

函数 Function

定义函数

函数声明

```
function sum(a, b) {  
  return a+b;  
}
```

函数表达式

```
var sum=function(a, b) {  
  return a+b;  
}
```

匿名函数

构造函数

```
var sum=new Function("a","b","return a+b");  
不推荐
```

声明提升

函数声明提升

函数声明语句将会被提升到外部脚本或者外部函数作用域的顶部，因此可以把函数声明放在调用它的语句后面。

```
f('superman');  
function f(name) {  
  console.log(name);  
} //superman
```

函数表达式则会报错

```
f('superman');  
var f= function(name) {  
  console.log(name);  
} //error
```

变量声明提升

[javascript](#)函数里的所有声明（只是声明，但不涉及赋值）都被提前到函数体的顶部，而变量赋值操作留在原来的位置。

```
var scope = 'global';  
function f() {  
  console.log(scope);  
  var scope = 'local';  
  console.log(scope);  
}
```

相当于：

```
var scope = 'global';  
function f() {  
  var scope;    //变量声明提升到函数顶部  
  console.log(scope);  
  scope = 'local'; //变量初始化依然保留在原来的位置
```

```
    console.log(scope);  
}
```

函数内部特殊对象

arguments 类数组对象，包含传入函数中的所有参数

属性：

length 传入参数的个数

callee 一个指针，指向拥有这个arguments对象的函数。常用于递归函数，消除递归函数执行时与函数名的紧密耦合现象

this 引用函数执行的环境对象

若该函数是绑定到某个对象上的方法：**this**指向该对象

若该函数在全局作用域中：**this**指向全局对象window

属性

caller 指向调用当前函数的函数

length 函数希望接收的参数个数

prototype 指向函数原型

方法

apply(函数运行的作用域(this指向)，参数数组(Array实例或arguments)) 在特定的作用域中调用函数

call(函数运行的作用域(this指向)，参数1，参数2，.....) 在特定的作用域中调用函数

bind(对象) 创建一个函数的实例，该实例的this值会被绑定到指定的对象上