

JavaScript 定时器 10

JS

定时器是什么

JavaScript中系统提供的函数，用来每隔一段时间或者延迟一段时间执行一段指定的代码，分为两种：

- 重复执行定时器 `setInterval`
- 延迟执行定时器 `setTimeout`

定时器的语法

重复执行定时器

```
1. //setInterval('要做的事儿','间隔时间');
2. //每个 间隔时间 做一次 要做的事。
3. // 间隔时间的单位是毫秒， 没有单位 1s == 1000ms
```

需求1：设置一个变量初始为0，让其每隔1秒自增依次，并且打印到控制台中。

```
1. var a = 0;
2.
3. /*setInterval(fn, 1000);
4.
5. function fn(){
6.     a++;
7.     console.log(a);
8. }*/
9.
10. setInterval(function (){
11.     a++;
12.     console.log(a);
```

```
13.     }, 1000);
14.
15.     /*setInterval('fn()', 1000);   // 虽然可以, 不要这么写
16.
17.     function fn() {
18.         a++;
19.         console.log(a);
20.     }*/
```

延迟执行定时器

需求2：页面上一个div，让其在3秒之后背景颜色由红色变为蓝色。

```
1.     var box = document.querySelector('.box');
2.
3.     setTimeout(function() {
4.         box.style.backgroundColor = 'blue';
5.     }, 3000);
```

清除定时器

```
1.     // 清除定时器也分为两种分别是
2.
3.     //clearInterval();
4.     //clearTimeout();
5.
6.     // 它们都有一个共同的特性, 就是脾气特别好, 给我什么都可以。
7.     // 如果参数是某个定时器的编号, 那么就会清除对应的定时器
8.     // 由于定时器的编号在跨浏览器的时候会有兼容性问题, 所以通常使用一个变量来存储定时器的返回值
```

需求3：设置一个变量初始为0，让其每隔1秒自增依次，并且打印到控制台中，当变量增加到5的时候停止定时器。

```
1.     var a = 0;
2.
3.     var timer = setInterval(function() {
4.         console.log(a++);
```

```
5.     if(a == 5){
6.         clearInterval(timer);
7.     }
8.     }, 1000);
```

需求4：延时菜单，使用以下结构，当移入按钮，让div显示，此时鼠标从按钮移开到div身上，div可以继续显示，当移开div之后，div消失，如果移开按钮的时候没有移入到div身上，div消失。

```
1.     <!DOCTYPE html>
2.     <html lang="en">
3.     <head>
4.         <meta charset="UTF-8">
5.         <title>延时菜单</title>
6.         <style>
7.             .box {
8.                 width: 100px;
9.                 height: 100px;
10.                background: red;
11.                display: none;
12.            }
13.        </style>
14.    </head>
15.    <body>
16.        <input type="button" value="按钮" class="btn">
17.        <div class="box"></div>
18.    </body>
19. </html>
```

关于定时器的最小时间间隔

定时器的时间间隔并不精确，不同的浏览器都有各自的最小间隔，最新的HTML5的规范规定了最小间隔为4ms。

获取元素计算后样式的数据

`getComputedStyle(元素).属性`

注意：无法获取 `transform` 的样式

封装CSS函数 (初步)

```
1.  function css(){
2.      var len = arguments.length;
3.      if(len < 2){
4.          return false;
5.      }
6.      if(len == 2){
7.          return Number.parseFloat(getComputedStyle(arguments[0])[arguments[1]]);
8.      }
9.      if(len == 3){
10.         arguments[0].style[arguments[1]] = arguments[2];
11.      }
12.  }
```

需求5：点击按钮让div向右以 `1px/20ms` 的速度运动。

```
1.  <!DOCTYPE html>
2.  <html lang="en">
3.  <head>
4.      <meta charset="UTF-8">
5.      <meta name="viewport" content="width=device-width, initial-
6.  scale=1.0">
7.      <meta http-equiv="X-UA-Compatible" content="ie=edge">
8.      <title>Document</title>
9.      <style>
10.         .box {
11.             width: 100px;
12.             height: 100px;
13.             position: relative;
14.             background: red;
15.             left: 0;
16.             top: 10px;
17.         }
18.     </style>
19. </head>
20. <body>
```

```
20. <input type="button" value="按钮" class="btn">
21. <div class="box"></div>
22. <script>
23.     var btn = document.querySelector('.btn');
24.     var box = document.querySelector('.box');
25.
26.     var t = null;
27.
28.     btn.onclick = function () {
29.         // 让box向右运动
30.         clearInterval(t);
31.         t = setInterval(function() {
32.             box.style.left = css(box, 'left') + 1 + 'px';
33.         }, 20);
34.         console.log(t);
35.     };
36. </script>
37. </body>
38. </html>
```