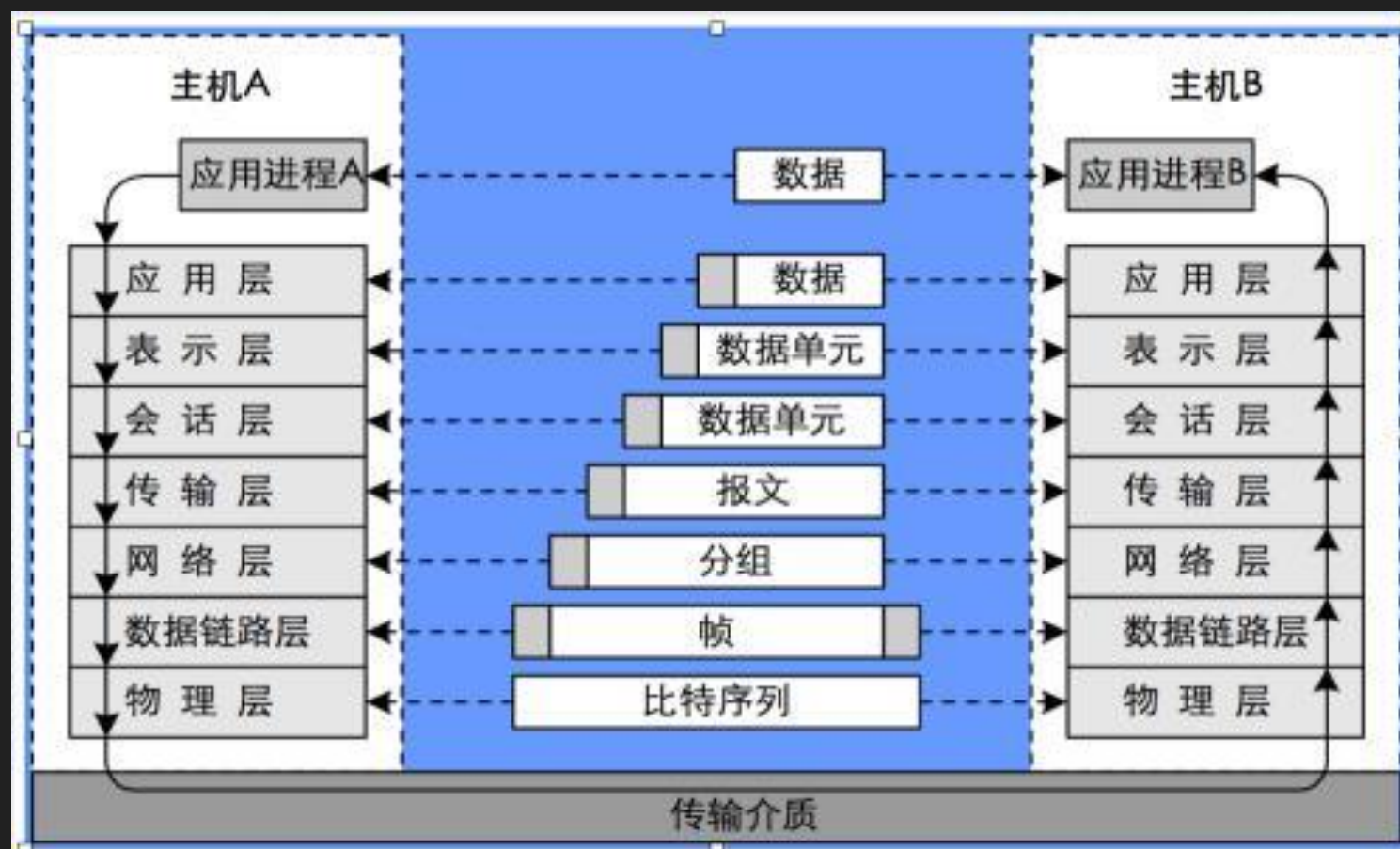


# Python与深度学习基础

张越一 zhyuey@ustc.edu.cn

# OSI模型



# TCP/IP协议

ISO/OSI 模型		TCP/IP 协议					TCP/IP 模型
应用层	文件传输 协议 (FTP)	远程登录 协议 (Telnet)	电子邮件 协议 (SMTP)	网络文件服 务协议 (NFS)	网络管理 协议 (SNMP)	应用层	
表示层							
会话层							
传输层	TCP                      UDP					传输层	
网络层	IP	ICMP	ARP      RARP			网际层	
数据链路层	Ethernet IEEE 802.3	FDDI	Token-Ring/ IEEE 802.5	ARCnet	PPP/SLIP	网络接口层	
物理层						硬件层	

