



# 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

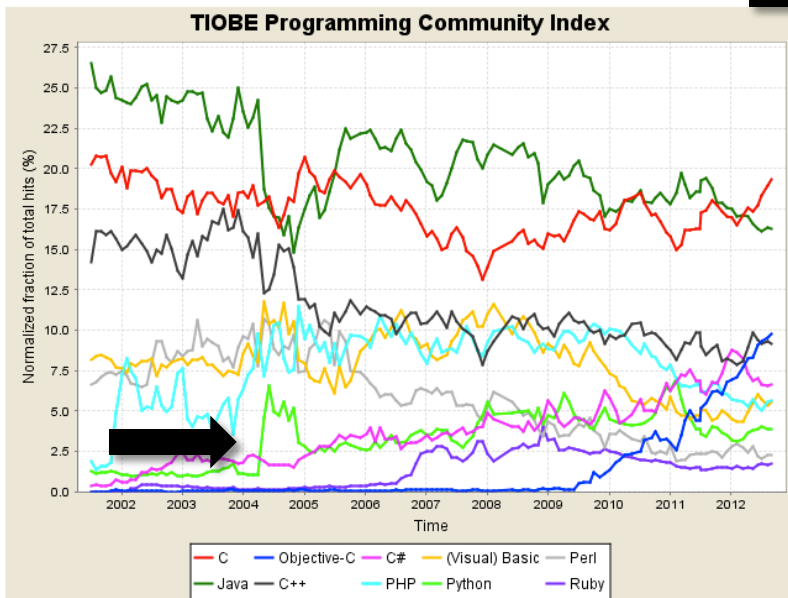
“Python is an experiment in how much freedom programmers need. Too much freedom and nobody can read another's code; too little and expressiveness is endangered.”

- [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

# http://python.org/

The screenshot shows a browser window with the URL `python.org`. The page features the Python logo and a search bar. A navigation menu on the left includes links for ABOUT, NEWS, DOCUMENTATION, DOWNLOAD, COMMUNITY, FOUNDATION, and CORE DEVELOPMENT. The main content area is titled "Python Programming Language – Official Website" and contains several sections: a introductory paragraph about Python's benefits, a "Python 3 Poll" section with a form to vote on Python 3 support, a "NASA uses Python..." section with an image of an astronaut, and a "What they are saying..." section with a quote from ITA Software. The footer includes logos for Visa and Mastercard, and a publication date of Tuesday, 12 Oct 2010.

Python Programming Language

python.org

python™

search

Advanced Search

Screen styles  
normal\* large userpref

ABOUT »

NEWS »

DOCUMENTATION »

DOWNLOAD »

COMMUNITY »

FOUNDATION »

CORE DEVELOPMENT »

Help

Package Index

Quick Links (2.7)

- » Documentation
- » Windows Installer
- » Source Distribution

Quick Links (3.1.2)

- » Documentation
- » Windows Installer
- » Source Distribution

Python Jobs

Python Merchandise

Python Wiki

Python 2 or 3?

Help Maintain Website

Help Fund Python

## 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 of its OSI-approved [open source license](#).

New to Python or choosing between Python 2 and Python 3? Read [Python 2 or Python 3](#).

The [Python Software Foundation](#) holds the intellectual property rights behind Python, underwrites the [PyCon conference](#), and funds other projects in the Python community.

[Read more](#), -or- [download Python now](#)

» **Python 3.2 alpha 3 released**  
The [third and final alpha release of Python 3.2](#) has been released for testing.

Published: Tue, 12 Oct 2010, 8:30 +0200

### Python 3 Poll

I wish there was Python 3 support in

(enter [PyPI package name](#))

[Results](#)

### NASA uses Python...



... joining users such as [Rackspace](#), [Industrial Light and Magic](#), [AstraZeneca](#), [Honeywell](#), and many others.

### What they are saying...

**ITA Software:**

Since then, we've changed how we use Python a ton internally. We have lots more production software written in Python.

# http://docs.python.org/

The image shows a browser window displaying the Python 2.7 documentation website. The browser's address bar shows 'docs.python.org'. The page has a dark blue header with 'Python v2.7 documentation' and 'modules | index' on the right. A left sidebar contains navigation links for 'Download', 'Docs for other versions', and 'Other resources'. The main content area is titled 'Python v2.7 documentation' and includes a welcome message, a 'Parts of the documentation:' section with links to various guides, and an 'Indices and tables:' section at the bottom.

