

函数6-30

2017年6月30日 15:35

```
function handle(){  
    alert(2)  
}
```

btn.onclick = handle; //函数调用正确姿势，不要写括号

实参：

实参的个数不固定，不确定的时候

使用函数内部内置的变量：

arguments 是函数实参的集合 类数组

和数组有一些相同的特性，不具备数组的全部的特性

相同特性：

结构 []

length

下标

不相同：

类数组没有数组的方法

函数的返回值

每一个函数被调用后都有一个返回值

1. 默认为undefined

2. 使用关键字return

只能在函数中使用，函数调用后返回return后面的值

如果return后没有值，函数返回值为undefined

return之后的代码停止执行

```
function add(a,b){           //参数相加  
    var n = 0;  
    for( var i = 0; i < arguments.length;  
i++ ){  
        n += arguments[i]  
    }  
    return n;  
}  
var m = add(1,2); // undefined  
console.log(m);
```

如果return后没有值，函数返回值为undefined
return之后的代码停止执行

获取计算后的样式

全局函数：

`getComputedStyle()`

参数：

是要获取计算后样式的元素

`getComputedStyle(元素)`

函数会返回一个对象

对象是这个元素所有计算后的样式

// 设置某一个元素指定的样式的值

```
function setStyle(element,prop,value){  
    element.style[prop] = value  
}  
  
setStyle(box,"width", '900px')
```

```
function css(element,prop){  
    return  
    parseFloat(getComputedStyle(element)[prop]);  
}  
console.log(css(box,"width"));  
封装一个函数，用来获取某一个元素指定的样式
```

// css 可以设置也可以获取指定元素的样式

```
function css(element,prop,value){  
    // 判断arguments的length  
    if(arguments.length>2){  
        // 设置  
        element.style[prop] = value;  
    }else{  
        // 获取  
        return parseFloat(getComputedStyle(element)[prop]);  
    }  
}
```

例子：点击div宽度加一百

```
<script>  
var btn=document.getElementsByTagName('button');  
var box=document.querySelector('.box')  
  
btn[0].onclick=function(){  
    var w=css(box,'width') ;  
    css(box,'width',(w+100)+'px');  
}
```

```
function css(element,prop,value){  
  if(arguments.length>2){  
    element.style[prop]=value;  
  }else{  
    return parseFloat(getComputedStyle(element)[prop])  
  }  
}  
</script>
```