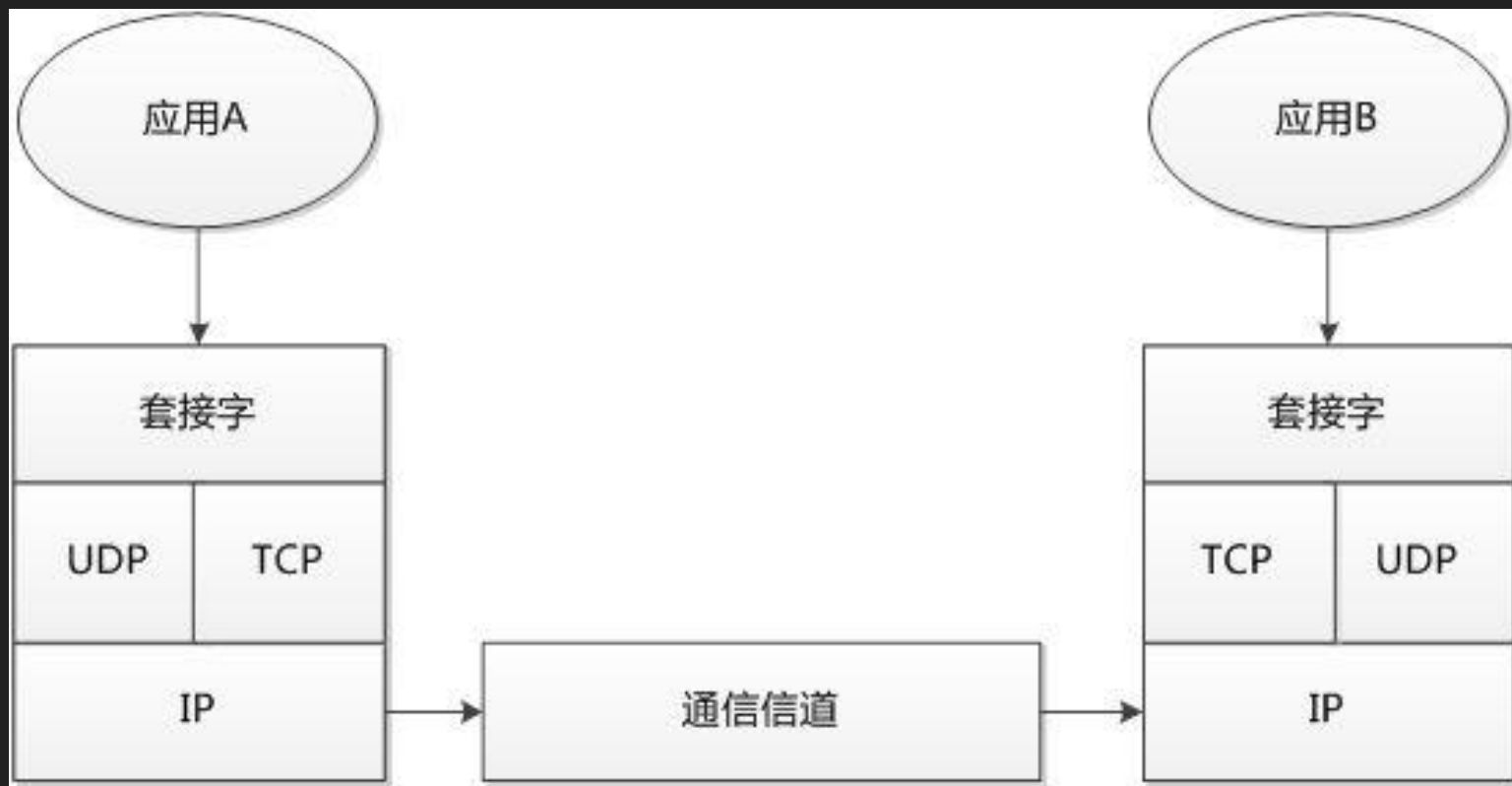
TCP/IP 模型与 OSI 模型的对比



**服务器之家**

