

# JavaScript 基础入门 06

JS

## 回顾 window

window 是JS中全局对象，凡是在全局中声明的变量，或者函数，最终都会成为它的属性（方法）。

- 全局：在JS中在script内，并且在函数外都算全局的范围。
- 对象：在现实中，对象是一个抽象的概念，可以指人或者事物，在JS中可以指标签元素等，都可以算是对象。
- 属性：一般都是名词，用来描述元素的特性的。
- 方法：在JS中就是指函数。

## 回顾for循环

对for循环使用的一些补充：

```
1.  <script>
2.    /*var i;
3.    for(i=0; i<3; i++){
4.        console.log(i);  // 0 1 2
5.    }*/
6.    // 循环完成后i的结果
7.    // console.log(i); // 3
8.
9.    /*for(var i=0; i<3; i++){
10.    // 相当于调用了3次函数 每次弹出i的值
11.        fn1();  // 0 1 2
12.    }
13.
14.    function fn1() {
15.        alert(i);
16.    }
```

```

17. // 在这调用函数 循环已经结束, 此时i的值是3
18. fn1(); //3*/
19.
20. /*for(var i=0; i<3; i++){
21.     function fn1() {
22.         alert(i); // 3
23.     }
24. }
25. // 上面这种写法, 相当于我特别傻的写了3次重复的下面的函数
26. function fn1() {
27.     alert(i);
28. }
29. function fn1() {
30.     alert(i);
31. }
32. function fn1() {
33.     alert(i);
34. }
35. // 在这调用函数 for循环已经结束此时i是3
36. fn1();*/
37.
38. for(var i=0; i<3; i++){
39. // 相当于用重复了3次下面的代码, 但是这个函数并没有触发
40.     document.onclick = function() {
41.         alert(i);
42.     }
43. }
44.
45. console.log(1); // 1
46.
47. /*document.onclick = function() {
48.     alert(i);
49. }
50.
51. document.onclick = function() {
52.     alert(i);
53. }
54.
55. document.onclick = function() {
56.     alert(i);
57. }*/
58. </script>

```

需求1: class为list的ul下面有5个li, 每个li里面的内容分别是0、1、2、3、4, 点击每个

li弹出其对应的内容。

```
1. <body>
2. <ul class="list">
3.     <li>0</li>
4.     <li>1</li>
5.     <li>2</li>
6.     <li>3</li>
7.     <li>4</li>
8. </ul>
9. <script>
10.     var lis = document.querySelectorAll('.list li');
11.
12.     console.log(lis);
13.
14.     /*lis[0].onclick = function () {
15.         alert(lis[0].innerHTML);
16.     };
17.     lis[1].onclick = function () {
18.         alert(lis[1].innerHTML);
19.     };
20.     lis[2].onclick = function () {
21.         alert(lis[2].innerHTML);
22.     };
23.     lis[3].onclick = function () {
24.         alert(lis[3].innerHTML);
25.     };
26.     lis[4].onclick = function () {
27.         alert(lis[4].innerHTML);
28.     };*/
29.
30.     for(var i=0; i<lis.length; i++){
31.         lis[i].onclick = function () {
32.             //alert(i)
33.             //因为这里的i是循环完成i的值
34.             alert(lis[/*填什么呢?*/].innerHTML);
35.         }
36.     }
37. </script>
```

## 初识 this

this 是JS中的一个关键字，是预先定义好的一个变量，并且这个变量储存的永远是一个对象，这个变量this储存的对象会随着不同的环境改变，也就是说它不是定值。这个this只能读，不能写。

## 关于this指向：(初步)

- 在全局下 this 就代表 window
- 对象.函数 函数里面的this就是这个对象。
- 元素.事件 = 函数，此时函数内的this就是这个元素对象。

```
1. // 全局
2. alert(this);
3. var a = 1;
4. alert(this.a)
```

```
1. // 通过.的方式调用函数
2. function fn(){
3.     alert(this); // window
4. }
5. window.fn()
```

```
1. // 事件调用
2. var btn = document.getElementById('btn');
3. btn.onclick = function (){
4.     alert(this); // btn
5. };
```

## 对需求1进行完善：

```
1. <script>
2.     var lis = document.querySelectorAll('.list li');
3.     for(var i=0; i<lis.length; i++){
4.         lis[i].onclick = function (){
5.             console.log(this);
6.             alert(this.innerHTML);
7.         }
8.     }
9. </script>
```

