PYTHON: WEB.PY/DJANGO/TORNADO

1 June 2017

咎妍

Python

Python

- Python是一门动态、面向对象的直译式语言。最初就是作为一门面向对象语言设计的。
- Python标准库很完备，Python甚至还自带服务器。
- Python的官方解释器是CPython，该解释器用C语言编写。
- Python的设计哲学是“优雅、明确、简单”，面临多种选择时解决的顺序可参考”Python格言”

- Guido希望找到像C语言能够全面调用计算机的功能接口，又像shell可以轻松的编程。
- Bourne Shell（ UNIX interpreter）本质是调用命令，虽然很像胶水，但功能不齐全（如没有数值类型）
- ABC语言虽然满足需求，但缺点过多：可拓展性差、读写文件困难、过度革新（符号太像自然语言了）、传播困难（编译器过大）
- 1989年圣诞开始设计，一种C和shell之间，功能全面，易学易用，可拓展的语言。

应用

- Web框架:
  - Python定义了WSGI标准应用接口来协调Http伺服器与基于Python的Web程式之间的沟通。
  - Django、Pyramid、TurboGears、Tornado、web2py、Zope、Flask等。
- GUI
  - Python本身包含的Tkinter库能够支持简单的GUI开发
  - wxPython或者PyQt等GUI套件来开发跨平台的桌面软件
- OS
  - 大多数Linux发行版和Mac OS X都集成了Python，可以在终端机下直接执行Python

语法

- 缩排(Off-side越位规则)：Python与其它大多数程序设计语言使用大括号不一样，使用缩排来定义语句块。
- 函数: Python的函数支持递归、默认参数值、可变参数、闭包，但不支持函数重载。
- 面向对象开发方法:绑定到对象的函数。

面向对象开发方法

- Instacne.method(arguments)
- Class.method(instance,arg)
- 定义对象方法

```
class Fish(object):
    def eat(self, food):
        if food is not None:
            self.hungry=False
class User(object):
    def __init__(myself, name):
        myself.name = name

#构造Fish的实例：
f=Fish()
#以下两种调用形式是等价的：
Fish.eat(f, "earthworm")
f.eat("earthworm")

u = User('username')

u.name
```

Python web框架

Python的web框架

- Python拥有足够多的免费数据函数库、免费的Web网页模板系统、还有与Web服务器进行交互的库、这些都可以设计到Web应用程序里面。
- Python有各种micro-framework、framework，虽然Ruby有Rails, PHP也有框架，但都不及Python，故Python web框架一直有很高的讨论度。
- (以下简称Python Web Framework为Python框架)

- Web应用程序可以使用基本HTTP应用服务器、存储机制如数据库、模板引擎、请求分配器、验证模块和Ajax工具包的组合。这些可以是单独的组件，也可以在高层框架中一起提供。
- 接下来介绍几个主流的Python框架: web.py，Django，Tornado

Web.py

Web.py

- "Think about the ideal way to write a web app. Write the code to make it happen.“
- Aaron Swartz设计
- Web.py是一个轻量级的开源Python Web框架，小巧灵活、简单并且非常强大，在使用时没有任何限制。
- web.py的设计理念力求精简，不像Pylons那样依赖大量的第三方模块，而是只提供的一个框架所必须的一些东西，如：URL路由、Template、数据库访问，其它的就交给用户自己去做好了。

- 一个框架精简的好处在于你可以聚焦在业务逻辑上，而不用太多的去关心框架本身或受框架的干扰，同时缺点也是得自己做出需要而框架沒有的東西。
- 目前Web.py被广泛运用在许多大型网站，如西班牙的社交网站Frinki、主页日平均访问量达7000万次的Yandex等。
- web.py最早是Aaron Swartz 在 reddit工作时做出来的，reddit在这个框架上顺利成为一家alexa排名头两百名内的网站。
- 案例: Frinki, Yandex, Make History, http://Oyster.com, 豆瓣, 早期的reddit

Django

Django

- 1.11.1(2017/04/04)
- Django是一个开源的Web框架，采用MVC设计模式，包含许多非常实用的库来加速Web开发。
- Google App Engine甚至Erlang都有框架受它影响。
- Django的文档最完善、市场占有率最高、python使用最广泛的开源框架之一。
- 案例: Instagram, pinterest

- 它最初是被开发来用于管理Lawrence出版集团旗下的一些以新闻内容为主的网站的，记者要求增加的特征或整个程序都能在计划时间内快速的被建立，通常只有几天或几个小时。
  - 从头开始编写网络应用程序。
  - 从头编写另一个网络应用程序。
  - 从第一步中总结（找出其中通用的代码），并运用在第二步中。
  - 重构代码使得能在第 2 个程序中使用第 1 个程序中的通用代码。
  - 重复 2-4 步骤若干次。
  - 意识到你发明了一个框架。

- 2005年7月在BSD许可证下发布。Django是以比利时的吉普赛爵士吉他手Django Reinhardt来命名的。

- Django的主要目标是简便、快速的开发数据库驱动的网站。
- Django注重组件的重用性和“可插拔性”，第三发插件和方便的开发工具包的特性使得Django有很强的可扩展性。
- 敏捷开发和DRY法则（Don‘t Repeat Yourself）。在Django中Python被普遍使用，甚至包括配置文件和数据模型，可说是为完美主义者所作的web框架。

