参考摘抄于：<https://es6.ruanyifeng.com/#docs/async>

1、基本概念

async函数是Generator函数的语法糖，使异步函数操作变得更加方便。async就是将Generator函数的（\*号）替换成async，将yield替换成await。

async函数对Generator函数的改进，体现在以下四个方面：

(1)内置执行器：Generator函数执行必须依赖执行器（如co模块），而async函数自带执行器，调用后会自动执行。

(2)更好的语义：async和await比起星号和yield语义更清楚了。async表示函数里有异步操作；await表示需要等待紧跟在其后的表达式执行。

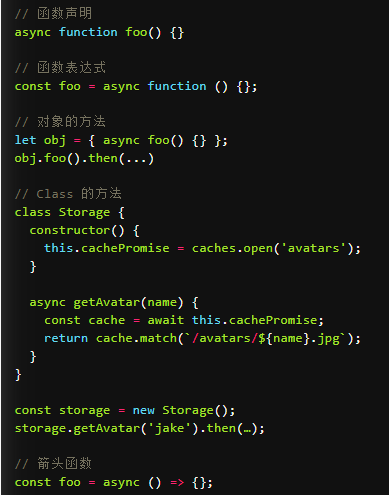
(3)更广的适用性：co模块约定yield后必须是Thunk函数或者Promise对象，而async函数的await后可跟Promise对象和原始类型的值（原始类型会自动转换为resolved的Promise对象）

(4)返回值是Promise对象：async函数的返回值是Promise对象，比Generator函数返回的Interator对象更方便。只要通过then方法就可以指定下一步操作。进一步说，async函数可以看成是多个异步函数包装成的一个Promise对象。

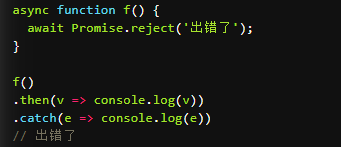
2、基本用法

async返回一个Promise对象，可以使用then函数添加回调函数。当函数执行的时候，一旦遇到await就先返回，等到异步执行完毕，再执行函数体内后面的语句。

使用方式:



注意：await命令后面的Promise对象如果变成reject状态，则reject的参数会被catch方法的回调函数接收到。

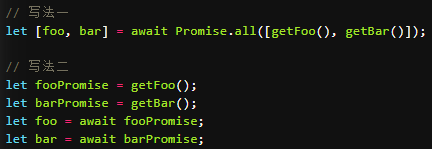


任何一个await语句后面的Promise对象变成rejected状态，那么整个async函数中断执行。（不再执行后续的代码）。如果想让代码继续执行，可以将第一个的await放在try…catch函数中，或者在第一个await方法后面跟着的Promise对象再跟一个catch方法，这样不论第一个await后面的代码执行是否报错，后面的代码都可以正常执行。

使用注意点：

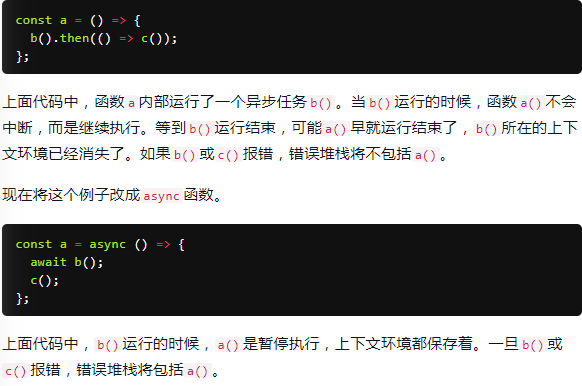
1>await命令后面的Promise对象执行运行结果可能是rejected，最好把await命令放在try…catch中。

2>多个await命令后面的异步任务，如果不是继发，最好让他们同时触发。



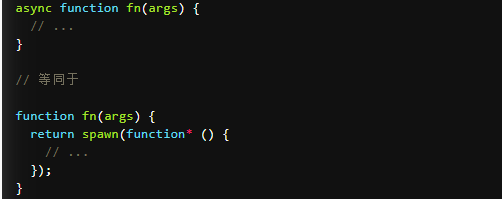
3>await命令只能用在async函数中，如果在普通函数中会报错。

4>async可以保留运行堆栈



3、async函数的实现原理

async函数的实现原理就是将Generator函数和自动执行器封装在一个函数里。



其中，spawn是一个自动执行函数。下面是spawn函数的实现。



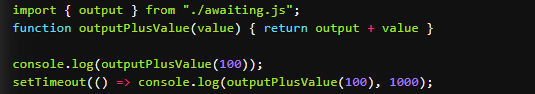
4、顶层await（语法提案）

借用await来解决模块异步加载的问题。若B文件导入A文件，A文件存在异步操作，且B文件代码执行依赖A文件异步操作的执行结果时，B文件的执行时机就会影响到实际执行结果。故引入了顶层await用来处理此类场景。

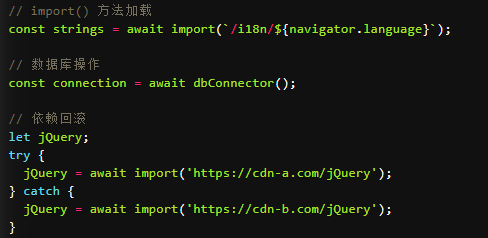
A文件代码



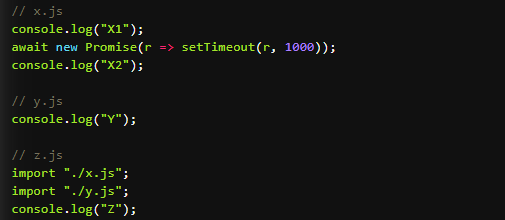
B文件代码



顶层await其他使用场景：



注意：如果加载多个包含顶层await命令的模块，加载命令是同步的。



输出结果：X1、Y、X2、Z

顶层await命令有点像交出代码的执行权给其他模块加载，等异步操作完成后，再拿回执行权继续执行后续代码。