node.js是一个基于Chrome v8引擎的javascript 运行环境。Node.js使用了一个事件驱动(执行事件决策的策略)、非阻塞式 I/O的模型，使其轻量又高效。（由c＋＋语言编写的）,包管理器npm

简单说：编写高性能网络服务器的javascipt工具包（用于js开发服务端程序）

单线程、异步、事件驱动。

可以辅助前端开发，代替后台开发

node-webkit／内核 nodeos／系统 express／框架

jade EJS ／模版 forever PM2 ／实现负载均衡

log.io／在浏览器打印各种输出的日志

grunt、gulp、webpack／自动化构建工具

mocha 、karma ／测试工具

特点：快、耗内存多——－网上一个百万级并发测试，未优化的情况下1M的链接消耗了16G内存

Nodejs的作用：node.js可以解析js代码（没有浏览器安全级别的限制），提供很多系统级别的api，如：文件的读写、进程的管理、网络通信。。。。

Node与php的差别

优点：

性能高

开发效率高

应用范围广

缺点：

新、会的人少

中间件少（函数）

IDE不完善（开发工具）

Const url=require(‘url’);

url——提供很多方法

url.parse(urlString)查询url信息

url.format(urlObject)将信息解析成正常的url地址

url.resolve(from, to)将两个url解析成一个完整的url from通常根url前 to通常跟path路径

querystring.escape(str)解析对象，字符串（汉字）

querystring.parse(str[, sep[, eq[, options]]])转换成对象

querystring.stringify(obj[, sep[, eq[, options]]])转换成字符串

querystring.unescape(str)解析escape转换的编码，解析成汉字

fs（file system）跟文件相关的方法—require(‘fs’)

1)stat() 获取文件的一些信息 isFile()

2)mkdir() 新增目录的

3)writeFile(url,con,fc) 书写文件

4)appendFile(url,con,fc) 追加文件内容

5)读取文件的内容：readFile(url,fc)

6)列出目录的东西：readdir

7)重命名目录或文件：rename

8)删除目录与文件：rmdir（目录）,unlink（文件）

readdirSync(src) 同步读取目录信息

event事件

class Aplayer extends aevent{};

var aplayer=new Aplayer();

aplayer.once('play',(abc)=>{

console.log('zzzzz'+abc);

})

aplayer.emit('play','hhhh');

aplayer.emit('play','hhhhaaaa');

gulp ：基于流的自动化构建工具； 提供丰富的插件

易于使用  
通过代码优于配置的策略，Gulp 让简单的任务简单，复杂的任务可管理。  
构建快速  
利用 Node.js 流的威力，你可以快速构建项目并减少频繁的 IO 操作。  
插件高质  
Gulp 严格的插件指南确保插件如你期望的那样简洁高质得工作。  
易于学习  
通过最少的 API，掌握 Gulp 毫不费力，构建工作尽在掌握：如同一系列流管道。

grunt： js的构建工具，一般用于：压缩文件，合并文件，简单语法检查；自动化。对于需要反复重复的任务，例如压缩（minification）、编译、单元测试、linting等，自动化工具可以减轻你的劳动，简化你的工作。当你在 Gruntfile 文件正确配置好了任务，任务运行器就会自动帮你或你的小组完成大部分无聊的工作。  
为什么要使用Grunt？

webpack主要吧浏览器不识别的语言（babel）转换成浏览器可识别的语言。webpack.config.js

－－提供了丰富的加载器

版本管理工具

Svn：集中式**SVN 断开网络或者断开VPN就无法commit代码，**

Git：分布式**但是Git 可以先commit到本地仓库。**

模块化开发规范

AMD－－异步加载规范，require。js 提前引入

CMD－－标准通用规范，sea。js 按需引入

commonjs－－谷歌提供的一种规范，核心 模块的接收（require）与暴露（module.exports）

（当一个js需要调用另一个js内的功能方法）