

Web基础概念

授课人 令狐东邪



版权声明

九章的所有课程均受法律保护,不允许录像与传播录像 一经发现,将被追究法律责任和赔偿经济损失





网络背后的世界



在我们的生活中,网络与我们的关系是如此密切,娱乐、学习、沟通、工作似乎都有网络的身影,网络对于我们是如此的熟悉,但网络背后的世界对于我们又是如此的陌生,下面让我们来了解一下网络背后的一些概念。





客户端&服务端

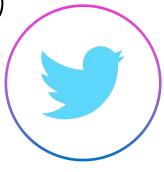




手机向推特请求最新、最热门的推文(Tweet)

«·—··—··—··—

推特把最新、最热门的推文返回给用户



在这个过程中,手机充当了 客户端(Client)的角色 而推特扮演**服务端(Server)** 的角色

为了能让客户端和服务端无障碍沟通,就需要一种双方都能理解的协议(Protocol)充当信息交换的媒介

而客户端和服务端的通讯方式使用的是HTTP协议!







我们知道了客户端(Client)和服务端(Server)之间的通讯需要使用到HTTP协议,那什么是HTTP协议呢



HTTP 的大名叫做超文本传输协议(Hypertext Transfer Protocol), 诞生于1989年, 那个时候互联网还属于少数人,正如其名,小H那时候就是用来传输超文本的(Html)。岁月如梭,时过境迁,来到21世纪,老H已经可以为我们传输诸如视频、音频和图片等等信息,已经是一个全能王。

HTTP协议是客户端和服务端之间通讯使用最多的协议,除此之外还有FTP,POP3等协议也可以用户C/S架构下的信息交换。



我FTP协议是文件传输小能手



我是POP3协议,你们发电子邮件可离不开我哦





HTTP的组成



在推特上看到的图片、视频、文字都离不开HTTP的传输,作为好奇宝宝的你,一定对HTTP协议的内涵充满了好奇!我们是怎么通过HTTP传递请求的?中间经历了哪些步骤?HTTP又是如何将我们的数据传递到我们的手中?让我们一起来窥探HTTP的 ""

通过下面的内容你将了解以下内容:

- URL的组成
- HTTP请求报文、响应报文及其组成
- DNS解析全过程







使用HTTP协议访问页面的时候,我们需要在浏览器中输入一个链接(URL)。关于URL的知识,你了解多少呢?





URL的组成

http://www.jiuzhang.com:80/path/index.html?key1=value1&key2=value2

- 协议 (http) 带上 http 表示我们使用的是 http 协议,如果是发邮件,就可以换成 pop3 协议。
- **主机名** (www.jiuzhang.com) 主机名(host)由**计算机名**(www)+域名(jiuzhangdahaofa.com)构成。
- | 端口 (80)
 - 一个计算机需要在同时提供很多的网络服务,你需要在一边聊天的时候,一边听歌,为了可以区别不同的网络服务,就出现了端口(port)的概念,每个端口都有各自的职能。 Django在自带的WSGI服务器使用的就是8000端口



URL的组成

http://www.jiuzhang.com:80/path/index.html?key1=value1&key2=value2

路径 (/path/index.html)

路径(Path)是资源在网站中的位置。比如,/path/index.html 这个路径,指向网站的/path子目录下面的网页文件index.html

查询参数 (key1=value1&key2=value2)

查询参数(Query Parameter)是提供给服务器的额外信息。参数的位置是**在路径后面**,两者之间使用?分隔,上例是?key1=value1&key2=value2。







信息的传递就像寄快递一个,需要给商品进行包装,加装泡沫隔板防碎,需要纸盒防脏,需要贴上快递信息,告诉快递员需要把商品送至何处,发货人是谁,收件人是谁等等。

HTTP协议请求数据和返回数据,这些数据都被封装在一个被称为报文(Message)的结构中。请求Web页面的报文被称为"请求报文(Request Message)"。相反,从服务端返回的报文被成为"响应报文(Response Message)"。

接下来让我们看看"请求报文"和"响应报文"的结构。





请求报文和响应报文



响

应

请求报文和响应报文都是由起始行、 **头部、空行和消息正文**组成。

请求报文的请求行+请求头部字段合称为请求头。

响应报文的状态行+响应头 部字段合称为<mark>响应头</mark>。 请求报文 Request message

建始行(Start-line) 请求 请求行 Request-Line 请求 请求头部字段 Request Header Fields

消息正文 (Body)

Blank Line

实体 Body

空行

响应报文 Response message

状态行 Status line

响应头部字段 Response Header Fields

空行

实体 Body





请求头的组成



请求头由请求行和请求头部字段组成

请求头 Request Header 请求行 Request-Line

请求头部字段 Request Header Fields Request Method: POST

Request URI: /

Request Version: HTTP/1.1

Connection: Keep-Alive\r\n

Content-Type: application/json\r\n

dw-protocol: 1.0\r\n
Content-Length: 868\r\n

Host: shuc-pc-hamlet.ksord.com\r\n

请求行包含:请求方法+请求

链接+请求协议版本 三部分

请求头部字段包含若干键

值对,中间用冒号:隔开





请求头的组成



请求方法 Request Method

HTTP/1.1 定义的请求方法有8种: GET、POST、PUT、DELETE、PATCH、HEAD、OPTIONS、TRACE

请求URI Request-URI

请求对应的URI地址,它和报文头的Host属性组成完整的请求URL

HTTP协议版本 Protocol Version

协议名称及版本号, 常见的有HTTP / 1.0 / 1.1 / 2.0, HTTP/1.1 是目前主流的 HTTP 协议版本。







对于资源(resource)的具体操作类型,由 HTTP 动词表示。常用的 HTTP 动词有下面四个:

