

# Introduction to *CASA* Scripting

The background of the slide features two large, white, parabolic radio telescope dishes. They are situated in a vast, arid desert landscape under a clear, light blue sky. The dishes are supported by complex metal structures and are angled towards the horizon. The overall scene is bathed in a soft, golden light, suggesting either dawn or dusk.

Josh Marvil

NRAO Data Reduction Workshop

April 9, 2013

# Outline of this Talk

---

The CASA Environment  
Writing and Running Scripts  
Basic Python in Scripts  
Writing CASA Tasks  
Examples

# The CASA Environment

---

Python 2.6 - Standard Library

iPython - Interactive Shell

Additional Python Modules

CASA Tasks and Tools

CASA Standalone Applications

— plotms, viewer, browser —

# The CASA Environment

---

- From the CASA command line
  - system shell access: `!du -hs *ms`
  - help documentation: `clean?`
  - TAB completion: `im<TAB>`
  - object inspection: `im.<TAB>`
  - auto-parentheses: `sin x`
  - searchable history: `im<UPARROW>`

# The CASA Environment

---

- A Python-aware text editor
  - syntax highlighting
  - auto-indentation
  - block quote
  - block indent

# A CASA Script

---

- Example script: G55.7 tutorial [[casaguides.nrao.edu](http://casaguides.nrao.edu)]

```
setjy(vis='G55.7+3.4_10s.ms', field='0542*',  
      spw='2~3,5~6', modimage='3C147_L.im')  
  
gaincal(vis='G55.6+3.4_10s.ms', caltable='G55.6+3.4_10s.G0',  
        spw='2~3,5~6', solint='int', calmode='p', field='0542*')
```

- Write your script with a text editor
- Name your script almost anything you want, e.g., myScript.py

# A CASA Script

---

- Run your script in CASA:

```
CASA>> execfile('myScript.py')
```

- Run your script from the terminal:

```
bash$ casapy -c myScript.py
```

- Run your script remotely:

```
bash$ nohup casapy -c myScript.py &
```

# A CASA Script

---

- Another example: listobs.last

```
taskname      = "listobs"  
vis           = "G55_split1.ms"  
verbose      = True  
listfile     = "G55_split1.ms.listobs.txt"  
  
#listobs(vis="G55_split1.ms", verbose=True,  
         listfile="G55_split1.ms.listobs.txt")
```



# A CASA Script

---

- Function call vs. command line style:

```
setjy(vis='G55.7+3.4_10s.ms', field='0542*',  
      spw='2~3,5~6', modimage='3C147_L.im')
```

```
default( 'setjy' )  
vis= 'G55.7+3.4_10s.ms'  
field = '0542*'  
spw = '2~3,5~6'  
modimage = '3C147_L.im'  
setjy()
```

# Basic Python in Scripts

---

- Common numerical types

```
x=1
```

```
x=1L
```

```
x=1.0
```

```
x=1e0
```

```
x = 1 + 0j
```

- Example strings

```
x= ''
```

```
x= '1'
```

```
x= '12'
```

- Example lists

```
x= []
```

```
x= [1]
```

```
x= [ 1.0, 2, '3' ]
```

# Basic Python in Scripts

---

- Index selection

```
x = 'abc'
```

```
x = [ 'a', 'b', 'c' ]
```

```
a = x[ 0 ]
```

```
b = x[ 1:2 ]
```

```
c = x[ -1 ]
```

```
ab = x[ :2 ]
```

```
bc = [ 1: ]
```

```
cba = [ ::-1 ]
```

# Basic Python in Scripts

---

- Adding strings and lists → concatenation

```
ab = 'a' + 'b'  
ab = 'ab'
```

```
x = [ 1, 2 ] + [ 3, 4 ]  
x = [ 1, 2, 3, 4 ]
```

- Multiplying strings and lists → replication

```
ab2 = 2 * 'ab'  
ab2 = 'abab'
```

```
x2 = 2 * [ 1, 2 ]  
x2 = [ 1, 2, 1, 2 ]
```

