复习：

HTTP协议 浏览器和web服务器之间通信的协议

浏览器发出请求，web服务器做出响应

请求方法 get/post....

请求-响应状态码

1\*\* 2\*\* 成功的响应 3\*\* 响应的重定向 4\*\* 客户端错误

5\*\* 服务器端错误

请求url

请求、响应的头信息

nodejs http模块 模拟浏览器发出请求 get

http模块 构建web服务器

createServer() listen(3000)

server.on('request',(req,res)=>{

req.method/url/headers

res.writeHead()/write()/end()

})

nodejs框架 express

创建web服务器 3.x **4.x**

路由功能：请求的方法(get/post..)，请求的url(/list)，响应的方法(回调函数)

app.get('/list',(req,res)=>{

req.method/url/headers/query/params

res.send('123')/sendFile(\_\_dirname+'/list.html')/redirect('/index')

})

1.请求的方法

post 只能使用表单提交的时候（现阶段）

get 可以通过表单提交，也可以在浏览器地址栏直接输入url

两者的区别？

post不会在地址栏暴露输入的数据，相对安全。get会在地址栏显示输入的数据，相对不安全。

一般对于一些安全较高的表单使用post，例如登录，注册...

对于一些安全较低的使用get，例如搜索关键字，查询

post执行速度相对于get，比较慢。

2.路由中的参数

服务器端配置路由参数

app.get('/list/:lid/:name',(req,res)=>{

//req.params 获取传递参数对象

})

练习：

路由： 购物车 /shopping get

把商品的价格(price)和商品的数量(num)以参数的形式传递给服务器，把接收的参数显示在浏览器中。

3.路由器

把相关的路由归纳到一起。

用户模块：注册，登录，列表，修改，删除/delete，检索

商品模块：列表，详情，修改，删除/delete

把商品路由器挂载到 /product下

路由器目的是便于管理路由，最终要用于web服务器。

4.require引入第三方模块机制

首先查找当前目录下是否有node\_moudules，如果没有往上一级目录寻找，直到根目录。

5.中间件

中间件就是express的函数，作用是为主逻辑所服务。中间件不能单独使用，必须要配合其它的中间件或路由实现一个完整的业务逻辑。

中间件拦截路由

app.use(fn); 拦截所有的路由

app.use('/list', fn); 拦截特定的路由(/list)

练习：

使用中间件实现计数器。

初始化一个变量用于记录访问次数——0；

在前置中间件中实现访问次数+1

在路由中(/user)，把+1的结果响应给浏览器。

**中间件分类**

应用级的中间件 app.use(fn)

路由级的中间件 —— 路由器的使用

app.use('/user', user)

内置的中间件

第三方的中间件

错误级的中间(课后自学)

①内置的中间件——static托管静态资源

sendFile只有在请求的时候，才会发送文件到浏览器。

app.use(express.static('目录路径'));

注意：可以同时托管多个静态目录，如果两个目录中有相同的文件名称，显示第一个目录下的文件。

②body-parser中间件的使用

引入中间件 const bodyParser=require('body-parser');

调用中间件

app.use(bodyParser.urlencoded({

extended: false

}));

获取post请求中表单的数据

req.body

课后任务：

(1)复习今天的内容，完成思维导图

(2)课后练习：

创建注册文件register.html(用户，密码，邮箱，电话);把这个文件作为静态资源托管，点击提交，使用post请求将数据发送给服务器；服务器使用body-parser中间件来获取数据，发送到浏览器“注册成功，你好+【用户名】”

(3)复习mysql中sql语句