需求2：现有5个50\*50的span，默认没有背景颜色，只有边框，点击其中一个span，当前被点击的这个背景颜色变为半透明的粉色，其它不做改变，变成粉色后再点击本身不做改变。再次点击另外一个，另外一个背景颜色变成半透明的粉色。其它的都变回没有颜色。

```
1.     <style>
2.         span {
3.             display: inline-block;
4.             width: 50px;
5.             height: 50px;
6.             border: 1px solid #000;
7.             margin: 5px;
8.         }
9.         .active {
10.             background: rgba(245, 61, 182, 0.64);
11.         }
12.     </style>
13.     <script>
14.         window.onload = function () {
15.             var spans = document.querySelectorAll('span');
16.
17.             /**
18.              * 第一种思路：点击当前的span给当前的span添加class
19.              * 在添加class之前 先把所有的span的class清空
20.              *
21.              * 第二种思路：定义一个上一个，点击当前的时候
22.              * 先清空上一个，然后给当前的添加class，然后把
23.              * 当前的变成上一个。
24.              */
25.
26.             // 思路1
27.             /*for(var i=0; i<spans.length; i++){
28.                 spans[i].onclick = function () {
29.                     // 清空一组span的class
30.                     for(var i=0; i<spans.length; i++){
31.                         spans[i].className = '';
32.                     }
33.                     // 给当前的点击的span添加class
34.                     this.className = 'active';
35.                 }
36.             }*/
37.
38.             // 思路2
```

```

39.         var prev = spans[0];
40.
41.         for(var i=0; i<spans.length; i++){
42.             spans[i].onclick = function () {
43.                 prev.className = '';
44.                 this.className = 'active';
45.                 prev = this;
46.             };
47.         }
48.
49.         // for(var i=0; i<3; i++){
50.         //     for(var i=0; i<3; i++){
51.         //         console.log(i);
52.         //     }
53.         // }
54.         // console.log(i);
55.     };
56. </script>
57. </head>
58. <body>
59.     <span></span>
60.     <span></span>
61.     <span></span>
62.     <span></span>
63.     <span></span>
64. </body>

```

需求3：生成10\*10方阵，拼成一张完整的图片，默认每个方块透明度为0.1，鼠标移入，让当前方块透明度变为1，同时放大1.2倍，移出后恢复为1倍，透明度还为1。

```

1. <!DOCTYPE html>
2. <html lang="en">
3. <head>
4.     <meta charset="UTF-8">
5.     <meta name="viewport" content="width=device-width, initial-
scale=1.0">
6.     <meta http-equiv="X-UA-Compatible" content="ie=edge">
7.     <title>Document</title>
8.     <style>
9.         .box {
10.             position: relative;
11.         }
12.         .box div {

```

```
13.     position: absolute;
14.     left: 0;
15.     top: 0;
16.     width: 50px;
17.     height: 50px;
18.     background-size: 500px 500px;
19.     background: url(pic.jpg) no-repeat;
20.     opacity: 0.3;
21.     transition: 500ms;
22. }
23. </style>
24. <script>
25.     window.onload = function () {
26.         var box = document.querySelector('.box');
27.
28.         var str = '';
29.         var t = -1;
30.         for(var i=0; i<100; i++){
31.             if(i%10 == 0) {
32.                 t++;
33.             }
34.             str += '<div style="left: '+(i%10 * 51)+'px; top: '+(t*51)+'px; bac
kground-position: '+( -i%10 * 50)+'px '+( -t*50)+'px;"></div>'
35.         }
36.
37.         box.innerHTML = str;
38.
39.         var divs = document.querySelectorAll('.box div');
40.
41.         /**
42.          * 移入每一个div 让当前移入的这个 透明度变为1 同时放大1.2倍
43.          * 移出当前移入的这个div 让它变回1倍
44.          */
45.
46.         for(var i=0; i<divs.length; i++){
47.             divs[i].onmouseover = function () {
48.                 this.style.opacity = 1;
49.                 this.style.transform = 'scale(1.2)';
50.             };
51.             divs[i].onmouseout = function () {
52.                 this.style.transform = 'scale(1)';
53.             };
54.         }
55.     };
56. </script>
```

```
57.     </head>
58.     <body>
59.         <div class="box"></div>
60.     </body>
61. </html>
```

## 从数组到对象

想找到数组中的元素，需要通过对应的索引值（0、1、2），这个索引值实质上就是这个元素对应的key。

0	1	2	3
'a'	'b'	'c'	'd'

对象和数组非常类似，但是可以自己定义每个数据对应的名字（key/属性名）。

### 创建自定义对象

```
var obj = {key:value};
```

```
1.  var obj = {
2.     a: 'a',
3.     b: 'b',
4.     c: 'c',
5.     d: 'd'
6. };
```