# Basic Python in Scripts

---

- Example script: G55.7 tutorial

```
setjy(vis='G55.7+3.4_10s.ms', field='0542*',  
      spw='2~3,5~6', modimage='3C147_L.im')
```

```
gaincal(vis='G55.6+3.4_10s.ms', caltable='G55.6+3.4_10s.G0',  
       spw='2~3,5~6', solint='int', calmode='p', field='0542*')
```

# Basic Python in Scripts

---

- Generalizing the script with Python strings:

```
vis = 'G55.7+3.4_10s.ms'  
field = '0542*'  
spw = '2~3,5~6'  
modimage = '3C147_L.im'
```

```
setjy(vis=vis, field=field, spw=spw, modimage=modimage)
```

```
gaincal(vis=vis, caltable=vis[:-2]+'G0', field=field,  
        spw=spw, solint='int', calmode='p')
```

# Basic Python in Scripts

---

- Conditional statements and logical operators

```
doThis = True
if doThis:
    print 'this will happen'
```

```
x = 3.0

if (x == '3'): print 'this will not happen'
elif (x > 5.0): print 'this will not happen either'
else: print 'this will happen'
```

# Basic Python in Scripts

---

- Objects with length can be iterated

```
x = 'abc'  
  
for item in x: print item
```

- *E.g.*, clean multiple fields

```
allFields = [ '0', '3', '7' ]  
  
for field in allFields:  
    clean( field = field ...
```



# Basic Python in Scripts

---

- The Python dictionary

```
x = { 'firstKey' : 3.0, 'secondKey' : 'a' }  
x[ 'thirdKey' ] = [ 1, 2 ]
```

```
a = x[ 'secondKey' ]  
x_keys = x.keys()
```

# Basic Python in Scripts

---

- Using dictionaries in CASA

```
raster = { 'file' : 'DSS_poss1_red.image',  
          'colormap' : 'Greyscale 1' }
```

```
contour = { 'file' : 'EVLA_Cband.image',  
           'levels' : [ 1, 2, 3, 5 ],  
           'unit' : 0.45,  
           'base' : 0 }
```

```
imview(raster=raster, contour=contour, out='filename.ps')
```

# Writing CASA Tasks

---

A Python Function  
The XML File  
buildmytasks

# Writing CASA Tasks

---

- Start with a CASA script

```
## script plotweather.py

vis = 'msName.ms'
seasonal_model = 0.5
doPlot = True

tb.open( vis + '/WEATHER' )
myTimes = tb.getcol( 'TIME' )
if doPlot:
    pl.plot( myTimes, ...
```

# Writing CASA Tasks

---

- Turn your script into a Python function

```
from casa import table as tb
import pylab as pl
...

def plotweather(vis='', seasonal_weight=0.5, doPlot=True):

    tb.open( vis + '/WEATHER' )
    myTimes = tb.getcol( 'TIME' )
    if doPlot:
        pl.plot( myTimes, ...
```

# Writing CASA Tasks

---

- Write the xml file (task interface)

```
[XML header -- copy from cookbook]
```

```
<task type="function" name="plotweather">
```

```
<shortdescription> short description </shortdescription>
```

```
<input> [details of each input parameter] </input>
```

```
<example>example text</example>
```

```
</task>
```

```
</casaxml>
```

# Writing CASA Tasks

---

- Inside the input tag

```
<param type="string" name="vis" kind="ms" mustexist="true">
<description>MS name</description>
<value></value>
</param>
...
[second input parameter]
...
[third input parameter]
```

