

IP[y]: A system  
for

IPython

Interactive

Scientific Computing

AUTHORS: FERNANDO PEREZ  
BRIAN E GRANGER  
(IEEE 2007)

PRESENTED BY: RASHMISNATA ACHARYYA

# Key Idea and Background

▶ What is Ipython?

▶ Why Ipython?

▶ How, when and by whom?

IP[y]:

IPython



# Features of a Good Interactive Computing Environment

# Access to all session states

```
$ ipython
Python 2.4.3 (Apr 27 2006, 14:43:58)
Type "copyright", "credits" or "license" for more
information.

IPython 0.7.3 -- An enhanced Interactive Python.
?          -> Introduction to IPython features.
%magic     -> Information about IPython magic %
functions.
Help       -> Python help system.
object?   -> Details about object. ?object also
works, ?? prints more.
In [1]:2**45+1
Out[1]:35184372088833L
In [2]:import cmath
In [3]:cmath.exp(-1j*cmath.pi)
Out[3]:(-1-1.2246063538223773e-16j)
# The last result is always stored as '_'
In [4]:_ ** 2
Out[4]:(1+2.4492127076447545e-16j)
# And all results are stored as N, where _N is
their number:
In [5]:_3+_4
Out[5]:1.2246063538223773e-16j
```

-The session input-outputs are stored in a table

-All outputs are accessible by `_N`

-save it into a log file

# A control system

```
In [2]: logstart -o -t ipsession.log
Activating auto-logging. Current session state
plus future input saved.
Filename           : ipsession.log
Mode               : backup
Output logging     : True
Raw input log      : False
Timestamping       : True
State              : active
```

-offers set of control (magic commands) improving usability

-Inspired from traditional Unix commands

# Operating System Access

```
In [36]: ls
tt0.dat tt1.DAT tt2.dat tt3.DAT
# 'var = !cmd' captures a system command into a
Python variable:
In [37]: files = !ls
==
['tt0.dat', 'tt1.DAT', 'tt2.dat', 'tt3.DAT']
# Rename the files, using uniform case and 3-digit
numbers:
In [38]: for i, name in enumerate(files):
.....:     newname = 'time%03d.dat' % i
.....:     !mv $name $newname
.....:
In [39]: ls
time000.dat time001.dat time002.dat time003.dat
```

-uses familiar commands (cd, ls )  
for navigation into file system and  
direct execution by OS

-python execution using unix like  
commands

-This feature has led users to use  
Ipython as their default shell

# Dynamic Introspection and Help



```
In [1]: from universe import DeepThought
In [2]: DeepThought. # Hit the Tab key here
        DeepThought._doc_ DeepThought.answer
        DeepThought.question
        DeepThought._module_ DeepThought.name
In [2]: DeepThought??
Type:          classobj
String Form:   universe.DeepThought
Namespace:    Interactive
File:         /tmp/universe.py
Source:
class DeepThought:
    name = "Deep Thought"
    question = None
    def answer(self):
        """Return the Answer to The Ultimate
Question Of Life, the Universe and Everything"""
        return 42
```

-Interactive Help

-Explore various types of objects in language

-Helpful in developing code, exploring problem or use an unfamiliar library

# Access to program Execution

```
fperez@ionigs:/home/fperez/test - Shell - Konsole
In [1]: pdb
Automatic pdb calling has been turned ON

In [2]: run error
reps: 5
-----
exceptions.ValueError                                Traceback (most recent call last)

/home/fperez/test/error.py
  66
  67 if __name__ == '__main__':
--> 68     main()
      main = <function main at 0x404ccaac>
  69
  70

/home/fperez/test/error.py in main()
  58     array_num = zeros(size, 'd')
  59     for i in xrange(reps):
--> 60         RampNum(array_num, size, 0.0, 1.0)
      global RampNum = <function RampNum at 0x404cca74>
      array_num = array([ 0.,  0.,  0.,  0.,  0.,  0.])
      size = 6
  61     RNTIME = time.clock()-t0
  62     print 'RampNum time:', RNTIME

/home/fperez/test/error.py in RampNum(result=array('built-in method typecode of array
object at 0x404ad6d8': [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, ...]), size=6, start=0.0, end=1.0)
  40     tmp = zeros(size-1)
  41     step = (end-start)/(size-1-tmp)
--> 42     result[:] = arange(size)*step + start
      result = array([ 0.,  0.,  0.,  0.,  0.,  0.])
      global arange = <built-in function arange>
      size = 6
      step = array([ 0.2,  0.2,  0.2,  0.2,  0.2,  0.2])
      start = 0.0
  43
  44 def main():

ValueError: frames are not aligned
- /home/fperez/test/error.py(42)RampNum()
  41     step = (end-start)/(size-1-tmp)
--> 42     result[:] = arange(size)*step + start
  43

ipdb> print start
0.0
ipdb> print step
[ 0.2  0.2  0.2  0.2  0.2  0.2]
ipdb> up
- /home/fperez/test/error.py(60)main()
--> 60     RampNum(array_num, size, 0.0, 1.0)
  61     RNTIME = time.clock()-t0

ipdb> print array_num
[ 0.  0.  0.  0.  0.  0.]
ipdb>
```

```
fperez@ionigs:/home/fperez/test - Shell - Konsole
longs[~/test]: python
Python 2.4.3 (#2, Apr 27 2006, 14:43:58)
[GCC 4.0.3 (Ubuntu 4.0.3-1ubuntu5)] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> execfile('error.py')
reps: 5
Traceback (most recent call last):
  File "<stdin>", line 1, in ?
  File "error.py", line 68, in ?
    main()
  File "error.py", line 60, in main
    RampNum(array_num, size, 0.0, 1.0)
  File "error.py", line 42, in RampNum
    result[:] = arange(size)*step + start
ValueError: frames are not aligned
>>>
```