Overview — Python v2.7 doc

docs.python.org

Python v2.7 documentation » modules | index

## Download

Download these documents

## Docs for other versions

- Python 2.6 (stable)
- Python 3.1 (stable)
- Python 3.2 (in development)
- Old versions

## Other resources

- FAQs
- Guido's Essays
- New-style Classes
- PEP Index
- Beginner's Guide
- Book List
- Audio/Visual Talks
- Other Doc Collections
- Report a Bug

## Quick search

Go

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

## Python v2.7 documentation

Welcome! This is the documentation for Python 2.7, last updated Nov 02, 2010.

### Parts of the documentation:

- [What's new in Python 2.7?](#)  
*or all "What's new" documents since 2.0*
- [Extending and Embedding](#)  
*tutorial for C/C++ programmers*
- [Tutorial](#)  
*start here*
- [Python/C API](#)  
*reference for C/C++ programmers*
- [Library Reference](#)  
*keep this under your pillow*
- [Installing Python Modules](#)  
*information for installers & sys-admins*
- [Language Reference](#)  
*describes syntax and language elements*
- [Distributing Python Modules](#)  
*sharing modules with others*
- [Python Setup and Usage](#)  
*how to use Python on different platforms*
- [Documenting Python](#)  
*guide for documentation authors*
- [Python HOWTOs](#)  
*in-depth documents on specific topics*
- [FAQs](#)  
*frequently asked questions (with answers!)*

### Indices and tables:

# The Python tutorial is good!

The screenshot shows a web browser window with the address bar displaying `docs.python.org/tutorial/index.html`. The page title is "The Python Tutorial" and it is part of the "Python v2.7 documentation". The left sidebar contains navigation links for "Previous topic" (What's New in Python 2.0), "Next topic" (1. Whetting Your Appetite), "This Page" (Report a Bug, Show Source), and "Quick search" (with a search input field and a "Go" button). The main content area features the title "The Python Tutorial" followed by "Release: 2.7" and "Date: November 02, 2010". The text describes Python as an easy-to-learn, powerful programming language with efficient high-level data structures and a simple but effective approach to object-oriented programming. It mentions that the Python interpreter and standard library are freely available in source or binary form for all major platforms. The tutorial is described as self-contained and suitable for reading off-line. It also references *The Python Standard Library*, *The Python Language Reference*, and *Extending and Embedding the Python Interpreter and Python/C API Reference Manual*.

Python v2.7 documentation » previous | next | modules | index

## The Python Tutorial

**Release:** 2.7  
**Date:** November 02, 2010

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.

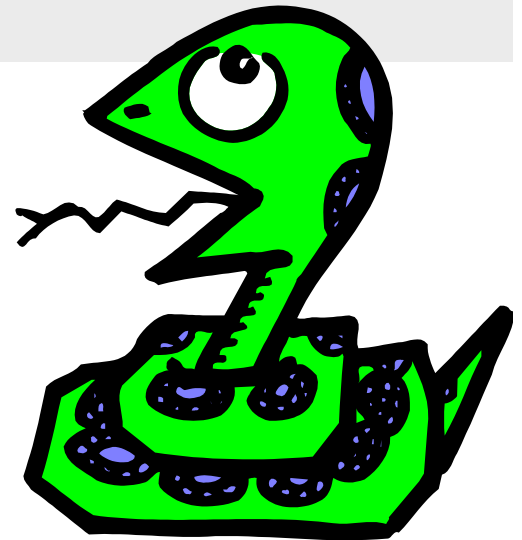
The Python interpreter is easily extended with new functions and data types implemented in C or C++ (or other languages callable from C). Python is also suitable as an extension language for customizable applications.

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 self-contained, so the tutorial can be read off-line as well.

For a description of standard objects and modules, see *The Python Standard Library*. *The Python Language Reference* gives a more formal definition of the language. To write extensions in C or C++, read *Extending and Embedding the Python Interpreter and Python/C API Reference Manual*. There are also several books covering Python



