**Web开发基础知识**

## | 概览

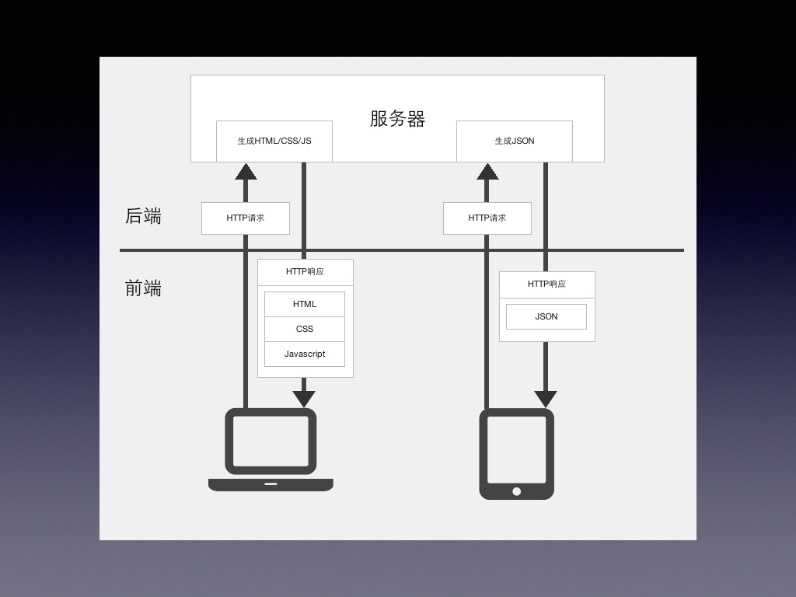
以下这张图就是元一分享的干货内容，它基本涵盖了一个初级码农需要知道的所有基础入门知识。但这张图的目的并非用来吓人，这其中的所有技术名词，将以最通俗易懂的方式串联起来—— 即我们上网时的慢动作解析：打开一个网页或 App，这背后都运用了那些技术来让这个网页和 App 的内容呈现在浏览器和手机上；驱动这些动作背后的技术名词都是什么，各自有着怎样的优缺点，彼此间是如何协作和运转的，以及怎样合理地评估技术能力和开发难度。当然，元一也推荐了丰富的学习资料。

这中间涉及的技术知识，前端包括 HTML，CSS，JavaScript，jQuery 以及 Bootstrap ；后端包括 HTTP 服务器，后端编程语言，数据库以及 Cookie 和 Session；移动开发分为原生，混合式，HTML5，以及不同的移动端技术选择在功能和开发成本上的比较。



## | 什么是前端？什么是后端？二者是如何配合运转的？

前后端的划分，可以简单地理解为凡是运行在用户设备上的技术都可以称为前端技术（ 比如 HTML / CSS / JS，甚至移动设备的 Obj-C / Swift ）；而后端的作用就是负责将这些东西封装在 HTTP 的数据包中然后通过网络传送到前端。当然除了这些前端文件，后端还有一个更重要的职能，即保存和提供用户数据，比如移动端常见的 JSON 就是目前最流行的在后端和前端之间传输的一个文件格式。

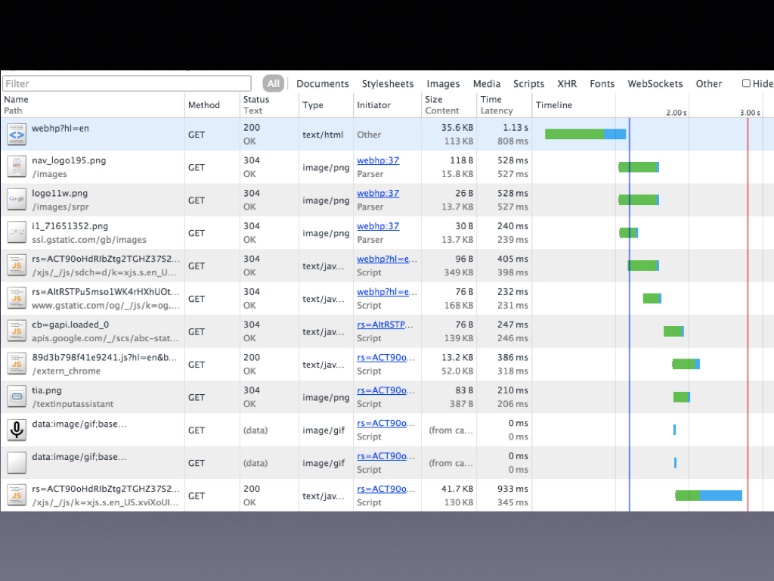


## | Web 前端的运行逻辑

假设我们要访问 Google，从我们在浏览器输入 Google.com 到最后这个页面出现在眼前，这其中涉及许多前端的技术反应和代码组合，总体而言可以简化为两步：

