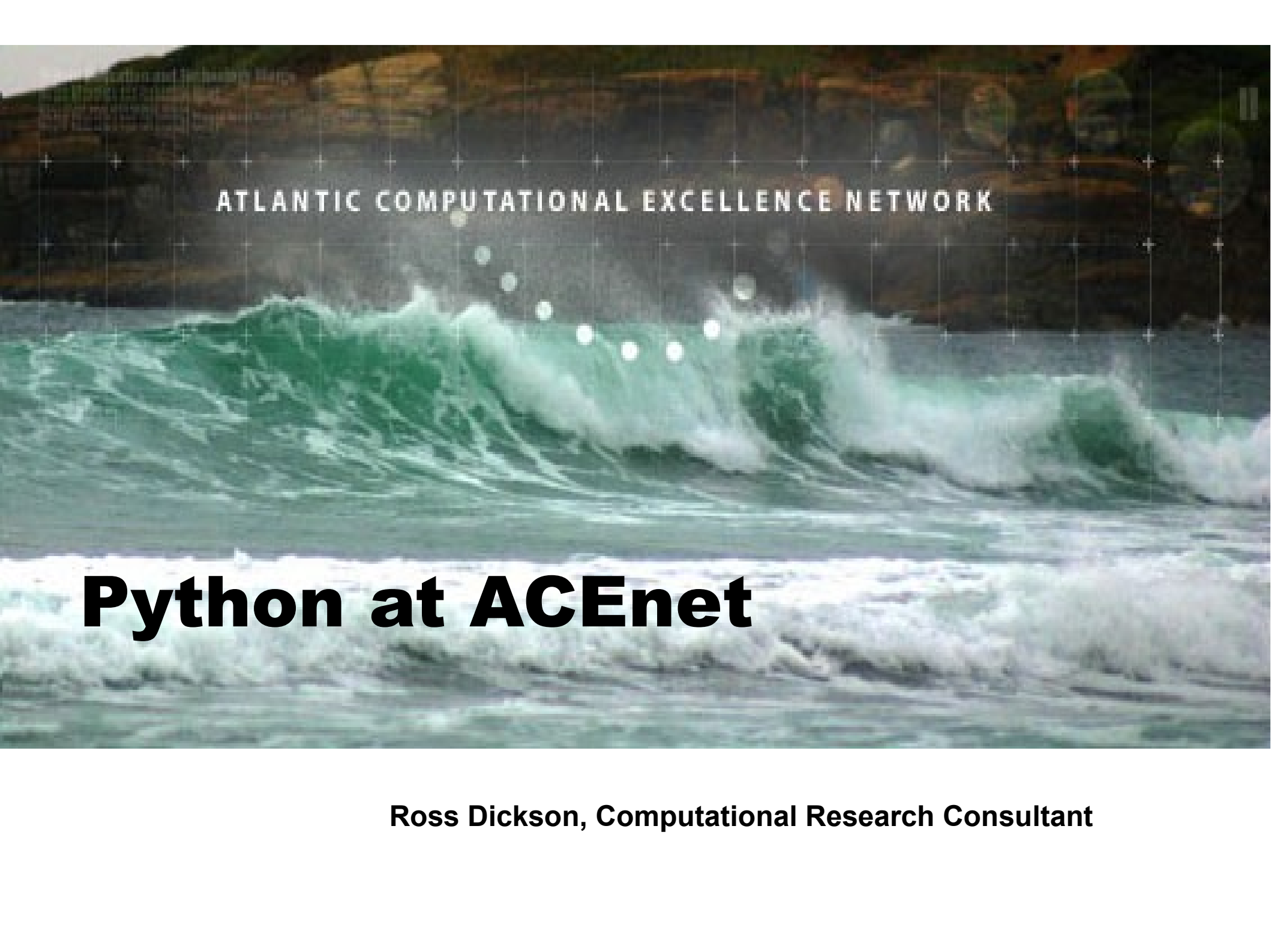




ATLANTIC COMPUTATIONAL EXCELLENCE NETWORK

Python at ACEnet

Ross Dickson, Computational Research Consultant

What about Python?