-Execution is done as if it was typed interactively

-Upon completion the program results update the session so that further exploration can be done

-ipdb is the default debugger which analyses from its in-memory state without the need of reloading libraries etc

-Need not reload libraries again and again.



**IPYTHON :**

**A BASE LAYER FOR INTERACTIVE  
ENVIRONMENT**

**easy  
modification and customization**

# Input Syntax Processing

```
In [1]: mass = 3 kg
In [2]: g = 9.8 m/s^2
In [3]: weight=mass*g
In [4]: weight
Out[4]: 29.4 m*kg/s^2
# We can see the actual Python code generated by
# IPython:
In [5]: %history # %history is an IPython "magic"
command
1: mass = PhysicalQuantityInteractive(3, 'kg')
2: g = PhysicalQuantityInteractive(9.8, 'm/s**2')
3: weight=mass*g
4: weight
```

-IPython ships with a physics profile

-installs a special input filter.

-This filter recognizes text sequences that appear to be quantities with units

-Was extensively used in SAGE (Software for Algebra and Geometric experimentation) for transforming Numerical quantity to exact integers, rationals, floats etc

# Error Handling

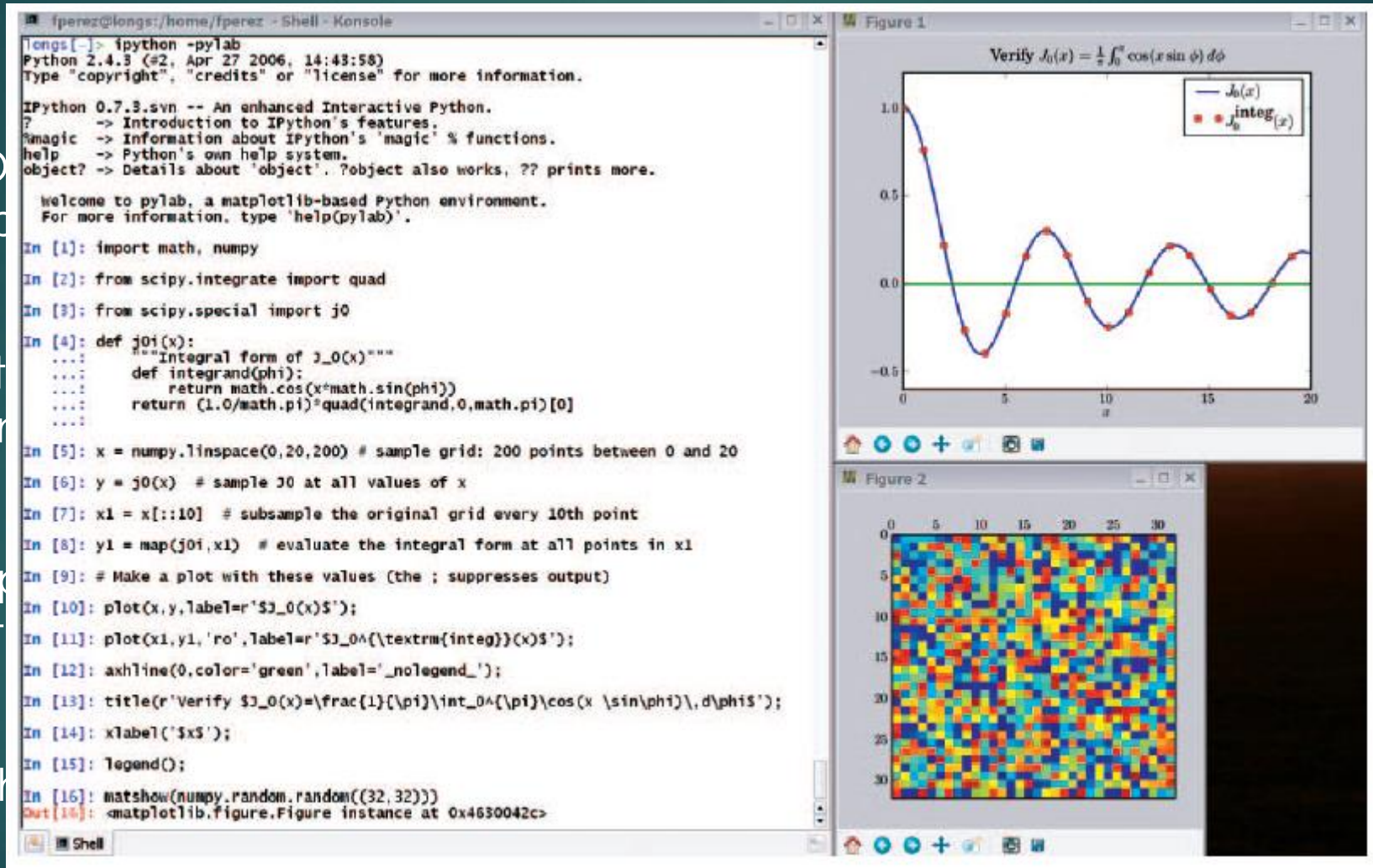
- ▶ Three exception handlers
- ▶ Users can register exception handlers to raise exceptions of registered type
- ▶ PyRaf interactive environment at Space Telescope Scientist Institute
- ▶ Handles internal errors separately from those meaningful to users

