前端

**HTML：**一种标记语言

**CSS**：定义样式

fixed 不动

left top 确定位置

weight height 确定宽高

background 背景颜色

（在 html 文件中用 <script> 关联进来就可）

HTML5与XHTML：

HTML5是一个通行标准；

XHTML是XML和HTML1的杂交，语法要求严格，与HTML有所不同

**JavaScript：**

HTML+CSS=一个静态的页面，于是有JavaScript（js）来给页面添加动态效果（如鼠标移动到哪里弹出小窗口）

1. →鼠标悬停到标签上时创建一个新的 <div> 小窗口
2. 用 JS 向知乎服务器发送一个请求，得到这个小窗口应该显示的数据，放在这个小窗口里（这就是所谓的AJAX，不用刷新就能与服务器进行交互，更新页面的一小部分~）

浏览器拿到这样的代码，就会解析并实现出相应的效果，即浏览器脚本。

（将XXX.js在 html 文件中用 <script> 关联进来就可）

后端

**Web Server 和 Web Services：**

浏览器给服务器发一个请求，服务器不是一看就知道怎么响应的。首先这些请求和响应要有一个通用的写法，也就是要有一个协议，常用的是 HTTP 协议。

HTTP 响应里还包括很多东西，比如 Content-type 表示服务器发过来的文件类型是什么（文本？动画？图片？音频？），这样发过去了人家浏览器好知道怎么展示给用户看。人家服务器怎么知道按协议要写什么东西进去呢，这就是 Web Server 干活的时候了。

客户端与服务器的交互：

1. 浏览器输入地址
2. AJAX
3. 微信之类的与服务器的交互
4. 网游与公司服务器
5. 搜索引擎用来搜集网页信息的程序（爬虫）与网站服务器
6. 地址+怎样访问对应服务器+权限，也可以自己去跟人家服务器交互（如微博API）
7. 服务器可以是大型机也可是个人电脑
8. 客户端可以是电脑、手机、平板、智能设备软件
9. 可以传HTML也可以传经过逻辑处理后生成的字符串或封装的数据

对于这些交互要规定一下，需要指明一种协议（比如 HTTP/HTTPS）和一种数据封装格式（比如 HTML/XML），Web Server 提供的 Web Service，指的就是这种协议+格式的交流体系

不过 Web Service 的生态系统和 HTML 的标准不一样，用户可以选择的协议和数据封装格式更多，普通的网站访问用的 HTTP + HTML 只是其中一种，一些封闭系统内的交流还可以自己定义一个协议和格式来用（比如 QQ）。

除了提供 Web Service， Web Server 还会兼顾很多功能，包括提供缓存，平衡负载，这样在访问量比较大的时候能有有条不紊地接客。常见的现成的 Web Server 有开源的 Apache、Nginx和微软的IIS，你也可以用一些工具（比如 Node.js ）自己定制一个。

**PHP ，服务器脚本，Web Framework：**

服务器脚本：

【生成如Web Service传输的数据，交由Web Server按协议套好相应格式（如不同用户登录后界面格式相同，具体内容不同），返回给客户端】

PHP：

写服务器脚本的语言

Web Framework：

我们在写服务器脚本的时候，通常还会用个同语言写的 Web Framework 来处理各种细节，防御一些常见的攻击，提供跨站认证（比如用已有的微博账号注册其他网站）的接口，利用cookie处理登陆状态和用户设置，生成网页模版之类的。

一个普通网站访问的过程

简单概括一下，对于我们普通的网站访问，涉及到的技术就是：

1. 用户操作浏览器访问，浏览器向服务器发出一个 HTTP 请求；
2. 服务器接收到 HTTP 请求，Web Server 进行相应的初步处理，使用服务器脚本生成页面；
3. 服务器脚本（利用Web Framework）调用本地和客户端传来的数据，生成页面；
4. Web Server 将生成的页面作为 HTTP 响应的 body，根据不同的处理结果生成 HTTP header，发回给客户端；
5. 客户端（浏览器）接收到 HTTP 响应，通常第一个请求得到的 HTTP 响应的 body 里是 HTML 代码，于是对 HTML 代码开始解析；
6. 解析过程中遇到引用的服务器上的资源（额外的 CSS、JS代码，图片、音视频，附件等），再向 Web Server 发送请求，Web Server 找到对应的文件，发送回来；
7. 浏览器解析 HTML 包含的内容，用得到的 CSS 代码进行外观上的进一步渲染，JS 代码也可能会对外观进行一定的处理；
8. 用户与页面交互（点击，悬停等等）时，JS 代码对此作出一定的反应，添加特效与动画；
9. 交互的过程中可能需要向服务器索取或提交额外的数据（局部的刷新，类似微博的新消息通知），一般不是跳转就是通过 JS 代码（响应某个动作或者定时）向 Web Server 发送请求，Web Server 再用服务器脚本进行处理（生成资源or写入数据之类的），把资源返回给客户端，客户端用得到的资源来实现动态效果或其他改变。