框架概览

网站搭建通常分为两个部分：服务器搭建（服务器端）+网页制作（客户端）。网页制作部分主要包括网页内容、样式设计；服务器部分负责监听端口，根据请求与本地内容生成相应的响应。网站框架示意图如下图所示：

一种简单的情景是：客户端通过访问特定IP以及端口发送HTML请求，监听该IP端口的服务器（这里使用NodeJS搭建）会接收并解析出HTML请求，然后根据请求的指定路径生成相应的HTML文件，通过HTTP协议返回到客户端，客户端的服务器再根据收到的HTML文件显示或者再次发送请求（依据HTML, CSS, javascript语法）。使用HTML5+CSS+javascript+NodeJS的客户端服务器端通信流程的简易示例图如下：

选用NodeJS搭建服务器端是因为其同样基于javascript语言，脚本式的语言极易上手，同时异步并发的特性天然支持网站服务器搭建，借助于众多成熟的包可以轻松愉快的搭建出性能不弱的网站服务器。其运行在服务器端，通过监听反馈对用户的需求做出响应。

前端网页制作依赖于HTML, CSS, javascript三大模块，其特性被绝大多数浏览器支持，也是网页制作的标准工具。根据W3C的建议，HTML5负责网页内容（例如包括什么模块、模块的相对位置、文章内容等等），CSS负责样式设计（例如字体大小、模块之间的间距、按钮大小颜色等等），JavaScript负责网页的动态相应部分（即根据用户的操作实时做出响应，例如修改网页内容和样式等等）。网页部分运行在客户端，通过浏览器分析展示给用户。

注意到通过JavaScript，一些无需加密且不依赖于服务器端数据的动态响应，可以在客户端实现即时响应。所以若仅需要静态网页，或者简易的动态效果，并不需要实现服务器的部分；通过拷贝文件，浏览器访问本地网页就可以进行简单的展示。（笔者因为展示的是搜索引擎，需要与本地的数据库、文本文件等做交互，仍然需要服务器端的搭建。）

比较重要知识的checklist如下：

2. HTML的基础语法，如HTML5声明，utf-8编码指定，网页标题设定，网页主要语言设定等。

3. tag中常用attribute的使用方式，如href, class, id, onChange等。

4. CSS在HTML文件中的链接方式，以及书写方式（style attribute等）。

5. CSS索引HTML文件中模块的方式，例如通过class，通过tag，通过id，以及多标签间 交并继承 等语法。

6. inline（如span）和block（如div） display的区别。

7. CSS的box model，理解margin和padding的区别。

了解HTML和CSS的基础语法之后，就可以制作出简单的网页了。如需自行设计的话，网上有一些使用dreamweaver，bootstrap等工具进行制作的教程。另一种方便简单的方法是在网上搜索类似的网站，下载好HTML以及CSS文件之后，对其内容格式做相应的修改，适用于无收益学习用的简单展示网页制作。

网站修改的时候强烈建议两个极其方便的工具：Visual Studio Code和Google Chrome。前者是微软出的跨平台编辑器，安装好HTML以及CSS的相关语法工具包后，即可实现语法补全、关键词加亮等功能；令笔者十分满足的一个功能是其支持代码格式的自动调整，部分下载好的HTML文件中所有代码都顺序排列在一行内，不易于代码阅读定位与调整，VS code可以一键将代码调整为易于阅读的格式，非常方便。Google Chrome作为前端工程师的神器，其使用方法入门教程网上有很多，这里简单介绍常用的方法：F12进入调试模式；其左上角的箭头可以点选网页内容之后定位其相关代码的具体位置；在elements页面中显示有HTML和CSS源文件，鼠标移动到HTML文件的相应模块时会在网页上对相关内容进行加亮，而且通过不同颜色区分margin/border/padding等，具体数值还会在elements中标识；点选模块之后会在elements页面中显示其相关的CSS代码，清晰的表示了各个属性的继承覆盖，并且可以直接修改查看即时效果；Console页面在JavaScript调试时常用；Network页面记录了客户端与服务器端的网络通信，在服务器代码调试时常用。