

# Python天文应用

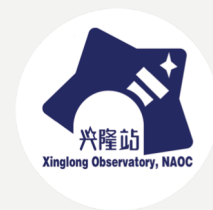
## ①语言基础

郑捷（小林）@ 中国科学院国家天文台兴隆观测基地

2018-12-17 NAOC

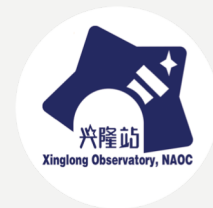
# 课件、资料

- [hebl.china-vo.org/course/astrodata2018/](http://hebl.china-vo.org/course/astrodata2018/)



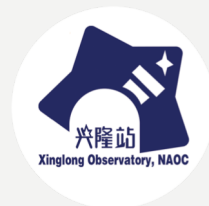
# Python天文应用

- 语言基础
- 天文工具包
- 天文数据处理与应用



# Python语言基础

- 简介
- 基本语法
- 流程控制
- 面向对象
- 模块

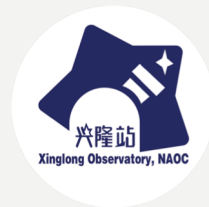




# 语言基础： 语简介

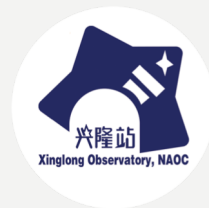
# 语言基础-简介

- Guido van Rossum, 1991
- 发展迅速, 语言优秀
- 应用领域
  - 数据分析, Web, 科学计算, ERP, 游戏, 数据库, 爬虫, 智能.....
  - 天文数据处理
- 2.x & 3.x



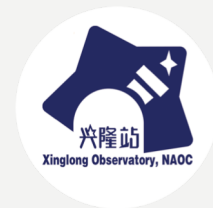
# 语言基础-简介

- 定位：脚本语言、解释型语言
- 语言本身较简单，依赖大量扩展包
- 开源、免费
- 全平台，适用于所有系统
- 在大部分Linux发行版、macOS预装
- 开发工具多样



# 语言基础-安装

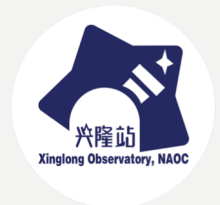
- 语言安装
  - Windows: 安装包
  - Linux: 预装; apt-get、yum等安装
  - macOS: 预装; 下载安装包
  - anaconda
- 扩展包安装
  - 单独安装包
  - 操作系统自带安装
  - pip、conda install





# 语言基础-运行环境

- python xxx.py
- iPython / python shell
- jupyter notebook / jupyter-lab
- IDE (Integrated Development Environment)
  - IDLE
  - PyCharm
  - vi/emacs/.....

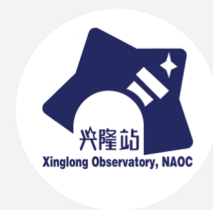


# 语言基础-对比

	Python	C/C++	Java	Php	.net	Matlab/idl
执行形式	解释	编译	编译	解释	编译	解释
速度	慢	快	中	中	中	中
易用性	好	不好	一般	不好	一般	好
跨平台	好	差	好	好	好	好
可读性	好	差	中	差	中	中
社区	好	好	好	好	好	不好
学习曲线	上手快	平台	快	快	快	平台
社会需求	高	低	高	中	高	低

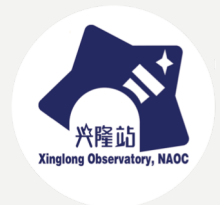
# 语言基础-缺陷


- 2到3的升级变化较大
- 运行速度较慢
- 同一功能被重复实现
- 调试功能有限
  
- 多语言综合使用



# 语言基础-建议环境

- 版本: 3.5-3.7, 除特殊情况外不建议2.x
- 安装: 自带, 或者anaconda
- 包管理: pip或conda
- 生产环境: 终端直接运行
- 测试环境: notebook或jupyter-lab
- 开发环境: pycharm
- 版本管理: git

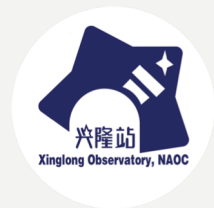




# 语言基础： 语法和计算

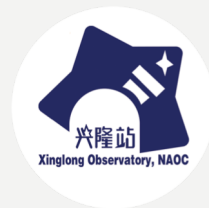
# 语言基础-语法

- 基本语法元素较少
- 面向过程、面向对象
- 支持Unicode字符作为标识符( $\geq 3.0$ )
- 通过缩进表达语句层次
- 单行多语句用分号;隔开
- 行末\  
表示续行, 或者自然续行
- 使用#表示注释
- 随处可见的doc-string



# 语言基础-变量和输出

- print函数
  - 注意2、3区别
- 变量类型
  - 整数（32位）
  - 浮点数（双精度，64位）
  - 字符串（Unicode）
  - 布尔（True/False）
- 组合类型
  - 列表 List
  - 元组 Tuple
  - 字典 Dict
  - 集合 Set
  - 数组 Array \*



# 语言基础-组合类型

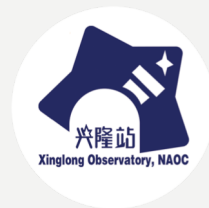
	数组 Array	列表 List	元组 Tuple	字典 Dict
元素数量	固定	可变	固定	可变
元素类型	相同	可不同	可不同	可不同
维数	可多维	一维	一维	一维
切片 Slice	可以	可以	可以	不可以
动态增减	不可以	可以	不可以	可以
元素赋值	可以	可以	不可以	可以
重复元素	可以	可以	可以	下标不可以，内容可以
访问	序号	序号	序号	关键字