1. 浏览器向 Google 的服务器发送了一个请求。

2. 服务器收到了一个 HTTP 响应，这个响应中就包含了执行这个命令所需要的所有资源（注：可以通过 Chrome 浏览器的开发者工具来进一步观察 HTTP 协议的运行情况；下图为 Google 的 HTTP 协议运行情况）。



上图这个界面看起来很复杂，但对于非程序员而言，HTTP 协议运行情况只要关注其中的几个关键部分：第一列，即资源的 URL；第四列是这个资源的类型。在第一个请求和后续的请求之间有一根蓝线，即进度条。而 HTTP 协议中运行的项目越少，浏览器加载的速度越快。图中 Google 就处理得很好，只有 10 个左右的请求。

## | Web 前端语言

#### HTML 和带样式的 HTML

HTML 就是一组标签和文本的组合，是一个最基本的网页。它已经包含了网页常见的元素，实际上在 Web 早期的很长一段时期内，网页都是这个样子。后来随着使用网络的人群越来越广泛，在 HTML3.0 中引入了对网页样式的定义，某种程度上可以说，也是从这个时候开始产生了网页设计师的角色。

#### CSS

带样式的 HTML 也拥有一个缺点，它需要为每个标题和文字都设定样式，工作量非常庞大。 CSS 就是在这样的情况下诞生了。CSS ，又称叠层样式表，简言之是一种用来表现 HTML 文件样式的样式设计语言。CSS 能够对网页中的对象的位置排版进行像素级的精确控制，实现基础的静态的交互设计；而CSS 目前的最新版本 CSS3 能够真正做到网页表现与内容分离。

#### Javascript

差不多在 CSS 诞生的同一时间，大家开始觉得这样静态的网页似乎略显无聊，能不能给网页加入一些可以动起来的元素？比如点击一个按钮之后变个颜色。当时网景公司的工程师Brendan Eich 就给他们自家的浏览器引入了这种实现动态效果的脚本语言，这就是 Javascript（简称 JS）的诞生。所以通俗来说，Javascript 就是用来给 HTML 网页增加动态功能，实现更炫酷的交互。

提到 Javascript ，就得提一下 jQuery 。 jQuery 是一个优秀的 Javascript 库。jQuery 使用户能更方便地处理 HTML ，它能够使用户的 HTML 页面保持代码和 HTML 内容分离,通过 jQuery ，可以不用在 HTML 里面插入一堆 JS 来调用命令，只需要定义 ID 即可。此外，由 Twitter 设计师 Mark Otto 和 Jacob Thornton 合作开发的 Bootstrap 也是一个受欢迎的前端框架。

## | HTML5 简史和响应式设计

HTML 在刚诞生的前 10 年发展是非常迅速的，在 1999 年，我们现在常说的 HTML5 的上一个版本 HTML4.0.1 就已经发布了，那么为什么从 4.0 到 5.0 会拖了 15 年之久？

首先，HTML4 的发布时间和门户时代（即 Web 1.0 时代）是基本吻合的，也就是说 HTML4 实际上是为门户型网站设计的。在门户网站经历的 4，5 的年发展之后，大家开始觉得只是单一接受信息的互联网太过无聊枯燥了，差不多 2004、2005 年开始，大家希望在网页中加入更多的互动元素，也就是我们常说的 Web 2.0。

但是这个时候大家发现，为 Web 1.0 设计的 HTML4 无法胜任这个工作，但是有另外一个技术却非常适合，那就是 Flash。所以在 Web 2.0 的早期，当时最炫酷的网站有很多是完全用 Flash 开发的，在以后的很长一段时间里，有很多网站都是 HTML 和 Flash 的混合式网站。所以在 2005 – 2010 年这段时间，HTML5 中的新标准主要是为了取代 Flash。

刚刚搞定了 Flash，又进入了移动开发时代，所以 HTML5 又花了 5 年时间制定各种针对移动平台的标准。但是到目前为止，虽然 HTML5 已定定稿，但是对移动平台的适应其实还在进行中，所以在未来很长一段时间内，就像当初的 Flash 一样，我们会看到越来越多的混合式应用。

在 iPhone 出现之前，大家访问 Web 的主要方式还是通过桌面浏览器，所以设计网页时只要考虑桌面浏览器的显示效果就足够了。但是在 iPhone 和 iPad 出现之后，就需要考虑同一个网页在不同设备上的显示效果，第一个问题的答案就是响应式，响应式的核心就是让同一个网页可以在不同设备上呈现出不同的显示效果，主要是通过CSS来实现的。