Django benefit

- Django拥有近乎完美的官方文档（包括Django book）。
- 强大的URL路由配置
- 全套的解决方案（full-stack framework + batteries included），开发网站应手的工具Django基本都做好了，因此开发效率非常高。
- Django是走大而全的方向，它最出名的是其全自动化的管理后台。

Django cons

- Django模板的设计哲学是彻底的将代码、样式分离；
  - (asp.net提倡将代码/模板分离，但技术上还是可以混合；而Django则是根本无法使用)
- Django提供的方便，也意味着Django内置的ORM跟框架内的其他模块耦合程度高。
- Django自带的ORM远不如SQLAlchemy强大。
- Template功能比较弱，不能插入Python代码，要写复杂一点的逻辑需要另外用Python实现Tag或Filter。
- URL配置虽然强大，但全部要手写。

Django 框架核心

- Django框架的核心包括：一个 面向对象 的映射器，用作数据模型（以Python类的形式定义）和关联性数据库间的媒介；一个基于正则表达式的URL分发器；一个视图系统，用于处理请求；以及一个模板系统。
- 核心框架中还包括：
- 一个轻量级的、独立的Web服务器，用于开发和测试。
- 一个表单序列化及验证系统，用于HTML表单和适于数据库存储的数据之间的转换。
- 一个缓存框架，并有几种缓存方式可供选择。
- 中间件支持，允许对请求处理的各个阶段进行干涉。
- 内置的分发系统允许应用程序中的组件采用预定义的信号进行相互间的通信。
- 一个序列化系统，能够生成或读取采用XML或JSON表示的Django模型实例。
- 一个用于扩展模板引擎的能力的系统。

Tornado

Tornado

- 4.5(2017/04/16)
- Tornado全称Tornado Web Server，是一个Web服务器兼Web应用框架，由FriendFeed公司在自己的网站中使用，被Facebook收购以后框架以开源软件形式开放给大众。
- Tornado與主流服務器的區別在於：拥有异步非阻塞IO的处理方式，速度快。
- 解决C10K problem
- 案例: FriendFeed, 知乎

- Tornado即是一个web server，同时又是一个类web.py的micro-framework，作为框架Tornado的思想主要来源于web.py
- **"[web.py inspired the] web framework we use at FriendFeed [and] the webapp framework that ships with App Engine..."**

  — Bret Taylor, co-founder of FriendFeed and original tech lead on Google App Engine

Tornado Cons

- Tornado走的是少而精的方向，它也有提供模板功能；
- 但它没有ORM（仅有一个mysql的超简单封装），甚至没有Session支持。
- 假设是一个大型网站，在高性能的要求下，框架的各个部分往往都需要定制，可以复用的模块非常少
  - (Tornado是Facebook開源出來的框架，其哲學跟Django近乎兩個極端。)

Tornado 服務器

- HTTP服務器：Tornado为了高效实现Comet/后端异步调用HTTP接口，是直接内嵌了HTTP服务器。
- 前端无需加apache / lighttpd / nginx等也可以供浏览器访问；但它并没有完整实现HTTP 1.1的协议，所以官方文档是推荐用户在生产环境下在前端使用nginx，后端反向代理到多个Tornado实例。

单线程异步程序

- Tornado本身是单线程的异步网络程序，它默认启动时，会根据CPU数量运行多个实例；充分利用CPU多核的优势。
- 网站基本都会有数据库操作，而Tornado是单线程的，这意味着如果数据库查询返回过慢，整个服务器响应会被堵塞。
- 数据库查询，实质上也是远程的网络调用；理想情况下，是将这些操作也封装成为异步的；但Tornado对此设计提供任何支持。
- 如果后端有查询实在是太慢，无法绕过，Tornaod的建议是将这些查询在后端封装独立封装成为HTTP接口，然后使用Tornado内置的异步HTTP客户端进行调用。

主要模块

- web- FriendFeed使用的基础web框架，包含Tornado的大多数重要功能
- escape- XHTML, JSON, URL的编码/解码方法
- database- 对MySQL的简单封装，使其更容易使用
- template- 基于Python的web模板系统
- httpclient- 非阻塞式HTTP客户端，它被设计用来和web及httpserver协同工作
- auth- 第三方认证的实现(包括google OpenID/ Oath、facebook platform、BBAuth、FriendFeed OpenID/OAuth、Twitter OAuth）
- locale - 针对本地化和翻译的支持
- options - 命令行和配置文件解析工具，针对服务器环境做了优化
- 底层模块
- httpserver - 服务于 web 模块的一个非常简单的 HTTP 服务器的实现
- iostream - 对非阻塞式的 socket 的简单封装，以方便常用读写操作
- ioloop - 核心的 I/O 循环
- Tornado 各模块

参考资料

- Python wiki: Web Frameworks for Python
- Web.py: philosophy
- Tornado: Tornado cn
- Wiki: python, Tornado, C10k problem
- Wiki Books: Python Programming
- 百度百科: BSD許可, Aaron Swartz
- Web.py
- The C10K Problem
- Programming Language Comparison
- Object-oriented programming in Python
- Web Framework Benchmarks

- 博客:简单而直接的Python web 框架：web.py
- Python Web部署方式总结
- Python（九）Tornado web 框架 其实很简单、深度应用
- Python六大框架对比
- 两个Python web框架：Django & Tornado比较
- 浅谈Python web框架
- Python从入门到放弃系列
- Python简史
- 深刻理解Python中的元类(metaclass)
- python 元编程
- C10k-problem

Questions?