**03**

复习:

|  |  |  |
| --- | --- | --- |
|  | 带路径 | 不带路径 |
| 文件模块 | require('./rect.js')  通常用于用户自定义模块 | require('querystring')  用于官方定义的核心模块 |
| 目录模块 | require('./1');自动引入1目录下的index.js或者查看package.json文件中的main属性 | require('2\_1')  要求放在当前目录下node\_module中 |

包和npm

包类似于模块

npm -->node package manage

工具模块

querystring ： parse stringify

url ： parse format

http://www.codeboy.com/products.html

fs 模块

查看状态

stat/statSync isDirectory() isFile()

对目录的操作

mkdir/rmdir/readdir

对文件的操作

readFile/writeFile/appendFile/unlink

http 模块

请求 - 响应

响应状态码：1xx 2xx 3xx 4xx 5xx

作为web客户端（请求），服务器端（响应）

http.get(url,(res)=>{

//响应的主体

res.on('data',(buf)=>{

console.log(buf.toString());

})

});

**今日学习目标**

**express 的使用**

**express 的路由**

**express的路由器**

**1.express的使用**

express是用于构建web服务器的一个框架

使用express构建web服务器

|  |
| --- |
| ①基于http模块，把express作为请求处理函数 |
| const express = require('express');  const http = require('http');  var app = express();  var server = http.createServer(app);  server.listen(3000,()=>{  console.log('服务器创建成功');  });  //简化版  http.createServer(app).listen(3000,()=>{  console.log('服务为创建成功');  }); |
| ②直接使用express构建web服务器 |
| var app = express();  app.listen(3000,()=>{  console.log('服务器创建成功');  }); |

**2.express中的路由**

http客户端发来消息，服务器端程序根据请求的方法和请求的url来确定一个响应的方法；

路由:请求方法，请求的url，响应的方法。

请求的对象

req.method 查看请求的方法

req.url 查看请求的url

req.headers 查看请求的头信息

req.query 查看请求的字符串，会被自动解析为对象

req.body 请求的主体（需要用到第三方插件）

req.param 查看路由的参数

响应的对象

res.send() 发送一个文本

res.sendFile() 发送一个文件，格式要求是绝对路径（\_\_dirname）

res.redirect() 重定向到另一个url

客户端向服务器端传参

①配置路由

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

req.params.lid//接受客户端所传递的参数

});

练习：把商品的价格和商品的数量以参数的形式传递

**3.路由器**

把相关模块下路由归纳一起，组成路由器

用户模块：注册，登录，用户列表，检索用户，删除用户，修改用户。

商品模块：

订单模块

练习：构建商品路由器（添加，删除，修改，检索）挂在到/product下