函数

function abs(x){

if(x>=0){return x;}

else{return –x; }

}

# 函数定义和调用

1.由于JavaScript允许传入任意个参数而不影响使用，因此传入的参数比定义的参数多也没问题，虽然函数内部并不需要这些参数。

Abs(10,’dsadsa’); //返回10

2.传入的参数必定的也没有问题。abs();// 返回NaN

3.arguments只在函数内部起作用，利用arguments可以获得调用者传入的所有参数，也就是说，即使不定义任何参数，还是可以拿到参数的值。它类似于数组但不是数组。

例如：function abs(){

If(arguments.length===0){return 0;}

var x=arguments[0];

retuen x>=0?x:-x;

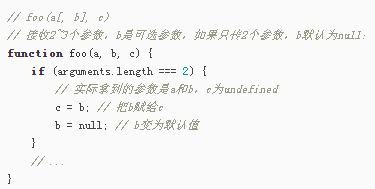
}

abs(); //0

abs(10); //10

abs(-9); //9

实际上，arguments最常用于判断传入参数的个数，



4.rest参数，由于JavaScript函数允许接收任意个参数，于是我们就不得不用arguments来获取所有参数。

function foo(a,b,...rest){

console.log(a);

console.log(a);

console.log(rest);

}

foo(1,2,3,4,5);

结果 1

2

Array[3,4,5]

foo(1);

结果 1

undefined

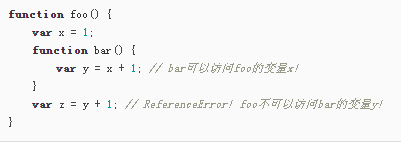
Array[]

rest参数只能写在最后，前面用...标识，从运行结果可知，传入的参数先绑定a,b,多余的参数以数组的形式交给rest。如果传入的参数连正常定义的参数都没填满，rest参数会返回一个空数组。

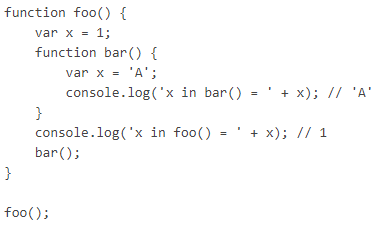
# 变量作用域与解构赋值

1.不同函数内部的的同名变量互相独立，互不影响。

2.由于JavaScript的函数可以嵌套，此时，内部函数可以访问外部函数定义的变量，反过来则不行。



3.如果内部函数定义了与外部函数重名的变量，则内部函数的变量将屏蔽外部函数的变量。

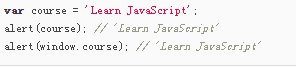


4.变量提升

在函数内部，JavaScript会将所有申明的变量提升到函数顶部，但是只提升变量的声明，不提升变量的赋值。所以我们应该在函数内部首先申明所有的变量。

5.全局作用域

不在任何函数内定义的变量就具有全局作用域，实际上，JavaScript默认有一个全局对象window,全局作用域的变量实际上被绑定到window的一个属性。



因此，访问alert();和访问window.alert();是一样的。所以，使用变量形式定义的函数实际上也是一个全局变量，顶层函数被定义成一个全局变量，绑定到window对象。

命名空间

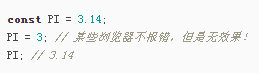
由于所有的全局变量都绑定到window，不同的文件中的全局变量可能会造成命名冲突，众多的JavaScript库将所有的代码放入唯一的命名空间。

局部作用域

由于var的作用域实际上是函数内部，无法定义具有局部作用域的变量，所以引入新的关键字，let，申明块级作用域的变量。

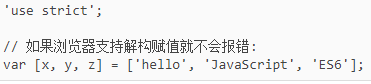
常量

const 和 let 都具有块级作用域，

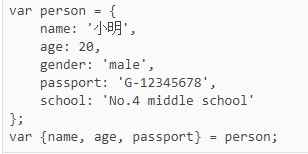


解构赋值

直接对多个变量进行赋值



对对象进行解构赋值

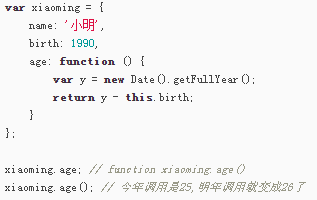


解构赋值的使用场景，交换两个变量的值，IMG_256

快速获取当前页面的域名和路径。IMG_256

## 方法

在一个对象中绑定函数，称为这个对象的方法。



在一个方法内部，this是一个特殊变量，他始终指向当前对象，也就是xiaoming这个变量。

**拆开写**

**Function** getAge(){

var y = new Date().getFullYear();

return y-this.birth;

}

var xiaoming={

name:’小明’，

birth:1990,

age:getAge

};

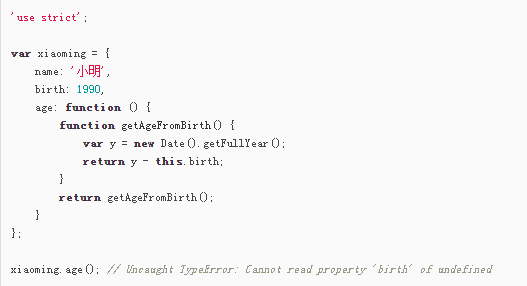
Xiaoming.age(); //25

getAge(); //NaN

如果以对象的方法形式调用，该函数的this指向被调用的对象，也就是xiaoming，这是符合我们预期的。如果单独调用函数，比如getAge(),此时，该户型的this指向全局对象，也就是window。

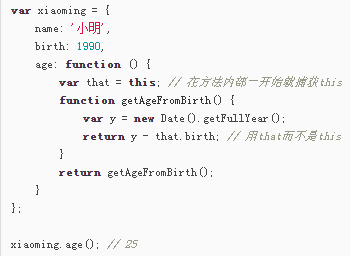
即使先IMG_256，要保证this指向正确，必须用obj.xxx();的形式调用。

**方法内嵌套**



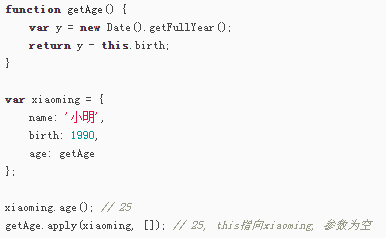
This指针只在age方法的函数内指向xiaoming，在函数内部定义的函数，this又指向undefined了！

**修复方法：用that变量首先捕获this**



## apply

要指定函数的this指向哪个对象，可以使用函数本身的apply方法，它接受两个参数，第一个参数就是需要绑定的this变量，第二个参数是array，表示函数本身的参数。



另一个和apply()类似的方法时call(),唯一的区别是:apply()将参数打包成Array在传入;call()把参数按顺序传入。比如调用Max.max(3,5,4),分别用apply()和call()实现如下:

IMG_256

**装饰器**

利用apply();我们还可以动态改变函数的行为。

JavaScript的所有对象都是动态的，即使内置的函数，我们也可以重新指向新的函数。