- GET (SELECT): 从服务器取出资源,比如获取最新的推文
- POST (CREATE): 在服务器新建一个资源,比如注册用户、发推文
- PUT(UPDATE): 在服务器更新资源,比如更换头像,更换昵称
- DELETE(DELETE): 从服务器删除资源。比如删除自己发的推文





请求头部字段



请求头头字段,说明服务器要使用的附加信息,比较重要的有Cookie、Referer、User-Agent等。

Cookie

这是网站为了辨别用户进行会话跟踪而存储在用户本地的数据。它的主要功能是**维持当前访问会话**。

Referer

标识请求是从哪个页面发过来的,服务器可以拿到这一信息并做相应的处理,如做来源统计、防盗链处理和反爬虫等。

User-Agent

它是一个特殊的字符串头,可以使服务器识别客户使用的操作系统及版本、浏览器及版本等信息。



HTTP协议的不足——Cookie



上面的内容让人如沐春风,让大家了解了请求报文的全貌,但是这里要重点在提一下**Cookie**,这是一个很重要的概念。

HTTP协议只是简单的将信息包装进请求报文,然后发送给服务器,然后或服务处理请求后向请求地返回响应报文。HTTP本身并不知道是哪个人在请求,可谓来者不拒。

这就带来了问题,大家设想一个场景。

我在推特上关注"九章算法"的账号,但是通过HTTP协议,服务器并不知道是谁想关注九章算法。为了解决这个我们,我们就需要一个被称为cookie的信息,当我们输入账号密码登录推特的时候,推特就会生成一个cookie并返回给我们,以后我们每次访问推特,在请求报文中都会自动捎上cookie,服务器通过cookie就可以知道是谁在访问了。





cookie访问过程



HTTP请求携带Cookie, 服务端就可以根据这个Cookie来判断是哪个用户在操作



双端交互及Cookie的产生过程



Cross-domain Cookie



Cookie具有一种Domain指令,该指令指示应向其发送Cookie的一个或多个域。默认情况下,domain设置为设置cookie的页面的主机名。想象一下一个网站https://google.com设置了以下标头:

Set-Cookie: id = 1234:

浏览器会将Cookie和所有后续请求发送到https://google.com域。由于使用了域的默认值,因此浏览器不会将此Cookie发送到google.com的任何子域。因此,浏览器不会将Cookie 发送给以下域的请求:

- https://mail.google.com
- https://drive.google.com
- https://files.drive.google.com

但是,如果https://google.com发送以下标头:

Set-Cookie: id = 1234; Domain = google.com

由于服务器已明确指定域值,因此浏览器将为 https://*.google.com的任何子域发送cookie。



Third Party Cookie



Cookie有三种主要用途:会话管理,个性化和跟踪。使用cookie进行个性化是理所当然的。它主要与用户偏好有关,例如语言,主题,颜色等。

- 1.大型出版网站(例如报纸,博客等)会在用户的浏览器上设置自己的Cookie。基本上,它是第一方Cookie。
- 2.此外,这些网站还会向其他网站(如Google)发送请求。这是通过发出一个大小为1px x 1px的简单图像请求来完成的。此请求返回的cookie存储在用户的计算机上。此外部请求设置的Cookie是第三方cookie。
- 3.现在,如果用户导航到其他网站(如果该网站碰巧也有向同一外部域的请求),则浏览器将与此请求一起发送现有的Cookie。
- 4.由于共享cookie,外部服务能够确定给定用户访问了哪些页面,网站,并且根据用户的喜好,这些公司开始向用户显示目标广告。
- 5.随着越来越多的第三方Cookie被保存在用户的计算机上,广告变得更强大,更有针对性。这些第三方cookie是永久性cookie,其中,第一方cookie通常是会话cookie。





响应头的组成

响应头的组成

响应头 Response Header



状态行 Status line

响应头部字段 Response Header Fields

响应头由状态行和响应头部字段组成

状态行包含:



(Status Code)及 状态(Description)描述 构成

Response Version: HTTP/1.1

Status Code: 200

[Status Code Description: OK]

Response Phrase: OK

Accept-Ranges: bytes\r\n

Content-Encoding: gzip\r\n
Content-Length: 232\r\n

Content-Type: text/plain\r\n

Date: Wed, 07 Apr 2021 06:21:46 GMT\r\n

响应头部字段包含若干个属性,格式为"属性名:

