# 上课笔记

## Nodejs：

是JS的一个运行环境；

Fs：是因为作者Cyan提供了对应的接口；

Js可以运行在浏览器端：JS解释器；

document.getElementById(‘mydiv’)：我们的js为什么可以获取到浏览器里面的节点？

因为浏览器提供了API；

Nodejs的核心是模块；

## 模块：

模块分类：

系统模块：fs，http，url，querystring等；

自定义模块：自己写的模块；

第三方模块：别人写的模块，express就是其中一个，要使用必须要安装；

模块的引入和导出：

require：引入模块，先找系统模块，然后node\_modules 目录，最后根据路径找；

module.exports：导出整个模块，常用；

export：导出模块，适合导出变量或者函数

## Express：

就是一个第三方模块，方便我们开发web服务；

官方文档：http://www.expressjs.com.cn/ 

jQuery----》JS

Express---》Nodejs

## 项目初始化：

npm init

## 安装Express：

npm i express --save

## 引入第三方模块：

const express = require(‘express’);

## 路由：

根据用户访问的地址，响应不同的内容；

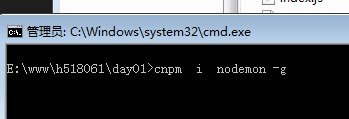
官方：路由是指如何定义应用的端点（URIs）以及如何响应客户端的请求。

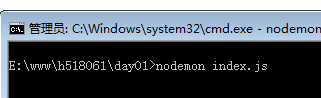
用户访问路径：除去根域名后的字符串 ，第一个字符是 **/**；

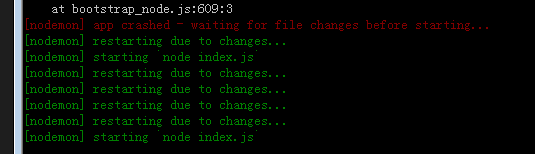
HTTP协议：

请求头，响应头

## 服务进程管理：进程自动化，进程守护







## Express的内容输出：

send：write end；

支持字符串、数组、对象等，最好不要输出数字类型，如果需要输出数组就用引号引起来；

send会把数字当成状态码；

json:专门输出JSON的数据；

## 继续执行：

//支持多函数

server.get('/fn', (req, res, next) => {

res.send('1');

}, (req, res, next) => {

res.send('2');

}, (req, res) => {

res.send('3');

});

next允许响应继续往下执行；

注意：整个请求只能有一个响应；

一般是前面的函数对数据进行加工处理，最后一次性输出；

一个请求只能有一个响应：多个响应会报错，代码应该如下；

//支持多函数

server.get('/fn', (req, res, next) => {

req.body = {};

// res.send('1');

req.body.username = '龙宇';

next();

}, (req, res, n) => {

// res.send('2');

req.body.passwd = '123456'

n();

}, (req, res, n1) => {

res.send(req.body);

});

## 中间件：

Express本身非常简介，他主要依赖中间键实现各种功能；

Express 是一个自身功能极简，完全是由路由和中间件构成一个的 web 开发框架：从本质上来说，一个 Express 应用就是在调用各种中间件。

**中间件（Middleware）** 是一个函数，它可以访问请求对象（[request object](http://www.expressjs.com.cn/4x/api.html#req) (req)）, 响应对象（[response object](http://www.expressjs.com.cn/4x/api.html#res) (res)）, 和 web 应用中处于请求-响应循环流程中的中间件，一般被命名为 next 的变量。

如果要使用中间件必须使用**use**；

## 静态资源托管：

static:少有的内置方法之一，可以实现静态资源托管；

static(‘path’:string, ‘放置静态资源的目录:string’);

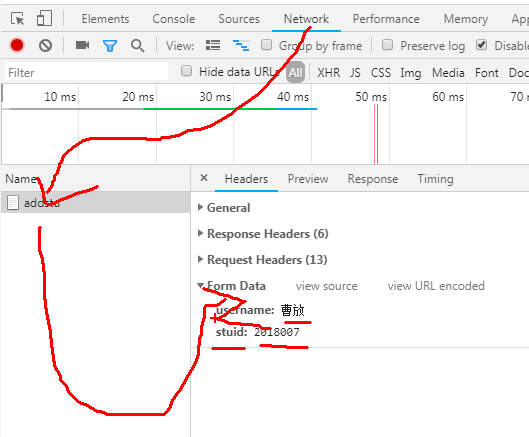
第一个参数path是可以省略的，省略的时候表示所有的访问；

我们的请求路径里面，不应该包含静态资源目录；

## 传值问题：

GET：直接在url里面传值，req.query;

