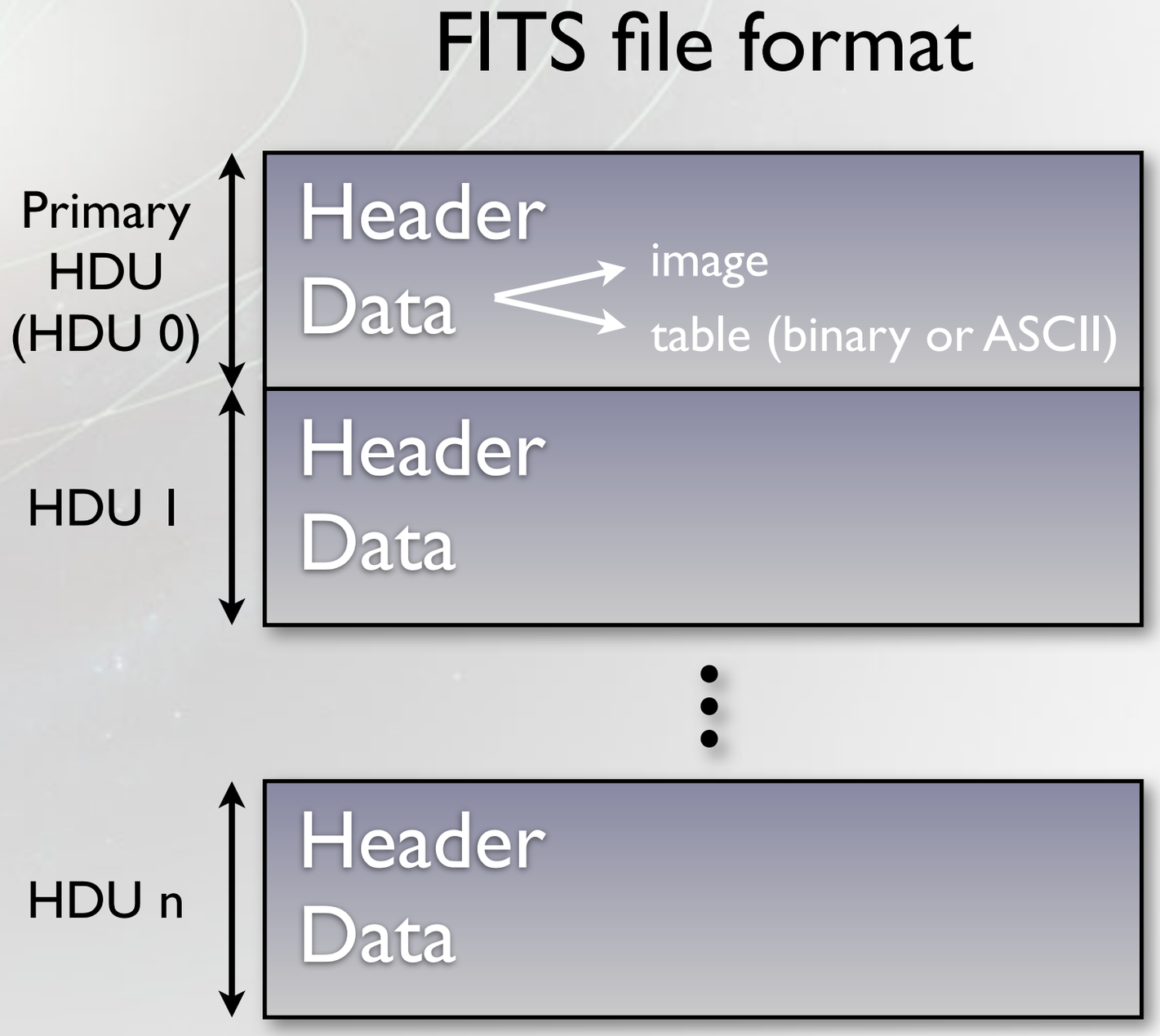
Reading FITS Files in Python

23 June 2010

Format of a FITS File

- FITS is a binary file format
- Text header + data
- HDU = Header Data Unit
- First is called “primary HDU” and is required.
- Supports n-dimensional data
- Data portion optional



FITS Headers

- Header consists of keyword/value pairs

```
KEYWORD = Value / comment
```

```
XTENSION= 'BINTABLE'           /Binary table written by MWRFITS v1.8
BITPIX  =                    8 /Required value
NAXIS   =                    2 /Required value
NAXIS1  =                 3022 /Number of bytes per row
NAXIS2  =                  721 /Number of rows
PCOUNT  =                    0 /Normally 0 (no varying arrays)
GCOUNT  =                    1 /Required value
TFIELDS =                  139 /Number of columns in table
```

- Keyword is uppercase, up to 8 characters, containing only [A-Z0-9_-]
- NAXIS keywords indicate data dimensions
- NAXIS1 - dimension of 1st axis, NAXIS2 - 2nd axis, etc.
- Last keyword always ‘END’

Reading FITS Files in Python

- Package: pyfits (http://www.stsci.edu/resources/software_hardware/pyfits)

```
import pyfits

filename = "data.fits"
dataHDUlist = pyfits.open(filename)

numHDUs = len(dataHDUlist)
print numHDUs
print dataHDUlist.info()

# print primary HDU header
for key, value in dataHDUlist[0].header.items():
    print key, value

dataHDUlist.close()
```

returns Python list of HDUs

structure of file (no. of HDUs, types, dimensions, etc.)

header of PrimaryHDU

close FITS file
(headers still available)

Exercise

- Write a Python script that takes one or more FITS files as arguments and prints out how many HDUs the file contains and the type of each one.

HDU Types

- Can store data as a binary table or as an image
  - Tables are organized by labeled columns and filled by adding rows of data
  - Images are simple arrays of pixel values

ImageHDU

- Image data is read in as a numpy array
  - Shape is specified by NAXIS keywords
  - Datatype is specified by BITPIX keyword

```
BITPIX    Numpy Data Type
8         numpy.uint8 (note it is UNsigned integer)
16        numpy.int16
32        numpy.int32
-32       numpy.float32
-64       numpy.float64
```

ImageHDU

- Read in image data:

```
hduList = pyfits.open('imageFile.fits')
# If you know hdu 1 is an image, then:
imageData = hduList[1].data


# get the pixel value at x=31, y=11
print imageData[10,30]

# print the 4th row
print imageData[3]

# extract a sub section
subImg = imageData[30:40,10:20]
```

TableHDU

- Tables can be read by column (field), or by row

```
hduList = pyfits.open('imageFile.fits')
# Assuming hdu 1 is a table
tbData = hduList[1].data

# print the first row in the table
print tbData[0]

# print the column 'OBJTYPE'
print tbdata.field('OBJTYPE')
```

TableHDU

```
import pyfits
import numpy

def tablehdu2dict(hdu):
    """Function that accepts a TableHDU object from Pyfits, and returns
    the table as an array of dictionaries. The keys in the dictionary
    represent the columns in the table, and each array index is a row
    of the table.
    """

    # Check to make sure the hdu isn't an Image
    if hdu._summary().split()[0].lower() == 'bintablehdu':
        retArray = numpy.array([])

        for row in hdu.data:
            tmp_dict = dict()
            for i,key in enumerate(hdu.data.names):
                tmp_dict[key] = row[i]
                retArray = numpy.append(retArray, tmp_dict)

        return retArray # an array of dictionaries for each row of the table

    else:
        print "HDU is not a Binary Table.\nSkipping..."
```