对象里面的数据包括 属性的名字（key），和属性对应的值（value），对象里面可以储存任意类型的数据，但是每个数据都必须对应一个名字。

### 给对象添加属性和方法

1. 在创建的时候直接写到对象里面

```
1.  var obj2 = {
```



```

2.     arr:[1,2,3,4],
3.     str:'abc',
4.     num:10,
5.     o:{},
6.     fn:function (){ // 对象里面写函数 需要这么写，不用写名字了。
7.
8.     }
9. }

```

1. 通过 对象.属性（方法）的方式给对象添加属性或者方法。

## 使用对象中的数据

通过 对象.属性名（方法名）

```

1.     var obj = {};
2.
3.     // 给obj添加一个属性a，对应的值是1
4.     obj.a = 1;
5.
6.     // 给obj添加一个方法
7.     obj.fn = function (){
8.         alert(1);
9.     }
10.
11.     //alert(obj.fn);

```

## 元素的自定义属性

从需求开始：

需求4：点击一个50\*50的span，默认只有边框，没有背景颜色的，点击的时候添加背景颜色，再次点击取消背景颜色。

```

1.     <head>
2.         <meta charset="UTF-8">
3.         <meta name="viewport" content="width=device-width, initial-
         scale=1.0">

```

```

4.     <meta http-equiv="X-UA-Compatible" content="ie=edge">
5.     <title>Document</title>
6.     <style>
7.         span {
8.             display: block;
9.             width: 50px;
10.            height: 50px;
11.            border: 1px solid #000;
12.            margin: 5px;
13.        }
14.        .active {
15.            background: rgb(250, 191, 15);
16.        }
17.    </style>
18. </head>
19. <body>
20.     <span class=""></span>
21.     <script>
22.         var span = document.querySelector('span');
23.         var flag = true;
24.         span.onclick = function () {
25.             if(flag) {
26.                 this.className = 'active';
27.             } else {
28.                 this.className = '';
29.             }
30.             flag = !flag;
31.         };
32.     </script>
33. </body>

```

需求5：class为box的div，里面有5个span，默认只有边框，没有背景颜色的，点击某一个span的时候添加背景颜色，再次点击添加过背景颜色的取消背景颜色。

1.

需求6：在需求2的基础上，添加点击被选中的可以取消选中，点击其它未被选中的，已经被选中的取消被选中的（去掉背景颜色）。

1. <body>

```

2.     <span></span>
3.     <span></span>
4.     <span></span>
5.     <span></span>
6.     <span></span>
7.     <script>
8.         // 获取到5个spans
9.         var spans = document.querySelectorAll('span'); // {flag:true}
10.
11.         // 定义上一个
12.         var prev = spans[0];
13.
14.         // 循环所有的span
15.         for(var i=0; i<spans.length; i++){
16.             // 给每个span添加自定义属性 flag 默认为true
17.             spans[i].flag = true;
18.             // 给每个span添加点击事件
19.             spans[i].onclick = function (){
20.                 // 如果当前点击的这个span身上的自定义flag 为true就执行
21.                 if(this.flag){
22.                     // 先清空上一个的className
23.                     prev.className = '';
24.                     // 由于上一个className被清空了，但是实际上没有className的时候
25.                     // 上一个的自定义属性flag对应的应该是个true，
26.                     // 所有要给它变回true
27.                     prev.flag = true;
28.                     // 给当前的这个点击的span添加一个className active
29.                     this.className = 'active';
30.                     // 让当前点击的这个span 变成上一个
31.                     // 以便下次点击的时候 可以找到正确的上一个
32.                     prev = this;
33.                 }else{
34.                     // 当这个span有className的时候，再次点击 由于 this.flag 变为了fals
35.                     e
36.                     // 那么就执行清空className的操作
37.                     this.className = '';
38.                 }
39.                 // 切换它的这个 开关
40.                 this.flag = !this.flag;
41.             }
42.         }
43.     </script>
</body>

