

Python 3.x

博士研究生讨论班分享课-01



Li.YJ  lyj8512@126.com

2018.9.20

在开始分享 python 内容之前,我们先来看看
Python 能做什么?

python 能做什么

科学计算

- LP, QP, QCP, MIP
(线性规划、二次方程规划、二次方程约束规划、混合整型规划)
- 微分方程, 矩阵运算等

示例介绍: [PyShare 01/example/MIP.ipynb](#)

python 能做什么

爬虫

- 实证研究数据来源方式之一
- 获取自己感兴趣的数据

示例介绍: *PyShare 01/example/Tmall_rateContent.ipynb*

python 能做什么

数据清洗

- 格式化处理, 比较转换, 单位转换, 正则处理文本

示例介绍: [PyShare 01/example/datatreating.ipynb](#)

python 能做什么

数据分析

- 统计数据, 分析数据

示例介绍: [PyShare 01/example/datanalysis.ipynb](#)

PyShare 课程适用对象

非计算机专业 0 基础编程经验学员
有兴趣愿意敲码捣鼓折腾的学员!

PyShare 课程定位

python 仅仅是研究工具, 快速了解 python 知识框架即可!
思考研究内容本身比学习 python 更加重要!

课程定位

- **内容:**python3.x & 爬虫相关: 快速入门
- **方式:**项目实现导向, 共同讨论交流学习
- **目的:**能写有助自身研究内容的代码即可!

PyShare 分享内容

Day 01 基础入门篇

Day 02 面向对象编程篇

Day 03 爬虫项目实施 01

Day 04 爬虫项目实施 02

Day 05 爬虫项目实施 03

Day 06 等补篇

课程分享内容

- 基本内容:

- 语法结构 → 数据类型 → 条件语句 → 流程控制 → 函数方法 → 包, 模块 → 文件操作

- 面向对象:

- 面向过程 & 面向对象区别 → 类和对象 → 封装 → 继承 → 类的特性 → 装饰器

- 爬虫项目 1:

- requests&pyquery → 项目分析 → 代码实现

- 爬虫项目 2:

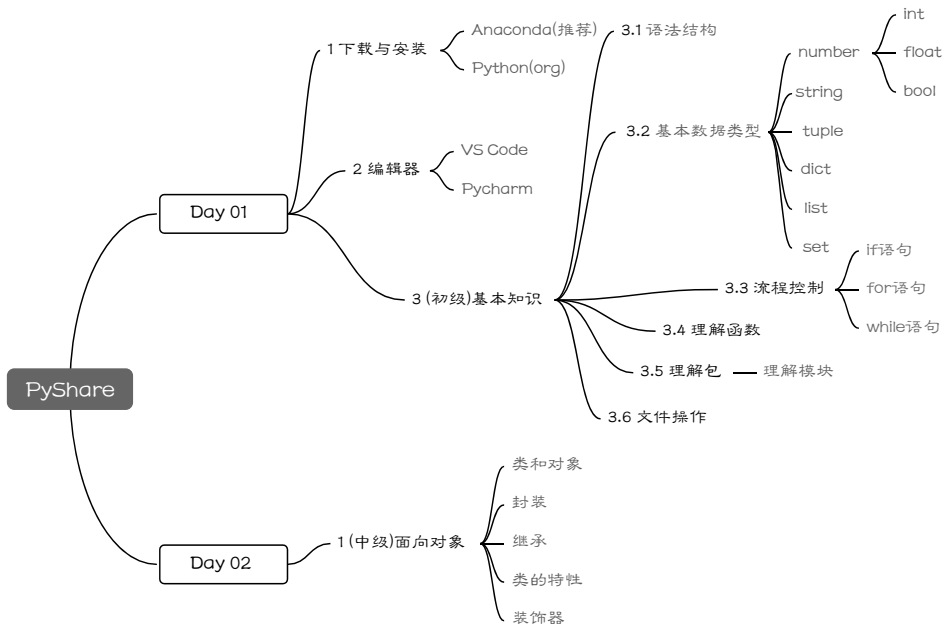
- JSON 文件 & re 正则表达式 → mongo 数据库 → 项目分析 → 代码实现

