一、几个概念

1. **网络通信规范：WSGI、uwsgi**

1）WSGI的全称是Web Server Gateway Interface（Web服务器网关接口），它不是服务器、python模块、框架、API或者任何软件，只是一种描述web服务器（如nginx，uWSGI等服务器）如何与web应用程序（如用Django、Flask框架写的程序）通信的规范。

2）uwsgi与WSGI一样，是uWSGI服务器的独占通信协议，用于定义传输信息的类型(type of information)。每一个uwsgi packet前4byte为传输信息类型的描述，与WSGI协议是两种东西，据说该协议是fcgi协议的10倍快。

1. **Web服务器：uWSGI 、Nginx**

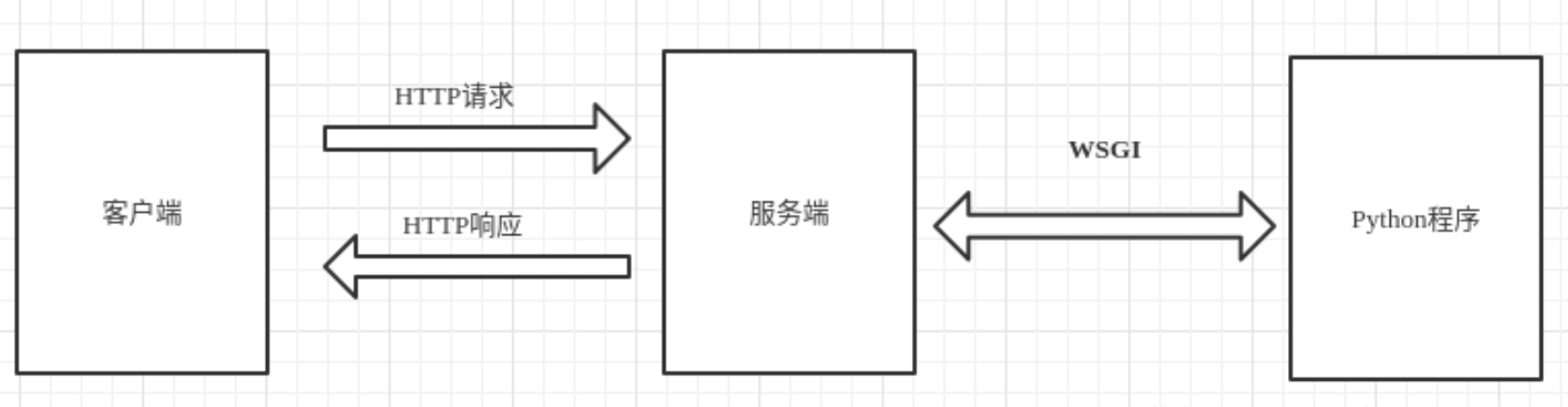
1）uWSGI是一个全功能的HTTP服务器，实现了WSGI协议、uwsgi协议、http协议等。它要做的就是把HTTP协议转化成语言支持的网络协议。比如把HTTP协议转化成WSGI协议，让Python可以直接使用。

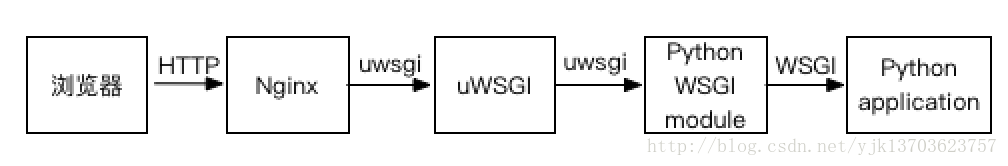
2）Nginx是一个Web服务器，其中的HTTP服务器功能和uWSGI功能很类似，但是Nginx还可以用作更多用途，比如最常用的反向代理功能。

1. **Web应用框架：Bottle、Flask、Django**

server和application的规范在PEP3333中有具体描述，要实现WSGI协议，必须同时实现web server和web application，当前运行在WSGI协议之上的web框架有Bottle, Flask, Django。

1. **访问过程**





1. Flask介绍

Flask是python的web应用框架。

1）创建Flask的实例

from flask import Flask

app = Flask(\_\_name\_\_)

2）建立URL与python函数的映射关系

通过装饰器的方式，使用路由(**Router**)实现，路由所对应的程序称为**视图函数**(view function)，即下面的index()函数。

@app.router(‘/’)

def index():

