

JavaScript 函数进阶 08

JS

函数的声明(完整)

```
1.  function sum(a, b) {  
2.      var c = a + b;  
3.      return c;  
4.  }
```

- 关键字：function
- 函数名字：例子中的 sum
- () 和 () 里面的参数，例子中的 a, b。一个函数可以有 0 个或者多个参数，每个参数之间用 "," 隔开。
- {}：利用用来写函数要执行的功能（代码）。
- return 子句：用来指定函数的返回结果（返回值），每个函数都有返回值，如果不指定函数的返回值，那么默认返回值就是 undefined。

函数参数

- 形参：就是函数定义时候的参数
- 实参：就是函数执行时候传入的参数

函数的形参和实参不一定是——对应的，在调用函数的时候可以传入任意多个实参。

函数的属性：

- fn.length: 代表函数形参的个数
- arguments: 函数内部的一个属性，它的类型是一个类数组，它的作用就是用来保存实参的集合。
 - length：就代表实参的个数。

关于arguments

函数之所以可以传入任意多个实参，实际上最终保存的地方都在这个类数组中。

函数的返回值

return的作用：

- 用来指定函数的返回值，return后面是什么，那么函数执行的结果就是什么。
- return 后面的代码不会再继续执行，也就说它有终止函数执行的作用。

练习：

需求1：页面上有3个计数器，分别实现点击-文本框内的数字-1；点击+，文本框内的数字+1。其中数字最小为0，最大为4。

```
1. <div class="item">
2.   <button>-</button>
3.   <input type="text" value="0">
4.   <button>+</button>
5. </div>
6. <div class="item">
7.   <button>-</button>
8.   <input type="text" value="0">
9.   <button>+</button>
10. </div>
11. <div class="item">
12.   <button>-</button>
13.   <input type="text" value="0">
14.   <button>+</button>
15. </div>
```

需求2：封装一个函数，参数为一个数组和一个布尔值，如果是true，那么就是求最大值，否则就是求最小值。

```
1. function result(arr, type){
```

```

2.     var n;
3.     if(type){
4.         n = -Infinity;
5.     }else{
6.         n = Infinity;
7.     }
8.     for(var i=0; i<arr.length; i++){
9.         if(type){
10.            if(n < arr[i]){
11.                n = arr[i];
12.            }
13.        }else{
14.            if(n > arr[i]){
15.                n = arr[i];
16.            }
17.        }
18.    }
19.    return n;
20. }
21.
22. var res = result([1,2,3,4]);
23.
24. console.log(res);

```

需求3：封装一个求和函数

```

1.     function sum(){
2.         var res = 0;
3.         for(var i=0; i<arguments.length; i++){
4.             res += arguments[i];
5.         }
6.         return res;
7.     }
8.
9.     console.log(sum(1,2,3,4,3434,657,34,65,234,65));

```

函数表达式

理解函数也是一种数据，可以向其它数据一样存放在变量中。

```
1.  var fn = function (a,b) {  
2.      var c = a + b;  
3.      return c;  
4.  };  
5.  
6.  var res = fn(1,2);  
7.  console.log(res);
```

函数表达式和函数声明的区别

函数声明的函数，可以在任意地方去调用。而函数表达式，只能在声明函数之后去调用这个函数。

1.

fuqiang@miaov.com