

Web 建站技术中，HTML、HTML5、XHTML、CSS、SQL、JavaScript、PHP、ASP.NET、Web Services 是什么？

建站有很多技术，如 HTML、HTML5、XHTML、CSS、SQL、JavaScript、PHP、ASP.NET、Web Services、浏览器脚...显示全部

关注问题

写回答


15 条评论

分享


邀请回答

...

查看全部 77 个回答



知乎用户
产品狗，机器学习初学者，略懂梅花
1,555 人赞同了该回答



爱学习，爱编程，爱咖啡可乐
爱挑战，爱钻研，爱打游戏
爱早起，也爱工作到深夜
我擅长技术，崇尚简单和懒惰
我神秘而孤僻，沉默而爱憎分明
DON'T PANIC! I'm a programmer

先普及用户通过浏览器访问网页的过程：

网页内容是通过服务器运算得出的结果，将结果(网页代码)传输给浏览器，网页代码再通过浏览器运算(计算、渲染)，最终展示在用户的眼前的。

至此，我们知道了有2个运算过程：

- 服务器运算
- 浏览器运算

而让电脑(网站服务器、你的个人电脑)乖乖执行运算，就需要编写程序脚本即程序代码。而编写代码的过程就叫做：编程。编写代码的人叫做：程序员(又戏称：程序猿、攻城狮)。

所以，由运算演变出：

- 服务器运算 -> 服务器脚本
- 浏览器运算 -> 浏览器脚本

在行业内，将 服务器运算 称为 后端；将浏览器运算 称为 前端；

后端更靠近服务器，前段更靠近用户，这样比较好记忆。

然后我们再来按照上述两个类别，对其他的东西分类：

- 服务器脚本(后端)：

PHP、ASP.NET

- 浏览器脚本(前端)：

HTML、CSS、JavaScript

HTML5、XHTML我们暂且将其统称为：HTML

细心的就发现了，SQL 没有提到。有了上面替



下载知乎客户端
与世界分享知识、经验和见解

关于作者



知乎用户
产品狗，机器学习初学者， ...

被收藏 1,197 次

编程，算法研究

2,131 人关注

知乎用户 创建

一些互联网知识

1,197 人关注

遥同学 创建

垄断对和错

174 人关注

zhen-liang 创建

前端攻城狮的成长之路

80 人关注

知乎用户 创建

Python&Flask

68 人关注

知乎用户 创建

相关问题

HTML、CSS、JavaScript、PHP、MySQL 的学习顺序是什么？

240 个回答

html css php js之间的关系？

10 个回答

为什么我在学习HTML\CSS的时候很快就能上手，可是JS我已经接触了2个月了，还是无从下手？

20 个回答

HTML & CSS 现在还多人用吗？

89 个回答

JSP 在走向衰落吗？以后是 HTML5、JavaScript 和 CSS 的天下吗？

9 个回答

相关推荐



陈丹青荐：中国美学80讲

曾荣荣

共 80 节课

试听

SQL，即数据库，用来存储和查询用户的数据，比如：

- 1、存储你注册的帐户信息(用户名、密码)；
- 2、存储你在知乎上回答的问题(标题、内容)；
- 3、查询本回答相关的评论。

SQL是安装在服务器上的，暂且也将它纳入后端范畴。

根据题主的问题，如果学习的话，有这么两种常见搭配：

- 1、PHP 搭配 MySQL；
- 2、ASP.NET 搭配 SQLServer。

MySQL和SQLServer都是SQL软件，都可以用来存储、查询数据。
至于为什么这么搭配你先不用管，总之上帝说要有光，那就**先光着**。



我们再来模拟一下访问网站的**用户登陆**过程，以PHP为例(ASP.NET同理)：

- 1、用户输入 用户名、密码 后 点<登陆>按钮；
- 2、后端程序开始运算：PHP运行，连接MySQL，在数据库中查询用户名、密码是否正确；
- 3、将运算数据返回给用户的浏览器，包含(HTML、CSS、JavaScript)；
- 4、浏览器将接收到的前端数据(HTML、CSS、JavaScript)，运算(解析)出来，成为网页。