# 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

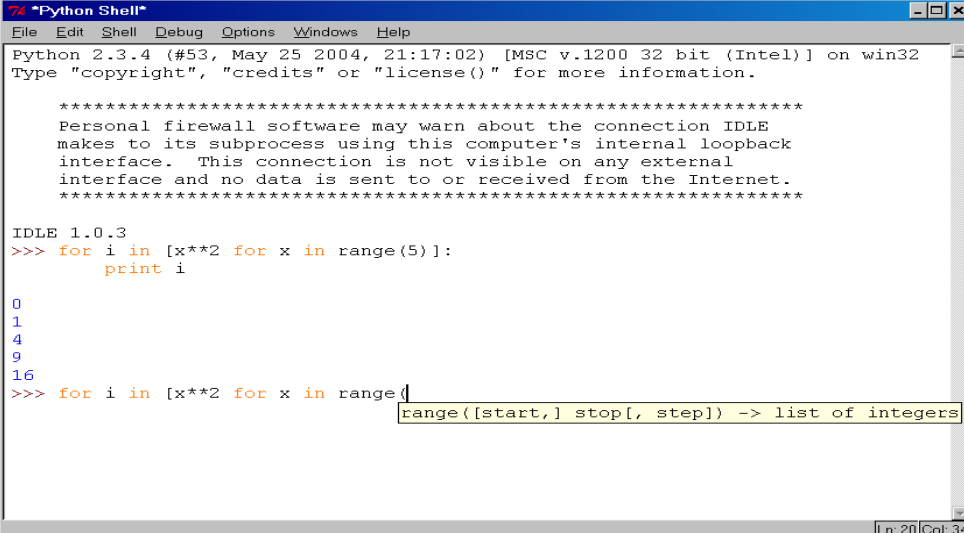
- Python (Cpython) is pre-installed on most Unix systems, including Linux and OS X
- Pre-installed version may not be most recent
- Two “latest versions” of Cpython:
  - v2.7.3 released April 2012 and v3.2.3
  - Python 3 is a non-backward compatible version which you should not use for 671
- Download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



```
Python Shell
File Edit Shell Debug Options Windows Help
Python 2.3.4 (#53, May 25 2004, 21:17:02) [MSC v.1200 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.

*****
Personal firewall software may warn about the connection IDLE
makes to its subprocess using this computer's internal loopback
interface. This connection is not visible on any external
interface and no data is sent to or received from the Internet.
*****

IDLE 1.0.3
>>> for i in [x**2 for x in range(5)]:
    print i
0
1
4
9
16
>>> for i in [x**2 for x in range(
range([start,] stop[, step]) -> list of integers

Ln: 20 | Col: 34
```

