# 第四天上课笔记

Nodejs：js的一个运行环境，事件驱动，非阻塞I/O；

模块：

系统模块；

自定义模块；

第三方模块；

关键词：

Require：引入模块，使用模块之前必须引入；

Exports：给我们一个自定义模块的机会；

Module.exports:导出自定义模块；

系统模块：

Fs：文件操作，读取，创建，写入文件；

Os：系统信息,EOL === end of line；

全局：global（window）、\_\_dirname、\_\_filename

### Node.js包管理器：

NodePackageManager === npm

cnpm：干的事情和npm完全一样，c === china

阿里巴巴搞的国内的npm服务器，每十分钟从npm官方网站同步一次最新的包；

cnpm install mysql

在安装一个包的时候，会把安装的包所依赖的包全部下载下来；

### 全局安装：

cnpm i react -g

全局安装的模块在每个项目里面都可以使用。

项目使用的依赖包最好是直接在项目里面进行安装，不要装成全局的。

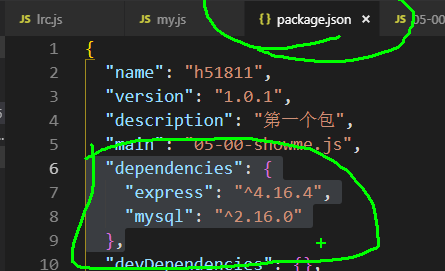
### 生成包的配置文件package.json：

npm init

一路回车即可，npm init -y表示直接生成配置文件。

### 把安装的包追加到依赖包里面去：

cnpm i mysql --save



### 如何把自己的项目代码分享给别人？

Node\_modules不要添加到压缩包里面 或者 git 上；

拿到代码之后,执行 cnpm i：会自动安装上所有的依赖包

### 删除不需要的第三方模块：

npm uninstall mysql

### 一次性安装多个第三方模块：

cnpm i mysql ejs vue react express --save ;

### 使用第三方模块有两个必须的步骤：

1. 安装：npm；
2. 引入：require；

最后，参考官方文档使用第三方模块，如果看不懂可以百度搜索一下；

### HTTP协议：

HTTP协议（HyperText Transfer Protocol 超文本传输协议）用于服务器端和客户端的通信。是B/S模式开发接触得最多的一种协议。

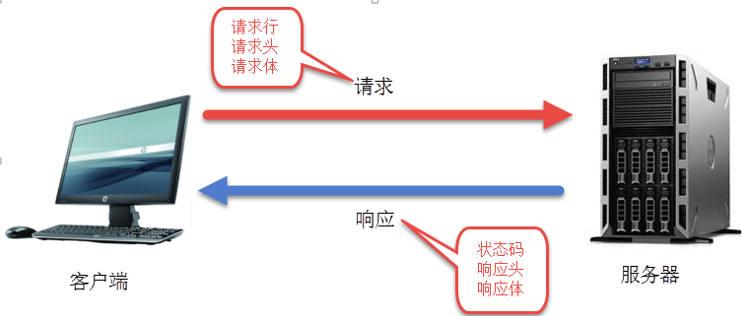
### HTTP协议是无状态的：

当一次访问完成之后，客户端和服务器端是断开的，不再有任何联系，当客户端再次访问服务器的时候，服务器不知道我是谁；

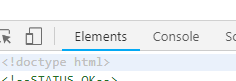
Cookie以及session实现了会话控制；

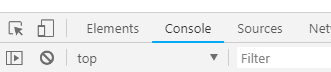
### 只能是客户端主动发起请求：

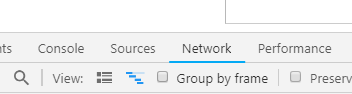
不能是服务器直接发送数据到客户端；



### 调试工具的使用：

HTML、CSS：

JavaScript：

HTTP：

### URL地址和请求的区别：

URL就是一个页面地址；

如果当前页面有1张图片，就需要发起1个请求； ===》2

如果当前页面有2个css文件，就需要发起2个请求； ====》4

如果当前页面有3个js文件，就需要发起3个请求；=====>7

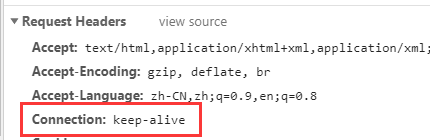
### 完成一个请求的步骤：

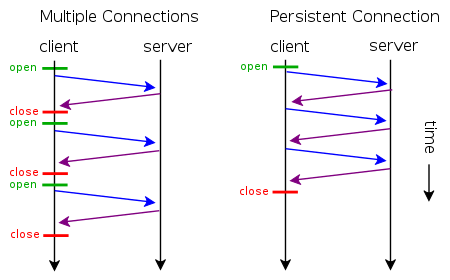
1. 建立连接：客户端和服务器端要连通；
2. 客户端发起请求：告诉服务器我要什么东西，request；
3. 服务器端响应请求：返回资源（html、css、js、图片等）到客户端，response；
4. 断开连接：释放资源，web服务器可以服务很多请求；

