



Python 0

Some material adapted
from Upenn cmpe391
slides and other sources



Overview

- History
- Significance
- Installing & Running Python
- Simple script examples

Brief History of Python

- Invented in the Netherlands, early 90s by Guido van Rossum
- Named after Monty Python
- Open sourced from the beginning, managed by [Python Software Foundation](#)
- Considered a scripting language, but is much more
- Scalable, object oriented and functional from the beginning
- Used by Google from the beginning

Python's Benevolent Dictator For Life

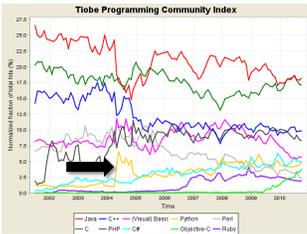
“Python is an experiment in how much freedom programmers need. Too much freedom and nobody can read another's code; too little and expressiveness is endangered.”

- [Guido van Rossum](#)



Python's place in the Market

- TIOBE has been collecting data on programming language "popularity" for many years
- Counts results for a query like "<language> programming" on popular search engines



Position Oct 2010	Position Oct 2009	Delta in Position	Programming Language	Ratings Oct 2010	Delta Oct 2009	Status
1	1	==	Java	18.166%	-0.48%	A
2	2	==	C	17.177%	+0.33%	A
3	4	↑	C++	9.802%	-0.08%	A
4	3	↓	PHP	8.323%	-2.03%	A
5	5	==	(Visual) Basic	5.650%	-3.04%	A
6	6	==	C#	4.983%	+0.55%	A
7	7	==	Python	4.860%	+0.96%	A
8	12	↑↑↑↑	Objective-C	3.706%	+2.54%	A
9	8	↓	Perl	2.310%	-1.45%	A
10	10	==	Ruby	1.941%	-0.51%	A
11	9	↓	JavaScript	1.659%	-1.37%	A
12	11	↓	Delphi	1.558%	-0.58%	A
13	17	↑↑↑↑	Lisp	1.084%	+0.48%	A-
14	24	↑↑↑↑↑	Transact-SQL	0.820%	+0.42%	A-
15	15	==	Pascal	0.771%	+0.10%	A-
16	18	↑↑	RPG (OS/400)	0.708%	+0.12%	A-
17	29	↑↑↑↑	Ada	0.704%	+0.40%	A-
18	14	↓	SAS	0.664%	-0.14%	B
19	19	==	MATLAB	0.627%	+0.05%	B
20	-	↑↑↑↑↑	Go	0.626%	+0.63%	B

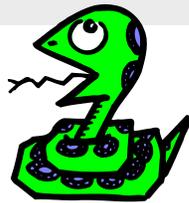
10/10 TIOBE Programming Community Index

http://python.org/

http://docs.python.org/

The Python tutorial is good!

Running Python



The Python Interpreter

- Typical Python implementations offer both an interpreter and compiler
- Interactive interface to Python with a read-eval-print loop

```
[finin@linux2 ~]$ python
Python 2.4.3 (#1, Jan 14 2008, 18:32:40)
[GCC 4.1.2 20070626 (Red Hat 4.1.2-14)] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> def square(x):
...     return x * x
...
>>> map(square, [1, 2, 3, 4])
[1, 4, 9, 16]
>>>
```

Installing

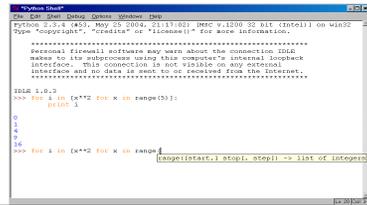
- Python is pre-installed on most Unix systems, including Linux and MAC OS X
- The pre-installed version may not be the most recent
- Two “latest versions”
 - 2.7 released 7/10 & 3.2 released 10/10
 - Python 3 is a non-backward compatible version which you should not use for 331
- Download from <http://python.org/download/>
- Python comes with a large library of standard modules

Python IDEs and Shells

- There are many Integrated Development Environments
 - IDLE
 - Eclipse + PyDev
 - Emacs
- As well as enhanced shells

IDLE Development Environment

- **IDLE** is the “official” IDE distributed with Python
- Preinstalled on MAC OS X
- Written in Python with the **Tkinter** GUI package
- Multi-window text editor with syntax highlighting, auto-completion, smart indent and other features
- Python shell with syntax highlighting, line recall, ...
- Integrated debugger with stepping, persistent breakpoints, and call stack visibility