- An increasingly popular language for scientific programming.
 - Also: web programming, teaching, *etc.*
- ACEnet provides support
 - I'll show you the flavour of the language,
 - what ACEnet provides, and
 - how to do more with it
- Learn it at www.python.org

http://docs.python.org/tutorial/

Firefox

The Python Tutorial — Python v2.7.3 do... +

docs.python.org/tutorial/

Python v2.7.3 documentation » previous | next | modules | index

The Python Tutorial

Release: 2.7
Date: April 18, 2012

Python is an easy to learn, powerful programming language. It has efficient high-level data structures and a simple but effective approach to object-oriented programming. Python's elegant syntax and dynamic typing, together with its interpreted nature, make it an ideal language for scripting and rapid application development in many areas on most platforms.

« The Python interpreter and the extensive standard library are freely available in source or binary form for all major platforms from the Python Web site, <http://www.python.org/>, and may be freely distributed. The same site also contains distributions of and pointers to many free third party Python modules, programs and tools, and additional documentation.

The Python interpreter is easily extended with new functions and data types implemented in C or C++ (or other languages callable from C). Python is also suitable as an extension language for customizable applications.

This tutorial introduces the reader informally to the basic concepts and features of the Python language and system. It helps to have a Python interpreter handy for hands-on experience, but all examples are

Previous topic
What's New in Python 2.0

Next topic
1. Whetting Your Appetite

This Page
[Report a Bug](#)
[Show Source](#)

Quick search

Go

Enter search terms or a module, class or function name.

http://www.python.org

Firefox

Python Programming Language – Offici... +

www.python.org

Google

python™

search

Advanced Search

ABOUT >>

NEWS >>

DOCUMENTATION >>

DOWNLOAD >>

下載 >>

COMMUNITY >>

FOUNDATION >>

CORE DEVELOPMENT >>

Help

Package Index

Quick Links (2.7.3)

- >> Documentation
- >> Windows Installer
- >> Source Distribution

Quick Links (3.2.3)

- >> Documentation
- >> Windows Installer
- >> Source Distribution

Python Programming Language – Official Website

Python is a programming language that lets you work more quickly and integrate your systems more effectively. You can learn to use Python and see almost immediate gains in productivity and lower maintenance costs.

Python runs on Windows, Linux/Unix, Mac OS X, and has been ported to the Java and .NET virtual machines.

Python is free to use, even for commercial products, because

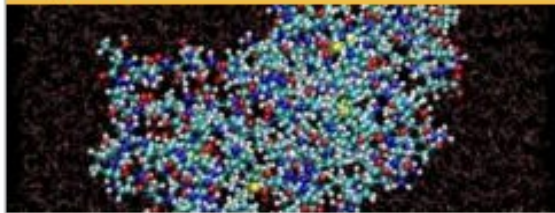
Python 3 Poll

I wish there was Python 3 support in

(enter [PyPI package name](#))

[Results](#)

Simulate biomolecules with Python...



... joining users such as [Rackspace](#), [Industrial Light and](#)

Invocation (at ACEnet)

```
$ module load python
```

```
$ python
```

```
Python 2.6.2 (r262:71600, May 26 2009,  
10:46:06)
```

```
[GCC 3.4.5 20051201 (Red Hat 3.4.5-2)] on  
linux2
```

```
Type "help", "copyright", "credits" or  
"license" for more information.
```

```
>>>
```

Or, "module load gcc python/2.7.2"

http://www.ace-net.ca/wiki/Modules

Firefox

Modules - ACEnet

www.ace-net.ca/wiki/Modules

Google

Search



Modules

Main page: [Software](#)

Over the next few months ACEnet will be gradually transitioning to using **Modules** to manage user environment. During this period we will also be updating documentation on the website. It's highly recommend that users start adopting new instructions on how to get access to different software packages. If you prefer using the old method of explicitly changing environment variables then you can keep using it. *December 16, 2011 (AST)*

Contents [\[show\]](#)

The Environment Modules package, or **Modules**, is a convenient tool to modify a user's environment. Modules allow to set or change different shell environment variables, such as `PATH` and `MANPATH`, in a clean fashion. ACEnet utilizes Modules to provide easy access to different pieces of software.

For every environment setting, there is a corresponding **modulefile**, which contains necessary instructions on how exactly a user's environment should be modified. The modulefiles can be loaded and unloaded dynamically and atomically, with the corresponding dependency and conflict resolution rules. This allows effective management of multiple versions of different applications and environments. Moreover, the Modules commands syntax is the same in every shell (whether it's `bash` or `csh`). You can find more information about Modules on the [Modules website](#). The version of Modules installed at ACEnet is a modified TCL 1.144 version.

