**闭包**

如下，外界无法更改counter变量，只能通过add来进行更改

var add = (function () {

    var counter = 0;

    return function () {

        return counter += 1;

    }

})();

add();

add();

add();

// 计数器为 3

**自我调用**

当执行如下代码时，函数将会被调用

(function () {

    console.log("Hello!!");

})();

**This的指向**

函数的this指向调用函数的对象

var myObject = {

    firstName: "John",

    lastName: "Doe",

    fullName: function () {

        console.log(this.firstName + " " + this.lastName);

    }

}

// 输出 John Doe

myObject.fullName();

var f = myObject.fullName;

// 输出 undefined undefined

f();

Lambda的this指向的是生成Lambda的this

var myObject = {

    firstName: "John",

    lastName: "Doe",

    // 如下，在生成 lambda 时，this指向是 window

    fullName: () => {

        console.log(this.firstName + " " + this.lastName);

    }

}

// 输出 undefined undefined

myObject.fullName();

**作为构造函数**

我们可以对函数使用new关键字，当使用new关键字时，函数内部的this指向为即将生成的对象

var Person = function(age, sex, name){

    this.age = age;

    this.sex = sex;

    this.name = name;

    this.say = function(){

        console.log(`my name is ${name}`);

    }

}

var person1 = new Person(20, '男', "cql");

person1.say();