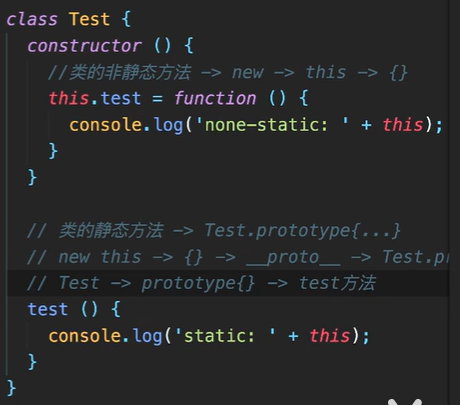
## 类

“**类的默认模式是严格模式**,不需要使用”use strict“指定运行模式。其本质是function

### 类的非静态属性和静态属性

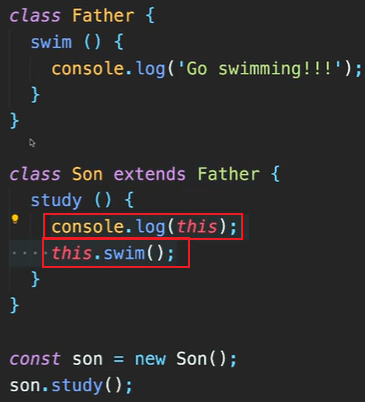
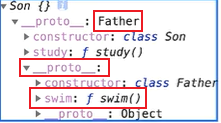