最后还有问题对吧？别急。

HTML、CSS、JavaScript又分别是什么呢？

HTML就是网页，一个静态的网页。(做任何操作页面上的东西都不会动，可以简单理解为：**没有特效**)

CSS、JavaScript就是完成「特效」的。

JavaScript 这个单词 好长 我打的好累，我为了偷懒，后面用缩写：JS

CSS：负责完成页面的美观度(颜色、布局)

JS：负责完成页面的操作需求

比如：点赞的时候，JS负责改变页面上的赞同按钮效果，同时告诉服务器：我赞同了！

=====

细心的又发现了，Web Services 我没有提到！
很多时候我们需要一个**数据**或者一个**功能**，而我们自己无法完成。

毛栗子A：天气预报

这个时候我们**自己的服务器上没有真实的天气数据**，这时就需要使用其他网站提供的「天气Web Services」。

而这个Web Services，提供一个**数据型**服务。

板栗B：发送短信验证码

这个时候我们**自己的服务器不能发送短信**，这时就需要使用其他网站提供的「短信Web Services」。

而这个Web Services，提供一个**功能型**服务。



★★★★★ 925 人参与



高效前端：Web 高效编程与优化实践
1,722 人读过 阅读

刘看山 · 知乎指南 · 知乎协议 · 应用 · 工作

申请开通知乎机构号

侵权举报 · 网上有害信息举报专区

违法和不良信息举报：010-82716601

儿童色情信息举报专区

联系我们 © 2018 知乎

之间有了统一的接口标准，有了互通的可能。

简单讲，同毛栗子A，我们知道了：

1、当你的网站开放一个Web Services接口，这个时候就可以允许别人查询到你数据库的数据。

2、当你的网站使用别人的Web Services时，这个时候就可以查询到别人的数据。

=====

表示层

访问 `http://www.victim.com`



呈现 HTML

Web 浏览器/呈现引擎

逻辑层

加载、编译并执行 `index.asp`

脚本引擎



发送 HTML

编程语言：C#、ASP、.NET、PHP、JSP 等

存储层

执行 SQL

RDBMS



返回数据

数据库：MSSQL、MySQL、Oracle 等

然后这一整套体系，有前端(表示层)、有后端(逻辑层)、有数据库(存储层)，一个独立的网站就开始运作了。

甚至还可以用Web Services去查一下别人网站的数据，综合一下展示出来，是不是感觉很屌？

屌个屁啊，此行有风险，入行需谨慎！！

特别鸣谢：[@马先森](#)