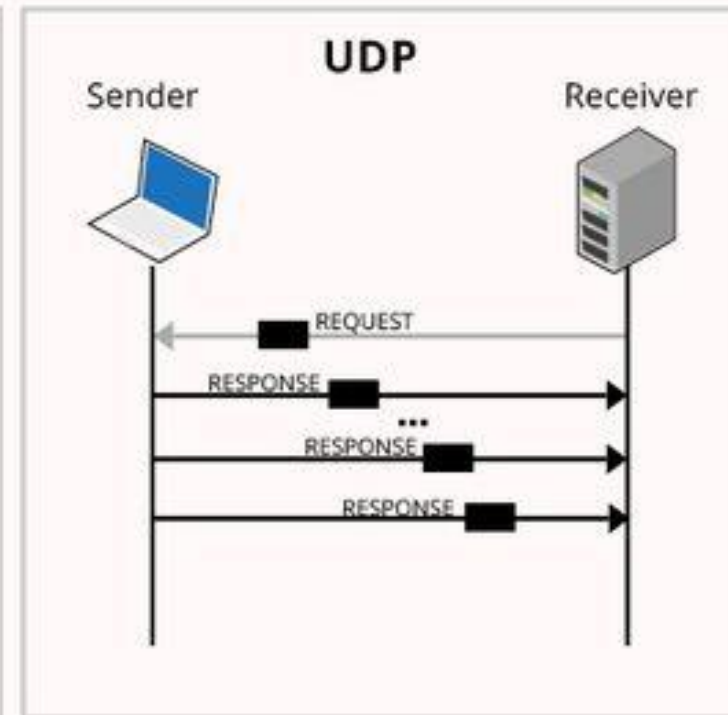
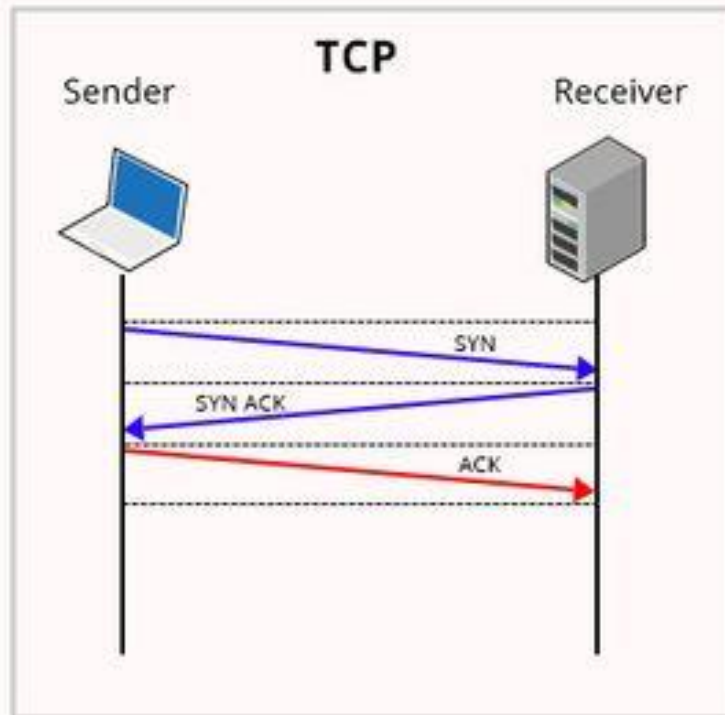
WWW.SERVER110.COM

