1. node的优点是什么？缺点是什么？

优点：<1高并发（最主要的一个优点）

<2适合I/O密集型应用

**缺点：**

1. 不适合CPU密集型应用；CPU密集型应用给Node带来的挑战主要是：由于JavaScript单线程的原因，如果有长时间运行的计算（比如大循环），将会导致CPU时间片不能释放，使得后续I/O无法发起；

解决方案：分解大型运算任务为多个小任务，使得运算能够适时释放，不阻塞I/O调用的发起；

2. 只支持单核CPU，不能充分利用CPU

3. 可靠性低，一旦代码某个环节崩溃，整个系统都崩溃

原因：单进程，单线程

解决方案：

（1）Nnigx反向代理，负载均衡，开多个进程，绑定多个端口；

（2）开多个进程监听同一个端口，使用cluster模块；

4. 开源组件库质量参差不齐，更新快，向下不兼容

5. Debug不方便，错误没有stack trace

1. Node有哪些特征，与其他服务器相比？

他是一个javascript运行环境，依赖于Chrome V8引擎进行代码解释

特征：单线程、事件驱动、非阻塞I/O，轻量，可伸缩，适于实时数据交互应用

node 无法直接渲染静态页面，提供静态服务

node 没有根目录的概念

node 必须通过路由程序指定文件才能渲染文件

node 比其他服务端性能更好，速度更快

1. commonJS中的require/exports和ES6中 import/export的区别是什么？

commonJS模块的重要特性是加载时执行，及脚本代码在require的时候，就会全部执行。一旦出现某个模块被“循环加载”就只输出已经执行的部分，还没有执行的部分是不输出的

ES6模块是动态引用，如果使用import从一个模块加载变量，那些变量不会缓存，而是成为一个指向被加载模块的引用,import/export最终都是编译为require/exports来执行的

1. 使用npm的好处是什么？

通过NPM，你可以安装和管理项目的依赖，并且能够指明依赖项的具体版本号，可以通过package.json文件来管理项目信息，配置脚本

1. 如何判断当前脚本运行在浏览器还是node环境中

通过判断 Global 对象是否为 window ，如果不为window ，当前脚本没有运行在浏览器中

1. 简述同步和异步的区别，如何避免回调地狱

同步方法调用一旦开始，调用者必须等到方法调用返回后，才能继续后续的行为

异步方法调用一旦开始，方法调用就会立即返回，调用者就可以继续后续的操作。而异步方法通常会在另外一个线程中，整个过程，不会阻碍调用者的工作

避免回调地狱：

1）Promise

2）async/await

3）generator

4）事件发布/监听模式

7. app.use和app.get区别

app.use(path,callback)中的callback既可以是router(路由)对象又可以是函数

app.get(path,callback)中的callback只能是函数

8．内置的fs模块架构是什么样子的？

fs模块主要由下面几部分组成:

1) POSIX文件Wrapper,对应于操作系统的原生文件操作

2) 文件流 fs.createReadStream和fs.createWriteStream

3) 同步文件读写,fs.readFileSync和fs.writeFileSync

4) 异步文件读写, fs.readFile和fs.writeFile

9. 实现一个简单的http服务器？

思路是加载http模块，创建服务器，监听端口

代码演示：

var http = require('http'); // 加载http模块

http.createServer(function(req, res) {

res.writeHead(200, {'Content-Type': 'text/html'}); // 200代表状态成功, 文档类型是给浏览器识别用的

res.write('<meta charset="UTF-8"><h1>我是标题啊！</h1><font color="red">这么原生初级的服务器，下辈子能用着吗，哈哈哈?!</font>'); // 返回给客户端的html数据

res.end(); // 结束输出流

}).listen(3000); // 绑定3000, 查看效果请访问 http://localhost:3000

10．express中如何获取路由的参数？

/users/:name使用req.params.name来获取;  
req.body.username则是获得表单传入参数username;  
express路由支持常用通配符 ?, +, \*, and ()

