

aaudio 范例: 纤程入门

```
//纤程入门
import console;
```

```
/*
一个运行时程序创建一个进程。
一个进程可包含多个可以并发运行的线程。
而一个线程可以包含多个有独立堆栈环境的纤程,纤程不能并发运行。
```

纤程存在的目的是在不同的函数间交换执行代码的控制权,
这类似迭代器,通过多个函数的分工合作的方式实现代码逻辑的分离。

但迭代器每次都要把一个函数执行完成,需要依赖for语句来重复的保存现场恢复现场。
所以初学者理解迭代器可能困难,但纤程就要简单的多启动纤程的函数只要执行一次。
纤程函数可以随时使用fiber.yield暂停执行,并可以通过fiber.resume唤醒。

fiber.yield与fiber.resume在两个纤程间可以交换执行代码的控制权。
并且他们的参数和返回值也可以互换 - 这就可以方便的在两个纤程间交换执行的进度和数据。

```
*/
var func = function(n){ //首次调用fiber.resume启动纤程时传入的参数 - 由这里启动函数的参数接收,即n=12
    for(i=1;n;1){
        fiber.yield (i) //休眠并将控制权与参数返回给调用者,此函数的返回值为下面代码中fiber.resume的参数
    }
}

//创建纤程,func作为启动函数
var fib = fiber.create(func)

while(
    var r,value;
    r,value = fiber.resume (fib,12); //控制权切到到纤程fib,此函数的返回值即为上面代码中fiber.yield的参数
    r && value
) {
    console.log(value)
}

console.pause();
```

[Markdown 格式](#)