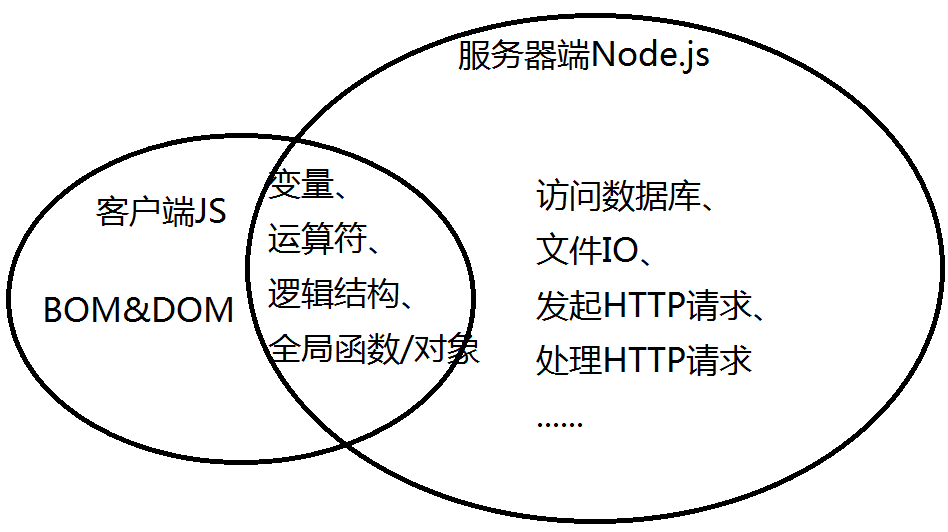
1. 使用服务器端编程语言：Node.js  
    Node.js是一个运行于V8引擎上的JS执行环境，提供了远超过经典客户端JavaScript API。  
    虽然名称里带着js，但是与经典的JavaScript完全不同——传统的JS运行于客户端浏览器中，主要用于实现页面交互效果；Node.js不属于客户端范畴，与Java/PHP/.NET语言类似，运行于服务器端，主要用于数据库操作、服务器端文件IO、网络访问等等。  
      
    英文官网：nodejs.org  
    中文镜像：nodejs.cn  
    使用Node.js的基本步骤：  
    ①下载并安装Node.js解释器：nodejs.org  
    ②在命令行界面中启动Node.js解释器，运行编写的JS代码即可  
    可以有两种运行方式：   
    方式1：交互式执行，输入一行代码回车，立即执行  
    方式2：脚本式执行，将所有要执行的代码写在一个.js文件中，一次性全部执行——推荐方式  
     
     
   二、使用Node.js创建服务器端项目的步骤

1.在服务器上安装Node.js解释器，确保在命令行中可以运行

Node -v

2.创建项目必需的文件夹，例如iweb\_backend，进入文件夹，创建Node.js项目必需的描述文件：package.json

可以手工创建该文件，并用记事本进行编辑；

也可以使用简化工具来创建此文件：npm init

1. 在项目根目录下，下载并安装必需的第三方模块，例如：

npm i mysql

npm i express

小知识：

可以使用npm i xxx自动从npmjs .com仓库中下载指定的第三方模块，也可以使用npm i自动读取package.json的指定的”依赖列表(dependecies)”查找必需的第三方依赖包，并自动从npmjs.com上下载

NPM与Node的关系：

Node.js的扩展模块有很多，为了简便开发者的查找和使用，有人专门为了创建了一个Node.js的扩展模块仓库（npmjs.com），并提供了一个命令行下载管理工具：Node Package Manager，简称NPM

后来版本的Node.js安装包中，已经默认把npm安装在一起，作为官方的扩展模块维护工具。

今晚任务：完成如下的服务器端数据API

1.API1.2：获取热门课程前12个GET /course/hottest

2.API3.1:获取讲师列表中的前5个 GET /teacher/list

3.API2.1：获取校区列表中的前5个 GET /school/list