this的理解

1. 什么是this

2. 全局

3. 局部

3.1 局部——函数自身调用

3.2 局部——事件调用函数

4. 构造函数

5 箭头函数，定义的时候

6. 修改this的方法

6.1 fn.call(this,参,参)

6.2 fn.apply(this,[参,参])

6.3 fn.bind(this)（）——闭包和call

bind内部返回值是一个函数，将返回的函数的this改成传入的对象

参考文档：https://developer.mozilla.org/zh-CN/docs/Web/JavaScript/Reference/Operators/this

1. 什么是this

环境变量，上下文对象；

this的指向取决与调用this的环境。

2. 全局

this => window

3. 局部

如果this出现在函数，那么this取决于函数的执行环境

3.1 局部——函数**自身调用**

严格模式'use strict';：this -> undefined

非严格模式: this -> window

3.2 局部——函数**事件调用**

this指向触发该事件的对象。

4. 构造函数

构造函数的this指向new构造函数出来的新对象。

5 箭头函数，定义的时候

this在定义的时候决定，由上下文决定

6. 修改this的方法

函数下的方法，每一个函数在js中也是一个对象，那么每一个函数下也有一些属性和方法，其中就有call与apply方法。

6.1 **fn.call(this,参,参)**

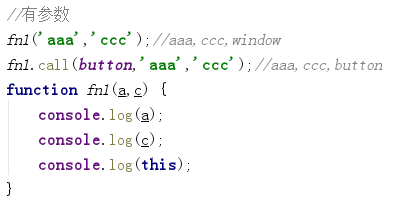
把函数作为一个对象去使用，调用该函数对象下的call方法,他的作用与直接调用函数类似，不过他可以在执行的时候传入参数。

**执行原理：**

fn在通过call执行的时候，会把函数执行过程中this指向call的第一个参数，同时会把第一个参数转成对应对象类型，所以如果我们传入了null，undefined，或不传任何参数，那么这个函数执行中的this与直接调用该函数执行中this是一致的。

call方法中除了第一个参数，后面的参数都会传递给函数的原参数。





6.2 **fn.apply(this,[参,参])**

这个方法与call一样，但是不同的是传参

apply方法可以把**arr**作为参数一个一个的传递给max。



6.3 **fn.bind(this)**

这个方法与call有点类似

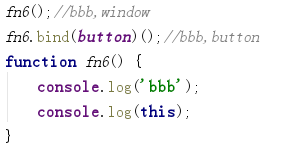
**区别：**

call与apply：this执行期绑定

bind: this声明期绑定

**执行原理：**

bind方法不改变原有函数的this，他会**返回一个绑定好的新函数**，这个新函数中的this永远都指向绑定的对象



用js实现find方法。