结果 

写在constructor中的非静态方法test()是在类new的过程中就一起创建了

写在外边的test()静态方法,是写在Test.prototype上的

### extends继承问题

Son继承了Father，

son.\_\_proto\_\_ 指向 Son.prototype

Son.prototype.\_\_proto\_\_ 指向 Father.prototype

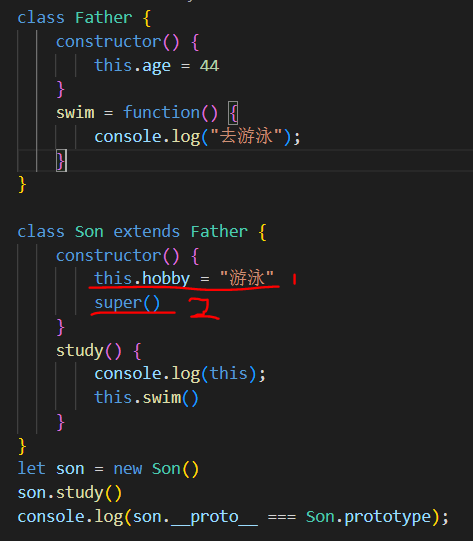
Father的swim方法是在其原型上的

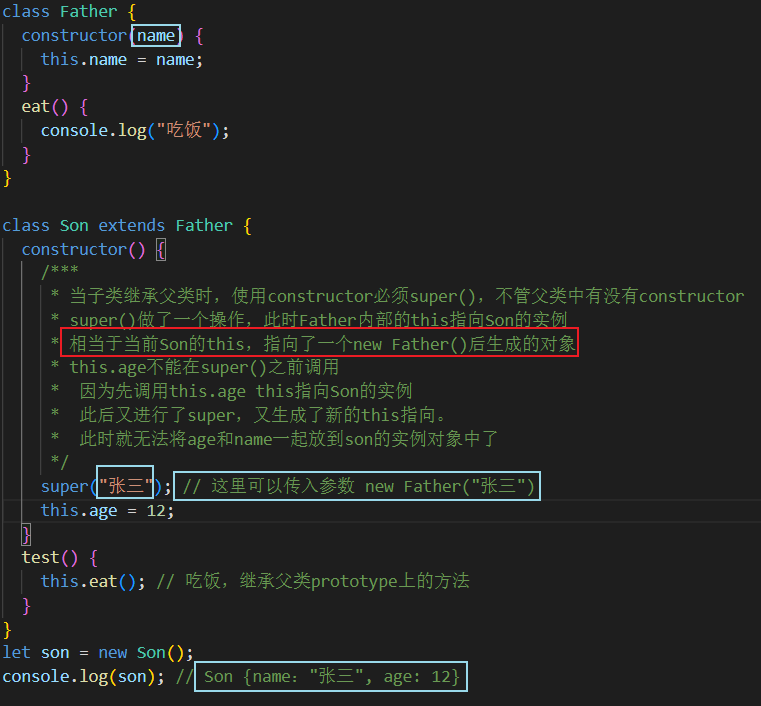
所以son实例对象可以顺着原型链使用到swim方法

### super()问题

**重点：**[this（2）\_哔哩哔哩\_bilibili](https://www.bilibili.com/video/BV1Gi4y1Z7pA?p=45&spm_id_from=pageDriver&vd_source=a67e813436aadc4608992466bdaea6bc) 18分钟

下面会报错，super()要先调用，原因看第二张图



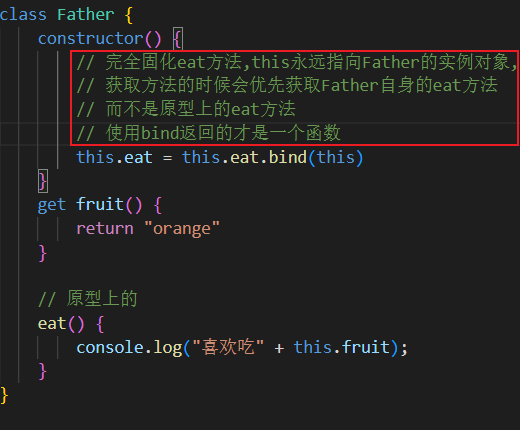


### 将一个类中的方法给到另外一个类

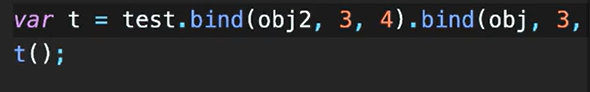


结果：

改变需求：需要完全使用Father上eat方法,水果也是Father的



## bind()问题

连用只有第一次调用的会生效



## 箭头函数this指向问题

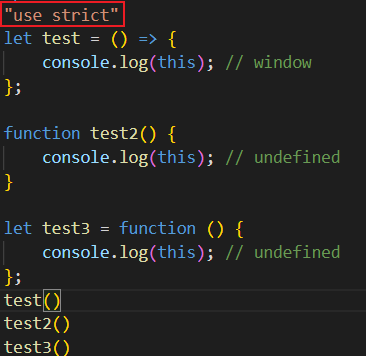
[【全网首发:已完结】『this指向问题』全解析【面试开发必备知识】\_哔哩哔哩\_bilibili](https://www.bilibili.com/video/BV1qp4y1Y7yQ?p=4&spm_id_from=pageDriver&vd_source=a67e813436aadc4608992466bdaea6bc)

箭头函数的this指向总是指向外层作用域的this指向(外层函数不能是箭头函数)

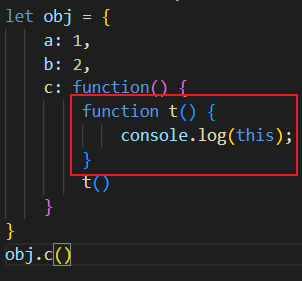
不能实例化对象

严格模式下函数声明和函数表达式this在全局中是undefined

箭头函数依然指向window



## this指向问题



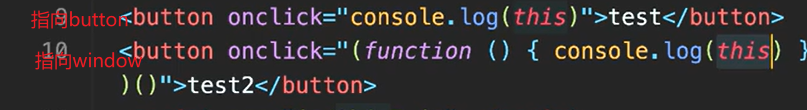
这图里面框起来的函数this指向window，它和普通函数在全局环境中调用是一样的

[this（5）\_哔哩哔哩\_bilibili](https://www.bilibili.com/video/BV1Gi4y1Z7pA?p=48&spm_id_from=pageDriver&vd_source=a67e813436aadc4608992466bdaea6bc) 8分钟