以上

更新

有人问我风险是什么，风险是：

你以后可能也许会像这样

▲ 1.6K

89 条评论

★ 收藏

♥ 感谢

收起

<https://www.zhihu.com/question/22689579/answer/27366799>

3/12



这样



这样



或者这样



总之最后会成为这样



编辑于 2017-01-20

更多回答

张秋怡
@ alinode

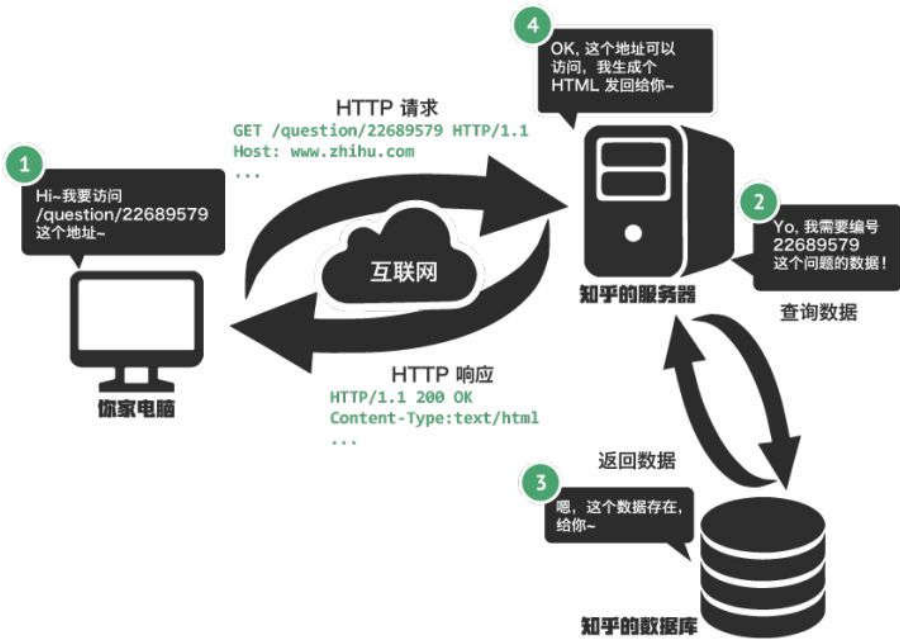
15,623 人赞同了该回答

一个学期前我也和楼主差不多不知道这些都是啥，一个学期之后差不多都弄懂了，来讲讲自己的理解吧 > < 因为接触的时间不是很长，有错误的地方欢迎指正~

首先要知道网站访问大概是个过程：
假设你在浏览器地址栏输入这个问题的地址

`http://www.zhihu.com/question/22689579`

访问过程和下图差不多，浏览器和服务器交流，服务器和数据库交流（有时候数据库就在服务器那台机子上）



HTML 与 CSS

你家公司拿到一个 html（就是上图 HTTP 响

1.6K

89 条评论

收藏

感谢

收起

染。HTML 就是一种标记语言。类比一下,大家在论坛上经常会用一些代码来添加富文本内容,举一些栗子:



- [img]图片地址[/img]

用来贴图

- [url]超链接[/url]

用来贴地址

- [del]呵呵[/del]

表示标签里的文字应该被删掉(默认添加删除线效果)

这种论坛代码也是一种标记语言,HTML的作用就和它差不多。服务器返回给你的html文件里面,写的是一些代码,大概是这样的:

```
1 <!DOCTYPE html>
2 <html lang="zh-CN" dropEffect="none" class="no-js" >
3 <head>
4 <meta charset="utf-8" />
5
6 <meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1" />
7
8 <title>
9
10 Web 建站技术中, html、html5、XHTML、CSS、SQL、JS、PHP、ASP.NET、Web Services 区别是什么
11
12 </title>
13
14 <meta name="apple-itunes-app" content="app-id=432274380, app-argument=zhihu://i
15
16
17
18
19 <meta http-equiv="mobile-agent" content="format=html5;url=http://www.zhihu.com
20 <meta id="znonce" name="znonce" content="05fe1c012d594aaaf65ab387d9b25ee">
21 <link rel="apple-touch-icon-precomposed" href="http://static.zhihu.com/static/i
22 <link rel="apple-touch-icon-precomposed" href="http://static.zhihu.com/static/i
23 <link rel="apple-touch-icon-precomposed" href="http://static.zhihu.com/static/i
24 <link rel="apple-touch-icon-precomposed" href="http://static.zhihu.com/static/i
25 <link rel="apple-touch-icon-precomposed" href="http://static.zhihu.com/static/i
26 <link rel="apple-touch-icon-precomposed" href="http://static.zhihu.com/static/i
27
28
29 <link rel="shortcut icon" href="http://static.zhihu.com/static/favicon.ico" ty
30
31 <link rel="search" type="application/opensearchdescription+xml" href="http://s
32
33
```

浏览器拿到这些代码之后,分析一下给你渲染好页面显示出来,如果不用css,效果是这样的,按照浏览器默认的样式显示出列表、图片、超链接、输入框、按钮等等:



Web 建站技术中, html、html5、X

是不是觉得默认样式有点看瞎狗眼呢.....所以很多时候我们需要自定义样式,现行通用的规定样式的语言是CSS,我们可以用它写一些定义样式的代码,然后在html文件里用一个<link>标签把这些规定样式的CSS代码与表达内容语义的HTML代码关联起来,然后你就能看到符合人类正常审美的页面了:



