

一、案例练习

二、函数

函数：能实现某个特定功能的代码块

(可以是一段代码，一段处理逻辑)

特点：封装

函数作用域

函数传递参数

```
1
2 function foo(index){ console.log(index) ;// undefined }
3 foo();//没有传递实参
4
5 function foo(index){ console.log(index) ;// 9 } //形参
6 foo(9)// 实参
```

函数返回值

return : 返回值

```
1 // 1.0 定义函数
2 function foo(){
3     //返回值
4     return 1111;
5 }
6
7 // 2.0 调用函数
8 var res = foo();
9 console.log(res);// 1111
10
11 备注：如果没有return语句，调用函数没有返回值，结果是undefined
12
13
14 // 思考：
```

```
15 // 调用 ce 函数 ,
16 // 返回一个宽度100px 、高度 100px, 背景色red 、外边距10px的div标签
```

什么情况下有返回值

```
1 <script>
2     // 1.0 定义函数
3     function foo(){
4         "aaa"
5     }
6     // 2.0 调用foo函数
7     var res = foo();
8     console.log(res);// undefined
9     // 同学们回答:
10         // 赌他有值          错误
11         // 有吧                错误
12         // undefined          正确
13         // 有, undefined
14         // 没有                正确
15         // undefined          正确
16         // undefined 原因: 没有声明?
17     // 1.0 定义函数
18     function foo2(){
19         // 返回值
20         return "aaa";
21     }
22     // 2.0 调用foo函数
23     var res2 = foo2();
24     console.log(res2);// aaa
```

return语句 有哪些作用

```
1 <p>返回值的作用（函数）</p>
2 <p>终止代码</p>
3 <p>在函数作用域的循环语句 跳出循环的作用</p>
4 <div>事件是函数，但是函数不一定是事件。事件作用域中的return 是没有返回值。是函数体</div>
5 <button>按钮</button>
6 <script>
7 // 1.0 定义函数
8 // 终止代码的作用
```

```
9 function foo(){
10     console.log("1");
11     return ;
12     console.log("2");
13 }
14 // 2.0 调用函数
15 foo()
16
17 // 3.0 定义函数
18 // 跳出循环的作用
19 function foo2(){
20     for(var i = 0 ; i < 5 ;i ++){
21         if(i == 3){
22             // 终止代码
23             return ;
24         }
25         console.log("i:",i);
26     }
27 }
28 // 4.0 调用函数
29 foo2();
30
31 // 5.0 定义函数
32 // 获取元素
33 var btn = document.getElementsByTagName("button")[0]
34 // 绑定事件
35 var res = btn.onclick = function(){
36     console.log("1111")
37     // 没有返回值，仅仅是终止代码的作用
38     return "abcd";
39     console.log("2222")
40 }
41 console.log(res);// 函数体
42 // 事件是函数，但不是所有函数都是事件。
43
44
45
46 // 定义函数 foo5
47 function foo5(){
48     // 没有返回值
```

```
49  }  
50  // 调用函数 foo5  
51  var res = foo5();  
52  console.log(res);// undefined  
53  // 函数不一定有返回值!  
54  </script>
```