除了响应式设计，HTML 在移动端遇到的另外两个问题就是如何利用移动设备的各种传感器，比如 GPS，摄像头等等；以及性能问题。为了解决这些问题，HTML5中添加了地理位置，拍照，3D 动画加速等等 API，可以部分的利用手机设备的一些新硬件，并且新的 API 还在不断的加入进来，这也是为什么现在的 HTML5 应用可以越来越炫酷的原因。但是，HTML5 并不是专为移动设备设计的，它是由 HTML5，CSS3 以及大量的 Javascript API 共同组成的一个标准合集，微信中的 HTML5 应用只是 HTML5 应用场景中的很小一部分。

## | 如何判断一个前端的能力？

关于前端，可以简单的把它理解为，凡是在我们的电脑，手机上运行的技术，HTML，CSS，Javascript，这些都属于前端技术，使用这些技术的我们就称为前端工程师。如何判断一个前端的能力呢？下面是一个简单的前端能力链：

* 1.只会 HTML/CSS 的，这种我们俗称切图的，基本上就是淘宝几十块切一张图的；
* 2. 懂一些简单的 Javascript，主要是使用一些现成的框架，比如 jQuery，bootstrap 等等；
* 3. 知道 jQuery 和 Bootstrap 的局限，必要时能写一些原生的 JS/CSS 代码；
* 4. 对JS/CSS非常了解，执着于使用浏览器的各种最新特性来实现各种炫酷效果，这种我们成为炫技派；
* 5. 可以自己写出类似 jQuery / Bootstrap 这样的前端框架供其他人使用。

## | 前端学习资料

[http://www.w3schools.com](http://www.w3schools.com/)

[http://onemonth.com](http://onemonth.com/)

<http://www.codecademy.com/>

<https://github.com/alex/what-happens-when>

<https://qdan.me/list/VNBugw7ObupFRdlE>

大家可能比较关心如果我要开发一个网站需要多少时间？这个问题虽然很难回答，元一还是来试着回答了。现在前端有了 jQuery，bootstrap 这样的框架，后端又有了 Ruby on Rails 这样的快速 Web 开发框架，如果从头学的话，像是一个简易的 Pinterest，大概一个月就可以了。如果是一个有经验的程序员，可能 1 个星期就可以开发出一个大概的原型出来。

## | 后端服务器

后端的任务实际上就是向前端提供需要显示网页和 APP 内容的数据，可能是 HTML，也可能是JSON 数据，也可以是音视频或者 PDF 文件。简单的来划分，一个服务器包含3个部分：

* 1. HTTP 服务器
* 2. 应用服务器
* 3. 数据库

HTTP 服务器的唯一任务就是把需要返回给客户端的资源文件封装在 HTTP 数据包里，这个资源有可能是它后面的应用服务器动态生成的，也有可能是保存在硬盘上的静态文件。这是所有后端程序都必须有的，也是直接和我们的浏览器通信，返回给我们数据的程序。它的作用就是把它后面的编程语言生成的各种 HTML/CSS/Javascript，打包成一个 HTTP 请求，然后再封装到一个 TCP/IP 的数据包里发回给我们。而最常用的两个 HTTP 服务器叫做 Apach 和 Nginx。

应用服务器就是通常意义上所说的码农负责的部分。他们的职责就是生成前端需要的HTML/CSS/JS 交给浏览器。

## | 后端语言

#### 1. .net/java

庞大，复杂。但 Java 的优点就是适合处理特别大的数据量，如果你的项目会很快实现大爆发，需要处理海量的请求，那么 Java 是一个不错的选择。

#### 2. PHP

可以快速上手，相比其他语言，可以更快的为应用添加各种新功能。当然，可维护性就另当别论了。

#### 3. Ruby

非常接近自然语言，基本上即使不懂编程，也能看明白 70% 或 80%。04 年出现了一个用 Ruby 编写的 Web 开发框架 Ruby on Rails，当时的效果是非常震撼的，以前需要一个团队才能搞定的事情，使用 Ruby on Rails 后 1 个人就可以胜任了，所以 Ruby on Rails 在极短的时间内就成为了 Ruby 的代名词，也成为了新手学习 Web 开发的不二选择，但是 Ruby 语言也并非十全十美，快的同时，他的最大短板就是性能。Twitter 最早就是使用 Ruby on Rails 开发的，但是随着用户数的逐步增长，Twitter 的宕机开始变得非常频繁，后来他们迫不得已将整个系统从 Ruby 迁移到到了一个从 Java 派生出来的语言 Scala。

#### 4. node.js

简单来说，可以把 node.js 理解为跑在服务器上的 javascript，再直白一点，就是一个跑在服务器上的浏览器，因为 node.js 最早就是从 chrome 浏览器的Javascript 引擎 V8 中剥离出来的。相比 Ruby，Node.js 程序可以获得更高的并发性能，这在一些高并发的场景下（比如群聊，多人协作等）会很有优势。

