

# Python 基础教学

余婷

## 如何找到服务器

网络中部署着各种各样的服务器，比如腾讯的服务器，百度的服务器

那么问题来了？

客户端如何找到想要连接的服务器呢？

客户端通过URL找到想要连接的服务器



# URL

---

- 什么是URL
  - URL的全称是Uniform Resource Locator（统一资源定位符）
  - 通过1个URL，能找到互联网上唯一的1个资源
  - URL就是资源的地址、位置，互联网上的每个资源都有一个唯一的URL
- URL的基本格式 = 协议://主机地址/路径

http://www.1000phone.com/img/logo.png

http://52.80.135.246/img/bdlogo.gif

- 协议：不同的协议，代表着不同的资源查找方式、资源传输方式
- 主机地址：存放资源的主机（服务器）的IP地址（域名）
- 路径：资源在主机（服务器）中的具体位置

## URL中常见的协议

---

- HTTP

- 超文本传输协议，访问的是远程的网络资源，格式是http://

- http协议是在网络开发中最常用的协议

- file

- 访问的是本地计算机上的资源，格式是file://（不用加主机地址）

- mailto

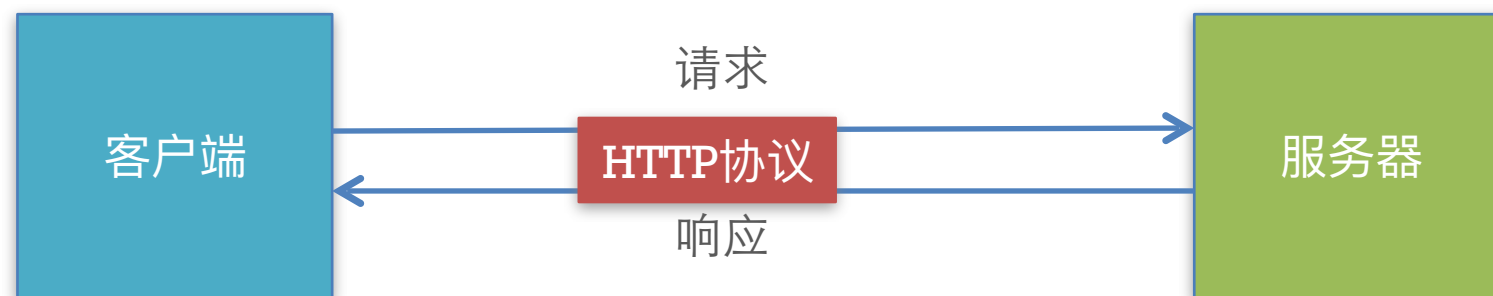
- 访问的是电子邮件地址，格式是mailto:

- FTP

- 访问的是共享主机的文件资源，格式是ftp://

## HTTP协议简介

- 不管是移动客户端还是PC端，访问远程的网络资源经常使用HTTP协议
- 访问千锋主页：<http://www.1000phone.com>
- 获得网易的新闻数据
- 获得优酷的视频数据
- HTTP协议的作用
- HTTP的全称是Hypertext Transfer Protocol，超文本传输协议
- 规定客户端和服务端之间的数据传输格式
- 让客户端和服务端能有效地进行数据沟通



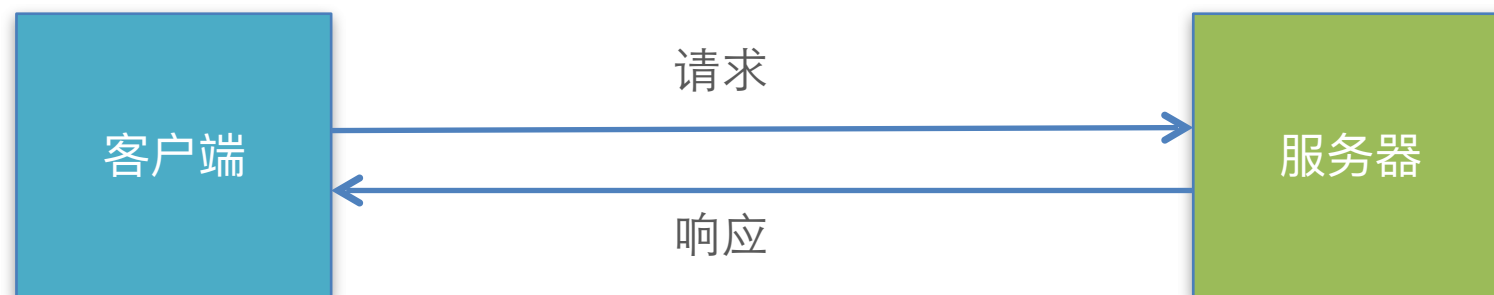
## HTTP协议的特点

---

- 简单快速
  - 因为HTTP协议简单，所以HTTP服务器的程序规模小，因而通信速度很快
- 灵活
  - HTTP允许传输各种各样的数据
- HTTP 0.9和1.0使用非持续连接
  - 限制每次连接只处理一个请求，服务器对客户端的请求做出响应后，马上断开连接，这种方式可以节省传输时间

## HTTP的基本通信过程

- 要想使用HTTP协议向服务器索取数据，得先了解HTTP的通信过程
- 完整的http通信可以分为2大步骤
  - 请求：客户端向服务器索要数据
  - 响应：服务器返回客户端相应的数据