- Account Renewal
- Cluster Status
- Ask Support
- Get an Account

Resources

- Compute Resources
- Videoconferencing
- Visualization
- Software

User Support

- User Guide
- Job Control
- FAQ
- Ask Support

News and Events

- Cluster Status
- Wavelets
- Events

Interactive use

- Can use Python interactively:

```
>>> a = 1
>>> b, c = "two", 3
>>> array=[a, b, c]
>>> array
[1, 'two', 3]
>>> (array[0]+array[2])/2.
2.0
>>>
```

- Like a calculator...
- ...but with strings, lists, loops, *etc. etc. etc.*

Language features

```
>>> weights = {'H':1.008, 'C':12.011, 'N':14.007}
>>> weights['C']
12.010999999999999
>>> weightFile = open('atomic_weights')
>>> weightFile
<open file 'atomic_weights', mode 'r' at
0x2a95854e00>
>>> weightFile.readline()
'H\t1.008\n'
```

- 'dictionary' *aka* associative array *or* hash
- *objects* (like the file) may have *methods* (like 'readline')
- "object oriented programming"

More language features

```
>>> for line in weightFile:
...     print(line.strip())
...
C      12.011
N      14.007
O      15.999
>>>
```

- **Indentation** delimits blocks (e.g. loop bodies)
 - No annoying braces or parens
 - Annoying spaces instead --- Take your pick...
- **for** loops:
 - **for thing in list:** idiom very common
 - **for number in range(n) :** also common

Modules

```
>>> a, b = 3, 4
>>> c = sqrt(a*a+b*b)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'sqrt' is not defined
>>> import math
>>> math.sqrt(a*a+b*b)
5.0
>>> from math import sqrt
>>> sqrt(a*a+b*b)
5.0
```

- Lots of functionality in modules
 - We'll come back to this

Programs

```
$ cat filter1.py
```

```
# From a file of numbers, one to a line, discard  
# values above 100 or below zero.
```

```
import sys
```

```
file = open(sys.argv[1], 'r')
```

```
for line in file:
```

```
    n = float(line)
```

```
    if 0 <= n and n <= 100:
```

```
        print n
```

```
$ python filter1.py numbers
```

```
1.0
```

```
50.0
```

```
99.0
```

```
$
```

Calling external programs

```
# Call an external program (qstat, in this example)
# and do something with the output.
```

```
from subprocess import Popen, PIPE

cmd = 'qstat -u *'
child = Popen(cmd.split(), stdout=PIPE)
linecount = 0
for line in child.stdout:
    linecount = linecount + 1
    if linecount % 10 == 1:
        print line.rstrip()
print "Line count: ",linecount
```

- Not as terse as Perl, but not too bad...