# 语言基础-赋值

- 普通变量
  - 赋值即创建、初始化
  - 类型可以变
- 组合类型元素
  - 先创建组合，然后才有内容
  - 创建的同时可以初始化
- 元组赋值
  - 同时给多个变量赋值



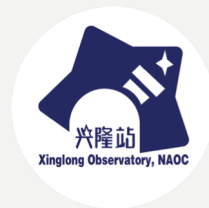
# 语言基础-布尔和判断

- bool: 表示真假
- 比较运算:  $>$   $<$   $==$   $<=$   $>=$   $!=$
- in运算
- 逻辑运算 and or not
  - not 不一定放前面, 也可以放中间
- is: 特殊运算, 对比是否是同一个对象
- ==和is的区别



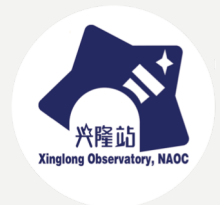
# 语言基础-特殊对象

- None: 表示空
- 对是否为空, 一般用is None判断



# 语言基础-格式化输出

- print 函数
- 格式字符串
  - 新办法 “xxx”.format(...)
  - 老办法 “xxx” % (...)
- 注意新老方法的区别
- 资源: <https://pyformat.info/>

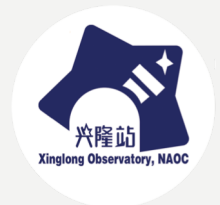




# 语言基础： 流程控制

# 语言基础-流程控制

- 缩进表达层次，而非符号
  - 控制语句结尾是冒号:
- 选择 if
  - if – elif – else 其他语言没有 elif
  - 没有 switch/case 类语句
  - 条件表达式
    - a if cond else b
- 逻辑 (布尔) 表达式



# 语言基础-流程控制

- for循环

- for x in xxx:

- x是单变量, xxx是组合

- 注意: 不仅仅是等差

- 列表生成器

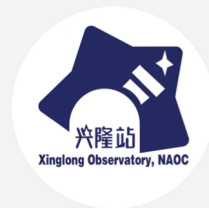
- [f(x) for x in xxx]

- 根据旧列表生成新列表



# 语言基础-流程控制

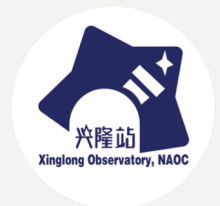
- while
  - 和其他语言没啥区别
- 循环控制
  - break
  - continue
  - 和其他语言没啥区别





# 语言基础-Pythonic

- Pythonic, 符合python要求的写法
- 例如:
  - 使用生成器, 而不是循环
  - 使用条件表达式而不是if语句
- 并非强制
- 但是可能提高效率



# 学习方式

- 学会基本语法
- 理解“嵌套”
- 知道有什么功能
- 会查手册
- 多使用
- 养成良好编程习惯

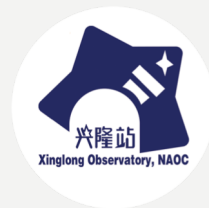




# 语言基础： 函数

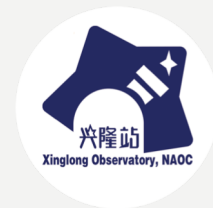
# 语言基础-函数


- 不区分函数和过程
- 使用def进行声明
- 函数可以带参数
- 参数传递分为按序和按名
- 形式参数可以带有默认值
- 用列表或者字典接受不定参数
- 用列表或者字典传递不定参数



# 语言基础-函数

- $\lambda$  函数
  - 一行之内定义
- 函数也是变量，也可以赋值和传递
- 函数是模块化程序设计的基础

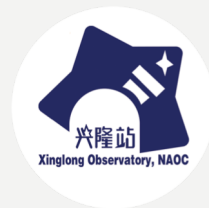




# 语言基础： 面向对象

# 语言基础-面向对象

- 面向对象就是以类和实例的方式编程
- 重点
  - 定义新数据类型
  - 封装
  - 继承
  - 重写父类函数



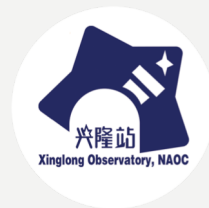


# 语言基础： 模块



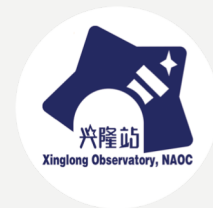
# 语言基础-模块

- 最简单的模块：文件
  - 每个文件就是一个模块
  - 全局信息（函数、变量）外部可用
  - 需要其他文件时，用import
- 被import的文件其实是被执行
- `if __name__ == "__main__":`
  - 作用？



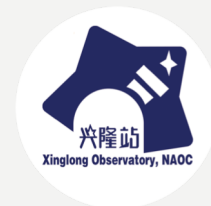
# 语言基础-模块

- import的四种写法
  - import xx
  - from xx import yy
  - import xx as zz
  - from xx import yy as zz



# 语言基础-模块

- 复杂的模块： 目录
  - 每个目录是一个模块
  - 目录中必须有 `__init__.py` 文件
  - 可以混合使用



谢谢