属性值 (key:value)",是服务端返回给客户端的

信息





状态行的组成



HTTP版本 响应码 响应码描述

HTTP版本 Protocol Version 服务器HTTP协议以及版本,常见的有HTTP / 1.0 / 1.1 / 2.0

响应码 Response Code 服务器返回的响应状态代码

响应码描述 Description 状态代码的文本描述



HTTP状态码



状态码Status Code是一个3位数字

(如200),它的第一位代表了不同的响应状态。响应状态共有5种,含义如下。

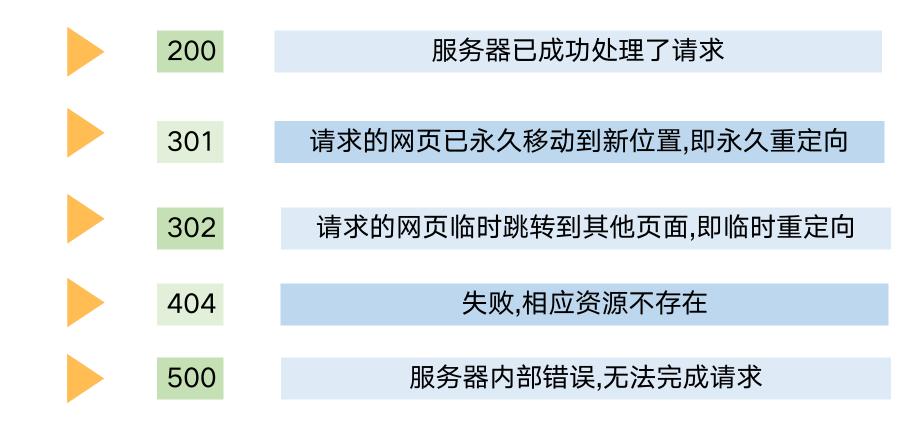
- 1 代表信息响应类,表示接收到请求并且继续处理,这类响应是临时响应。
- 2 代表处理成功响应类,表示动作被成功接收、理解和接受。
- 3 代表重定向响应类,为了完成指定的动作,必须接受进一步处理。
- 4 代表客户端错误,表示客户请求包含语法错误或者是不能正确执行的请求。
- 5 代表服务器端错误,服务器不能正确执行一个正确的请求。





常见响应状态码



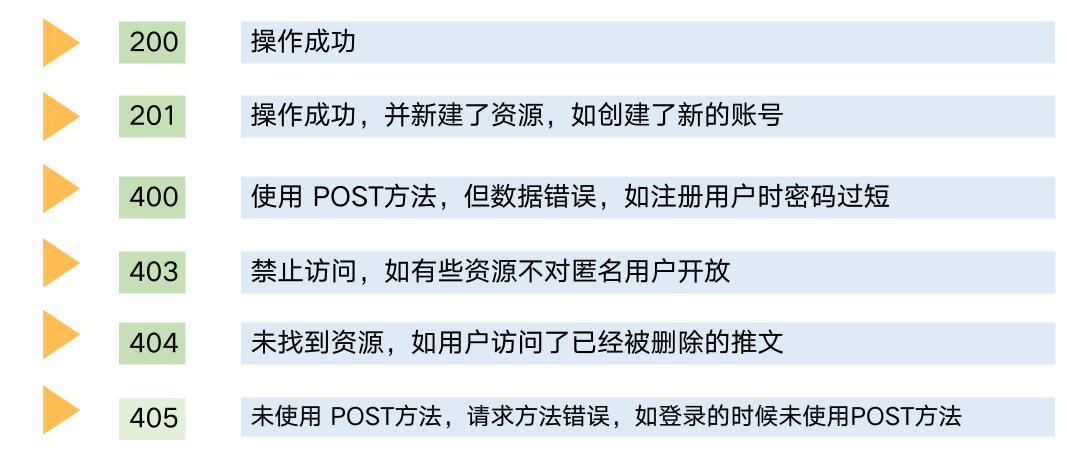






推特项目用到的状态码









常见响应头部字段



响应头头字段,服务器返回的附加信息,重要的有Set-Cookie、Content-Encoding等。

Content-Encoding

文档的编码(encode)方法。只有在解码之后,才可以得到 Content-Type头指定的内容类型

Content-Length

内容长度。只有当浏览器使用持久HTTP连接时才需要这个 数据

Set-Cookie

进行重定向或在创建某个新资源时指定位置 设置与页面关联的 Cookie



HTTP&HTTPS



1.HTTPS

HTTPS (Hyper Text Transfer Protocol Secure,超文本传输安全协议),比HTTP多了个"安全"。因为HTTP的传输是明文传输的,这很不安全,毕竟谁也不想被别人看光光!

而在HTTP之上增加的 SSL / TLS 层为我们的数据传输进行了加密。

2.HTTPS是如何实现"安全"的呢?

我们先来看看 HTTP 是怎么传输数据的: HTTP 将应用程序提供的数据封装后,明文交给位于运输层的TCP,而后发送到网络上。由于是明文传输,发送的信息可以在传输过程中被任意篡改,甚至被完全替换,安全性低,这就是HTTP的主要缺点。

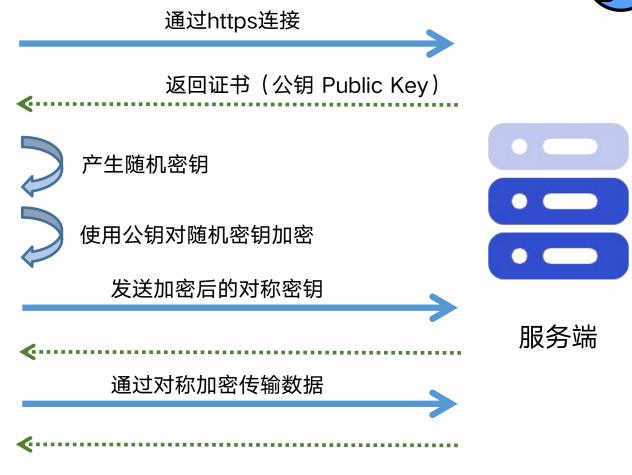


HTTPS建立连接的流程



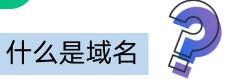
而HTTPS通过传送证书, 非对称加密、对称加密等 技术,来对数据加密,只 有数据的发送者和接收者 才能解密被加密的内容, 对于信息传递的中间人看 到的是被加密后的乱码。