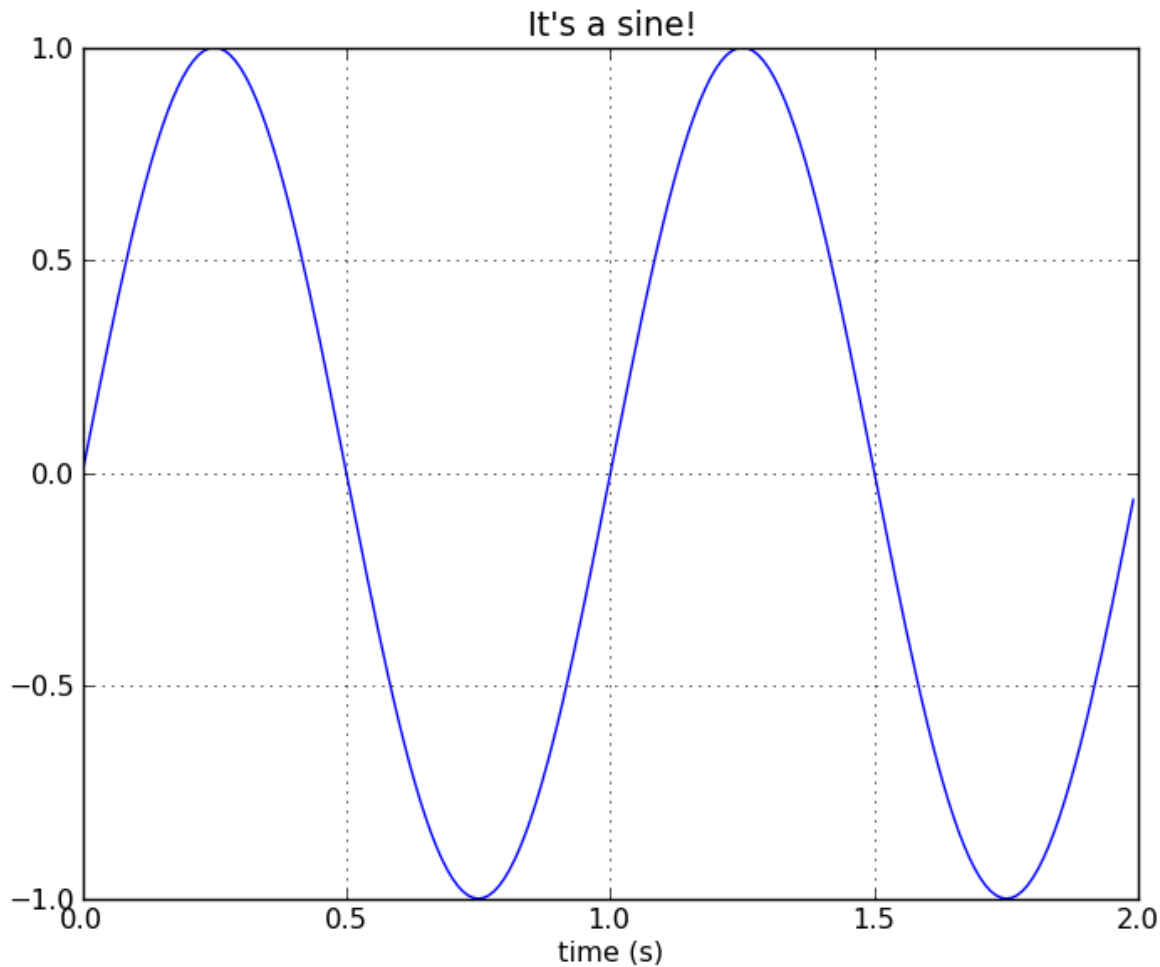
Python modules

Something for everyone



- numpy – matrix manipulation & numerics
- scipy – special functions, integration, optimization, interpolation, Fourier transforms, *etc. etc.* ...
- Scientific Python – not the same thing
- matplotlib *aka* pyplot
