

```
<!DOCTYPE html>
```

```
<html>
```

```
    <head>
```

```
        <meta charset="utf-8">
```

```
        <title>js的函数学习</title>
```

```
    <!--
```

js的函数学习:

作用: 功能代码块的封装, 减少代码的冗余

### 1. 函数的声明

方法一: function 函数名(形

参名1...) {执行体}

方式二: var 函数名=new

Function(" 形参1 "... "函数执行体")

注意: 此

声明表明在js中函数是作为对象存在的。

方式三: var 函数名

=function (形参1...) {执行体}

### 2. 函数的参数

js中函数在调用时, 形参可以不赋值, 不会报错;

js中函数在调用时, 形参可以不全赋值, 不会报错,

但是实参会依次进行赋值

### 3. 函数的返回值

在js中如果函数有返回值则直

接返回, 没有的话默认返回undefined

注意:

js的代码声明区和执行区是一起的, 都是在js代码的代码域

中

```
-->
```

```
<!--声明js的代码域-->
```

```
<script type="text/javascript">
```

```
    //1. 声明函数
```

```
    //方法一:
```

```
    function test1(a1, a2) {
```

```
        alert("函数声明");
    }
    //方法二:
    var test02 = new Function("a1", "a2", "alert('函数声明
二')");

    //方式三:
    var test03=function(a1,a2){
        alert("函数声明三");
    }
    // 函数的调用
    test1(1, 2);
    test1="abc";
    alert(test1);
    test02(1, 2);
    test03(1,2);

    //2. 函数的参数
    function testParam(a1,a2){
        alert("函数的形参学习");
    }
    testParam();

    //3. 函数的返回值
    var testReturn=function(){
        alert("函数的返回值学习");
        return "js";
    }
    alert(testReturn());
</script>
</head>

<body>
</body>

</html>
```

[http://127.0.0.1:8020/js/10-js-](http://127.0.0.1:8020/js/10-js-is%E7%9A%84%E5%87%BD%E6%95%B0%E5%AD%A6%E4%B9%A0.html)  
[is%E7%9A%84%E5%87%BD%E6%95%B0%E5%AD%A6%E4%B9%A0.html](http://127.0.0.1:8020/js/10-js-is%E7%9A%84%E5%87%BD%E6%95%B0%E5%AD%A6%E4%B9%A0.html)