## HTTP通信过程-请求

- HTTP协议规定：1个完整的由客户端发给服务器的HTTP请求中包含以下内容

□ 请求头：包含了对客户端的环境描述、客户端请求信息等

GET /minion.png HTTP/1.1 // 包含了请求方法、请求资源路径、HTTP协议版本

**Host:** 120.25.226.186:32812 // 客户端想访问的服务器主机地址

**User-Agent:** Mozilla/5.0 // 客户端的类型，客户端的软件环境

**Accept:** text/html, \*/\* // 客户端所能接收的数据类型

**Accept-Language:** zh-cn // 客户端的语言环境

**Accept-Encoding:** gzip // 客户端支持的数据压缩格式

□ 请求体：客户端发给服务器的具体数据，比如文件数据(POST请求才会有)



## HTTP通信过程-响应

- 客户端向服务器发送请求，服务器应当做出响应，即返回数据给客户端
- HTTP协议规定：1个完整的HTTP响应中包含以下内容

□ 响应头：包含了对服务器的描述、对返回数据的描述

HTTP/1.1 200 OK // 包含了HTTP协议版本、状态码、状态英文名称

Server: Apache-Coyote/1.1 // 服务器的类型

Content-Type: image/jpeg // 返回数据的类型

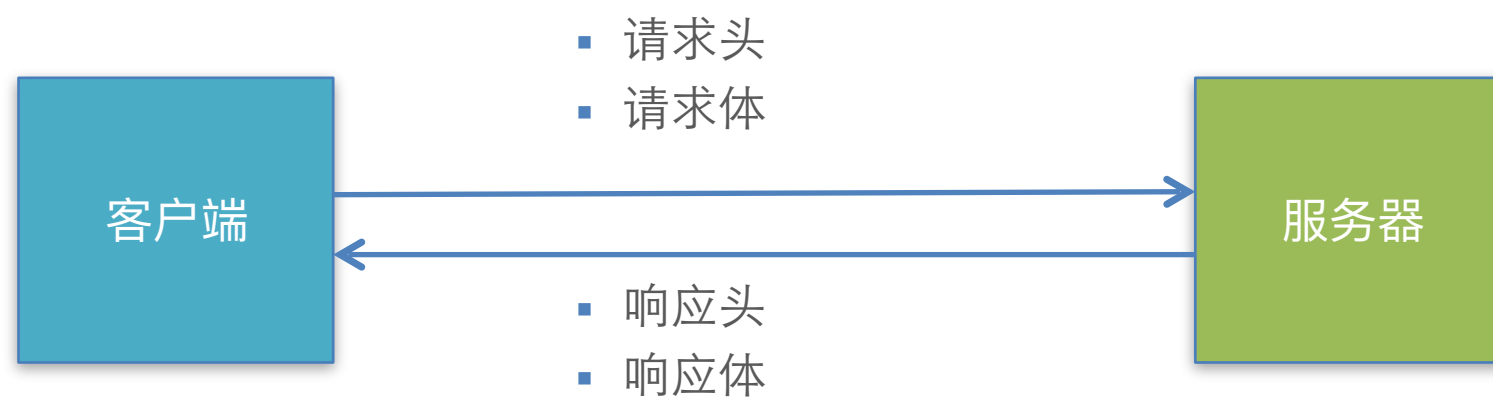
Content-Length: 56811 // 返回数据的长度

Date: Mon, 23 Jun 2014 12:54:52 GMT // 响应的时间

□ 响应体：服务器返回给客户端的具体数据，比如文件数据

## HTTP通信过程

---



## 常见响应状态码

状态码	英文名称	中文描述
200	OK	请求成功
400	Bad Request	客户端请求的语法错误，服务器无法解析
404	Not Found	服务器无法根据客户端的请求找到资源
500	Internal Server Error	服务器内部错误，无法完成请求

## 发送HTTP请求的方法

- 在HTTP/1.1协议中，定义了8种发送http请求的方法
  - ▣ GET、POST、OPTIONS、HEAD、PUT、DELETE、TRACE、CONNECT、PATCH
  - ▣ 根据HTTP协议的设计初衷，不同的方法对资源有不同的操作方式
    - ✓ PUT：增
    - ✓ DELETE：删
    - ✓ POST：改
    - ✓ GET：查
  - ▣ 最常用的是GET和POST（实际上GET和POST都能办到增删改查）
- 要想使用GET和POST请求跟服务器进行交互，得先了解一个概念
  - ▣ 参数
    - ✓ 就是传递给服务器的具体数据，比如登录时的帐号、密码

## GET和POST对比

---

- GET和POST的主要区别表现在数据传递上

### □ GET

- ✓ 在请求URL后面以?的形式跟上发给服务器的参数，多个参数之间用&隔开，比如

`http://www.test.com/login?username=123&pwd=234&type=JSON`

- ✓ 由于浏览器和服务对URL长度有限制，因此在URL后面附带的参数是有限制的，通常不能超过1KB

### □ POST

- ✓ 发给服务器的参数全部放在请求体中
- ✓ 理论上，POST传递的数据量没有限制（具体还得看服务器的处理能力）

## GET和POST的选择

---

- 选择GET和POST的建议
  - 如果要传递大量数据，比如文件上传，只能用POST请求
  - GET的安全性比POST要差些，如果包含机密\敏感信息，建议用POST
  - 如果仅仅是索取数据（数据查询），建议使用GET
  - 如果是增加、修改、删除数据，建议使用POST