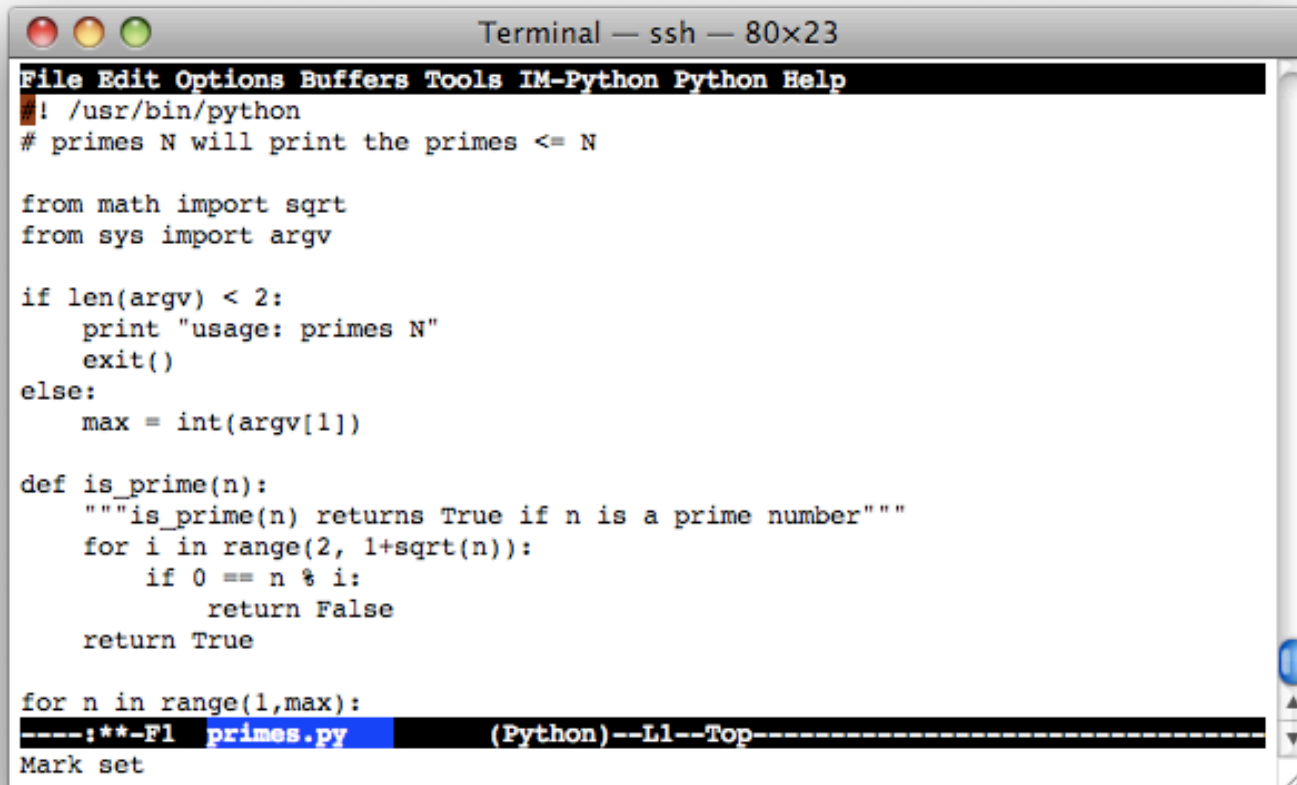
# Eclipse + Pydev



- 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.



The image shows a terminal window titled "Terminal — ssh — 80x23" with an Emacs-style menu bar: "File Edit Options Buffers Tools IM-Python Python Help". The terminal displays Python code for finding primes. The code includes imports for sqrt and argv, a usage message, a function is\_prime, and a loop to print primes up to a given number. The terminal also shows Emacs-style status information at the bottom: "----:\*\*-F1 primes.py (Python)--L1--Top-----" and "Mark set".

```
Terminal — ssh — 80x23
File Edit Options Buffers Tools IM-Python Python Help
! /usr/bin/python
# primes N will print the primes <= N

from math import sqrt
from sys import argv

if len(argv) < 2:
    print "usage: primes N"
    exit()
else:
    max = int(argv[1])

def is_prime(n):
    """is_prime(n) returns True if n is a prime number"""
    for i in range(2, 1+sqrt(n)):
        if 0 == n % i:
            return False
    return True

for n in range(1,max):
----:**-F1 primes.py (Python)--L1--Top-----
Mark set
```

# 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

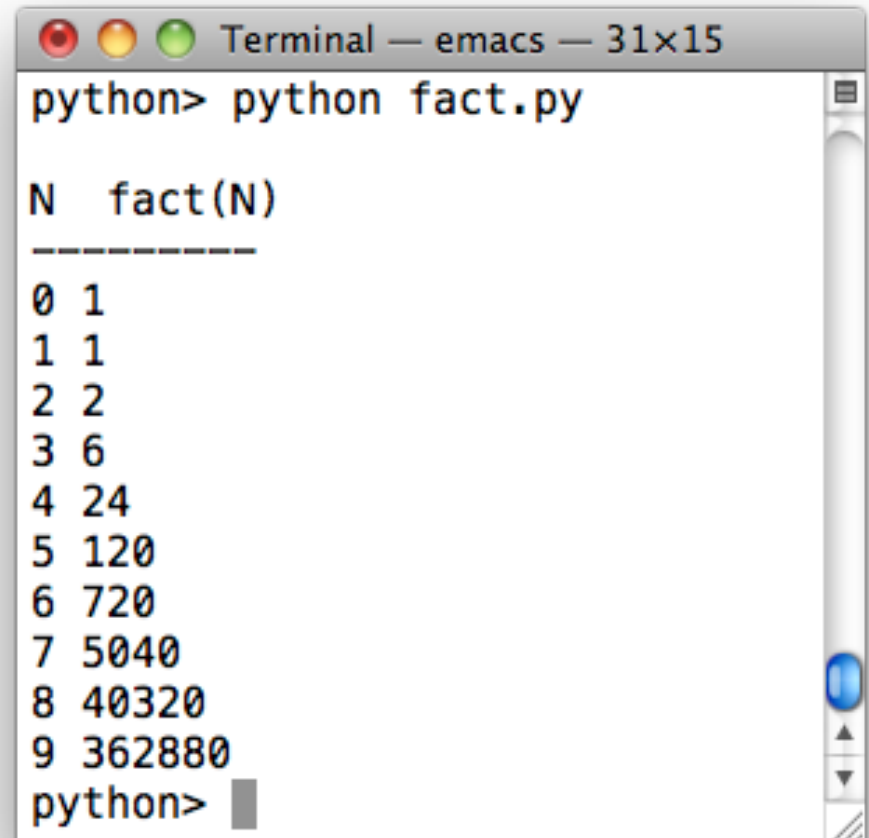
```
% fact.py
```

# 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 named fact.py. The prompt is "python> python fact.py". The output is a table with two columns: "N" and "fact(N)". A dashed line separates the header from the data. The data rows are: 0 1, 1 1, 2 2, 3 6, 4 24, 5 120, 6 720, 7 5040, 8 40320, 9 362880. The prompt "python>" is followed by a cursor.

```
Terminal — emacs — 31x15
python> python fact.py

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

[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