CSS 代码的格式基本是

属性:值

比如知乎顶上那个蓝色的导航条，它的 CSS 大约是这样的

```
元素 {
}
.zu-top {
  position: fixed;
  left: 0px;
  top: 0px;
  z-index: 20;
  width: 100%;
  height: 45px;
  background: -moz-linear-gradient(center top ,
    rgb(8, 110, 213), rgb(5, 93, 181)) repeat-x scroll 0%
    0% rgb(7, 103, 200);
  border-bottom: 1px solid rgb(4, 78, 151);
  box-shadow: 0px 1px 2px rgba(0, 0, 0, 0.25), 0px
    1px 0px rgba(255, 255, 255, 0.15) inset;
}
```

第一个属性对应的代码翻译成人话，就是它的位置（position）应该是不动（fixed）的，浏览器会兢兢业业地去实现这段代码要求实现的效果，所以你在页面上翻来翻去顶上那个导航条都会死死地黏在窗口顶部不跟着滚动。再拿里面的其他几个属性做栗子解释一下就是：

- left 和 top
为零指明这个导航条要紧贴着窗口的左上角
- width 和 height 指定这个导航条的宽和高
- background
指明这个导航条的背景是一种渐变的蓝色

浏览器就会根据这些 CSS 代码，“画”出对应的样式。

HTML 5 与 XHTML

像人的语言一样，网络上的网页里的 HTML 代码也不一定是标准的，好比有时候你发音不太标准，别人会去猜测你说的到底是什么一样，有些时候前端程序猿不小心写错了 HTML，浏览器也会试图猜测他们原来想写的是什么，做对应的渲染，而猜是要有一个常识做依据的。加上有些浏览器支持一些标签，有一些又不支持，以及其他混乱的情况，为了防止大家鸡同鸭讲，我们需要对 HTML 代码里能有什么标签，标签怎么写，标签可以有什么属性这些东西有一定的共识，建立一个通行的标准，HTML5 就是其中一个比较新的标准。这个标准新加了很多可以用的标签和属性，然后各大浏览器也吭哧吭哧按这个标准去实现了很多这些新加的标签和属性，本来前端程序员要写一堆代码去实现的效果，现在浏览器都给你实现好了，只要写两三行，调用一下浏览器给你实现的部分就能搞定，简单愉快，所以很多人都在热情地推广这个标准~（当然新标准也不可能是完美的，总会有一些问题，怕跑题这里按下不表~）

至于 XHTML，就是 HTML 的近亲 XML 和 HTML 自己的杂交品种，对语法要求比较严格，并且为了兼容 XML，在语法上与 HTML 有一些不同~

JavaScript 与浏览器脚本

有了表示内容和语义的 HTML，规定样式的 CSS 还是可以有一些动画的，不过这个跑题了

调的网页怎么能展现我大智人种族的创造性口胡！于是我们有了 Javascript（JS）来给页面添加一些动态的效果，比如知乎问题的标签，鼠标移上去会弹出一个小窗口，这个就是 JS 实现的效果啦。



浏览器都会帮你实现一些 JS 可以用的工具（函数，对象什么的），你只要写一些 JS 的代码，保存在 xxx.js 里，在 html 文件中用 `<script>` 关联进来就可以用了，像上图这个效果应该就包括了

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~）

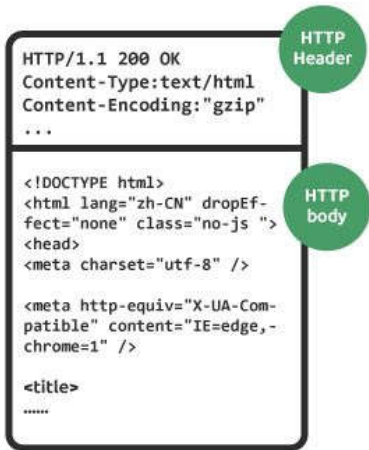
浏览器拿到这样的代码，就会解析并实现出相应的效果，楼主问的浏览器脚本，指的就是这样子的代码。其实用来写浏览器脚本的，也不是非得JavaScript不可，当年还为到底用什么语言写这种代码有过一段战火纷飞的时期，但现在尘埃落定，各大浏览器都默认了：请用 JS 写这些动态效果的代码给我解析~