# Writing CASA Tasks

---

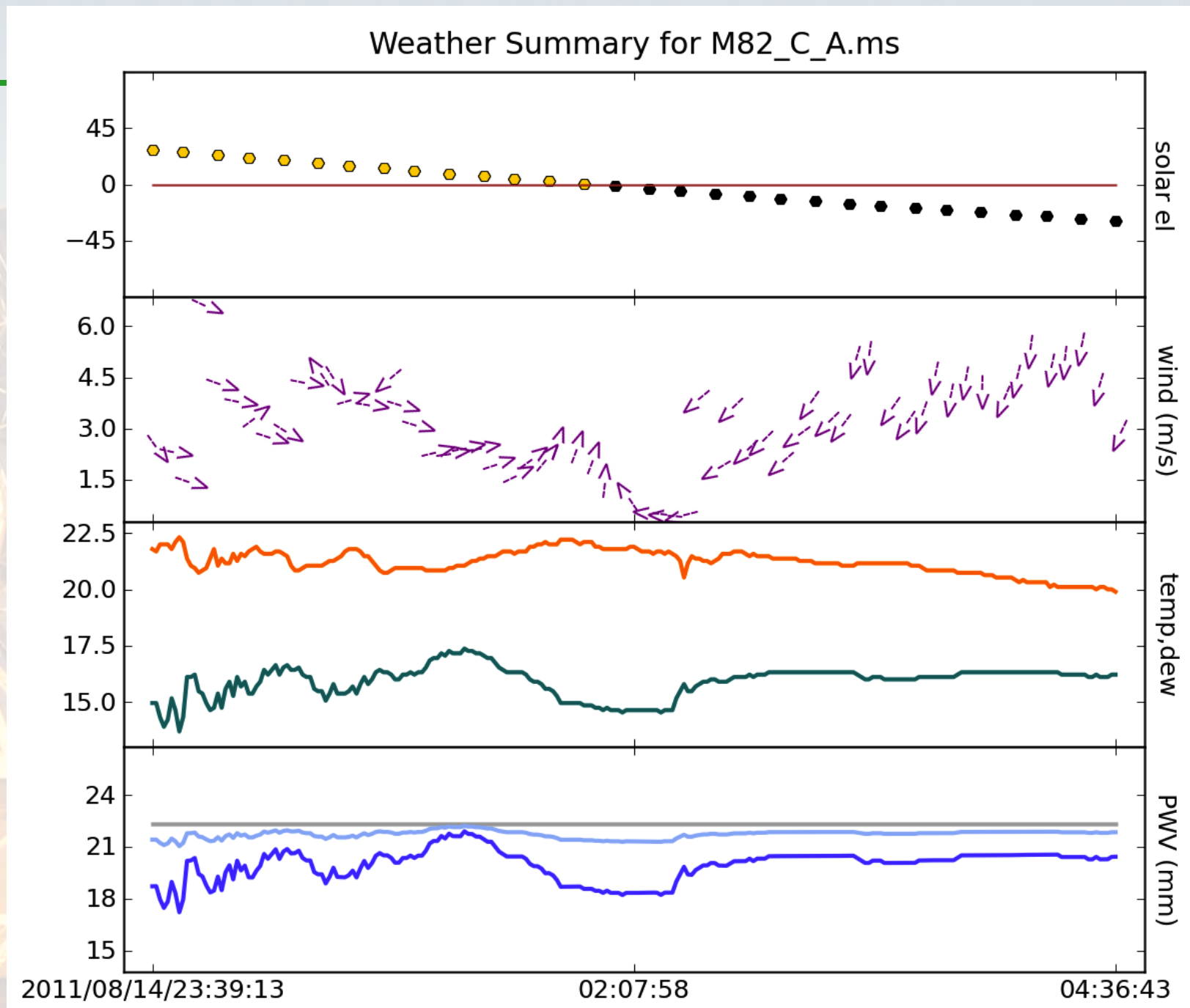
- Name your function file `task_taskname.py`
- Name your xml file `taskname.xml`
- Building and importing your task

```
CASA>> !buildmytasks taskname  
CASA>> execfile( 'mytasks.py' )  
CASA>> inp taskname
```



# plotweather

- the tb tool
- the qa tool
- the me tool
- the pl tool



# Examples

---

- The quanta tool (qa)

```
x = qa.quantity( 12.4, 'deg' )  
x_rad = qa.convert( x, 'rad' )  
x_dms = qa.angle( x )  
x_hms = qa.time( x )
```

```
mp = qa.constants( 'mp' )  
c2 = qa.pow( qa.constants('c'), 2 )  
E = qa.mul( mp, c2 )  
qa.convert( E, 'MeV' )
```

# Examples

---

- The measures tool (me)

```
me.doframe( me.observatory( 'VLA' ) )  
me.doframe( me.epoch( 'utc', 'today' ) )  
me.measure( me.direction( 'SUN' ), 'AZELGEO' )
```

```
M81 = me.direction( 'B1950', '9:51:27', '69.18.08' )  
M82 = me.direction( 'J2000', '9:55:53', '69.40.46' )  
me.separation( M81, M82 )
```