域名(英语: Domain Name)是由一串用点分隔的名字组成的 Internet上某一台计算机或计算机组的名称,用于在数据传输时标识计算 机的电子方位(有时也指地理位置)。

域名的组成

www.twitter.com







三级域名

二级域名

顶级域名





常见的顶级域名













小明完成了他的网站开发,希望他的网站可以被大家访问。这时他可以去AWS / Aliyun购买一台云服务器,厂商会送一个公网IP绑定这台云服务器(Cloud Server),比如141.213.021.098

他的朋友就可以通过 141.213.021.098 访问小明的网站了,但是这个网址并不容易记忆,小明为了方便朋友,就去购买了一个域名 xiaoming.com ,这下每个人都能记住小明的网站地址了

那么,他人是怎么通过 xiaoming.com 访问到小明的服务器的呢?秘密就在DNS,在浏览器访问xiaoming.com 的时候, xiaoming.com 被翻译为 141.213.021.098,浏览器会通过 141.213.021.098 去访问小明部署在上服务上的网站





什么是域名解析(DNS)





DNS (Domain Name System)就像一个电话本,当我们需要给一个人打电话的时候,光知道名字是不行的,还需要通过电话簿查询到他的电话号码

DNS就相当于手机里电话簿,域名就是人名,而对应的IP地址就是电话号码。

通过ping 命令可以获取一个 网址 对应的 ip 地址, 聪明的你可以打开终端尝试一下

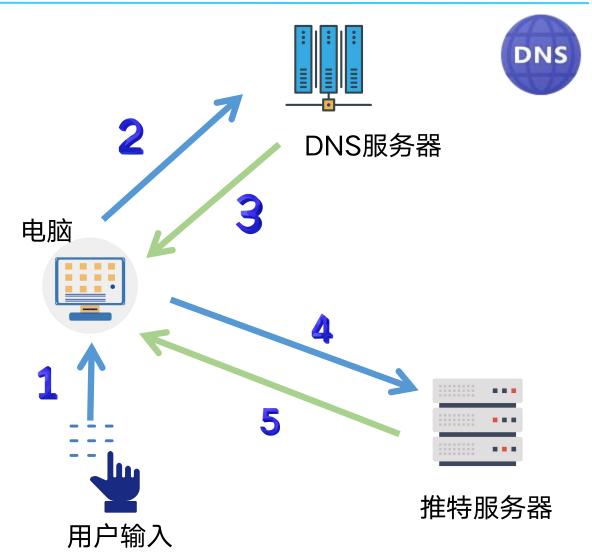
>>> ping www.twitter.com PING www.twitter.com (108.160.170.33) 56(84) bytes of data.





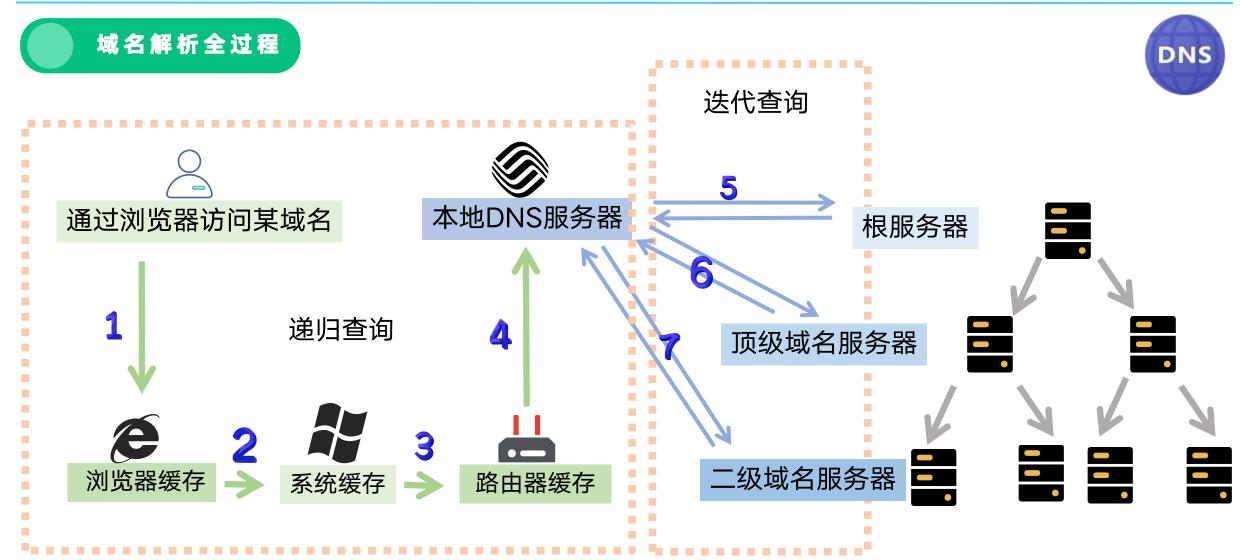
网页的浏览的背后

- 1. 输入www.twitter.com并回车
- ▶ 2. 浏览器发起域名解析 (DNS)
- 3. 从DNS服务器获取www.twitter.com 对应的ip地址(108.160.170.33)
- 4. 浏览器向108.160.170.33请求网页
- 5. 推特服务器响应请求