```

# 自定义属性的应用

## 选项卡

```
1.  <!DOCTYPE html>
2.  <html lang="en">
3.  <head>
4.      <meta charset="UTF-8">
5.      <meta name="viewport" content="width=device-width, initial-
scale=1.0">
6.      <meta http-equiv="X-UA-Compatible" content="ie=edge">
7.      <title>Document</title>
8.      <style>
9.          button {
10.              width: 70px;
11.              height: 40px;
12.          }
13.          .active {
14.              background: rgba(58, 219, 65, 0.65);
15.          }
16.          div {
17.              width: 220px;
18.              height: 220px;
19.              font: 49px/220px arial;
20.              text-align: center;
21.              border: 2px solid #000;
22.              margin-top: 5px;
23.              display: none;
24.          }
25.          .show {
26.              display: block;
27.          }
28.      </style>
29. </head>
30. <body>
31.     <button class="active">1</button>
32.     <button>2</button>
33.     <button>3</button>
34.     <div class="show">1111</div>
35.     <div>2222</div>
36.     <div>3333</div>
37.     <script>
38.         /**
```

```

39.      * 需求：选项卡
40.      * 当鼠标移入对应的按钮的时候，让对应的div显示出来，然后不是对应的div隐藏
41.      */
42.      var btns = document.querySelectorAll('button');
43.      var divs = document.querySelectorAll('div');
44.
45.      /**
46.       * btns
47.       * 0      btns[0]
48.       * 1      btns[1]
49.       * 2      btns[2]
50.       */
51.
52.      /**
53.       * divs
54.       * 0      divs[0]
55.       * 1      divs[1]
56.       * 2      divs[2]
57.       */
58.
59.      // 1 先做鼠标移入按钮 移入的按钮 有背景颜色 其它没有
60.      // 2 将所有的btns 和 div 对应起来
61.
62.      for(var i=0; i<btns.length; i++){
63.          // 给每个btns添加一个自定义的属性 叫 index，每次循环正好等于当前这个i的
值
64.          btns[i].index = i; // 自定义索引值
65.          // 给每个btns添加鼠标移入事件
66.          btns[i].onmouseover = function (){
67.              //console.log(this.index);
68.              // 先清空所有btns的className，以及让所有的div都隐藏（清空所有div的className）
69.              for(var i=0; i<btns.length; i++){
70.                  btns[i].className = '';
71.                  divs[i].className = '';
72.              }
73.              //给当前的按钮添加背景 也就是 className为active
74.              this.className = 'active';
75.              //给当前这个按钮对应的这个div添加一个用来显示出来的className为show
76.              divs[this.index].className = 'show';
77.          };
78.      }
79.      </script>
80. </body>
81. </html>

```

```
1.  <!DOCTYPE html>
2.  <html lang="en">
3.  <head>
4.      <meta charset="UTF-8">
5.      <meta name="viewport" content="width=device-width, initial-
scale=1.0">
6.      <meta http-equiv="X-UA-Compatible" content="ie=edge">
7.      <title>Document</title>
8.      <style>
9.          button {
10.              width: 70px;
11.              height: 40px;
12.          }
13.          .active {
14.              background: rgba(58, 219, 65, 0.65);
15.          }
16.          div {
17.              width: 220px;
18.              height: 220px;
19.              font: 49px/220px arial;
20.              text-align: center;
21.              border: 2px solid #000;
22.              margin-top: 5px;
23.              display: none;
24.          }
25.          .show {
26.              display: block;
27.          }
28.      </style>
29.  </head>
30.  <body>
31.      <button class="active">1</button>
32.      <button>2</button>
33.      <button>3</button>
34.      <div class="show">1111</div>
35.      <div>2222</div>
36.      <div>3333</div>
37.      <script>
38.          /**
39.           * 需求：选项卡
40.           * 当鼠标移入对应的按钮的时候，让对应的div显示出来，然后不是对应的div隐藏
41.           */
42.          var btns = document.querySelectorAll('button');
43.          var divs = document.querySelectorAll('div');
44.
```

```
45.      // 清空上一个
46.
47.      var prevBtn = btns[0];
48.      var prevDiv = divs[0];
49.
50.      for(var i=0; i<btns.length; i++){
51.          /**
52.           * 什么时候用自定义索引值：
53.           * 当一组元素去控制另外一组元素的时候，首先想到自定义属性（索引值）
54.           * 给控制的一组元素添加索引值，而不是被控制的。
55.           */
56.          // 给每个按钮添加一个自定义索引值 0 1 2
57.          btns[i].index = i;
58.          btns[i].onmouseover = function (){
59.              // 清空上一个（第一个，默认显示的那个）的className
60.              prevBtn.className = '';
61.              // 让上一个（默认显示的div）的className 清空
62.              prevDiv.className = '';
63.              // 给当前的按钮添加className
64.              this.className = 'active';
65.              // 让当前这个按钮对应的div显示出来（添加className为show）
66.              divs[this.index].className = 'show';
67.              // 更新上一个
68.              prevBtn = this;
69.              prevDiv = divs[this.index];
70.          };
71.      }
72.      </script>
73. </body>
74. </html>
```