以上就是前端部分的内容，下面简述一下后端的東西吧 >

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。
像最前面的图，服务器的响应写了一个状态码 200 OK，是 HTTP 协议里约定俗成的一个东西，服务器写 200 OK 在响应里，表示“你请求的这个东西我有”，如果是404 Not Found，就是“你请求的这个东西我这里没有”。
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。

形象化一下HTTP响应，大概就长这样：



再上个锤子，浏览器和服务器之间请求响应的过程大致是长这样的。右下角的那些东西就是由 Web Server 生成的（服务器脚本可以做一些改动，

▲ 1.6K

89 条评论

★ 收藏

♥ 感谢

收起



再比如说很多时候你访问一个网站，浏览器里输的地址并没有写明你请求的文件，比如这个问题的地址是：

`http://www.zhihu.com/question/22689579`

但知乎的服务器其实返回了一个html给你，服务器怎么知道这个地址对应要返回什么样的 html 代码给你的？也是 Web Server 干的活。

除了浏览器输地址敲回车这种赤裸裸的访问，客户端与服务器的交互还有很多种，比如：

- 前面提到的用 JS 完成的 AJAX，有点像浏览器和服务器之间的悄悄话~

还有其他应用软件与服务器的交互，比如：

- 微信、QQ 与腾讯的服务器的交互
- 网游客户端与网游公司服务器的交互
- 搜索引擎用来搜集网页信息的程序（爬虫）与各种各样的网站服务器的交互
- 只要你知道用什么地址访问、怎样访问人家的服务器，并且有相应权限，你也可以自己写一些程序去和他们的服务器交互（比如用[微博API](#) - [新浪微博API](#)获取微博，开发第三方应用或者做数据分析）。

从这些栗子里可以看出，客户端与服务器的交互的主体、客体、载体是五花八门的：

- 服务器可以是大型机也可以是个人电脑，只要能跑相应的程序就行
- 客户端像前面举的栗子一样，可以是各种软件，而且这些软件不一定运行在个人电脑上，也可以是手机、平板、智能穿戴设备等等
- 有时候不是传生成好的 HTML 或者其他服务器上已经有的文件，而是传输经过一定逻辑处理后生成的字符串或者其他各种封装好的数据

像前面提到的 HTML 需要有一定标准一样，为了防止混乱和鸡同鸭讲，我们又需要先对这些机器需要怎么交互达成一定共识，再让它们进行交流。人与人之间通信，需要先有一种大家都认识的写法（比如简体字/繁体字）和一种彼此都懂的语言（比如普通话/广东话）。要让这些形形色色的机器能够通过网络进行交互，我们就需要指明一种协议（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）。

Web Service 传输的数据再经由本地客户端（浏览器、QQ/微信，网游客户端等）的分析渲染，就能够以普通人能够理解的形式展现出来。此外还有一些 Web Service 并不是为普通用户设计的，像前面提到的微博API，是用来给程序猿进行二次开发的~

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。因为 Web Server 需要比较好的性能，所以投产时用的 Web Server 通常是C/C++/Java写的，但是其实很多语言都可以写，而且配合上语言底层的优化和好的模型，其他语言写的 Web Server也可以有不错的表现。

PHP，服务器脚本，Web Framework

开头那张图里服务器接到请求之后可以给访客发送对应的文件，但21世纪的服务器怎么可能只会“接请求-发文件”这么弱智的一招呢，人家还可以处理你上传来的文件的！还可以接受你发过来的各种请求，去操作服务器本地的文件or数据库的！要干这些事，自然服务器那边也少不了要有代码了，这些代码就是服务器脚本。前面说的 Web Service 传输的数据，主要也是由服务器脚本生成，再交由 Web Server，按照某种协议套好整个响应的格式，返回给客户端的。

同一个网址，每个人看到的页面不一定是一样的，比如知乎首页的网址都是



但是没登陆和登陆之后看到的東西不一樣，登陸之後每個人看到的導航欄的用戶信息，關注的動態，都不一樣。服務器腳本可以對這些不同的狀態，生成不同的頁面，交給 Web Server 返回給瀏覽器。