DNS





DNS



迭代查询

顶级域名服务器

二级域名服务器

本地DNS服务器

域名解析全过程

世界上一共有13个服务器,聪明的你想过一个问题吗这13个根服务器是怎承受住全世界电脑的DNS解析的? 毕竟我们经常会访问一些冷门网站

这个问题的答案是,对于一些经常访问的网站,会有各级缓存。 当然缓存有失效时间,这就需要进行DNS解析了,还有就是访问

以前没有访问过的网站,此时我们的本地DNS服务器就会帮助我们,本地DNS服务往往由ISP厂商提供,例如你使用了中国移动的宽带,那就会通过中国移动的DNS服务器经常解析,而这个服务器往往不只一台,各省各市都有对于的节点。除此之外,我们还可以指定第三方本地DNS服务器,比如国内著名的114.114.114.114,谷歌的8.8.8.8等等。

如果本地DNS找不到对于的记录,才回去根服务器查询,而每个根服务器背后都是一堆计算机 集群,可以顶住很大访问量。

课程版权归属于九章算法(杭州)科技有限公司,贩卖和传播盗版将被追究刑事责任

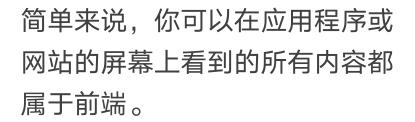




前端和后端的区别









后端,也称为"服务器端开发", 也就是在系统"后面"所发生的 事情。









HTML (HyperText Markup Language) 是超文本标记语言。



CSS (Cascading Style Sheets)是级联样式表。



JavaScript 是一种脚本语言,主要用于前端页面的DOM处理。

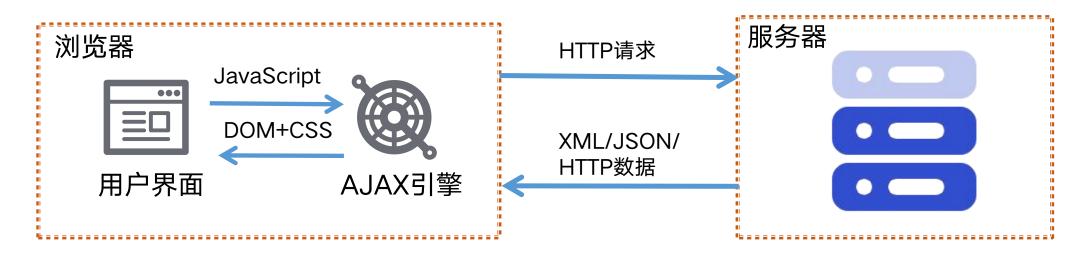






AJAX, Asynchronous JavaScript and XML是在不重新加载整个页面的情况下,与服务器交换数据并更新部分网页的技术。

AJAX 技术出现前,将一个商品添加至购物车会使得整个页面刷新,导致用户等待时间变长,降低用户体验。同时服务器的负载量也大。







生活中的Ajax——推特的关注

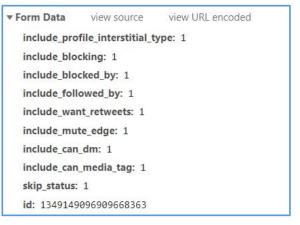




1 关注某个推特用户,按下 Follow 按钮

2 JS 为我们发起了 AJAX 请求,使用 post 方法提交了参数。





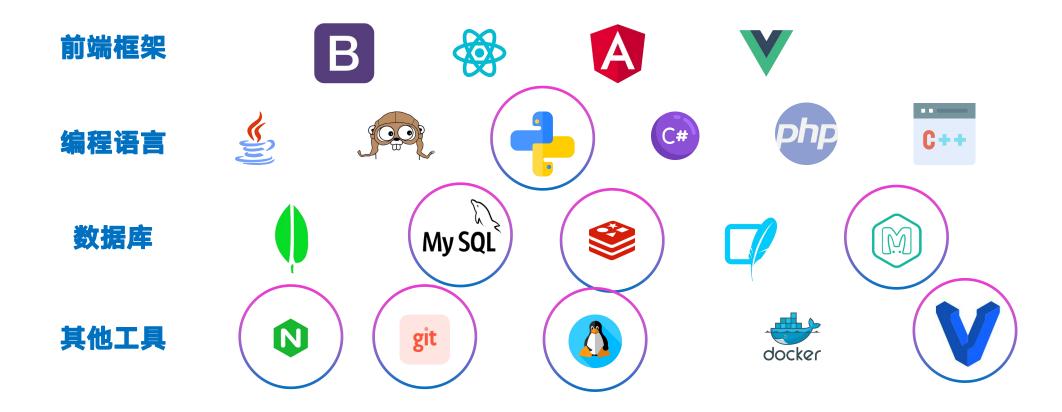