# Tab Completion

- ▶ PyMAD at neutron scattering Facility of Institute of Laue Langvin in Grenoble, France
- ▶ Ipython runs on a system connecting Neutron spectrometer over a network.
- ▶ Users interact with remote systems like local
- ▶ Tab Completion operates over a network to fetch info of remote objects for display in local console

# Graphical Interface ToolKit and Plotting

- ▶ In D app
- ▶ Ipython (using
- ▶ 3D plotting with ipython
- ▶ Both



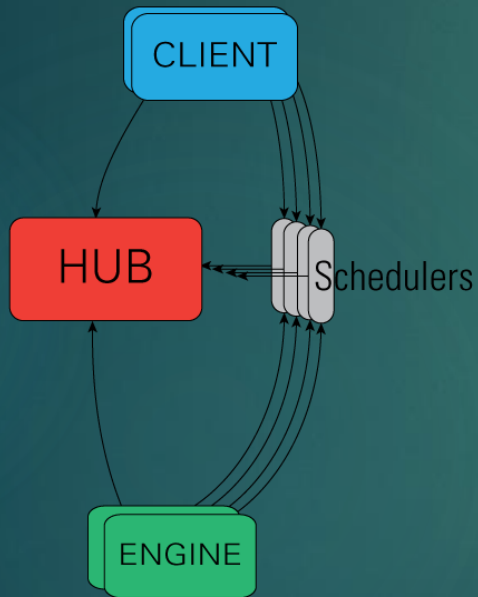
# Interactive Parallel and Distributed Computing

## Styles of parallelism

- ▶ Single program, multiple data (SPMD) parallelism.
- ▶ Multiple program, multiple data (MPMD) parallelism.
- ▶ Message passing using MPI.
- ▶ Task farming.
- ▶ Data parallel.
- ▶ Combinations of these approaches.
- ▶ Custom user defined approaches

IPython enables all types of parallel applications to be developed, executed, debugged and monitored *interactively*.

# Interactive Parallel and Distributed Computing(cont..)



-The IPython engine is a Python instance that takes Python commands over a network connection.

-The IPython controller processes provide an interface for working with a set of engines

-Hub is the process that keeps track of engine connections, schedulers, clients, as well as all task requests and results.

- While the engines themselves block when user code is run, the schedulers hide that from the user to provide a fully asynchronous interface to a set of engines.

Source: [http://ipython.org/ipython-doc/2/parallel/parallel\\_intro.html#examples](http://ipython.org/ipython-doc/2/parallel/parallel_intro.html#examples)

# Explore:

- ▶ Fernando Perez: IPython in depth: high productivity interactive and parallel python - PyCon 2014

<https://www.youtube.com/watch?v=XFw1JVXKJss>

- <http://ipython.org/>

- References:


<http://ipython.org/>

Wikipedia




Thank  
you





ask yourself  
the hard questions



esa