# TCP and UDP



# TCP VS UDP

## TCP Vs UDP Communication



# TCP VS UDP

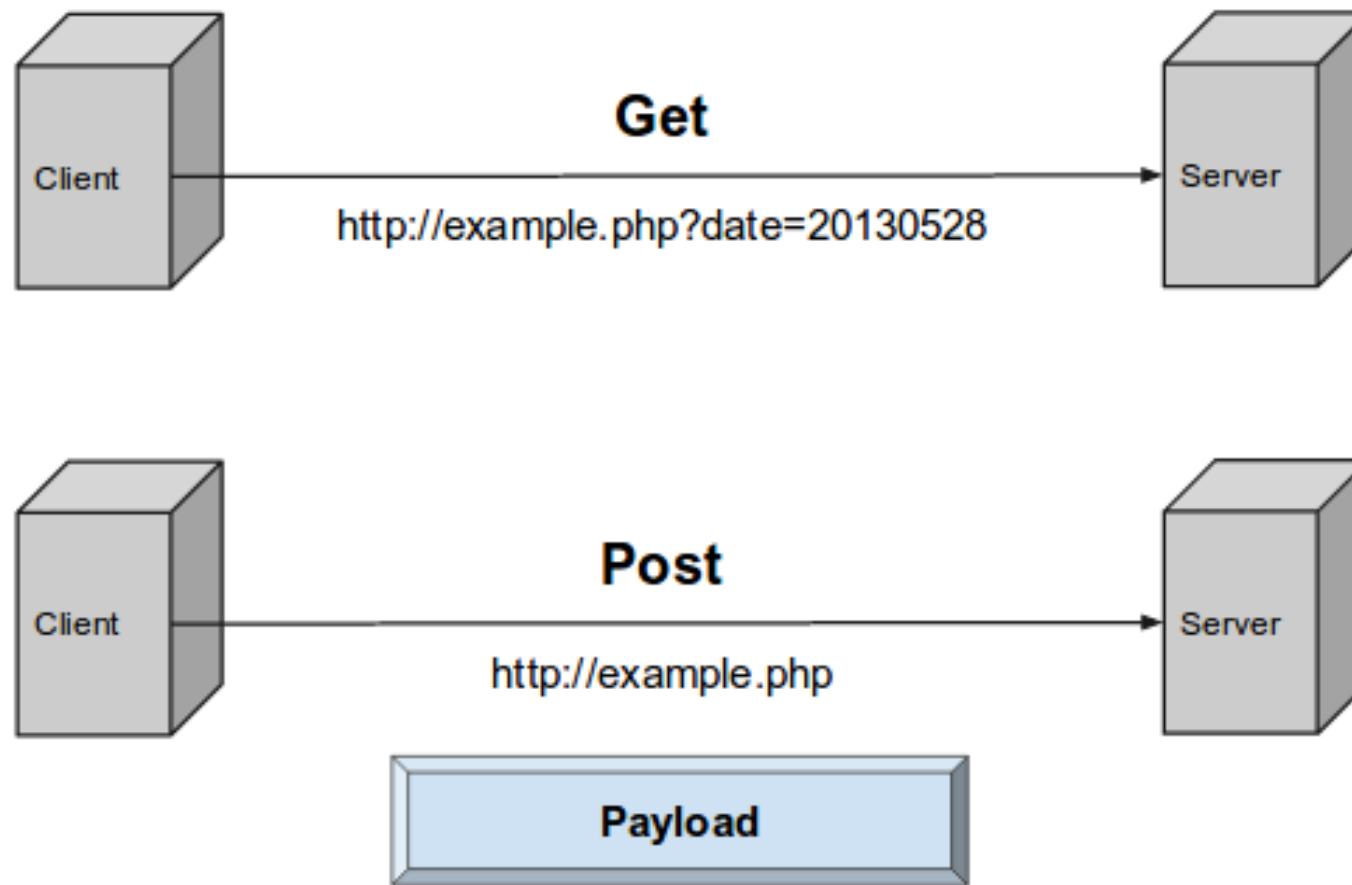
## TCP Vs. UDP

No.	TCP	UDP
1.	Connection Oriented Protocol	Connection-less Protocol
2.	Connection in byte stream	Connection in message stream
3.	It does't support multicasting and broadcasting	It supports broadcasting
4.	It provides error control and flow control	Error Control and Flow control is not provided
5.	Supports full Duplex	Does not support full Duplex
6.	TCP packet is called as Segment	UDP packet is called as User Datagram





