# 上课笔记

Express：极简 web开发框架

路由：根据用户访问的地址来响应不同的内容：GET、POST、ALL；

中间件：对我们的请求头，响应头进行加工，use；

Express.static()：express本身就有极少数的中间件，可以指定访问路径，参数指定静态资源的存放路径；

Body-parser:接收post过来的数据，并把数据追加到req的body属性里面；

Ejs：模板引擎，把我们的数据和模板文件结合起来；

Mysql：负责操作数据库；

Async：变异步为同步；

MySQL：数据库管理系统 DBMS

数据库服务器，数据库管理系统，数据库，表，字段（属性，列），记录，数据；

使用SQL语句实现数据的操作：增删查改；

增加：INSERT INTO tablename(f1, f2, f3…) VALUES (v1, v2, “v3”…)

删除：DELETE FROM tablename WHERE 判断条件；

修改：UPDATE tablename SET f1 = nv1, f2 = nv2……. WHERE 判断条件；

查询：

SELECT \* FROM tablename WHERE 判断条件；

排序：

对查询结果进行排序；

SELECT \* FROM tablename WHERE 判断条件 ORDER BY f1 ASC|DESC；

ASC:默认值，升序；

DESC：降序

**SELECT** \* **FROM** students **WHERE** 1 **ORDER** **BY** **status** **ASC**, sid **ASC**

中文字段排序：

**SELECT** \* **FROM** students **WHERE** 1 **ORDER** **BY** **CONVERT**(stuname **USING** **GBK**) **ASC**

随机排序：

**SELECT** \* **FROM** students **WHERE** 1 **ORDER** **BY** **RAND**()

**SELECT** \* **FROM** students **ORDER** **BY** **rand**() **LIMIT** 1

指定查询范围：

SELECT \* FROM tablename WHERE 判断条件 LIMIT start, num

Start：开始位置，可以省略，表示从0开始

Num：取多少条，不是end

**SELECT** \* **FROM** students **WHERE** 1 **LIMIT** 2, 3

pagenum = 30 每页显示信息数

Page=1：**SELECT** \* **FROM** students **WHERE** 1 **LIMIT 0, pagenum**

Page=2: **SELECT** \* **FROM** students **WHERE** 1 **LIMIT 30, pagenum**

page=n**：SELECT** \* **FROM** students **WHERE** 1 **LIMIT pagenum\*（n-1）, pagenum**

**统计数量**：

SELECT COUNT(sid) FROM students WHERE 1

**SELECT** **COUNT**(sid) **AS** totalnums **FROM** students **WHERE** 1

## AJAX：Async Javascript and XML

XMLHttpRequest ---》 XHR

方法：

Open:建立对服务器的调用

Send：发起请求

SetRequestheader：设置请求头

属性：

onreadystatechange：状态改变事件

readystate：获取状态的，4表示完成

status：服务器响应状态

responseText：服务器的响应内容

MVC：模型-视图-控制器 开发模式

编程思想；

M：Model，模型，负责数据部分，DBer；

V：View，视图，页面，HTML+CSS，前端工程师；

C：Controller，控制器，流程控制，后台开发工程师；

~~MVVM：react angular vue~~

**HTTP协议**：主动权在客户端

建立连接

发起请求

服务器响应请求；

**断开**；

**http协议是无状态的**；

淘宝怎么知道这个单是我下的？？

会话控制：cookie、 session

Cookie：服务器保存到客户端的标识；

当用户再次发起请求的时候，请求头里面会自动加上cookie信息，把cookie信息发送到服务器端；

存储的大小是有限制的：4K；

所以说，安全性没有保障；

//启用cookie处理的中间件

let signed = 'www.jyqh.170815h';

server.use(cookieParser(signed));

SESSION：

存在服务器上；

没有大小限制；

安全性相对比较高；

依赖与cookie：存用户对应的SESSION的主键；

每个页面都可以使用；

//启用SESSION

server.use(session({

secret: signed,

name:'sessionID',

resave: false,

saveUninitialized: true,

cookie: {maxAge:7\*24\*3600\*1000}

}));

登录：

账号 密码

**验证码**：图形

文字 或者 数字信息；

输入框里面输入图形上的信息 进行验证；

随机生产字符串，然后把这个字符串显示图形上；

同时需要把该字符串信息保存到**session**里面；

SVG：图形格式，jpg，gif，png都是像素图形

SVG是矢量图，就是说不会失真；