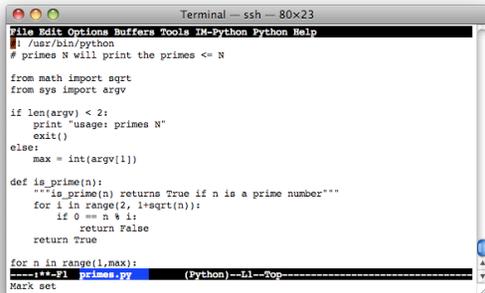
Eclipse + Pydev



- Pydev is an Eclipse plugin for Python
- Download from <http://pydev.org/>
- Syntax highlighting, code completion, goto function, debugger, ...

Editing Python in Emacs

- Emacs *python-mode.el* has good support for editing Python, enabled by default for .py files
- Features: completion, symbol help, eldoc, and inferior interpreter shell, etc.



Emacs as a Python IDE

- You can fire up a shell in emacs via M-x python-shell
- You can also set up a more powerful Python IDE environment in EMACS
 - **Pymacs** allows two-way communication between Emacs Lisp and Python
 - **Ropemacs** provides advanced features such as completion, refactoring, etc

Running Interactively on UNIX

On Unix...

```
% python
>>> 3+3
6
```

- Python prompts with '>>>'.
- To exit Python (not Idle):
 - In Unix, type CONTROL-D
 - In Windows, type CONTROL-Z + <Enter>
 - Evaluate exit()

Running Programs on UNIX

- Call python program via the python interpreter

```
% python fact.py
```
- Make a python file directly executable by
 - Adding the appropriate path to your python interpreter as the first line of your file

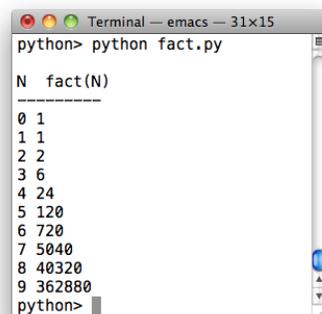
```
#!/usr/bin/python
```
 - Making the file executable

```
% chmod a+x fact.py
```
- Invoking file from Unix command line

```
% fact.py
```

Example 'script': fact.py

```
#!/usr/bin/python
def fact(x):
    if x == 0:
        return 1
    return x * fact(x - 1)
print "\nN fact(N)"
print "-----"
for n in range(10):
    print n, fact(n)
```



A terminal window titled 'Terminal -- emacs -- 31x15' showing the execution of a Python script. The prompt 'python>' is followed by 'python fact.py'. The output is a table with two columns: 'N' and 'fact(N)'. The output is as follows:

N	fact(N)
0	1
1	1
2	2
3	6
4	24
5	120
6	720
7	5040
8	40320
9	362880

The terminal prompt 'python>' is visible at the bottom of the window.

Python Scripts

- When you call a python program from the command line the interpreter evaluates each expression in the file
- Familiar mechanisms are used to provide command line arguments and/or redirect input and output
- Python also has mechanisms to allow a python program to act both as a script and as a module to be imported and used by another python program

Another Script Example

```
#!/usr/bin/python
""" reads text from stdin and outputs any email
    addresses it finds, one to a line """
import re
from sys import stdin

# a regular expression for a valid email address
pat = re.compile(r'[-\w][-\w]*@[-\w][-\w.]+[a-zA-Z]{2,4}')

for line in stdin:
    for address in pat.findall(line):
        print address
```

results

```
python> python email0.py <email.txt
bill@msft.com
gates@microsoft.com
steve@apple.com
bill@msft.com
python>
```

Getting a unique, sorted list

```
import re
from sys import stdin

pat = re.compile(r'[-\w][-\w]*@[-\w][-\w.]+[a-zA-Z]{2,4}')
# found is an initially empty set (a list w/o duplicates)
found = set()
for line in stdin:
    for address in pat.findall(line):
        found.add(address)

# sorted() takes a sequence, returns a sorted list of its elements
for address in sorted(found):
    print address
```

results

```
python> python email2.py <email.txt
bill@msft.com
gates@microsoft.com
steve@apple.com
python>
```

Conclusion: Python is ..

- Popular as a scripting language
- Popular as a general purpose language
- Open sourced
- Interesting from a program language perspective
- Easy to learn and use, so being used in many CS 101 courses