

- **1.python历史**

- Python的意思是蟒蛇
- Python的作者是Guido van Rossum（龟叔）
- Python是龟叔在1989年圣诞节期间，为了打发无聊的圣诞节而用C编写的一个编程语言
- Python正式诞生于1991年
- Python的解释器如今有多个语言实现，我们常用的是CPython（官方版本的C语言实现），其他还有Jython（可以运行在Java平台）、IronPython（可以运行在.NET和Mono平台）、PyPy（Python实现的，支持JIT即时编译）
- Python目前有两个版本，Python2和Python3，最新版分别为2.7.12和3.7.3，由于py2在2019年年底将会停止使用，现阶段大部分公司开始由py2转向py3;

- **Python优缺点**

- **优点**

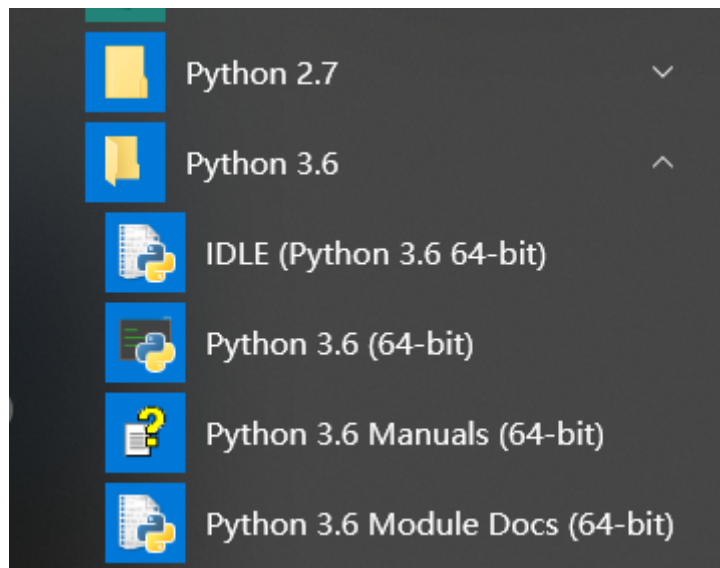
- 简单——Python是一种代表简单主义思想的语言。阅读一个良好的Python程序就感觉像是在读英语一样，尽管这个英语的要求非常严格！Python的这种伪代码本质是它最大的优点之一。它使你能够专注于解决问题而不是去搞明白语言本身。
- 易学——就如同你即将看到的一样，Python极其容易上手。前面已经提到了，Python有极其简单的语法。
- 免费、开源——Python是FLOSS（自由/开放源码软件）之一。简单地说，你可以自由地发布这个软件的拷贝、阅读它的源代码、对它做改动、把它的一部分用于新的自由软件中。FLOSS是基于一个团体分享知识的概念。这是为什么Python如此优秀的原因之一——它是由一群希望看到一个更加优秀的Python的人创造并经常改进着的。
- 高层语言——当你用Python语言编写程序的时候，你无需考虑诸如如何管理你的程序使用的内存一类的底层细节。
- 可移植性——由于它的开源本质，Python已经被移植在许多平台上（经过改动使它能够工作在不同平台上）。如果你小心地避免使用依赖于系统的特性，那么你的所有Python程序无需修改就可以在下述任何平台上运行。这些平台包括Linux、Windows、FreeBSD、Macintosh、Solaris、OS/2、Amiga、AROS、AS/400、BeOS、OS/390、z/OS、Palm OS、QNX、VMS、Psion、Acom RISC OS、VxWorks、PlayStation、Sharp Zaurus、Windows CE甚至还有PocketPC、Symbian以及Google基于linux开发的Android平台！

- 解释性——一个用编译性语言比如C或C++写的程序可以从源文件（即C或C++语言）转换到一个你的计算机使用的语言（二进制代码，即0和1）。这个过程通过编译器和不同的标记、选项完成。当你运行你的程序的时候，连接/转载器软件把你的程序从硬盘复制到内存中并且运行。而Python语言写的程序不需要编译成二进制代码。你可以直接从源代码运行程序。在计算机内部，Python解释器把源代码转换成称为字节码的中间形式，然后再把它翻译成计算机使用的机器语言并运行。事实上，由于你不再需要担心如何编译程序，如何确保连接转载正确的库等等，所有这一切使得使用Python更加简单。由于你只需要把你的Python程序拷贝到另外一台计算机上，它就可以工作了，这也使得你的Python程序更加易于移植。
- 面向对象——Python既支持面向过程的编程也支持面向对象的编程。在“面向过程”的语言中，程序是由过程或仅仅是可重用代码的函数构建起来的。在“面向对象”的语言中，程序是由数据和功能组合而成的对象构建起来的。与其他主要的语言如C++和Java相比，Python以一种非常强大又简单的方式实现面向对象编程，python一切皆对象。
- 可扩展性——如果你需要你的一段关键代码运行得更快或者希望某些算法不公开，你可以把你的部分程序用C或C++编写，然后在你的Python程序中使用它们。
- 丰富的库——Python标准库确实很庞大。它可以帮助你处理各种工作，包括正则表达式、文档生成、单元测试、线程、数据库、网页浏览器、CGI、FTP、电子邮件、XML、XML-RPC、HTML、WAV文件、密码系统、GUI（图形用户界面）、Tk和其他与系统有关的操作。记住，只要安装了Python，所有这些功能都是可用的。这被称作Python的“功能齐全”理念。除了标准库以外，还有许多其他高质量的库，如wxPython、Twisted和Python图像库等等。
- 规范的代码——Python采用强制缩进的方式使得代码具有极佳的可读性。
- 缺点
 - 运行速度，有速度要求的话，用C++改写关键部分吧。
 - 国内市场较小（国内以python来做主要开发的，目前只有一些web2.0公司）。但时间推移，目前很多国内软件公司，尤其是游戏公司，也开始规模使用他。
 - 中文资料匮乏（好的python中文资料屈指可数）。托社区的福，有几本优秀的教材已经被翻译了，但入门级教材多，高级内容还是只能看英语版。
 - 构架选择太多（没有像C#这样的官方.net构架，也没有像ruby由于历史较短，构架开发的相对集中。Ruby on Rails 构架开发中小型web程序天下无敌）。不过这也从另一个侧面说明，python比较优秀，吸引的人才多，项目也多。
 - 运行速度，有速度要求的话，用C++改写关键部分吧。

