

# Python天文应用

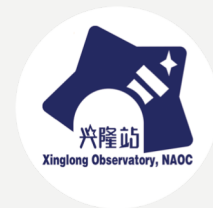
## ③天文数据处理

郑捷（小林）@ 中国科学院国家天文台兴隆观测基地

2018-12-18 NAOC

# Python天文应用

- 语言基础
- 天文工具包
- 天文数据处理和应用





# 天文数据处理 概述

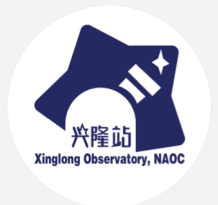
# 天文数据处理-概述

- astropy包
- 汇集了此前的各种零散包
  - pyfits
  - sep
  - .....
- 还有大量的相关包
  - <http://www.astropy.org/affiliated/index.html>
- pyraf



# astropy-概述

- astropy.io.fits
- astropy.io.ascii
- astropy.io.votable
- astropy.wcs
- astropy.time
- astropy.coordinates
- ...

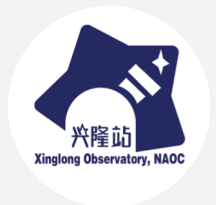




# 天文数据处理 fits文件

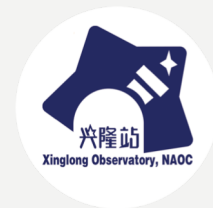
# Fits文件概述

- Flexible Image Transport System
- 在DOS的8.3文件名时期一般用扩展名fit
- 1982 IAU
- FITS文件广泛用于天文学描述数据
  - 图像
  - 表格
  - 光谱
  - ...



# FITS 文件逻辑结构

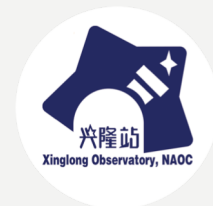
- 分为简单图像和多扩展数据
- 简单图像： 只有一个数据段
  - 头部 header
  - 数据 data
- 多扩展数据（多子图、多层）
  - 一个主头部 primary header
  - 若干个数据段，每个包括头部和数据
  - 主头部没有数据





# FITS头部

- 对数据的描述
- 表现为一组文本记录
- 每个记录为
  - 字段名=值/注释
  - 注释
  - 历史
  - 其他说明
- 值类型包括： 整数、浮点数、字符串



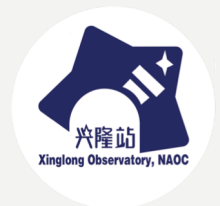
# FITS头部：字段分类

- 数据基本描述
  - SIMPLE、BITPIX
  - NAXIS、NAXIS1、NAXIS2
- 观测信息
  - 观测目标、时间
  - 望远镜信息，终端信息
- 数据处理过程信息
  - 处理软件、时间、方法等
- 图像定标信息
  - 流量定标
  - 天体测量定标



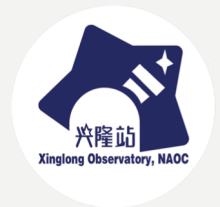
# FITS格式： 图像

- 测光图像、抽谱前光谱等
- 一般表现为二维位图，即每像素灰度
  - BITPIX 每像素数据类型
    - 8 16 32 整数
    - -16 -32 -64 浮点数 (实测-16没法用)
  - NAXIS=2, NAXIS1、NAXIS2为x/y大小



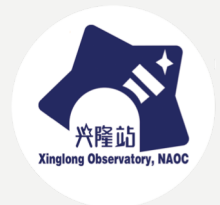
# FITS格式：二进制表格

- 二进制表格就是二维表
- 每行一记录，每列一字段
- 表格必须作为扩展存储



# FITS格式-数据表达

- BITPIX
- 一般整数是带符号整数
- 对无符号整数, 用BZERO调整
  - BITPIX = 16
  - BZERO = 32768
- 所以一个ext, 只能有一种无符号整数



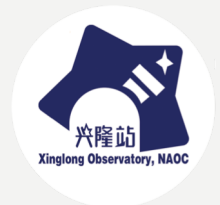


# 天文数据处理

## FITS 读写

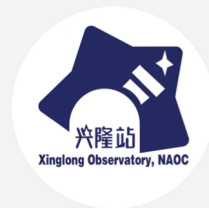
# FITS读写

- `import astropy.io.fits as fits`
- `from astropy.io import fits`
- `hdulist = fits.open('xxxx.fits')`
  - 读入后为HeaderUnit对象组成的列表
  - 每个ext读入后是一个hdu
  - 哪怕只有一个简单图像，也是一个列表
  - 采用了延后读取策略，不用不会真的读



# FITS读写-访问头部

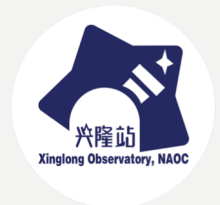
- `hdulst[0].header` 头部
  - 每个记录被叫做一个卡片 `card`
  - `header` 是一个卡片数组 (字典)
  - `hdulst[0].header['xxxx']`
  - 可读写
- 用 `append`, `update` 等方法进行改写





# FITS读写-访问数据

- `hdulst[0].data`
  - 数组形式
  - 具体形式和`ext`类型有关
- 二维表
  - `hdulst[0].data[y, x]`
  - **注意: python的下标和fits是相反的!!**
- 表格
  - 按照`np`表格方式访问



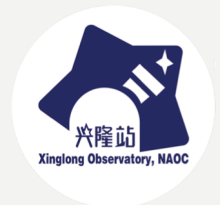
# FITS读写-快捷读取

- astropy提供了简化读写方式
  - fits.getheader
  - fits.getdata
  - fits.getval
- 写
  - fits.setval
  - fits.delval
- 不需要打开整个文件



# FITS读写-保存

- 如要修改，在open时要加update=True
- 直接创建新fits文件并保存
  - `hl = fits.HDUList()`
  - `hdu = fits.PrimaryHDU()`
  - `hdu.header = fits.Header()`
  - `hdu.data = ....`
  - `hl.append(hdu)`
  - `hl.writeto('xxxxx.fits')`

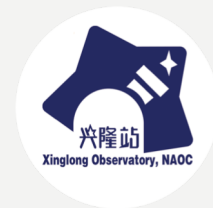




# 天文数据处理 案例

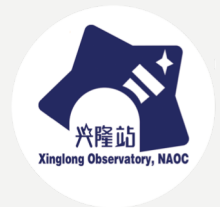
# 案例-概述

- 假设有一个fits图像需要进行找星和测光
- 测光后，将图像绘制出来，并且标上找到的星
- 绘制仪器星等-误差图
- 绘制天球坐标分布图
- 星表输出成文本和fits格式



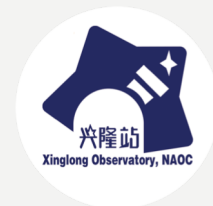
# 案例-实现

- 调用SExtractor
- 读图像和星表
- 绘制
- 保存



# 案例-调用

- 从notebook到.py
- python终端直接调用
- shell调用
- 跨目录调用



谢谢