POST：直接使用第三方模块body-parser



我们在做项目的时候，会有很多POST请求以及传递数据；

那不是所有post请求的数据都在req的body属性里了？

接收POST：body-parser

Ajax的时候，一般来讲我们操作是JSON格式的数据，此时服务器端最好使用：

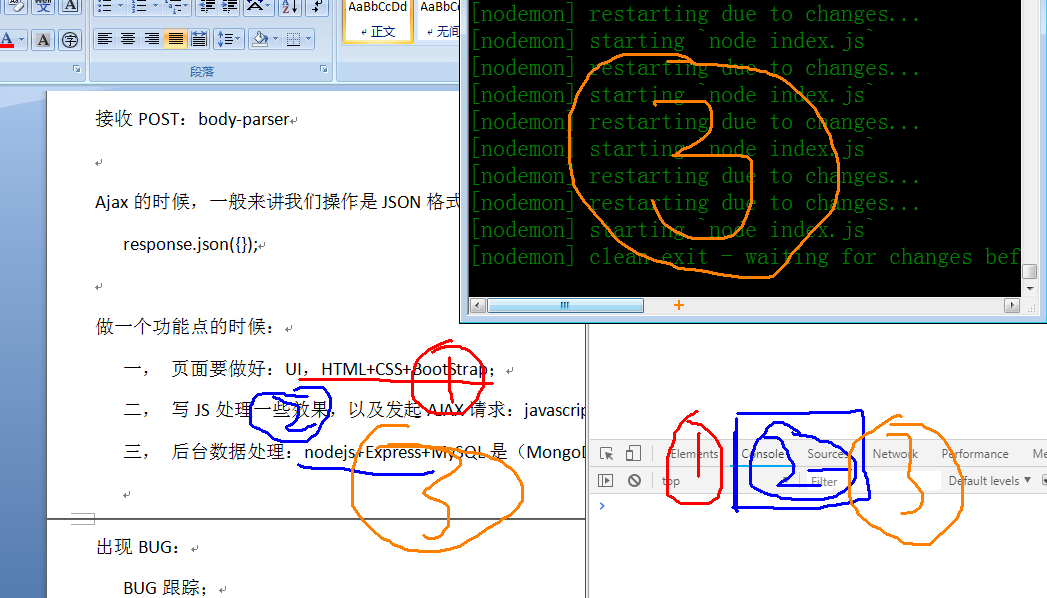
response.json({});

## 做一个功能点的时候：

1. 页面要做好：UI，HTML+CSS+BootStrap；
2. 写JS处理一些效果，以及发起AJAX请求：javascript+jQuery；
3. 后台数据处理：nodejs+Express+MySQL是（MongoDB，SQLServer）

出现BUG：

BUG跟踪；



## MVC：

模型-视图-控制器 项目架构思想

Model：模型，数据处理，增删查改，数据库工程师；

View：视图，页面，前端工程师；

Controller：流程控制,后台开发者；

表单直接提交改造成AJAX处理：

Submit 改成 button；

<button type=”**button**”>确定</button>

**Content-Type:**

application/x-www-form-urlencoded

## 作业：

实现登录；

Let memebers =

[‘刘桐’:’123456’, ‘曹华胜’:’123789’,’张明浩’:’789456’, ‘王小浦’:’123000’];

AJAX的方式实现登录：使用原生态的

账号不存在；

密码错误；

登录成功；



老王去买一包烟；

小卖部：没烟了

**等着**:对老王来讲是阻塞的;

小卖部的老板说：你回去把，有烟了我给你打电话，此时老王可以回家，老王是非阻塞；

打电话：异步；

老王自己时不时的去小卖部看一下：同步；

非阻塞：线程

阻塞：

异步：消息

同步：