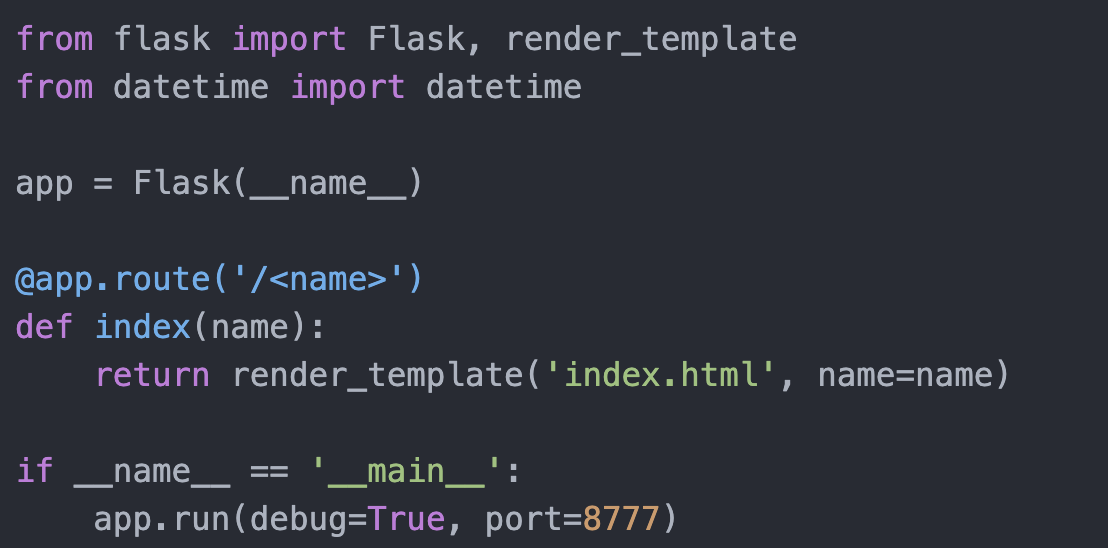
return ‘hello world!’

3）通过运行Flask实例的run()方法，启动服务器

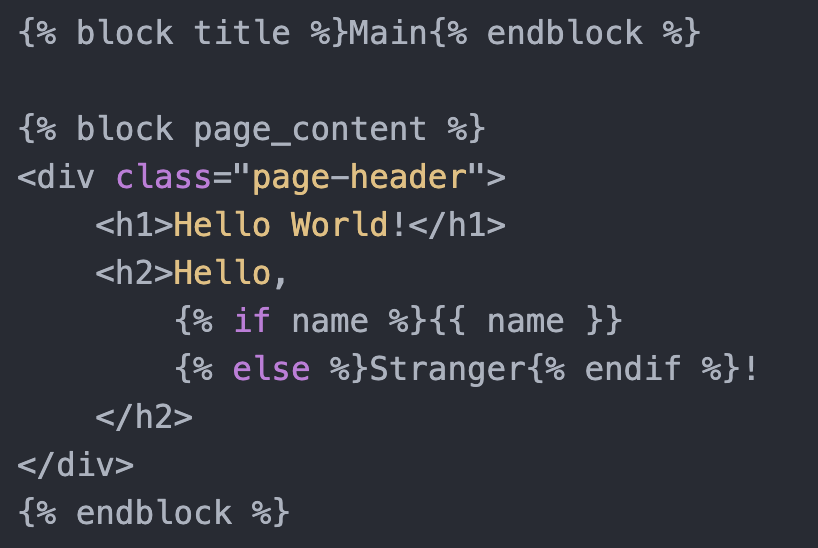
if \_\_name\_\_ == ‘\_\_main\_\_’:

app.run(debug=True, port=80)

4）使用模版定义网页格式



其中，index.html文件如下，需放在程序所在文件夹下的templates文件夹内。



5）访问

网页打开 <http://127.0.0.1:5000/> 进行访问。

三、使用uWSGI部署Flask应用

1）uWSGI安装

pip install uwsgi

2）配置uwsgi.ini文件

[uwsgi]

# uwsgi 启动时所使用的地址与端口

#http = 172.28.135.117:8005

#超时时间默认3s

#socket-timeout=4

#杀死超过3秒的请求

#harakiri = 4

# 指向网站目录

chdir = ./

# python 启动程序文件

wsgi-file = main.py

# python 程序内用以启动的 application 变量名

callable = app

# 处理器数

processes = 4

# 线程数

enable-threads = true

socket= 127.0.0.1:8001

备注：

http 的话，nginx和uwsgi中间走的是http协议，文本

socket 的话，nginx和uwsgi中间走的是wsgi协议，二进制，wsgi 协议效率会更高一点

2）部署执行和终止

uwsgi uwsgi.ini #执行

pkill -f -9 uwsgi #终止

nohup uwsgi conf/uwsgi\_config.ini > /dev/null 2>&1 & #执行，无标准输出和错误输出

**参考**：<https://www.jianshu.com/p/0f528c47c5bf>