE html>
<html>
<body>
    <button onclick="timedText()">Try it</button>
    <p id="demo">Click on "Try it". I will display when two, four, and six
seconds have passed.</p>
    <script>
        function timedText() {
            setTimeout(myTimeout1, 2000)
            setTimeout(myTimeout2, 4000)
            setTimeout(myTimeout3, 6000)
        }
        function myTimeout1() {
            document.getElementById('demo').innerHTML = "2 seconds";
        }
```

```
        function myTimeout2() {
            document.getElementById('demo').innerHTML = "4 seconds";
        }
        function myTimeout3() {
            document.getElementById('demo').innerHTML = "6 seconds";
        }
    </script>
</body>
</html>
```

A clock created with a timing event

```
<!DOCTYPE html>
<html>
<head>
    <script>
        function startTime() {
            var today = new Date();
            var h = today.getHours();
            var m = today.getMinutes();
            var s = today.getSeconds();
            m = checkTime(m);
            s = checkTime(s);
            document.getElementById('txt').innerHTML = h + ":" + m + ":" +
s;
            var t = setTimeout(startTime, 500);
        }
        function checkTime(i) {
            if (i < 10) { i = "0" + i };  // add zero in front of numbers <
10
            return i;
        }
    </script>
</head>
<body onload="startTime()">
    <div id="txt"></div>
</body>
</html>
```

오션,, Javascript, Timing, Events

From:
http://125.132.25.164/dokuwiki/ - **문제를 잘 정의하는 것은 문제를 절반 해결한 것이다. - 2023.12**

Permanent link:
http://125.132.25.164/dokuwiki/doku.php?id=wiki:javascript:javascript_note:js_timing&rev=1618215159

Last update: **2022/03/10 19:52**