3 此时页面并没有刷新,但是 Follow 按钮变成了 Following, 这就是 AJAX 异步请求给我们 带来的局部刷新技术





后端概览







2021全球操作系统市场份额



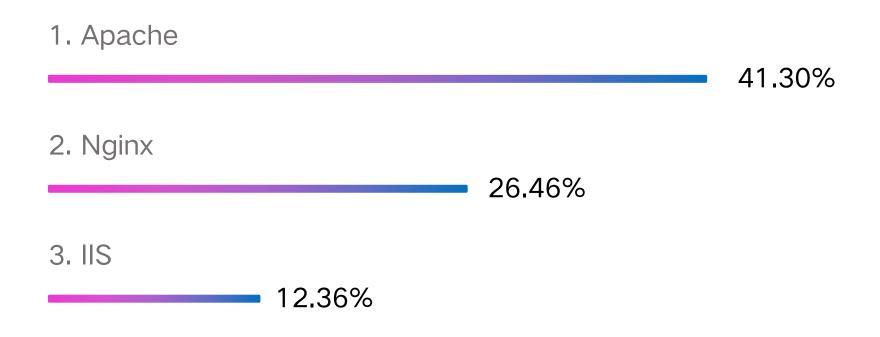
- 2. Windows Server
- **———** 12.55%
- 3. Unix
- **2.24**%

Linux 是服务器操作系统领域的领头羊





2021全球网络服务器市场份额



Apache 和 Nginx 是最流行的网络服务器





常见关系型数据库

大系型蚁塘库 Relational database



Oracle

MySQL

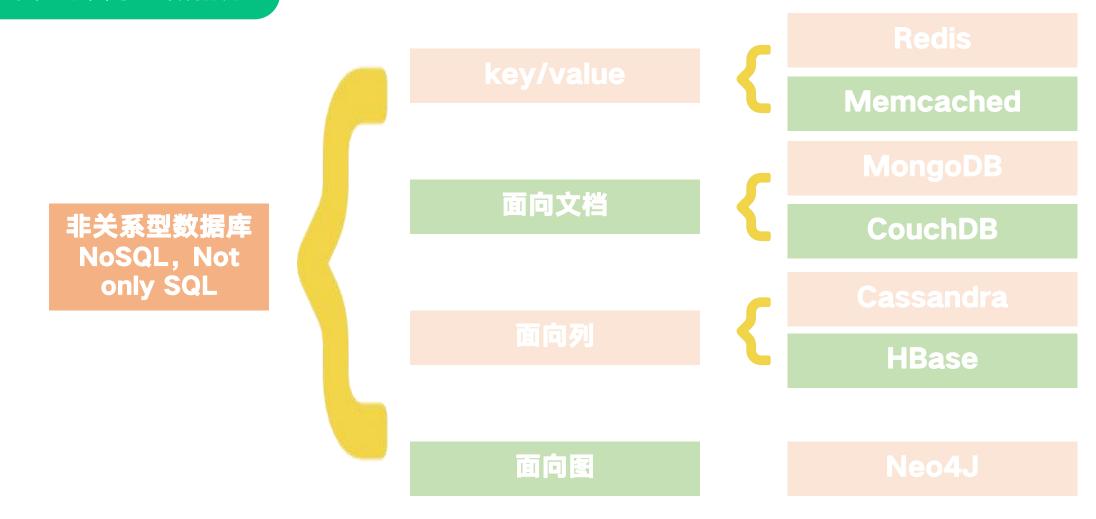
SQL Server

PostgeSQL





常见非关系型数据库

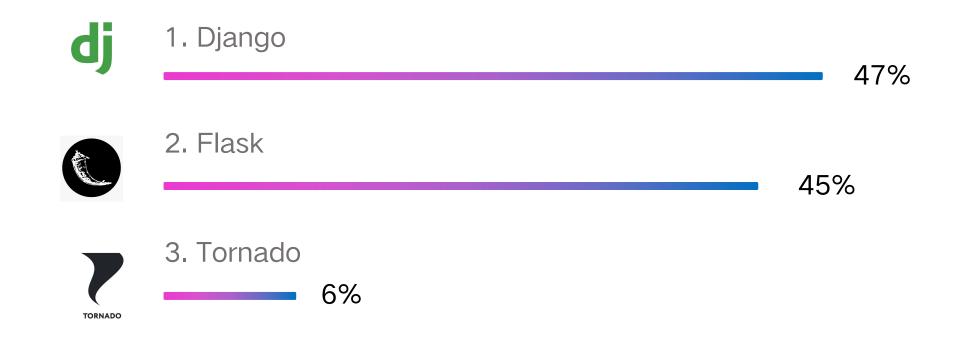


课程版权归属于九章算法(杭州)科技有限公司,贩卖和传播盗版将被追究刑事责任





2021 Python Web框架排行









我们编写的Django代码最后都要变成程序运行在服务器上,这就涉及到了使用服务器部署和运行我们的程序。Linux占据服务器市场超过80%的市场份额,接下来让我们一起熟悉一下Linux的命令。





磁盘管理



pwd (英文全拼: print work directory)

用于显示工作目录

语法

pwd [--help][--version]

实例

pwd # 输出结果 /root/test





磁盘管理



cd (英文全拼: change directory)

用于切换当前工作目录

语法

cd [dirName]

dirName: 要切换的目标目录。

实例

cd /usr/bin # 跳到 /usr/bin/

cd ~ # 跳到 home 目录

cd .. # 跳到父目录







mkdir (英文全拼: make directory)

用于创建目录

语法

mkdir [-p] dirName

实例

mkdir runoob # 在当前目录下,建立一个名为 runoob 的子目录:





磁盘管理



S(英文全拼: list files)

命令用于显示指定工作目录下之内容 (列出目前工作目录所含文件及子目录)

语法

Is [-alrtAFR] [name...]