- basemap – world maps under your data

matplotlib



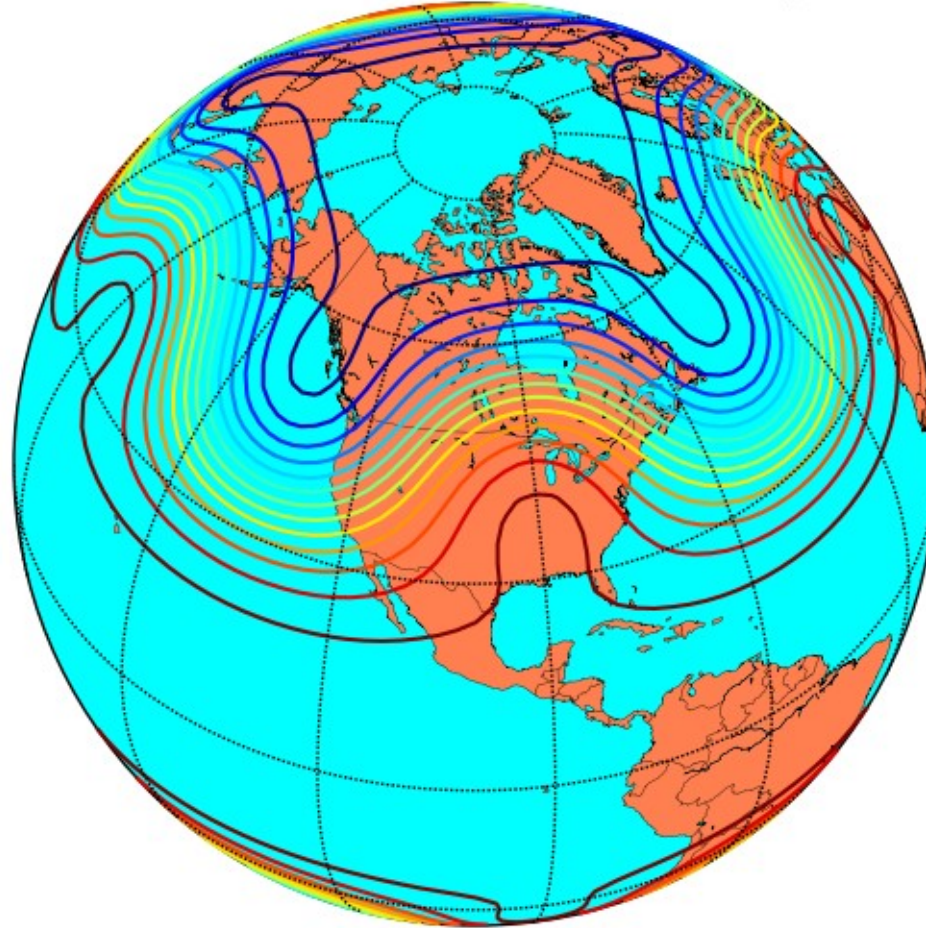
```
from pylab import *
```

```
t = arange(0.0, 2.0, 0.01)  
s = sin(2*pi*t)  
plot(t, s, linewidth=1.0)
```

```
xlabel('time (s)')  
ylabel('voltage (mV)')  
title('It\'s a sine!')  
grid(True)  
show()
```