## 事件处理函数的this

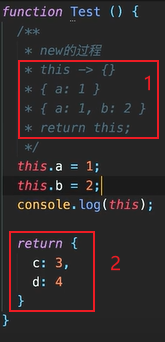






绑定事件的处理函数，而不是绑定事件？待解答

## 构造函数new的过程理解



1 构造函数在实例化对象时，内部所作的一个操作

创建一个this指向一个空对象

向下执行 分别将a 和 b 一起加入到这个对象中

隐式的返回this

2 修改了返回值，那么实例化的对象名就指向这个修改后的对象了

## this指向问题

1、

[鼠标事件深入、点击与拖拽分离、双击事件 (qq.com)](https://ke.qq.com/webcourse/386920/104499667#taid=13292245176674152&vid=5285890787799483220) 2小时07分

2、

let \_self = this

obj.addEventListener(“click”, function(){

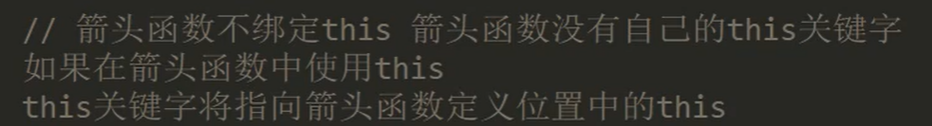
fn.call(\_self)

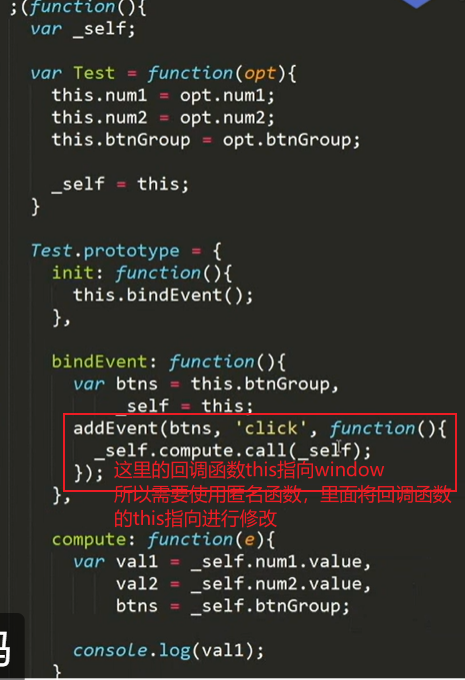
}) 这样子写可以

obj.addEventListener(“click”, fn.call(this))

这里直接通过call修改指向是不行的.该方法会直接运行一遍函数

可以通过bind来修改this指向。



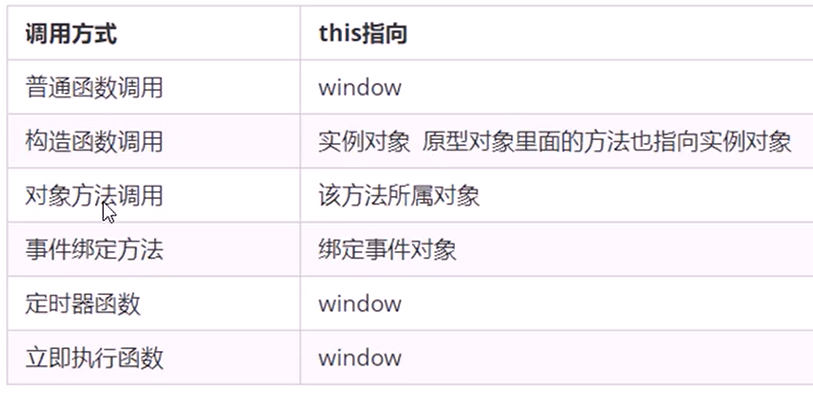


返回构造函数到GO去

用init去执行事件方法

事件处理函数都放这里

window.Text = Text



## 通过对象调用一个函数。函数中的this指向该对象

通过函数来调用一个方法，方法内部的this指向该函数。因为函数也是对象

function Fn(){}

Fn.test = function(){

console.log(this)  
}

Fn.test()