- 国内市场较小（国内以python来做主要开发的，目前只有一些web2.0公司）。但时间推移，目前很多国内软件公司，尤其是游戏公司，也开始规模使用他。
- 中文资料匮乏（好的python中文资料屈指可数）。托社区的福，有几本优秀的教材已经被翻译了，但入门级教材多，高级内容还是只能看英语版。
- 构架选择太多（没有像C#这样的官方.net构架，也没有像ruby由于历史较短，构架开发的相对集中。Ruby on Rails 构架开发中小型web程序天下无敌）。不过这也从另一个侧面说明，python比较优秀，吸引的人才多，项目也多。

• 搭建编程环境

- 安装下载好的python3.x版本，安装时勾选add python 3.X to path选项；安装完成将会在菜单生成python3.x的目录；



- 在win+r, cmd中输入python
- 显示下图，表示安装完成，环境变量配置成功；

```
C:\WINDOWS\system32\cmd.exe - python
Microsoft Windows [版本 10.0.17134.765]
(c) 2018 Microsoft Corporation. 保留所有权利。

C:\Users\93999>python
Python 3.6.5 (v3.6.5:f59c0932b4, Mar 28 2018, 17:00:18) [MSC v.1900 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> _
```

- 在终端输出hello world

```
Microsoft Windows [版本 10.0.17134.765]
(c) 2018 Microsoft Corporation. 保留所有权利。

C:\Users\93999>python
Python 3.6.5 (v3.6.5:f59c0932b4, Mar 28 2018, 17:00:18) [MSC v.1900 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> print("hello world")
hello world
>>>
```

- 命令行执行代码

-
-

```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [版本 10.0.17134.765]
(c) 2018 Microsoft Corporation. 保留所有权利。

C:\Users\93999>F:\python_learn001\python_001.py
C:\Users\93999>python F:\python_learn001\python_001.py
hello world!
C:\Users\93999>
```

- 安装pycharm

- 下载pycharm专业版，特殊方法使用或使用社区版；
- pycharm快捷键及常用配置

- **1、编辑 (Editing)**

- Ctrl + Space 基本的代码完成 (类、方法、属性)
 - Ctrl + Alt + Space 快速导入任意类
 - Ctrl + Shift + Enter 语句完成
 - Ctrl + P 参数信息 (在方法中调用参数)
 - Ctrl + Q 快速查看文档
 - Shift + F1 外部文档
 - Ctrl + 鼠标 简介
 - Ctrl + F1 显示错误描述或警告信息
 - Alt + Insert 自动生成代码
 - Ctrl + O 重新方法
 - Ctrl + Alt + T 选中
 - Ctrl + / 行注释
 - Ctrl + Shift + / 块注释

- Ctrl + W 选中增加的代码块
- Ctrl + Shift + W 回到之前状态
- Ctrl + Shift +]/[选定代码块结束、开始
- Alt + Enter 快速修正
- Ctrl + Alt + L 代码格式化
- Ctrl + Alt + O 优化导入
- Ctrl + Alt + I 自动缩进
- Tab / Shift + Tab 缩进、不缩进当前行
- Ctrl+X/Shift+Delete 剪切当前行或选定的代码块到剪贴板
- Ctrl+C/Ctrl+Insert 复制当前行或选定的代码块到剪贴板
- Ctrl+V/Shift+Insert 从剪贴板粘贴
- Ctrl + Shift + V 从最近的缓冲区粘贴
- Ctrl + D 复制选定的区域或行
- Ctrl + Y 删除选定的行
- Ctrl + Shift + J 添加智能线
- Ctrl + Enter 智能线切割
- Shift + Enter 另起一行
- Ctrl + Shift + U 在选定的区域或代码块间切换
- Ctrl + Delete 删除到字符结束
- Ctrl + Backspace 删除到字符开始
- Ctrl + Numpad+/- 展开折叠代码块
- Ctrl + Numpad+ 全部展开
- Ctrl + Numpad- 全部折叠
- Ctrl + F4 关闭运行的选项卡
- **2、查找/替换(Search/Replace)**
- F3 下一个
- Shift + F3 前一个
- Ctrl + R 替换
- Ctrl + Shift + F 全局查找
- Ctrl + Shift + R 全局替换
- **3、运行(Running)**
- Alt + Shift + F10 运行模式配置
- Alt + Shift + F9 调试模式配置
- Shift + F10 运行
- Shift + F9 调试
- Ctrl + Shift + F10 运行编辑器配置