# GET VS POST



*Form Data, JSON Strings, Query Parameters, View States, etc*

# GET VS POST

	GET	POST
后退按钮/刷新	无害	数据会被重新提交（浏览器应该告知用户数据会被重新提交）。
书签	可收藏为书签	不可收藏为书签
缓存	能被缓存	不能缓存
编码类型	application/x-www-form-urlencoded	application/x-www-form-urlencoded 或 multipart/form-data。为二进制数据使用多重编码。
历史	参数保留在浏览器历史中。	参数不会保存在浏览器历史中。
对数据长度的限制	是的。当发送数据时，GET 方法向 URL 添加数据；URL 的长度是受限制的（URL 的最大长度是 2048 个字符）。	无限制。
对数据类型的限制	只允许 ASCII 字符。	没有限制。也允许二进制数据。
安全性	与 POST 相比，GET 的安全性较差，因为所发送的数据是 URL 的一部分。  在发送密码或其他敏感信息时绝不要使用 GET ！	POST 比 GET 更安全，因为参数不会被保存在浏览器历史或 web 服务器日志中。
可见性	数据在 URL 中对所有人都是可见的。	数据不会显示在 URL 中。



# Json

- JSON(JavaScript Object Notation) 是一种轻量级的数据交换格式。易于人阅读和编写。同时也易于机器解析和生成。它基于JavaScript Programming Language, Standard ECMA-262 3rd Edition - December 1999的一个子集。

```
{ "employees":  
  [ { "firstName":"Bill" ,  
    "lastName":"Gates" },  
    { "firstName":"George" ,  
    "lastName":"Bush" },  
    { "firstName":"Thomas" ,  
    "lastName":"Carter" } ] }
```

# Network Package

urllib, urllib2 ?

No

We want something simple!!!

# Requests 让 HTTP 服务人类

- [http://cn.python-requests.org/zh\\_CN/latest/](http://cn.python-requests.org/zh_CN/latest/)
- Requests 允许你发送纯天然，植物饲养的 HTTP/1.1 请求，无需手工劳动。你不需要手动为 URL 添加查询字符串，也不需要为 POST 数据进行表单编码。Keep-alive 和 HTTP 连接池的功能是 100% 自动化的，一切动力都来自于根植在 Requests 内部的 urllib3。
- Twitter、Spotify、Microsoft、Amazon、Lyft、BuzzFeed、Reddit、NSA、女王殿下的政府、Amazon、Google、Twilio、Mozilla、Heroku、PayPal、NPR、Obama for America、Transifex、Native Instruments、Washington Post、Twitter、SoundCloud、Kippt、Readability、以及若干不愿公开身份的联邦政府机构都在内部使用。



# BeautifulSoup

- <https://www.crummy.com/software/BeautifulSoup/bs4/doc/>
- Beautiful Soup 是一个可以从HTML或XML文件中提取数据的Python库.它能够通过你喜欢的转换器实现惯用的文档导航,查找,修改文档的方式.Beautiful Soup会帮你节省数小时甚至数天的工作时间.
- Latest: 4.7.1
- Beautiful Soup provides a few simple methods and Pythonic idioms for navigating, searching, and modifying a parse tree
- Beautiful Soup automatically converts incoming documents to Unicode and outgoing documents to UTF-8.
- Beautiful Soup sits on top of popular Python parsers like lxml and html5lib, allowing you to try out different parsing strategies or trade speed for flexibility.



