

目录

第 1 章 入门准备

01 开篇词：你为什么要学 Python ？

02 我会怎样带你学 Python ？

03 让 Python 在你的电脑上安家落户

04 如何运行 Python 代码 ？

第 2 章 通用语言特性

05 数据的名字和种类—变量和类型

06 一串数据怎么存—列表和字符串

07 不只有一条路—分支和循环

08 将代码放进盒子—函数

09 知错能改—错误处理、异常机制

10 定制一个模子—类

11 更大的代码盒子—模块和包

12 练习—密码生成器

第 3 章 Python 进阶语言特性

13 这么多的数据结构（一）：列表、元祖、字符串

14 这么多的数据结构（二）：字典、集合

15 Python大法初体验：内置函数

16 深入理解下迭代器和生成器

17 生成器表达式和列表生成式

18 把盒子升级为豪宅：函数进阶

19 让你的模子更好用：类进阶

20 从小独栋升级为别墅区：函数式编程

28 Web 开发

更新时间：2019-11-01 09:51:58



“ 勤学如春起之苗，不见其增，日有所长。 ”
——陶潜

Web 开发，即 Web 应用开发，简单来说就是开发网站。

对于一个网站项目，其工作内容可以划分为两个主要部分——前端和后端。



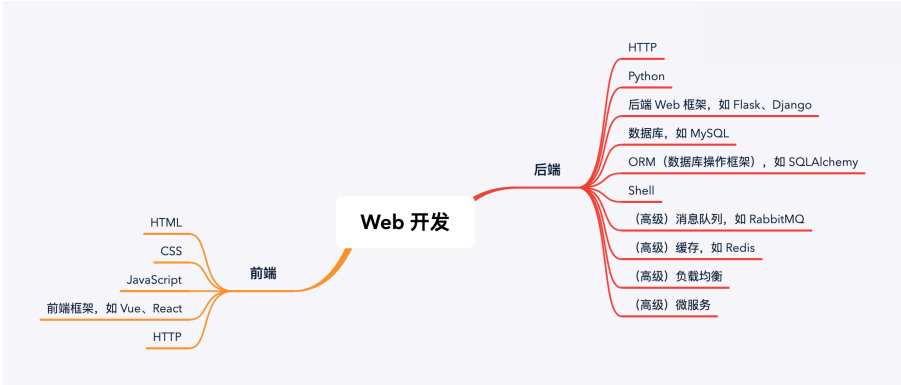
前端负责在浏览器中，向用户展示网页的样式、内容，并与用户交互，处理用户的点击和输入。

以一个简单的博客网站为例，前端负责将网站的样式、博客列表、博客内容等信息展示到浏览器中，并接受用户点击阅读帖子的操作，以及创建帖子和评论的操作。

后端负责接受前端的任务请求，根据需要提取、转换数据并返回给前端；或将前端发送过来的数据记录下来，长久保存到数据库中；或在背后处理一些任务。

在博客网站中，前端向后端请求博客列表数据、博客内容数据，后端从数据库中提取出这些数据，加工后返回给前端。用户创建帖子或评论时，前端接受输入并发送给后端，后端加工后将其记录在数据库中。

