

Countdown: Mehrere Zähler aktualisieren

== Info ==

Der Code benutzt das Observer Design Pattern.
Die Geschwindigkeit kann manipuliert werden.

== Konfiguration ==

Das Update Interval wird im Konstruktor übergeben. Im Beispiel wird pro Sekunde 1x runtergezählt:

Quellcode

```
1. var up = new UpdateCounters(-1);
```

Optional könnt ihr das Updateintervall in Millisekunden auch als 2ten Parameter angeben. Folgendes Beispiel macht ein Update, alle 2 Sekunden.

Quellcode

```
1. var up = new UpdateCounters(-1, 2000);
```

Jedes Objekt das aktualisiert werden soll, muss registriert werden. Das funktioniert wie folgt:

Quellcode

```
1. <script type="text/javascript">up.push('update1');</script>
```

Das update1 referenziert einen Container mit der ID "update1". Es kann ein Blockelement (DIV), aber auch ein Inlineelement (SPAN) sein.

Quellcode

```
1. <div id="update1">01:02:03</div>
```

== HTML Datei ==

Quellcode

```
1. <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN" "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
2. <html xmlns="http://www.w3.org/1999/xhtml" dir="ltr" xml:lang="de">
3. <head>
4. <title>Mehrere Counter runterzählen</title>
5. <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
6. <script type="text/javascript" src="update-counters.js"></script>
7. <script type="text/javascript">
8. var up = new UpdateCounters(-1);
9. </script>
10. </head>
11. <body onload="up.start();">
12. <h2>Das Update ist auf -1 konfiguriert, es wird also runtergezählt</h2>
13. <div id="update1">01:02:03</div>
14. <script type="text/javascript">up.push('update1');</script>
15. <div id="update2">04:05:06</div>
16. <script type="text/javascript">up.push('update2');</script>
17. <div id="update3">07:08:09</div>
18. <script type="text/javascript">up.push('update3');</script>
```

```

23. <p>
25. <a href="#" onclick="up.fire()">force single update</a> ::
26. <a href="#" onclick="up.start()">start autoupdate</a> ::
27. <a href="#" onclick="up.stop()">disable autoupdate</a>
28. </p>
29. <p>
30. <a href="#" onclick="up.speedUp()">increase speed</a> ::
31. <a href="#" onclick="up.speedDown()">decrease speed</a>
32. </p>
33. </body>
34. </html>

```

Alles anzeigen

== JavaScript Datei ==

Quellcode

```

1. /**
2.  * xxx [...] by http://www.easy-coding.de
3.  *
4.  * @param poll integer update interval in microsecons
5.  */
6. function UpdateCounters(update, poll) {
7.   this.update = update;
8.   this.poll = poll ? poll : 1000;
9.   this.list = [];
10.  this.timer = null;
11.  this.format = /(\d+):(\d+):(\d+)/;
12. /**
14.  * adds elem
15.  * @param id string as id
16.  */
17.  this.push = function(id) {
18.   this.list.push(id);
19.  };
20. /**
22.  * gets seconds from string "h:m:s"
23.  * parseInt is forced to use decimal system
24.  */
25.  this.fromFormat = function(str) {
26.   this.format.exec(str);
27.   return (parseInt(RegExp.$1, 10) * 60 * 60)
28.   + (parseInt(RegExp.$2, 10) * 60)
29.   + (parseInt(RegExp.$3, 10));
30.  }
31. /**
33.  * gets string "h:m:s" from seconds
34.  */
35.  this.toFormat = function(s) {
36.   s = s > 0 ? s : 0;
37.   var h = m = 0;
38.   if(s > 59) {
39.    m = Math.floor(s/60);
40.    s = s - ( m*60 );
41.   }
42.   if(m > 59) {
43.    h = Math.floor(m/60);

```

```

44. m = m - ( h*60 );
45. }
46. if(h < 10) {
47. h = "0"+h;
48. }
49. if(s < 10) {
50. s = "0"+s;
51. }
52. if(m < 10) {
53. m = "0"+m;
54. }
55. return h + ":"+ m + ":"+ s;
56. }
57. /**
58. * sends single request
59. * @param loop boolean
60. */
61. this.fire = function() {
62. var i, dom;
63. for(i=0; i<this.list.length; i++) {
64. dom = document.getElementById(this.list[i]);
65. dom.innerHTML = this.toFormat(this.fromFormat(dom.innerHTML) + this.update);
66. }
67. };
68. /**
69. * stops auto updater
70. */
71. this.stop = function() {
72. window.clearTimeout(this.timer);
73. };
74. this.speedUp = function(factor) {
75. this.stop();
76. this.poll /= factor || 2;
77. this.start();
78. }
79. this.speedDown = function(factor) {
80. this.stop();
81. this.poll *= factor || 2;
82. this.start();
83. }
84. /**
85. * starts auto updater
86. */
87. this.start = function() {
88. this.timer = window.setInterval(function(up) {
89. return function() {
90. up.fire();
91. };
92. }(this), this.poll);
93. };
94. }

```

Alles anzeigen

== Demo ==

demo.easy-coding.de/javascript/update-counters/

Download: demo.easy-coding.de/javascript/update-counters/download.zip