知乎的主頁給大家看到的 html 整體來說是不差的，都有導航欄，左邊是關注的動態，右邊是廣告和邊欄，每一塊的整體構造大同小異，只是一些地方內容有所區別。服務器腳本就是利用已知的數據，在這些因人而異的地方填入相應的內容，生成給每個人看的頁面。

比如我的主頁，導航欄右邊的頭像和名字跟別人看到的不一樣，就是因為這塊地方有一個放圖片的 `` 標簽和一個寫名字的 `` 標簽，服務器腳本在查詢本地的數據之後給我返回的頁面里 `` 的標簽填了我頭像的圖片鏈接，`` 標簽里填了我的名字，給別人的頁面就填其他鏈接、其他名字，這樣每個人看到的頁面就不一樣了。



```
aria-activedescendant="">
<span class="name">
  张秋怡
</span>
</img>
```

PHP 就是一種常見的用來寫服務器腳本的语言，其實只要是能拿來寫大家傳輸數據的通用接口（CGI）的语言都可以用來寫服務器腳本（也就是說幾乎所有編程语言都可以寫 = =b），只是因為現成工具的豐富程度和專攻程度不一樣，所以有一些语言在寫服務器端腳本的時候會比較熱門。

為了方便，我們在寫服務器腳本的時候，通常還會用個同语言寫的 Web Framework 來處理各種細節，防禦一些常見的攻击，提供跨站認證（比如用已有的微博賬号注册其他网站）的接口，利用 cookie 處理登陸狀態和用戶設置，生成網頁模版之類的。如果你用 C# 或者 Visual Basic 寫服務器腳本，就可以用 ASP.NET 這個框架實現這些功能，幫你省點麻煩。不過現在不少人是反過來了一個好用的 Web Framework 去選擇它對應的服務器腳本语言的。

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. 用戶操作瀏覽器訪問，瀏覽器向服務器發出一個 HTTP 請求；
- 2. 服務器接收到 HTTP 請求，Web Server 進行相應的初步處理，使用服務器腳本生成頁面；
- 3. 服務器腳本（利用Web Framework）調用本地和客戶端傳來的數據，生成頁面；
- 4. Web Server 將生成的頁面作為 HTTP 響應的 body，根據不同的處理結果生成 HTTP header，發回給客戶端；
- 5. 客戶端（瀏覽器）接收到 HTTP 響應，通常第一個請求得到的 HTTP 響應的 body 里是 HTML 代碼，於是對 HTML 代碼開始解析；
- 6. 解析過程中遇到引用的服務器上的資源（額外的 CSS、JS代碼，圖片、音視頻，附件等），再向 Web Server 發送請求，Web Server 找到對應的文件，發送回來；
- 7. 瀏覽器解析 HTML 包含的內容，用得到的 CSS 代碼進行外觀上的進一步渲染，JS 代碼也可能会对外觀進行一定的處理；
- 8. 用戶與頁面交互（點擊，懸停等等）時，JS 代碼對此作出一定的反應，添加特效與動畫；
- 9. 交互的過程中可能需要向服務器索取或提交額外的數據（局部的刷新，類似微博的新消息通知），一般不是跳轉就是通過 JS 代碼（響應某個動作或者定時）向 Web Server 發送請求，Web Server 再用服務器腳本進行處理（生成資源或寫入數據之類的），把資源返回給客戶端，客戶端用得到的資源來實現動態效果或其他改變。

注意這只是小網站里比較常見的模型，大網站為了解決規模問題還会有很多處理，每個環節都會有一些細微的差異，中間還會使用各種各樣的工具減輕服務器的壓力，提高效率，方便日常維護~

延伸阅读 —— 那些看花眼的名词

為了方便調試，很多 Web Framework 會自帶一個簡單的 Web Server，或者有些 Web Server 會自帶一個簡單的 Web Framework，實際部署到服務器上開放使用的时候為了性能或者安全等多方面的考慮，可以把內置的 Web Server 換成其他的，比如

