

# What is Python?

- ▶ Object-oriented, rapid prototyping language
- ▶ Multi-purpose: Web, GUI, scripting, science, etc.
- ▶ Interpreted and interactive; very easy to learn
- ▶ Strongly typed and dynamically typed
- ▶ Focus on readability and productivity
- ▶ Extensive standard library and community
- ▶ CPython, Jython, IronPython, PyPy, etc.
- ▶ Home page: <http://www.python.org/>

## Who uses Python?

- ▶ Google (various projects)
- ▶ NASA (several projects)
- ▶ NYSE (one of only three languages “on the floor”)
- ▶ Industrial Light & Magic (everything)
- ▶ Yahoo! (Yahoo mail & groups)
- ▶ RealNetworks (function and load testing)
- ▶ RedHat and Ubuntu (Linux installation tools)
- ▶ LLNL, Fermilab (steering scientific applications)

*More success stories at <http://www.pythonology.com/>*

# Python vs Java

## Python

- ▶ Dynamically typed  
(assign names to values)
- ▶ Concise: “expressing much in a few words; implies clean-cut brevity, attained by excision of the superfluous”

```
# open the file d.txt  
myFile = open("d.txt")
```

## Java

- ▶ Statically typed  
(must declare variables)
- ▶ Verbose: “abounding in words; using or containing more words than are necessary”

```
import java.io.*;  
...  
// open the file d.txt  
BufferedReader myFile =  
    new BufferedReader(  
        new FileReader("d.txt"));
```

<http://pythonconquerstheuniverse.wordpress.com/category/java-and-python/>

## Some Basics

- ▶ Indentation matters (no braces!)
- ▶ Type matters (must be consistent)
- ▶ Comments start with a #
- ▶ Strings use 's or "s (or ""s)
- ▶ `print` (vs `System.out.println`)
- ▶ `None`  $\approx$  `null`, `self`  $\approx$  `this`
- ▶ `and`, `or`, `not` instead of `&&`, `||`, `!`
- ▶ `[1, 2, 3]` is a list ( $\approx$  `ArrayList`)

# Discrete Math Example

```
"""Simple demo of the Caesar Cipher."""

def encrypt(msg, k):
    """Shifts msg by k letters, ignoring spaces."""
    ans = ""
    for c in msg:
        if 'a' <= c <= 'z':
            p = ord(c) - ord('a') # char to int
            q = (p + k) % 26
            c = chr(q + ord('a')) # int to char
        ans += c
    return ans

if __name__ == "__main__":
    print encrypt("hello world", 3)
```

# Editors & Tools

Recommended: your favorite editor

- ▶ Linux: <http://www.geany.org/>
- ▶ Windows: <http://notepad-plus-plus.org/>
- ▶ Mac: <http://www.barebones.com/products/textwrangler/>

Optional: heavyweight IDE

- ▶ Eclipse + PyDev: <http://pydev.org/>

Command Line

- ▶ IPython <http://ipython.org/>
  - ▶ Mac/Windows: requires `readline` module
- ▶ PyChecker <http://pychecker.sourceforge.net/>

# Ready, Set, Learn!

## Official Documentation

- ▶ Tutorial: <http://docs.python.org/2/tutorial/>
- ▶ Reference: <http://docs.python.org/2/reference/>
- ▶ Modules: <http://docs.python.org/2/py-modindex.html>

## Free Textbooks

- ▶ Dive Into Python <http://diveintopython.org/>
- ▶ How to Think Like a Computer Scientist  
<http://www.greenteapress.com/>

## Online Tutorials

- ▶ <http://www.codecademy.com/tracks/python>
- ▶ <http://www.learnpython.org/>