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

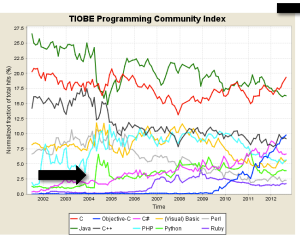
- Invented in the Netherlands, early 90s by Guido van Rossum
- Named after Monty Python
- Open sourced from the beginning, managed by Python Software Foundation
- Considered a scripting language, but is much more
- Scalable, object oriented and functional from the beginning
- Used by Google from the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

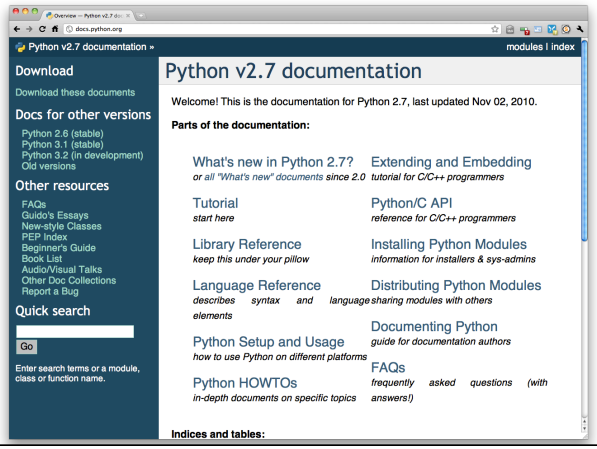


Position Sep 2012	Position Sep 2011	Delta in Position	Programming Language	Ratings Sep 2012	Delta Sep 2011	Status
1	2	↑	C	19.295%	+1.29%	A
2	1	↓	Java	16.267%	-2.49%	A
3	6	↑↑↑	Objective-C	9.770%	+3.61%	A
4	3	↓	C++	9.147%	+0.30%	A
5	4	↓	C#	6.596%	-0.22%	A
6	5	↓	PHP	5.614%	-0.98%	A
7	7	=	(Visual) Basic	5.528%	+1.11%	A
8	8	=	Python	3.861%	-0.14%	A
9	9	=	Perl	2.267%	-0.20%	A
10	11	↑	Ruby	1.724%	+0.29%	A
11	10	↓	JavaScript	1.328%	-0.14%	A
12	12	=	Delphi/Object Pascal	0.993%	-0.32%	A
13	14	↑	Lisp	0.969%	-0.07%	A
14	15	↑	Transact-SQL	0.875%	+0.02%	A
15	39	↑↑↑↑↑↑↑↑↑↑	Visual Basic .NET	0.840%	+0.53%	A
16	16	=	Pascal	0.830%	-0.02%	A
17	13	↓↓↓↓	Lua	0.723%	-0.43%	A-
18	18	=	Ada	0.700%	+0.02%	A--
19	17	↓↓	PL/SQL	0.604%	-0.12%	B
20	22	↑↑	MATLAB	0.563%	+0.02%	B

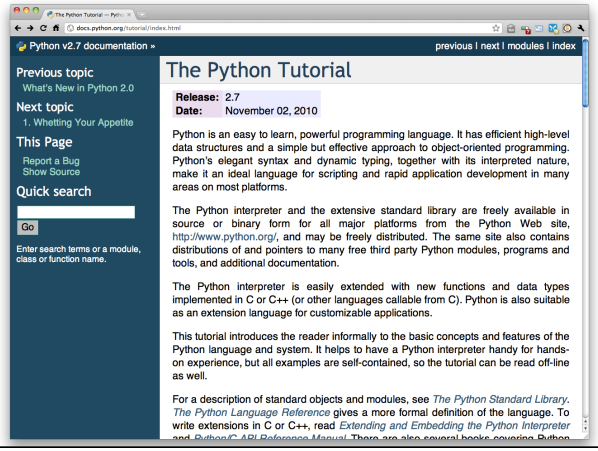
09/12 TIOBE Programming Community Index

http://python.org/

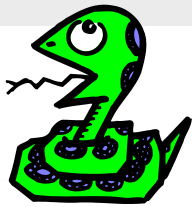
http://docs.python.org/



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

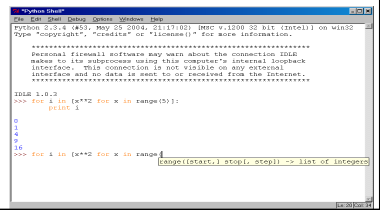
- Python (Cpython) is pre-installed on most Unix systems, including Linux and OS X
- Pre-installed version may not be most recent
- Two "latest versions" of Cpython:
  - v2.7.3 released April 2012 and v3.2.3
  - Python 3 is a non-backward compatible version which you should not use for 671
- Download 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
  - iPython
- Most expert Python programmers I know use emacs

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

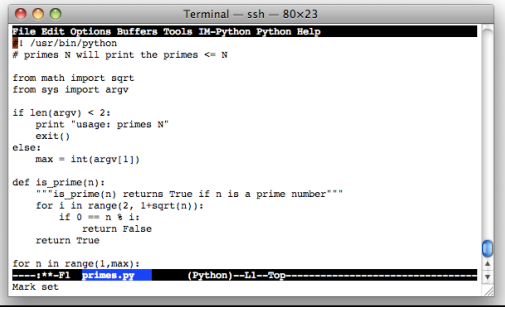


Eclipse + Pydev

- 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 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

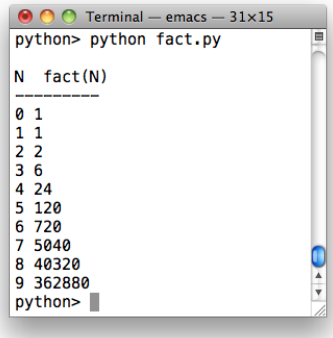
```
% fact.py
```

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)
```



fact.py

Python Scripts

- When you call a python program from the command line the interpreter evaluates each expression in the file
- For output, call print or write explicitly
- Familiar mechanisms provide command line arguments and/or redirect input and output
- Python has a convention 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
```

email0.py

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
```

email1.py

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
- Fast enough for most purposes
- Interesting from a program language perspective
- Easy to learn and use, so being used in many CS 101 courses