JavaScript中的this作用

# JS中函数的定义方法

## 函数的定义方法一：

**function funName(params,…){**

**…**

**}**

## 函数的定义方法二：

**var funName = function(){**

**…**

**}**

# JavaScript中类的定义

## 最简单的类的定义：类内为空，在函数外赋予属性和方法。

**function Person(){}**

**创建对象：**

var person = new Person();

person.name = “Joy”;

person.age = 10;

person.sex = “man”;

person.sayHello = function(){

window.alert(“Hello World!”);

}

window.alert(person.name+”,”+person.age+”,”+person.sex);

person.sayHello();//调用执行sayHello()方法。

## 在类内定义属性和方法

### 公开属性和方法：this的作用

**在类内部，利用this定义属性和方法，这些属性和方法是公开的，在外部可以调用。这也是最常用的定义类的方式。**

function Student(){

this.name = "Joy";

this.age = 12;

**this.sayHello = function(){**

**window.alert("Hello World!");**

**}**

}

### 私有属性和方法

在类内部定义**私有属性和方法**。

function Teacher(){

var name = "Joy";//私有属性

var age = 12;//私有属性

function sayHello(){//私有方法

window.alert("Hello World!");

}

**this.sayHi** = function(){//公开方法，供外界调用私有属性和私有方法。

sayHello();

window.alert(name+","+age);

}

}

**类内部的私有属性和私有方法只能在类内部使用**，在外部要想调用这些私有属性和私有方法，必须定义一个公开方法，公开方法内部调用这些私有属性和私有方法。

## 带参数的类的定义

function Police(name,age){

this.name = name;

this.age = age;

this.sayHello = function(){

window.alert("Hello World!"+"I'm "+name+","+age+"岁了。");

}

}

测试函数：

function test5(){

**var police = new Police("Joy",12);**

police.sayHello();

}

## this的注意事项

this不能在类定义的外部使用，只能在类定义的内部使用。