实例

Is / #显示根目录下所有的目录和文件

bin dev lib media net root srv upload www boot etc lib64 miscopt sbin sys usr home lost+found mnt proc selinux tmp var





文件管理



cat (英文全拼: concatenate)

查看文件内容

语法

cat [-AbeEnstTuv] [--help] [--version] fileName

实例

cat .bashrc # 查看.bashrc中的内容 # ~/.bashrc: executed by bash(1) for non-login shells.





cat命令实战



cat 显示文件内容

在终端输入如下命令,即可查看ubuntu当前的软件源

>>> cat /etc/apt/sources.list

实例

.

deb http://mirrors.aliyun.com/ubuntu focal main restricted deb http://mirrors.aliyun.com/ubuntu focal-updates main restricted deb http://mirrors.aliyun.com/ubuntu focal universe deb http://mirrors.aliyun.com/ubuntu focal-updates universe





文件管理

rm (英文全拼: remove)

删除文件或者文件夹

语法

rm [options] name...

实例

rm .bashrc # 查看.bashrc中的内容 rm -r cpp # 加r参数,表示删除文件夹





文件管理



mV (英文全拼: move file)

移动文件或者文件夹, 也可以用户重命名

语法

mv [options] source dest mv [options] source... directory

实例

mv aaa bbb # 将文件 aaa 改名为 bbb mv info/ logs # 将 info 目录放入 logs 目录中





文件管理



CP (英文全拼: copy file)

复制文件或目录

语法

cp [options] source dest
cp [options] source... directory

实例

cp -r test/ newtest #将当前目录 test/ 下的 所有文件复制到新目录 newtest 下





文件管理



in (英文全拼: copy file)

为某一个文件在另外一个位置建立一个同步的链接

语法

In [参数][源文件或目录][目标文件或目录]

实例

In -s log2013.log link2013 # 给文件创建软链接 In log2013.log ln2013 # 给文件创建硬链接





In命令实战

创建一个指向python3的软连接

应用场景

在最新的发行版本终端输入python命令,出现了如下的提示

Command 'python' not found, did you mean:

command 'python3' from deb python3 command 'python' from deb python-is-python3

解决办法

在命令行输入如下命令:

>>> In -s /usr/bin/python3 /usr/bin/python

此命令会在为/usr/bin/python3在/usr/bin路径下创建了一个名为python的软连接,下次直接输入python就可以调用python3







Grep (英文全拼: Global Regular Expression Print)

用于查找文件里符合条件的字符串

语法

grep [-abcEFGhHilLnqrsvVwxy][-A<显示行数>][-B<显示列数>][-C<显示列数>][-d<进行动作>][-e<范本样式>][-f<范本文件>][--help][范本样式][文件或目录...]

实例

-rw-r--r-- 1 csy csy 3818 Mar 21 14:03 .bashrc



grep命令实战

查看所有和python有关的命令

应用场景

有时我们需要查看某个路径下有否有我们需要的文件,但又不想从众多文件中大海捞针,此时grep命令可以帮助我们只留下匹配的内容

>>> Is /usr/bin | grep python python3 python3-config python3.8 python3.8-config x86_64-linux-gnu-python3-config x86_64-linux-gnu-python3.8-config





系统管理



PS(英文全拼: process status)

显示当前进程的状态, 类似于 任务管理器

语法

ps [options] [--help]

实例

ps -ef | grep php #显示 php 的进程 root 794 1 0 2020? 00:00:52 php-fpm: master process (/etc/php/7.3/fpm/php-fpm.conf) www-data 951 794 0 2020? 00:24:15 php-fpm: pool

WWW





系统管理



kill

终止执行中的程序或工作

语法

ps [options] [--help]

实例

最常用的信号

1 (HUP): 重新加载进程。

9 (KILL): 杀死一个进程。

15 (TERM): 正常停止一个进程。

kill 12345 # 杀死pid为12345的进程 kill -HUP 12345 #发送SIGHUP信号,杀死pid为12345的进程





为什么要使用 Git

git

- 1 你是否有过关闭编辑器之后就无法回退代码?
 - 2 你是否有过把代码传到服务器却被繁琐的过程折磨?
 - 3 你是否经历过和他人合作写代码却把时间都浪费在了 代码合并上?