### 长连接：

HTTP1.0是每个资源请求完成之后，就断开；

HTTP1.1默认是资源请求完成之后，如果还有请求，会保持连接；





长连接的性能明显优于每次都断开重新连接。

Keep-Alive: timeout=5, max=100  
Timeout：过期时间5秒；

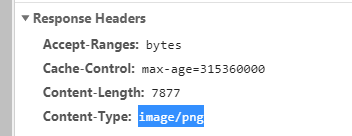
Max：最多一百次请求，强制断掉连接；

### 请求头：

请求头里面的客户端信息：

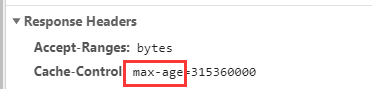


### 响应头：

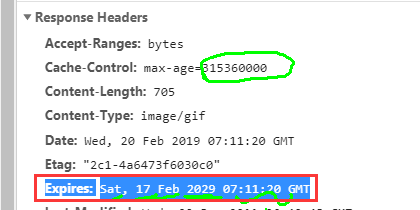


Content-Type：定义响应内容类型

Cache-Control: 有没有缓存图片、js、css等静态资源；

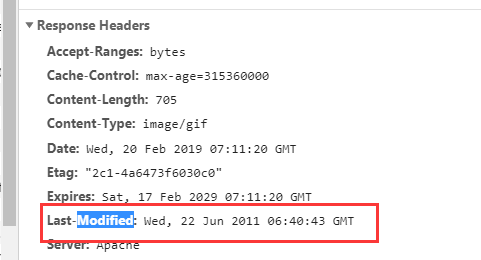


定义缓存时间：从当前时间开始，多少s之内有效



在什么时间之前有效：有效期，具体的时间，参考的是本地时间；

如果说expires和max-age冲突了，expires失效，优先使用max-age。



当前文件在服务器上的最后修改时间。

**CTRL+F5**：强制刷新，不会使用本地cache的内容，全部从服务器重新加载，在开发的时候常用，防止页面上的效果不是代码的最新效果；

### 如何优化我们的前端页面提升页面性能？

1. 可以把多个小图标放到一张图里面，一次性加载；background-position
2. 直接使用svg描述图片，不用加载资源；
3. 使用字体库实现图标效果；
4. Css和js文件可以合并到一个文件；
5. Css、js、小图片等等可以直接打包成一个js文件；
6. Ajax可以实现局部刷新，减少HTTP请求；

### HTTP请求状态码：

200：成功；

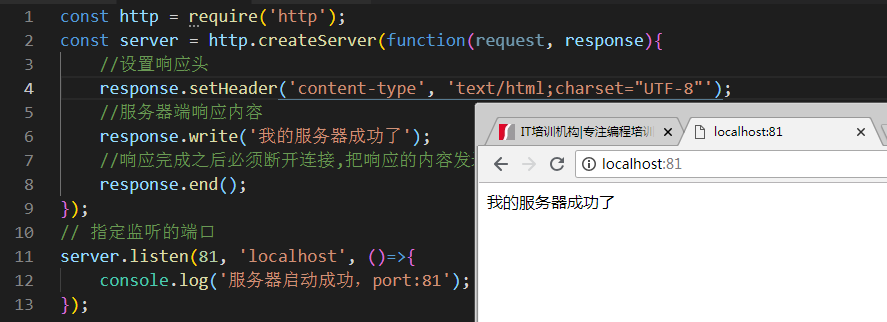
302：Redirect，跳转到新的页面；

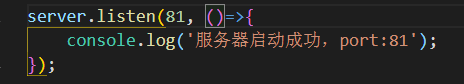
404：找不着，客户端错误；

500：服务器内部错误；

### 使用nodejs的http模块，实现http协议：

最基本的web服务器：





可以访问的方式：

<http://localhost:81>

<http://lulaoshi:81> 给你的计算机取个名字，建议使用这种方式访问

<http://127.0.0.1:81>

<http://192.168.7.103:81/>

URL地址的基本结构：

协议://域名:端口号/路径?查询信息

**路由：**当用户在访问不同的地址的时候，很明显，我们应该给她响应不同的内容。

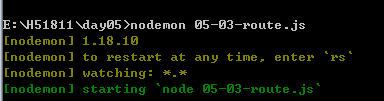


nodemon：实现nodejs的进程维护



使用：

nodemon 文件名



### 传值的方式

表单提交：get、post

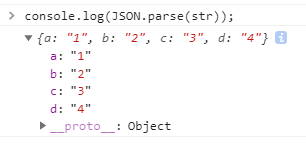
Get请求：

传值：?后面传值 ?a=1&b=2&c=3&d=4

### 对象和字符串的相互转换：

JSON.stringify()：把对象转换成字符串，我们称这个字符串是对象格式的字符串；

JSON.parse()：把对象（JSON）格式的字符串转换成真正对象；



### 做一个简单的登录功能：

1. 做个登录页面；
2. 登录数据处理；

### Querystring:

可以对查询信息进行解析