# Examples

---

- Opening tables with browsetable

```
CASA>> browsetable( 'myTable' )
```

```
bash$ casabrowser myTable
```

- Opening tables with the tb tool:

```
tb.open( 'myTable' )  
myData = tb.getcol( 'DATA' )  
tb.close()
```

# Examples

---

- Using the table query language (TaQL)

```
tb.open( 'myTable' )  
stb = tb.query( 'ANTENNA1 == 3 && FIELD_ID == 1' )  
myGains, myTimes = stb.getcol( 'GAIN' ), stb.getcol( 'TIME' )  
tb.close()
```

# Examples

---

- Iterate over large data columns

```
ms.open( 'MSname' )
ms.iterinit( interval=1000 )
ms.iterorigin()

moretodo = True
while moretodo:
    myData = ms.getdata( items=['data'] )
    ...
    moretodo = ms.iternext()

ms.close()
```

# Examples

---

- Opening images with the ia tool:

```
ia.open( 'myImage' )  
mySummary = ia.summary()  
myData = ia.getregion( region = 'myRegion' )  
myStats = ia.statistics( region = 'myRegion' )  
myCsys = ia.coordsys()  
ia.regrid( outfile = 'newImage.image', csys = 'newCsys' )  
ia.close()
```

# Examples

---

- The pl tool (numpy)

```
x = pl.array( [1,2,3,4] )  
y = 2*x**3 - 4
```

```
y2 = y[ y > 2 ]  
x2 = x[ y > 2 ]
```

```
x = pl.linspace( 1, 10, 100 )  
x_log = pl.logspace( 0, 1, 100 )  
noise = pl.randn( 100 )
```



# Examples

---

- The pl tool (matplotlib)

```
pl.plot( x )  
pl.plot( x, y, 'k--' )  
pl.errorbar( x, y, xerr=xerr, yerr=yerr )  
  
pl.semilogy( x, y, 'bo', ms=4 )  
pl.hist( x, bins=pl.linspace(0,20,21) )  
pl.imshow( X, cmap = 'gray')
```

# Examples

---

- More with matplotlib

```
p1.plot( x1, y1, 'bo', label= 'firstLabel' )  
p1.plot( x2, y2, 'gd', label= 'secondLabel' )  
p1.legend()
```

```
p1.title( 'Sample Title' )  
p1.xlabel( 'Sample Label' )  
p1.text( 3, 4.5, 'Sample Text' )
```

# Examples

---

- There is also scipy

```
from scipy.optimize import curve_fit
from scipy.integrate import odeint
from scipy.special import gamma
from scipy.interpolate import interp1d
from scipy.stats import ks_2samp
```

# Examples

---

- Searching strings

```
x = 'abc'  
myIndex = x.find( 'b' )
```

- Searching lists

```
x = [ 'a', 'b', 'c' ]  
myIndex = x.index( 'b' )
```

# Examples

---

- Handling errors in your script:

```
x = [ 2, 3 ]
searchThis = 1
stopOnError = True

try:
    myIndex = x.index( searchThis )

except:
    print 'index not found: ', searchThis
    if stopOnError: raise
```

# Examples

---

- Open and parse a text file:

```
for line in open('myText.txt', 'r'):
    line1 = line.split(' ')
```

- Append a value to a text file:

```
x = 3.0
out1 = open('myText.txt', 'a')
out1.write( str(x) + '\n' )
out1.close()
```

# Examples

---

- Running commands from the system shell

```
os.system( 'xv myPlotFile.png &' )
```

```
os.system( 'pdflatex myTexFile.tex' )
```

```
os.system( 'mutt -s '+thisSubject+' -a '+thisAttachment+ \  
          ' '+thisAddress+' < '+thisBody )
```

# References

---

- Python Documentation

- <http://docs.python.org/release/2.6/>
- <http://docs.scipy.org>
- <http://matplotlib.org>

- CASA Documentation

- [http://casa.nrao.edu/casa\\_cookbook.pdf](http://casa.nrao.edu/casa_cookbook.pdf)
- <http://casa.nrao.edu/docs/casaref/CasaRef.html>
- <http://casaguides.nrao.edu>