深刻的来理解函数的三种角色  
 var num = 0;  
 function Fn(num) {  
 arguments[0] = 100;  
 this.num = 200;  
 console.log(num);   
 console.log(this.num); return num;  
 }

**这里将Fn(m)当做一个普通函数，预解释首先将全局变量付给函数形参，执行到函数时形参传递值为num=100;全局变量this.num=200；**

**所以console.log(num)=100;**

**Console.log(this.num)=200;**

Fn.num = 400;  
 Fn.prototype.num = 300;  
 Fn.prototype.getNum = function () {  
 console.log(this.num) ;};  
**这里Fn是一个类，Fn中有一个私有变量Fn.num=400;类Fn的有num=300;原型上有个得到num的方法，执行console.log(this.num=300;)** var f = new Fn(num);  
 console.log(f.num);  
 f.getNum();  
 Fn.prototype.getNum(); console.log(num);

**在这里Fn有一个实例f，它有Fn的私有变量num=100,console.(f.num)=100;num上边的函数只定义没有执行num在这个还是0变量执行后console.log(num)=0** var n = Fn(num);  
 console.log(n);  
 console.log(num);

**在这里,函数赋给n，输出console.log（n）=console.log(Fn(num))=100;这里num是私有变量**

**Console.log(num)=200这里num是全局变量**