- Ctrl + Alt + R 运行manage.py任务
- **4、调试(Debugging)**
- F8 跳过
- F7 进入
- Shift + F8 退出
- Alt + F9 运行游标
- Alt + F8 验证表达式
- Ctrl + Alt + F8 快速验证表达式
- F9 恢复程序
- Ctrl + F8 断点开关
- Ctrl + Shift + F8 查看断点
- **5、导航(Navigation)**
- Ctrl + N 跳转到类
- Ctrl + Shift + N 跳转到符号
- Alt + Right/Left 跳转到下一个、前一个编辑的选项卡
- F12 回到先前的工具窗口
- Esc 从工具窗口回到编辑窗口
- Shift + Esc 隐藏运行的、最近运行的窗口
- Ctrl + Shift + F4 关闭主动运行的选项卡
- Ctrl + G 查看当前行号、字符号
- Ctrl + E 当前文件弹出
- Ctrl+Alt+Left/Right 后退、前进
- Ctrl+Shift+Backspace 导航到最近编辑区域
- Alt + F1 查找当前文件或标识
- Ctrl+B / Ctrl+Click 跳转到声明
- Ctrl + Alt + B 跳转到实现
- Ctrl + Shift + I 查看快速定义
- Ctrl + Shift + B 跳转到类型声明
- Ctrl + U 跳转到父方法、父类
- Alt + Up/Down 跳转到上一个、下一个方法
- Ctrl +]/[跳转到代码块结束、开始
- Ctrl + F12 弹出文件结构
- Ctrl + H 类型层次结构
- Ctrl + Shift + H 方法层次结构
- Ctrl + Alt + H 调用层次结构
- F2 / Shift + F2 下一条、前一条高亮的错误

- F4 / Ctrl + Enter编辑资源、查看资源
- Alt + Home显示导航条F11书签开关
- Ctrl + Shift + F11书签助记开关
- Ctrl + #[0-9]跳转到标识的书签
- Shift + F11显示书签

• 6、搜索相关(Usage Search)

- Alt + F7/Ctrl + F7文件中查询用法
- Ctrl + Shift + F7文件中用法高亮显示
- Ctrl + Alt + F7显示用法

• 7、重构(Refactoring)

- F5复制F6剪切
- Alt + Delete安全删除
- Shift + F6重命名
- Ctrl + F6更改签名
- Ctrl + Alt + N内联
- Ctrl + Alt + M提取方法
- Ctrl + Alt + V提取属性
- Ctrl + Alt + F提取字段
- Ctrl + Alt + C提取常量
- Ctrl + Alt + P提取参数

• 8、控制VCS/Local History

- Ctrl + K提交项目
- Ctrl + T更新项目
- Alt + Shift + C查看最近的变化
- Alt + BackQuote(`)VCS快速弹出

• 9、模版(Live Templates)

- Ctrl + Alt + J当前行使用模版
- Ctrl + J 插入模版

• 10、基本(General)

- Alt + #[0-9]打开相应的工具窗口
- Ctrl + Alt + Y同步
- Ctrl + Shift + F12最大化编辑开关
- Alt + Shift + F添加到最喜欢
- Alt + Shift + I根据配置检查当前文件
- Ctrl + BackQuote(`)快速切换当前计划
- Ctrl + Alt + S 打开设置页

- Ctrl + Shift + A查找编辑器里所有的动作
- Ctrl + Tab在窗口间进行切换
- **一些常用设置:**
 - 1. pycharm默认是自动保存的, 习惯自己按ctrl + s 的可以进行如下设置:
 - 1. file -> Setting -> General -> Synchronization -> Save files on frame deactivation 和 Save files automatically if application is idle for .. sec 的勾去掉
 - 2. file -> Setting -> Editor -> Editor Tabs -> Mark modified tabs with asterisk 打上勾
 - 2. Alt + Enter: 自动添加包
 - 3. 对于常用的快捷键, 可以设置为visual studio(eclipse...)一样的:
 - file -> Setting -> Keymap -> Keymaps -> visual studio -> Apply
 - 4. Pycharm中默认是不能用Ctrl+滚轮改变字体大小的, 可以在file -> Setting -> Editor -> Mouse中设置
 - 5. 要设置Pycharm的字体, 要先在file -> Setting -> Editor -> Editor中选择一种风格并保存, 然后才可以改变
 - 6. 在setting中搜索theme可以改变主题, 所有配色统一改变