unction(resolve)

**谈谈对ES6的了解.**

es6是一个新的标准，它是一种新的javascript规范,它包含了许多新的语言特性和库.

Let和Const

let用来声明变量。特点：

(1)不存在变量提升

(2)存在暂时性死区，ReferenceError

(3)不允许重复声明,let与val，let与let，let与参数

const用来声明常量：特点：

(1)一旦声明必须初始化，常量的至不能变。

(2)只在声明的块级作用域内有效

(3)不会提升，存在暂时性死区，先声明后使用 。

(4)不可重复声明。

块级作用域

Let和const声明的变量均具有块级作用域.

为什么需要块级作用域？

(1)内层变量可能会覆盖外层变量

(2)用来计数的循环变量可能泄露为全局变量。

箭头函数

箭头函数this的指向是固定化的，这非常有利于封装回掉函数。实际原因是箭头函数根本没有自己的this，导致内部函数的this就是外层代码块的this，正因为它没有this，所以不能作为构造函数。

(1)函数体内的this对象，就是定义时所在的对象，而不是运行时所在的对象。

(2)不可以当作构造函数，就是不能使用new命令.

Promise

Promise是异步编程的一种解决方案，比传统的解决方案--回调函数和事件监听--更合理和强大。

(1)对象的状态不受外界的影响。promise对象代表一个异步操作，有3种状态：pending(进行中)、Fulfiled(已成功)、Rejected(已失败)。

(2)一旦状态发生改变就不会再变，任何时候都可以得到这个结果。Promise对象的状态改变只有两种可能：从Pending变为Fulfiled和从Pending到Rejected。

(3) Promise.then():then()方法可以接收两个函数，一个是处理成功后的函数，一个是处理错误结果的函数。

(4) Promise.resolve():Promise成功的回调函数,接受一个参数值，可以是普通的值，具有then()方法的对象和Promise实例。正常情况下，它返回一个Promise对象，状态为fulfilled。但是，当解析时发生错误时，返回的Promise对象将会置为rejected态.

(5) Promise.reject():

(6) Promise.all():Promise.all()接收一个参数，它必须是可以迭代的，比如数组。它通常用来处理一些并发的异步操作.

(7)Promise.race():Promise.race()和Promise.all()类似，都接收一个可以迭代的参数，但是不同之处是Promise.race()的状态变化不是全部受参数内的状态影响，一旦参数内有一个值的状态发生的改变，那么该Promise的状态就是改变的状态。