

# 函数

2017年6月29日

17:20

## 函数

可存放多行代码，形成代码块

是一个对象，可执行的对象 可反复使用

可以作为值使用

使用关键字 `function`

语法：

```
function [函数名]() {  
  
}
```

完整写法：

```
function [函数名]([参数1, 参数2, . . . . .]) {  
    // 函数体 代码块  
    [return 值]  
}
```

函数定义：

函数声明

```
function 函数名() {  
  
}
```

注意：函数声明必须带上函数名

函数表达式

当在js中使用一个单词的时候

1. 当成变量

2. js语法规定的一些直接可以使

用的单词

必须写成字符串

select选中文本框中的值

```
btn.onclick = function () {  
    message.select();  
}
```

把函数作为值使用，赋值为变量

```
var fn = function (){  
}
```

fn为函数名

函数定义以后不会执行函数中的代码，必须调用

## 函数调用

函数定义后，不会执行函数中的代码，只能调用才能执行

函数名()

调用：函数名()

调用位置：

函数声明调用：

可以在函数声明之前和之后都可以调用

函数表达式调用：

必须在赋值之后调用

## 函数参数

在函数共用的时候，在调用函数的时候要执行不同的值，通过传参的形式解决

形参

```
function func(miaov,abc){  
    console.log(miaov,abc);  
}  
func(1,2);
```

## 形参

函数定义的时候，写在小括号中

```
function func([形参1,形参2]){
```

```
}
```

形参的命名规则，要符合变量的命名规则

形参就是在函数的内部声明的变量,默认值为undefined

形参只能在函数中使用

```
        console.log(miaov,abc);  
    }  
func(1,2);
```

## 实参

在调用函数时，写在小括号中

会把这个实参赋值给形参

形参和实参是一一对应的，以逗号隔开