Apache 或者 Nginx（举个栗子，知乎用的就是 Apache 和 Nginx）



见知乎使用了哪些框架和开源库?)。如果是开源的东西，还可以在遵守开源协议的前提下自己改一下再用~

因为后端不像前端已经有 HTML + CSS + JS 这样的既定事实标准，服务器脚本与 Web Framework

的选择很多，所以新手会听到很多眼花缭乱的技术名词的地方多在这里~

举一些栗子，早年常见的服务器端语言有：

- 开源的 PHP
- Sun 公司的 JSP 中使用的 Java
- 微软的 ASP 中使用的 VBScript

现在在这方面的应用热起来的语言有

- Python，对应常见的 Framework 包括知乎和Quora有用到的 Tornado（其实是自带 Framework 的 Web Server），社区很成熟的 Django（用户包括 Instagram、Pinterest）等
- Ruby，一般都用 Rails 这个 Framework，用户包括 Github、早期的 Twitter 等
- 逆天的 JavaScript，有了 Node.js 这个平台，Web Server、服务器脚本和浏览器脚本全都可以用 JavaScript 来写.....Node.js上最常用的 Framework是Express
- 微软家的则跟着 ASP.NET 转移到了C# 或者 Visual Basic
- Erlang，擅长大规模的并发，不少游戏公司拿来写服务器，靠几十个工程师支撑几亿用户的 WhatsApp也是用的这个~

几种常见的架构包括：

- LAMP = Linux + Apache + MySQL + PHP（P还可能是Python或Perl。有时候L会改成 W=Windows。）, 也就是服务器上的操作系统是 Linux，Web Server 用 Apache，数据库用 MySQL，服务器脚本用 PHP，这些都是开源技术，网站起步时用起来的成本会比较低，所以是普通网站里非常常见的架构（虽然对于发展得很大的网站会遇到很多瓶颈），Facebook 就是这种，淘宝也曾经是。
- J2EE，Java 世界的架构，通常是企业用的（银行、大型公司.etc），比较常见地还会搭配一种 UNIX 做操作系统，Apache 做 Web Server，Tomcat 转换 JSP 到 Java 给服务器程序用（其实它也自带 Web Server），Oracle 数据库等等。不一定拿来建站，常常用来提供企业里的各种需要用到网络的业务。我们学校教务系统就是用J2EE做的=。= 淘宝现在也是从LAMP转型到了这个。
- ASP.NET，微软家的架构，通常会搭配 Windows Server 操作系统，SQL Server 数据库，IIS 做 Web Server。StackOverflow和京东（曾经）就是这个架构。
- 神奇的MEAN架构，MongoDB做数据库，Express做 Web Framework，Angular 做前端的 JavaScript 框架，Node.js 用于编写 Web Server。神奇之处在于这几个东西的语言都是 JavaScript（MongoDB的实现不是，但与外界沟通用的语言是）。因为是比较新的架构，还有待时间的考验，不过被很多人（尤其是靠 JavaScript 吃饭的前端程序猿们）热切关注。
- 一般来说重点不在技术而且在乎成本的新网站比较喜欢用 LAMP，重视安全稳定和速度的企业和机构喜欢 J2EE，想省事的网站喜欢 ASP.NET，比较 Geek 的网站和创业公司喜欢折腾各种 Python、Ruby、Node.js世界的东西，Google 这样现成的技术都解决不了需求的超大型网站就自己折腾解决方案。

虽然可以用的语言和所属体系五花八门，其实服务器端程序要做的事情本质上都差不多的，就好比自然世界中要表达“吃过了没”这句话的意思，你可以用各种各样的语言在各种各样的场景里表达出来~

编辑于 2014-12-10

▲ 15K ▼ 605 条评论 分享 ★ 收藏 ♥ 感谢 收起 ^



胖胖小
码农

563 人赞同了该回答

就说下前台这三个，html是名词，css是形容词，javascript是动词。三个互相配合才是一句话

发布于 2015-01-07

▲ 563 ▼ 40 条评论 分享 ▲ 1.6K 89 条评论 ★ 收藏 ♥ 感谢 收起

查看全部 77 个回答