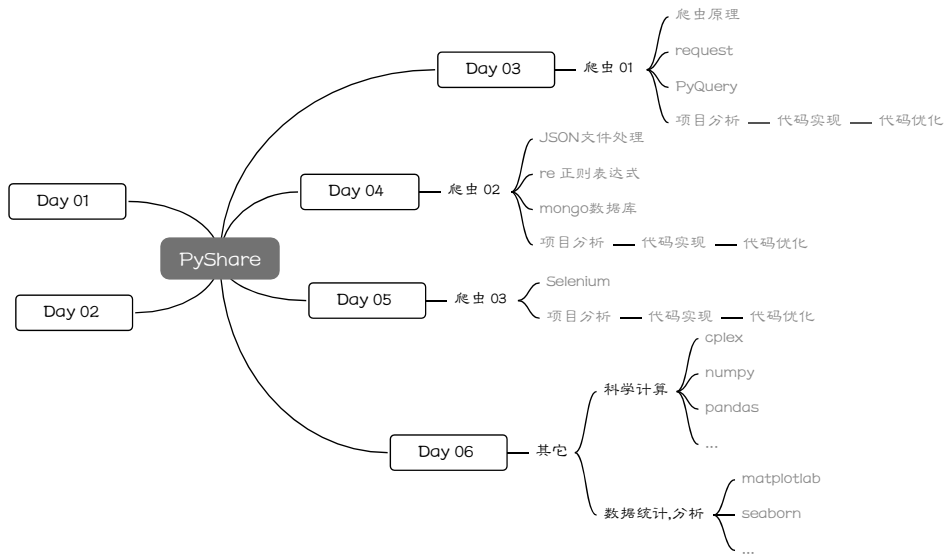
- 爬虫项目 3:

- selenimu → 项目分析 → 代码实现 → 代码优化

- 其它: 待定

- 科学计算数据后处理, 数据统计分析, 绘图等





Python 学习有门槛吗? 要怎么学?

学习要求

有编程语言基础:

- 简单过一篇语法, 就能上手解决问题

无编程语言基础:

- 那也没关系, 跟着分享课程节奏敲码练习即可

Python 那么多库, 那么多函数, 今天记下来明天又忘记, 怎么办?

- 练习很重要
- 案例积累很重要
- 一个好的开发环境, 更加重要

快速认识 python

认识 python

- 近些年相当流行的语言, 但最早可以追溯到上世纪 90 年代...
- 根据最新 TIOBE 语言热度排行榜,python 已经高居第 3 位
- pthon 拥有大量的第三方库, 可以帮助大家解决各类研究问题

python 语法特点

- 简洁, 优雅, 易阅读
- 易于学习
- 丰富的标准库与第三方库
- 跨平台运行
- python 具有动态脚本的特性, 又有面向对象的特性

python 是什么语言类型

- 编译型语言

编译型语言在执行代码之前, 会有一个预告编译的过程通常来说它会把我们写的源代码编译成相应环境的机器码, 由于机器码是非常接近于底层的, 因此它的执行效率是非常高的

python 是什么语言类型

- 解释型语言 (高级语言)

解释型语言是不会有预先编译的过程的, 也就是说每一次去执行这个程序的时候, 都会去解释这样的一个源代码, 所以相应的执行效率是要比编译型语言远要低的.

python 缺点

- 慢

我们说一种语言快或者是慢, 肯定是有会有一个比较对象的, 在这里我们说 python 慢是相对哪种语言来说的呢, 主要呢是相较于 C C++ JAVA 语言来说

python 真的很慢吗?

疑问?