上面这个过程会涉及哪些技术呢？

<div>← 慕课专栏</div> <div>目录</div> <div>第 1 章 入门准备</div> <div>01 开篇词：你为什么要学 Python？</div> <div>02 我会怎样带你学 Python？</div> <div>03 让 Python 在你的电脑上安家落户</div> <div>04 如何运行 Python 代码？</div> <div>第 2 章 通用语言特性</div> <div>05 数据的名字和种类—变量和类型</div> <div>06 一串数据怎么存—列表和字符串</div> <div>07 不只有一条路—分支和循环</div> <div>08 将代码放进盒子—函数</div> <div>09 知错能改—错误处理、异常机制</div> <div>10 定制一个模子—类</div> <div>11 更大的代码盒子—模块和包</div> <div>12 练习—密码生成器</div> <div>第 3 章 Python 进阶语言特性</div> <div>13 这么多的数据结构（一）：列表、元组、字符串</div> <div>14 这么多的数据结构（二）：字典、集合</div> <div>15 Python大法初体验：内置函数</div> <div>16 深入理解下迭代器和生成器</div> <div>17 生成器表达式和列表生成式</div> <div>18 把盒子升级为豪宅：函数进阶</div> <div>19 让你的模子更好用：类进阶</div> <div>20 从小独栋升级为别墅区：函数式编程</div>	<div>三 你的第一本Python基础入门书 / 28 Web 开发</div> <div>1. 前端展示网页样式和内容，用到 CSS、HTML、前端框架</div> <div>2. 前端处理用户交互，用到 JavaScript、前端框架</div> <div>3. 前端与后端交换数据，用到 JavaScript、HTTP（RESTful API）</div> <div>4. 后端处理前端的请求，用到 Python、后端 Web 框架</div> <div>5. 后端提取已被长久保存的数据，或记录需长期保存的数据，用到数据库，如MySQL</div> <div>6. 网站的部署，用到一些 Linux 命令或 Shell 脚本</div> <div>7. 当网站规模很大时，后端可能会用到消息队列、缓存、负载均衡、微服务等高级技术</div> <div><pre>graph LR Web[Web 开发] --- Front[前端] Web --- Back[后端] Front --- HTML Front --- CSS Front --- JS[JavaScript] Front --- Framework[前端框架, 如 Vue、React] Front --- HTTP Back --- HTTP2[HTTP] Back --- Python Back --- Framework2[后端 Web 框架, 如 Flask、Django] Back --- DB[数据库, 如 MySQL] Back --- ORM[ORM (数据库操作框架), 如 SQLAlchemy] Back --- Shell Back --- MQ["(高级) 消息队列, 如 RabbitMQ"] Back --- Cache["(高级) 缓存, 如 Redis"] Back --- LB["(高级) 负载均衡"] Back --- Micro["(高级) 微服务"]</pre></div> <p>其中，如果前端与后端交换数据时使用 RESTful API，那么后端部分将变得更加通用，不仅能应用于网站，也能应用于 iOS 应用、安卓应用、小程序，也就是说后端的技术栈在这些应用中是通用的。</p> <p>看看可看出，Python 的职责在后端部分，如果你想成为一个 Python 后端开发者，只需要关注和学习后端部分即可，至于前端部分，由前端开发者完成，之后直接通过 HTTP 接口与其对接。如果你想成为全栈开发，能独立开发一个完整网站的全能型选手，那前、后端技术都应该去掌握。</p> <p>以下提供一些入门资料，比较基础易学，供你参考：</p> <ul style="list-style-type: none">• HTML、CSS：慕课网《初识HTML+CSS》• JavaScript：慕课网《JavaScript入门篇》《JavaScript进阶篇》• 前端框架：慕课网 Vue《vue2.5入门》，React《React16.4 快速上手》，二选一即可• HTTP：图书《图解HTTP》• 后端 Web 框架：图书《Flask Web开发：基于Python的Web应用开发实战》（强烈推荐）• 数据库 MySQL：慕课网《与MySQL的零距离接触》• ORM 数据库操作框架：翻译文档《Flask-SQLAlchemy 快速入门》• Linux：慕课网《Linux达人养成计划 I》 <p>加油吧！</p> <div>← 27 这么多条路，你想走那一条？</div> <div>29 数据科学 →</div>

目录	
第 1 章 入门准备	
01 开篇词：你为什么要学 Python ？	 目前暂无任何讨论
02 我会怎样带你学 Python ？	
03 让 Python 在你的电脑上安家落户	
04 如何运行 Python 代码？	千学不如一看，千看不如一练
第 2 章 通用语言特性	
05 数据的名字和种类—变量和类型	
06 一串数据怎么存—列表和字符串	
07 不只有一条路—分支和循环	
08 将代码放进盒子—函数	
09 知错能改—错误处理、异常机制	
10 定制一个模子—类	
11 更大的代码盒子—模块和包	
12 练习—密码生成器	
第 3 章 Python 进阶语言特性	
13 这么多的数据结构（一）：列表、元祖、字符串	
14 这么多的数据结构（二）：字典、集合	
15 Python大法初体验：内置函数	
16 深入理解下迭代器和生成器	
17 生成器表达式和列表生成式	
18 把盒子升级为豪宅：函数进阶	
19 让你的模子更好用：类进阶	
20 从小独栋升级为别墅区：函数式编	