11.express response有哪些常用方法？

res.download() 弹出文件下载

res.end() 结束response

res.json() 返回json 在这里插入代码片

res.jsonp() 返回jsonp

res.redirect() 重定向请求

res.render() 渲染模板

res.send() 返回多种形式数据

res.sendFile 返回文件

res.sendStatus() 返回状态

12.es5和es6的区别？

ECMAScript5，即ES5，是ECMAScript的第五次修订，于2009年完成标准化

ECMAScript6，即ES6，是ECMAScript的第六次修订，于2015年完成，也称ES2015

ES6是继ES5之后的一次改进，相对于ES5更加简洁，提高了开发效率

ES6新增的一些特性：

　　1）let声明变量和const声明常量，两个都有块级作用域  
　　ES5中是没有块级作用域的，并且var有变量提升，在let中，使用的变量一定要进行声明

　　2）箭头函数  
　　ES6中的函数定义不再使用关键字function()，而是利用了()=>来进行定义

　　3）模板字符串  
　　模板字符串是增强版的字符串，用反引号（`）标识，可以当作普通字符串使用，也可以用来定义多行字符串

　　4）解构赋值  
　　ES6 允许按照一定模式，从数组和对象中提取值，对变量进行赋值

　　5）for of循环  
　　for...of循环可以遍历数组、Set和Map结构、某些类似数组的对象、对象，以及字符串

　　6）import、export导入导出  
　　ES6标准中，Js原生支持模块(module)。将JS代码分割成不同功能的小块进行模块化，将不同功能的代码分别写在不同文件中，各模块只需导出公共接口部分，然后通过模块的导入的方式可以在其他地方使用

　　7）set数据结构  
　　Set数据结构，类似数组。所有的数据都是唯一的，没有重复的值。它本身是一个构造函数

　　8）... 展开运算符  
　　可以将数组或对象里面的值展开；还可以将多个值收集为一个变量

　　9）修饰器 @  
　　decorator是一个函数，用来修改类甚至于是方法的行为。修饰器本质就是编译时执行的函数

　　10）class 类的继承  
　　ES6中不再像ES5一样使用原型链实现继承，而是引入Class这个概念

　　11）async、await（是重点）  
　　使用 async/await, 搭配promise,可以通过编写形似同步的代码来处理异步流程, 提高代码的简洁性和可读性  
　　async 用于申明一个 function 是异步的，而 await 用于等待一个异步方法执行完成

　　12）promise（是重点）  
　　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）更合理、强大

　　13）Symbol  
　　Symbol是一种基本类型。Symbol 通过调用symbol函数产生，它接收一个可选的名字参数，该函数返回的symbol是唯一的

　　14）Proxy代理  
　　使用代理（Proxy）监听对象的操作，然后可以做一些相应事情

13. var,let,const的区别？

var声明变量可以重复声明，而let不可以重复声明

　　var是不受限于块级的，而let是受限于块级

　　var会与window相映射（会挂一个属性），而let不与window相映射

　　var可以在声明的上面访问变量，而let有暂存死区，在声明的上面访问变量会报错

　　const声明之后必须赋值，否则会报错

　　const定义不可变的量，改变了就会报错

const和let一样不会与window相映射、支持块级作用域、在声明的上面访问变量会报错

14.使用箭头函数应该注意什么？

（1）用了箭头函数，this就不是指向window，而是父级（指向是可变的）

　　（2）不能够使用arguments对象

　　（3）不能用作构造函数，这就是说不能够使用new命令，否则会抛出一个错误

　　（4）不可以使用yield命令，因此箭头函数不能用作 Generator 函数

15. 介绍下 Set、Map的区别？

应用场景Set用于数据重组，Map用于数据储存

Set：

　　（1）成员不能重复  
　　 （2）只有键值没有键名，类似数组  
　　 （3）可以遍历，方法有add, delete,has

Map:

　　（1）本质上是健值对的集合，类似集合  
　　 （2）可以遍历，可以跟各种数据格式转换

16.Promise构造函数是同步执行还是异步执行，那么 then 方法呢

promise构造函数是同步执行的，then方法是异步执行的

17. promise有几种状态，什么时候会进入catch？

三个状态：pending、fulfilled、reject

　　两个过程：padding -> fulfilled、padding -> rejected

当pending为rejectd时，会进入catch

18. Promise 中reject 和 catch 处理上有什么区别

reject 是用来抛出异常，catch 是用来处理异常

reject 是 Promise 的方法，而 catch 是 Promise 实例的方法

reject后的东西，一定会进入then中的第二个回调，如果then中没有写第二个回调，则进入catch

网络异常（比如断网），会直接进入catch而不会进入then的第二个回调

19. forEach、for in、for of三者区别

forEach更多的用来遍历数组

for in 一般常用来遍历对象或json

for of数组对象都可以遍历，遍历对象需要通过和Object.keys()

for in循环出的是key，for of循环出的是value

查漏补缺：

**1 javascript部分：**

**重点 js 高级程序设计 3 - 7 章，10 - 13 章，21 章（有js高级程序设计这本书）**

**2 网络部分**

**重点图解 http，极客时间的趣谈网络协议，博客**

**3 html，css部分**

**Css 权威指南 2、3、7、8、10 章，w3c，博客**

**4 es6部分**

**重点 ES6 标准入门 2、3、7、11、14、16、18、19、22、23**

**5 算法**

**重点剑指 offer！！**