让我们来一起学习git吧,上诉的问题通通可以被这个工具解决





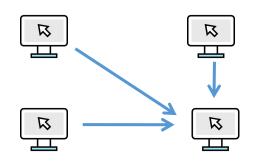
Git 介绍

Git是目前世界上最先进的分布式版本控制系统。

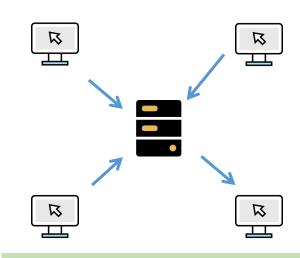
SVN 与 Git 最主要的区别

SVN是集中式版本控制系统,版本库是集中放在中央 服务器的。

Git是分布式版本控制系统,那么它就没有中央服务器的,每个人的电脑就是一个完整的版本库。



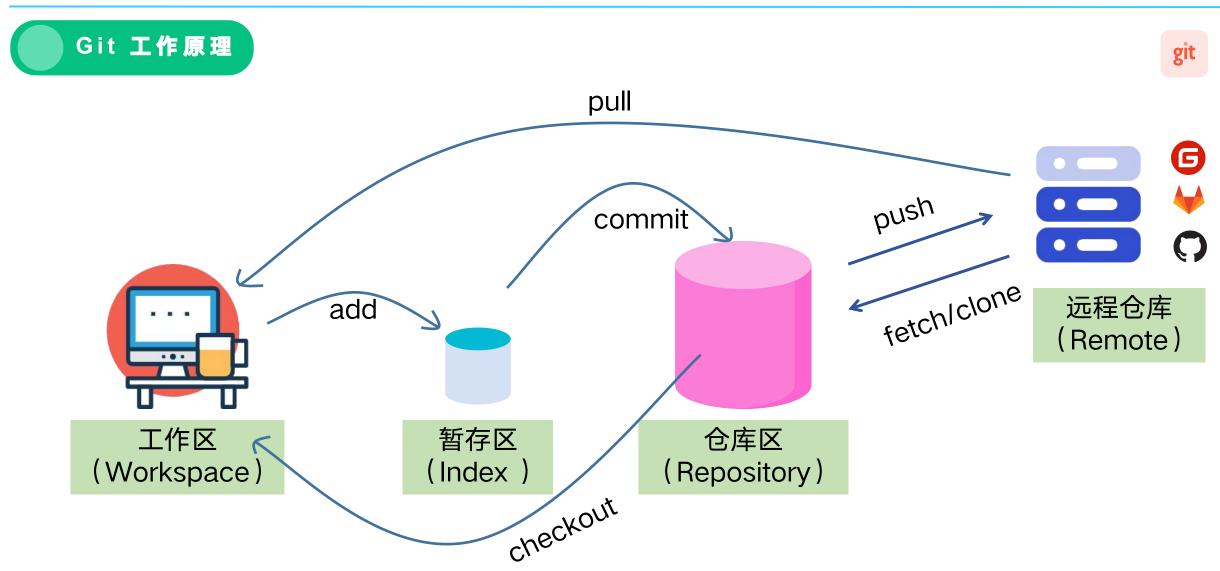
Git管理模式(去中心化)



SVN管理模式(有中心)

git









git init 把目录变成git可以管理的仓库,会在当前路径下建立一个.git文件夹

git add 添加文件到暂存区, reset命令可以将文件恢复到工作区

git commit 将暂存区内容添加到仓库中,checkout命令可以恢复到工作区

git push 上传远程代码并合并到工作区

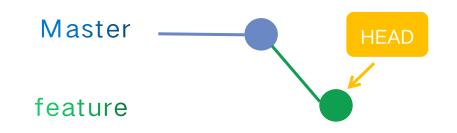
git pull 下载远程代码并合并到工作区







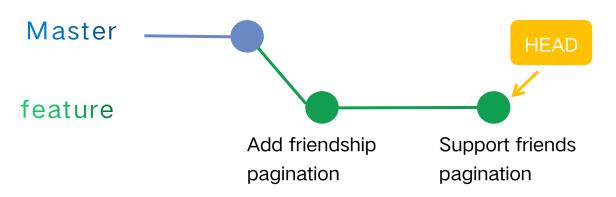




1. 从 master 拉出新分支, 并切换到新分支上进行开发。





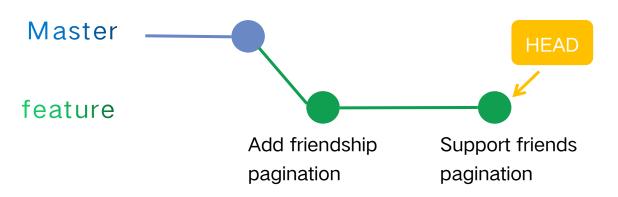


- 1. 从 master 拉出新分支, 并切换到新分支上进行开发。
- 2. 在新分支上进行开发。



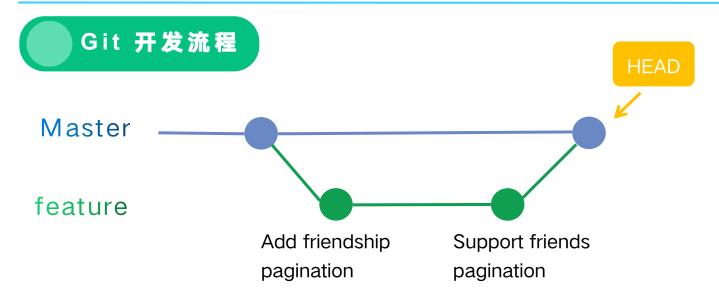
Git 开发流程

git



- 1. 从 master 拉出新分支, 并切换到新分支上进行开发。
- 2. 在新分支上进行开发。
- 3. 新分支开发完成后,就向 master 发起一个 pull request (简称PR),项目负责人会对你开发的代码进行审阅。





- 1. 从 master 拉出新分支, 并切换到新分支上进行开发。
- 2. 在新分支上进行开发。
- 3. 新分支开发完成后,就向 master 发起一个 pull request (简称PR),项目负责人会对你开发的代码进行审阅。
- 4. 如果你的 PR 被接受, 就合并进 master, 原来你拉出来的那个分支就被删除。

课程版权归属于九章算法(杭州)科技有限公司,贩卖和传播盗版将被追究刑事责任





Git 分支操作

git

- 查看分支: git branch
- **创建分支: git branch name**
- 切换分支: git checkout name
- 创建+切换分支: git checkout −b name
- 合并某分支到当前分支: git merge name
- 删除分支: git branch d name