**Node.js(服务器端)**

**基于谷歌的V8引擎，一种运行在服务器端的JS解释器。**

**可以实现所有后端语言的功能，例如：Java，PHP等能做的Node.js也可以。**

**（1）对比JS和Node.js**

**1）JS运行在客户端浏览器，存在多款浏览器兼容性问题。**

**Node.js运行在服务器端，只能谷歌V8引擎一种解释器，不存在兼容问题。**

**2）两者都有ES（内置）对象，自定义对象，不同的宿主对象。**

**3）JS用于开发浏览器端的交互效果**

**Node.js用于服务器的操作，例如：数据库访问，网络操作等…**

**Node.js适合做IO（输入/输出）操作为主的应用。**

**不适合做CPU密集型的应用**

**（2）Node.js的运行模式**

**脚本模式：**

**node 文件路劲 回车**

**交互模式：**

**node 回车 进入到交互模式**

**两次Ctrl+c退出交互模式**

**全局对象**

**（1）Node.js——global**

**交互模式是在全局作用域下，变量和函数都是全局global下。**

**eg：global.a global.fn()**

**脚本文件中所处在一个模块作用域下，里面的变量和函数都是局部的。**

**（2）JS——window （浏览器端）**

**在JS脚本文件中所处在全局作用域下，里面的变量和函数都是全局的。**

**eg：window.a window.fn()**

**console对象**

**console.log(1); //打印输出**

**console.info(2); //输出消息**

**console.warn(3); //输出警告**

**console.error(4); //输出错误**

**console.time('字符串') //开始计时**

**console.timeEnd('字符串') //结束计时**

**process对象**

**进程对象**

**process.arch 查看当前CPU的架构**

**process.platform 查看当前的操作系统**

**process.version 查看当前node.js版本号**

**process.pid 查看当前进程的编号**

**process.kill() 结束指定编号的进程**

**Buffer对象**

**缓冲区，缓冲器：内存中的一块区域，用于临时存储数据，往往存储一些网络传输时的资源。**

|  |
| --- |
| **//创建Buffer大小为5，并存储数据**  **let buf=Buffer.alloc(5,'abcde');**  **//将buf转为字符串**  **buf.toString() 或者 String(buf)** |

**定时器**

**（1）一次性定时器**

|  |
| --- |
| **开启**  **let timer=setTimeout(回调函数,间隔时间);**  **当间隔时间到了，会执行一次回调函数，单位是毫秒。**  **清除**  **clearTimeout(timer);** |

**（2）周期性定时器**

|  |
| --- |
| **开启**  **let timer=setInternval(回调函数,间隔时间);**  **每隔一段时间，执行一次回调函数**  **清除**  **clearInterval(timer);** |

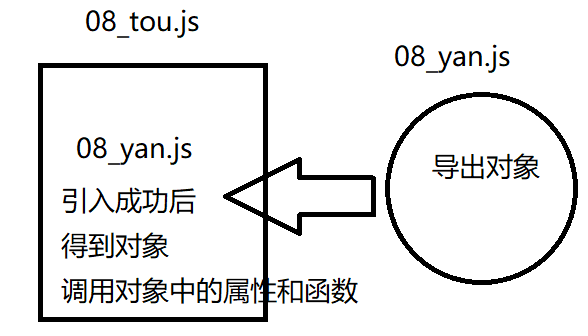
**强制结束运行用 Ctrl+c**

**模块系统**

**每个文件就是一个模块，每一个模块是一个独立的功能体。**

**Node.js下模块分为自定义模块，核心模块，第三方模块**

|  |
| --- |
| **require 用于引入一个模块，得到的是导出的对象**  **module 当前模块的对象**  **module.exports 导出的对象，默认是一个空对象，如果要导出内容，只需要把内容添加到对象中** |



课后任务

(1)复习今天内容，整理思维导图

(2)练习

创建两个模块 main.js(主模块)，circle.js(功能模块)，在功能模块下导出两个函数，传递半径，计算圆的周长和面积。在主模块下引入，并调用两个函数

(3)预习核心模块 querystring、url、fs模块；同步和异步原理