Python for Earth Scientists

Andrew Walker

andrew.walker@bris.ac.uk
Python is:

A dynamic, interpreted programming language.
Python is:
A dynamic, interpreted programming language.

Source code → Compiler → Object code → Linker → Executable

Static, compiled language

Data → Output
Python is:

A dynamic, interpreted programming language.

The program can change at run time.
Python is:

A dynamic, interpreted programming language.

The program can change at run time.
Python is:

A dynamic, interpreted programming language...

...strongly typed but dynamically typed.
Python is:

A dynamic, interpreted programming language...

...strongly typed but dynamically typed.

Some typing styles

Weak

x = 1 + "2.0"

Dynamic

a = 5
...
a = "cat"

Static

real(dp) :: r
...
r = 5.2_dp
Python is:

A dynamic, interpreted programming language...

... strongly typed but dynamically typed ...

... imperative and object oriented.
Python is:

A dynamic, interpreted programming language...

... strongly typed but dynamically typed ...

... imperative and object oriented.

(Can look procedural)
Python is:

A dynamic, interpreted programming language...

...strongly typed but dynamically typed...

...imperative and object oriented...

...fun to learn and use.

http://xkcd.com/353/
Python is multi-paradigm, designed for rapid development, comes with batteries included, modern, free, open, portable, testable, opinionated, evolving, good as 'glue', and encourages rapid (agile) development.
Today

Now.  I’ll talk for ~30 mins on Python’s syntax and landscape

2 pm.  Practical 1: First steps

3 pm.  I’ll say something about types, modules, scripts and the standard library

4 pm.  Practical 2: Scripts and modules

Thursday

1 pm.  I’ll explain object oriented programming

2 pm.  Practical 3: Object oriented programming

3 pm.  I’ll introduce SciPy, NumPy and Matplotlib

4 pm.  Practical 4: NumPy, SciPy and Matplotlib

We should finish by 5 pm. Don’t worry if you don’t finish all the exercises.
Syntax

print "Hello, world!"

a = 1 # an integer
b = 2.7 # a float
c = "Hello, world" # a string
print a
print b
print c
x1 = 6 # This is OK
1x = 7 # This is an error

Starting with this is compulsory

Variable assignment, comments and three types
Some mathematical and logical operators.

```python
a = 1
b = 10
c = a + b  # What is c?
(4 < 5)  # True
(4 > 5)  # False
((4 < 5) or (4 > 5))  # True
```
if sky == 'blue':
    birdsong = True
elif sky == 'black':
    birdsong = False
else:
    pass  # do nothing

If, elif, else. Note indentation. No case statement.
Syntax

```python
a = 0
while (a < 10):
    print a
    a = a + 1
```

While loop

```
for a in range(10):
    print a
```

Iteration
Syntax

def add_ten (value):
    value = value + 10
    return value

Function definition

b = add_ten(5)
print b

Function use
#!/usr/bin/env python

"""
List files in the working directory
"""
import os
def main():
    print os.getcwd()
    for fn in os.listdir('.'):  
        print fn
if __name__ == '__main__':
    main()
#!/usr/bin/env python

"""
List files in the working
directory
"""

import os

def main():
    print os.getcwd()
    for fn in os.listdir('.'):  
        print fn
    if __name__ == '__main__':
        main()
The interpreter

Language parser and implementation

CPython
C
Reference implementation. Runs on Windows, Unix, Mac etc. Extension modules are .so/.dll

Jython
Java
Targets Java virtual machine. Can load Java libraries as extension modules

IronPython
C#
Targets .NET framework. Can integrate with Silverlight, Visual Basic etc.

PyPy
Python
Compiler and implementation. Includes JIT compiler and can be very fast

for Earth Scientists:
27 & 29 Sept. 2011
Standard library

~300 modules designed to be shipped with interpreter

- Mathematical functions
- Advanced string processing
- Many data types and specialised algorithms
- Threading and multiprocessing
- Access to OS facilities

- Networking: sockets, http, email, ftp
- Data storage, database access
- Compression, encryption, hashing
- File formats, XML
- Graphics

http://docs.python.org/library/
Package index

16725 free to use packages (collections of modules) in a searchable index

http://pypi.python.org/

for Earth Scientists: 27 & 29 Sept. 2011
The interpreter

Language parser and implementation

Integrated development environment

Standard library modules

Third party modules

Module management and upgrade tool

Interpreter ± collection of modules ± other stuff = Python distribution
Some distributions

- Official CPython distribution
- Distribution with OS: Mac, Linux
- Commercial: ActivePython, Enthought
- Specialised: stackless, embedded
Versioning

In principle language version is separate from interpreter version and versioning of modules. However, CPython and Python stay in lock-step.
Development

Interpreter developed in a similar way to other large pieces of open source code (distributed version control, bug tracker, etc.)

Language developed by the adoption of Python Enhancement Proposals (PEPs) - open process but ultimately decided on by “BDFL”

(Python Language Moratorium until late 2012)

http://www.python.org/dev/peps/
http://docs.python.org/devguide/#
http://docs.python.org/

http://www.python.org/
Enthought Python Distribution

Commercial python distribution aimed at scientists. Free for academics. What we will use because it is easy to install and has all the modules we need.