# Flask

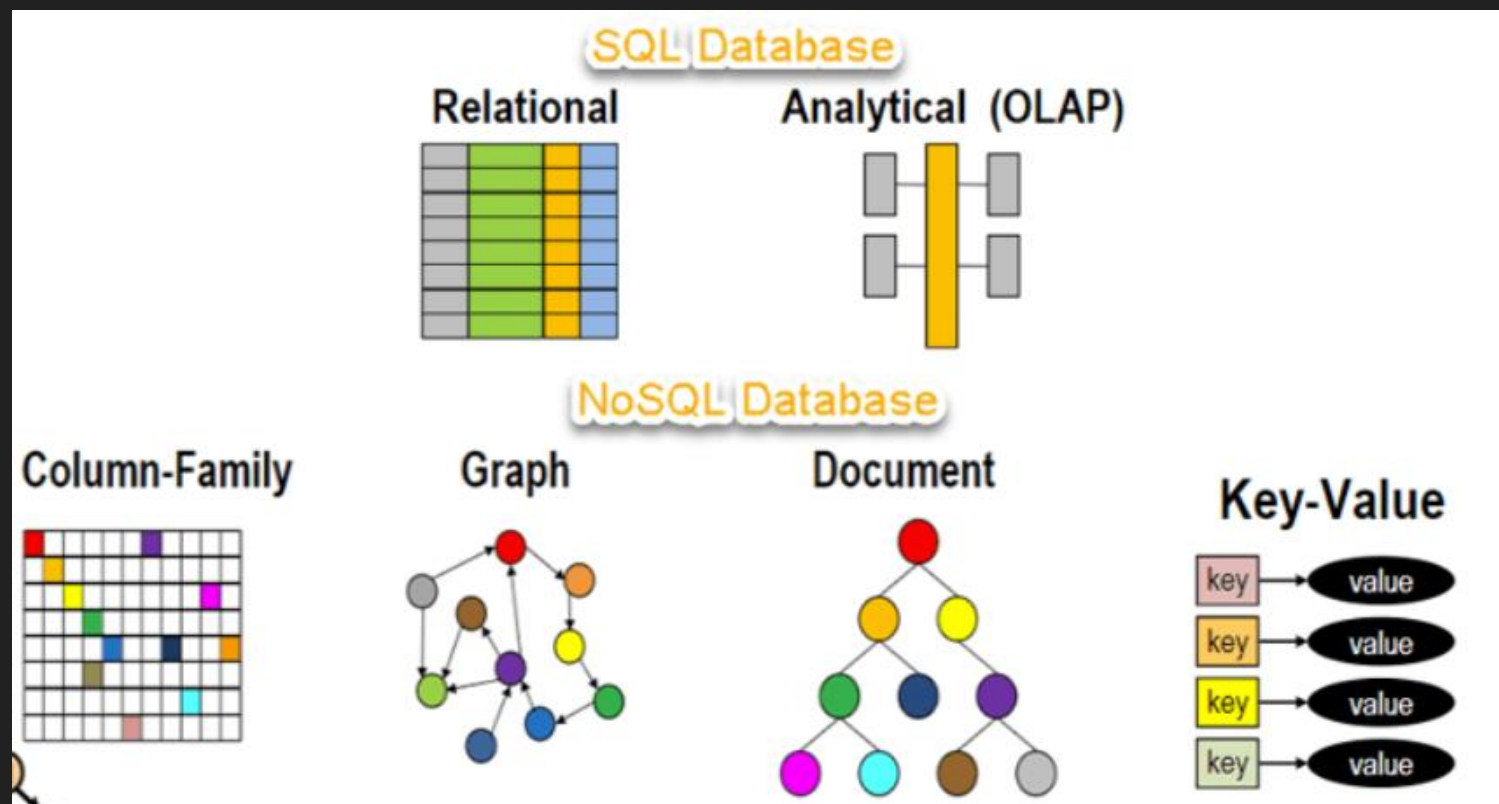
- Flask是一个基于Python实现的web开发的'微'框架
- 诞生于2010年，作者是Armin Ronacher
- 文档齐全，上手方便
- 开发效率高
- 优秀的拓展机制
- Jinija2模板
- PyCharm有原生支持
- <http://localhost:5000> or <http://127.0.0.1:5000>





# SQL and NoSql

- 结构化查询语言(Structured Query Language)简称SQL(发音: /'es kju: 'el/ "S-Q-L"), 是一种特殊目的的编程语言, 是一种数据库查询和程序设计语言, 用于存取数据以及查询、更新和管理关系数据库系统
- NoSQL, 泛指非关系型的数据库。面对超大规模和高并发的数据, SQL力不从心了。NoSQL数据库的产生就是为了解决大规模数据集合多重数据种类带来的挑战, 尤其是大数据应用难题。





# SQL Database



ORACLE®

PostgreSQL



Microsoft®  
SQL Server®

TURBODB  
Embedded



MySQL™

# NoSQL Database



CouchDB  
relax



*Cassandra*



mongoDB



membase



# SQLite

- a small, fast, self-contained, high-reliability, full-featured, SQL database engine
- SQLite is the most used database engine in the world.
- SQLite is built into all mobile phones and most computers and comes bundled inside countless other applications that people use every day.



# Great Python

- 1. Python Has A Rich And Supportive Community
- 2. Easy To Code And Write
- 3. Open-Source And Availability
- 4. Standard Library
- 5. Cross-Platform Language