basemap

contour lines over filled continent background



Installing modules

- We provide several modules
 - See <http://www.ace-net.ca/wiki/Python>
- ...but you can also install your own
- Check out PyPI
 - <http://pypi.python.org/pypi>
- General instructions are on our wiki (above) and here:
 - <http://docs.python.org/release/2.7.2/install/index.html>

Installing a module

Example: PIL, "Python Imaging Library"

```
$ module load gcc python/2.7.2
$ wget http://effbot.org/downloads/Imaging-1.1.7.tar.gz
. . .
$ tar xzf Imaging-1.1.7.tar.gz
. . .
$ cd Imaging-1.1.7
$ view README
. . .
$ python setup.py install -home=~
$ python
>>> import Image
>>> im = Image.open("Images/lena.png")
>>> im.show()
```

ipython

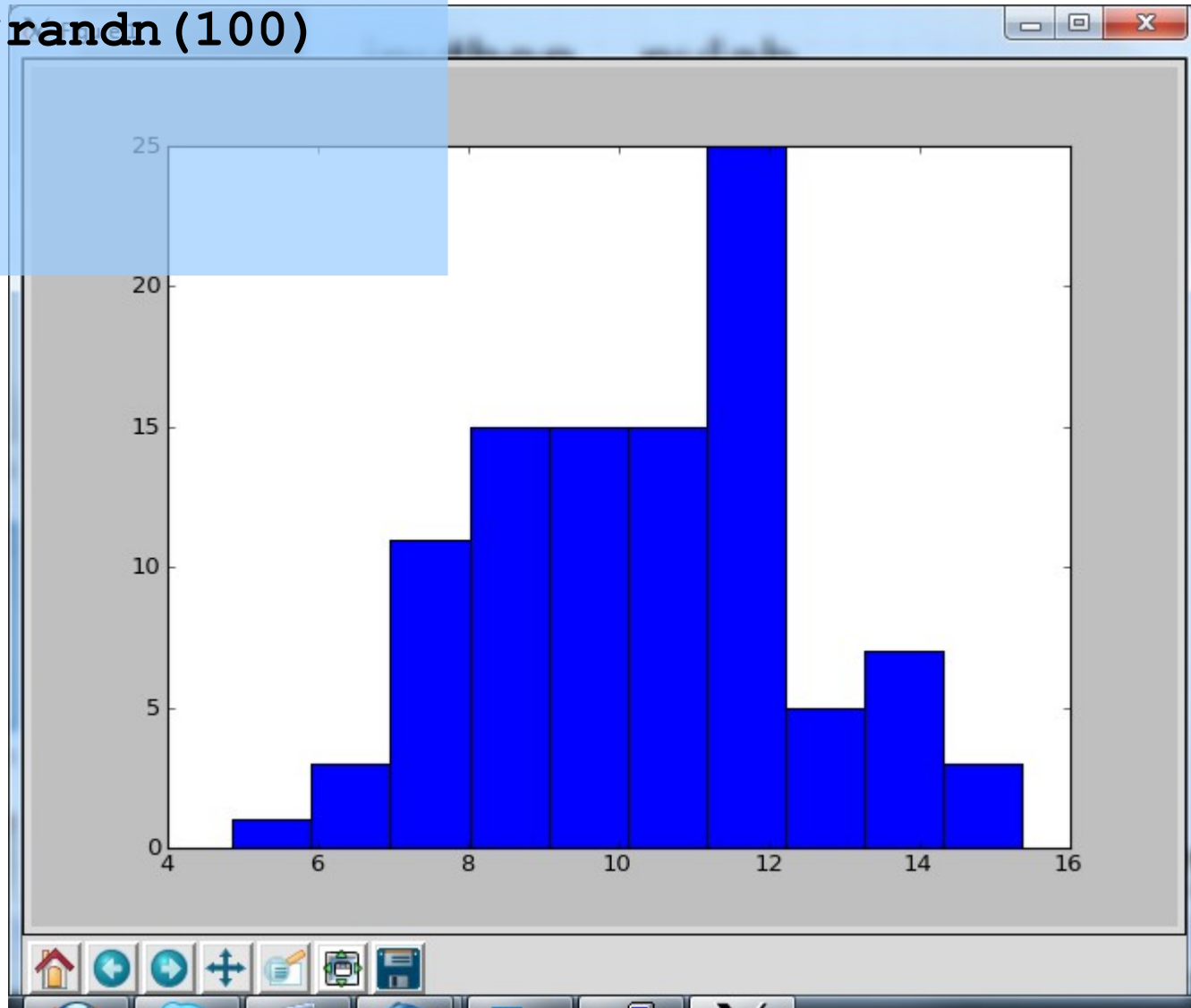
- Brings some shell operations inside Python
 - `'cd'`, `'ls'`, *etc.* at Python prompt
 - tab completion
- Introspection: `object_name?`
- Edit - save - reload - run
- Built-in debugger
- Integrates several popular modules
 - *e.g.* pylab

ipython --pylab

```
$ ipython --pylab
```

```
In [1]: x = 10 + 2*randn(100)
```

```
In [2]: hist(x, 10)
```



http://ipython.org/

The screenshot shows a Firefox browser window with the address bar displaying 'IPy IPython: Productive Interactive Computi...' and 'ipython.org'. The page title is 'IPython: Productive Interactive Computing'. The main content area features the IPython logo and a navigation bar with links: Home, Download, Documentation, Videos, Wiki, News, and Citation. Below this is a large heading 'IPython: Productive Interactive Computing' followed by a paragraph describing IPython's purpose and a list of its features. On the right side, there is a search bar, a 'VERSIONS' section with links for 'Stable' (0.12 - December 2011) and 'Development' (pre-0.13), and a 'LINKS' section with links for 'Issue tracker', 'Mailing lists' (User, Dev), and 'IRC channel'. At the bottom, it states 'IPython is open source (released under the revised BSD license)'.

Firefox

IPy IPython: Productive Interactive Computi... +

IPy ipython.org

IP[y]: IPython

Interactive Computing

[Home](#) · [Download](#) · [Documentation](#) · [Videos](#) · [Wiki](#) · [News](#) · [Citation](#)

IPython: Productive Interactive Computing

IPython provides a rich toolkit to help you make the most out of using Python interactively. Its main components are:

- Powerful Python shells (terminal- and Qt-based).
- A web-based notebook with the same core features but support for rich media, text, code, mathematical expressions and inline plots.
- Support for interactive data visualization and use of GUI toolkits.
- Flexible, embeddable interpreters to load into your own projects.
- Tools for high level and interactive parallel computing.

IPython is open source (released under the revised BSD license)

Google™ Custom Search x

VERSIONS

Stable
0.12 – December 2011
[Download](#)

Development
pre-0.13
[Github](#)

LINKS

[Issue tracker](#)
[Mailing lists:](#)
[User](#) · [Dev](#)
[IRC channel](#)

Example code

- On ACEnet clusters...
 - brasdor
 - courtenay
 - fundy
 - glooscap
 - mahone
 - placentia
- ...in `/home/rdickson/public/demo/Python/
cp /home/rdickson/public/demo/Python/* .`