

螺丝工作室第10周例会

基本介绍

Python 是什么



大蟒蛇 (X)

编程语言 (O)

脚本小子常用工具 (OO)

官方解释

Python is an **interpreted**, **** high-level**** and **general-purpose** programming language. Python's design philosophy emphasizes code readability with its notable use of significant whitespace. Its language constructs and **object-oriented** approach aim to help programmers write clear, logical code for small and large-scale projects.

Interpreted: (绝大多数情况下是) 解释语言, 而非大家所熟悉的C/C++等编译性语言j

high-level: Python是一种“高级语言”, 抽象层次高

general-purpose: Python可跨平台、用途极广、生态良好, 特别是近年在数据科学领域热度极高

Python当前有Python2与Python3两个大版本, 当前主流开发均基于Python3, 直接学3就好, 基本不用考虑2。

Python可以用来做什么

- Web 应用开发
- 自动化运维
- 数据分析/科学计算。。。
- 各种脚本

Python最主流的解释器(CPython)基于C语言开发，与C/C++/Fortran等语言有极好的亲和性，互相调用非常方便（当下最推荐使用pybind11）

Python有极强的扩展性，其他语言编写的模块/框架可以较容易地与其进行衔接，有非常强大的生态环境，日常工作中相当多领域的工具工具都提供了Python接口。

Python的优势/劣势

- 优势：跨平台；编程灵活、简便；开源生态好，同时主流厂商均大力推广
- 劣势：男默女泪的性能（性能瓶颈部分往往需要利用C/C++等语言实现，然后在Python中调用）；动态性导致代码规模变大后潜在bug多；只会python容易变成脚本小子

开发环境搭建

概述

当下主流的两种Python环境搭建方法：

1. 纯粹的Python环境，直接装在系统中，手动管理环境
2. 使用conda, virtualenv等工具进行管理

如果只是临时一用，或者没有使用不同版本Python和package的需求，可以采用方法（1），不过长期使用的话最好还是选择（2），其中推荐conda。

conda介绍与安装



conda是一个python环境管理工具，我们在安装时一般会选择以conda为基础的安装套件，比如anaconda（有图形界面，但是做的很烂，bug比feature多，不推荐）或者miniconda（相当于前者的无图形界面精简版，但是其他功能很完整，推荐安装）。

miniconda在国内清华等镜像站上都有定期更新的资源，可以直接下载安装，安装过程中一般不会遇到什么阻碍。最后记得一点，如果安装时没有选择自动添加环境变量的话，则需要手动添加一下，否则没法直接使用conda命令。

利用conda进行环境配置

conda安装完成后（一般会安装在用户根目录，miniconda安装的话就会有~/miniconda这个目录），会默认带有一个叫做root的环境，可以通过以下命令查看

```
conda env list
```

可以看到类似这样的输出

```
# conda environments:
#
root                  *  /home/xsxsx/anaconda3
```

除了root外，我们可以新建Python环境，使用以下命令：

```
conda create -n testenv
```

之后就可以看到

```
# conda environments:
#
testenv               /home/xsxsx/anaconda3/envs/testenv
root                  *  /home/xsxsx/anaconda3
```

```
conda list
```

命令可以用来查看当前环境里面都有哪些packages。

要切换到root以外的环境，windows下使用

```
activate <envname>
```

linux下使用

```
source activate <envname>
```

要切换回root环境，只需要分别使用

```
deactivate
```

和

```
source deactivate
```

即可。

root以外的环境都位于miniconda/envs/下面，例如testenv就是miniconda/envs/testenv/，删除对应文件夹就可以移除该环境相关的所有内容，当然更合适的方法是

```
conda remove --name <envname> --all
```

常用命令就是以上几个，如果需要指定安装python版本等，可以自行查询参数。

包管理

使用conda后，Python主要有两种包管理方式，pip（Python原生）和conda。两者均有大量国内镜像站，推荐切换至国内源，速度会快很多。

两者使用方法类似，但是要注意两者的下载源并不相同，虽然大多数情况下不会引起问题，可以放心混用，但是偶尔也会有冲突的情况。

个人经验：稳定性上，pip官方源（PyPI）> pip第三方源>conda官方源>>conda第三方源，pip可以满足需求的时候尽量用pip。

IDE/编辑器/交互式环境

IDE范畴内有且只有一个推荐：Pycharm，全面超越其他所有Python专用IDE，不要用anaconda自带的spider等IDE，用户体验差距极大。



通过学生邮箱注册账号可以免费使用专业版（推荐使用，不过大工邮箱不知道为什么被ban了），嫌麻烦也可以用教育版，对于现阶段而言没有多大区别。

编辑器推荐VScode，在写脚本/轻量级代码的时候比Pycharm更加灵活。

Python的交互式开发环境推荐使用ipython和jupyter notebook/lab，不推荐直接使用python自带的命令行。

!python是一个增强的python命令行，临时写脚本一般都用这个；jupyter notebook和jupyter lab则是一个基于网页的python交互式环境（当然现在功能丰富了很多，lab目前已经支持了很多js插件），后者是前者的升级版，配置文件都有很大一部分是通用的。

PYTHON语法

Python语法非常简单，我们就基于菜鸟教程来进行讲解。学习Python基础部分要注意下面这些要点：

- Python动态性和抽象层次很高，学习时不要受到C/C++思维上的约束。
- 学Python一定要注重代码规范（其实其他语言也一样），详细可参考PEP8。
- 一定要清醒地认识到用Python调package绝不意味着自己掌握了这个package的内容，否则容易成为脚本小子

内容总结：

1. 基础语法
2. 变量类型
3. 控制流
4. 字符串
5. 组合数据类型：列表，元组，字典，集合