#### 5. 其它（python，closure 等）

#### 6. 无后端（leancloud）

无后端编程是最近的一个新趋势，但她并非说是真的没有后端，而只是把后端交给一些第三方的云平台，比如 Leancloud，Firebase 等。如果你开发一个手机 App，这样的好处就是你可以在早期没有后端程序员的情况下快速开工，像Leancloud 这样的云平台已经可以胜任大部分的应用场景，如果后期业务逻辑复杂之后再寻找合适的后端工程师迁移也不迟。

#### 7. 最强编程语言 Lisp

如果要评选一个最强的编程语言，该是哪个呢？答案就是Lisp。为什么是 Lisp？Lisp 的作者在很早以前就从数学的层面总结了一个完美的编程语言应该具备的 9 种能力，而 Lisp 就是为了配合他的这个理论而产生出来的语言。Hacker News 是由 YC 的创始人 Paul Graham 开发的，而 Paul Graham 本身就是一个 Lisp 程序员，他为了开发 Hacker News，专门发明了一种新语言叫做 Arc，但因为它是基于 Lisp 的，所以也被归为了 Lisp 的方言之一。

## | 数据库

我们平常访问的大部分网站都是需要登录操作的，登录之后我们看到的就是只和自己相关的那部分内容。这些用户信息是保存在什么地方的呢？这就需要用到数据库。关于数据库，代表性的有两个：

* 1. MySQL
* 2. MongoDB

MySQL 是最常用的结构化数据库，也是大多数创业公司的选择。为什么是结构化的？就是说它的表的结构是固定的，比如我们常见的 User 表在 MySQL 中就是这样的：

id name email password

1 一元 y@modao.io $2a$10$ZxNhLyDTdagtUc...

2 产品集小妹 xiaomei@36kr.com $2a$10$zK7TnBXYYFz...

如果我们需要取得一条用户记录来检查他输入的密码是否正确，这时我们就需要使用 SQL，SQL 就是结构化查询语言。

简单来说，SQL 数据库保存的是结构化数据，NOSQL 数据库则可以保存非结构化数据。举个例子，还拿上面的用户表来举例，如果我们现在想要给产品集小妹增加一些额外的属性，比如她给某个产品点赞可以效果 x2，那么如果是 SQL 数据库，我们就需要给数据库增加一个新的字段来保存这个属性：

id name email x2

1 一元 y@modao.io false

2 产品集小妹 xiaomei@36kr.com true

但是如果是 MongoDB 这样的 NOSQL 数据库，我们就不需要给所有用户都增加一个x2的属性，只需要给产品集小妹单独增加就可以了，NOSQL 中保存到数据是如下这个样子的：

{id: ‘1’, name: ‘一元’, email: 'y@modao.io’, password: ‘…’}

{id: ‘1’, name: ‘产品集小妹’, email: 'xiaomei@36kr.com’, password: ‘…’, x2: true}

## | Cookie 和 Session

服务器要处理成千上万用户的请求，那么他是如何区分每个用户，并返回给每个用户他所需要的内容的 ？这就要涉及到 Cookie 和 Session。我们可以将 Cookie 理解为是服务器给每个用户分配的唯一 ID，这个 ID 由用户浏览器保存，而 Session 则是服务器为了维护这个会话在服务器端保存的与 cookie 对应的用户数据。

## | 移动开发

移动端和浏览器的区别就在于，大部分 App，我们打开的一瞬间，就已经看到了它的界面，而不用再去向服务器来拿显示界面的 HTML 等文件。所以移动端，开发原生应用所运用到的技术（比如 Objective C，swift）就相当于前端的 HTML，只不过它是直接保存在应用本地的。这样就产生了一个问题：如何来获取应用数据？如果是网页应用，我们可以直接将数据包含在HTML 中一并反馈给浏览器；但是对于移动应用就需要有一个专门的协议来传送应用需要的数据，这就是 JSON。

移动应用的前端技术，目前来说主要有以下三种：

* 1. 原生
* 2. 混合式
* 3. HTML5

HTML5 必经要经过浏览器这个中间层，所以在性能上多少会有些损失，所以如果你的应用对性能特别敏感，原生就会是比较好的选择；对于普通的性能要求没那么严格的应用来说，HTML5是完全可以满足的。而如果已经有了一个移动端的网站，这种情况下混合式就会是一个比较好的选择，它可以最大程度的利用已有的资源。如果说你是从头开发一个移动应用，并且这个应用对用户体验的要求也不是特别严格，那么 HTML5 就会是一个很好的选择，HTML5 移动应用比较显著的应用就是 Dailycost 。

如果说开发一个原生应用需要 4-6 周，那么同样功能的应用如果我们把其中的一部分用 HTML来实现，那么可能就只需要 3-4 周的时间，但是如果我们全部使用 HTML ，可能就只需要1-2 周。