- 既然编译型语言执行效率更高, 那么为什么还会有解释型语言的存在

这是开发效率 VS 执行效率的问题

python 下载, 安装

二选一即可, 推荐安装 python 发行版 anaconda

- python 官网, <https://www.python.org>
- anaconda 官网, <https://www.anaconda.com/download>

python 编辑器

- [anaconda](#), 安装完成后 [Jupyter](#), 也就可以使用了
交互编程, 一般用于代码演示, 教学
- [VS Code](#), <https://code.visualstudio.com>
代码补全, 调试, 友好界面... 等等
- [pycharm](#), <http://www.jetbrains.com/pycharm>

PyShare 基本语法

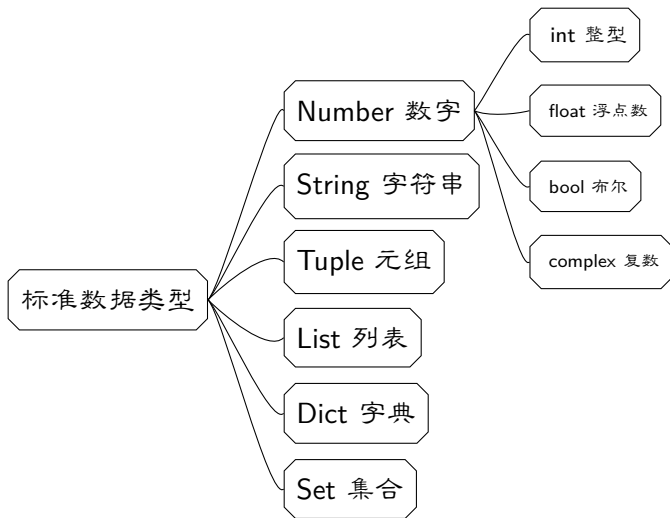
基本语法

- 字符命名
- 保留字关键字
- 注释
- 行与缩进
- 多行语句
- 变量
- 表达式
- 代码块

示例介绍: *PyShare 01/000.ipynb*

PyShare 数据类型

标准数据类型



标准数据类型

数字 number

- int 整型
- float 浮点型
- bool 布尔类型
- complex 复数

示例介绍: [PyShare 01/001.ipynb](#)

字符串 string

- 认识字符串
- 高级字符串
- 字符串操作
- 字符串内置函数

示例介绍: [PyShare 01/002.ipynb](#)

元组 tuple

- 认识元组
- 元组操作
- 元组内置函数

示例介绍: *PyShare 01/003.ipynb*

列表 list

- 认识列表
- 列表操作
- 列表内置函数

示例介绍: *PyShare 01/004.ipynb*

字典 dict

- 认识字典
- 字典操作
- 字典内置函数

示例介绍: *PyShare 01/005.ipynb*

集合 set

- 认识集合
- 集合操作
- 集合内置函数

示例介绍: *PyShare 01/006.ipynb*

PyShare 运算符

运算符

- 算术运算符
- 赋值运算符
- 比较运算符
- 逻辑运算符
- 成员运算符
- 身份运算符

示例介绍: [PyShare 01/007.ipynb](#)

PyShare 流程控制

流程控制

- if 条件语句
- for 循环语句
- while 循环语句
- 枚举函数, 迭代器, 生成器

示例介绍: *PyShare 01/008.ipynb*

PyShare 包模块函数方法

包模块函数方法

- 理解包与模块
- 导入模块
- 从包里导入模块
- 导入多个模块
- 从包导入全部模块
- 包的内置函数

示例介绍: *PyShare 01/009.ipynb*

包模块函数方法

- 理解变量
- 理解函数
- 匿名函数 (lambda)
- 理解作用域

示例介绍: [PyShare 01/010.ipynb](#)

PyShare 文件读写

文件读写

- 原始模式
- 简洁模式
- 文件目录操作

示例介绍: *PyShare 01/011.ipynb*

PyShare Day 01 练习

估算 π 的数值

```
import random  
random.uniform(-1,1)
```

PyShare Day 01 总结

本节结束, 你应该知道以下内容:

知道 python 基础知识

开始能看懂 py 脚本文件

能创建 py 文件, 运行并调试 py 文件, 知道怎么 Jupyter 做自己代码笔记

能使用函数, 减少代码重复量

知道文件操作方式

Thank you!