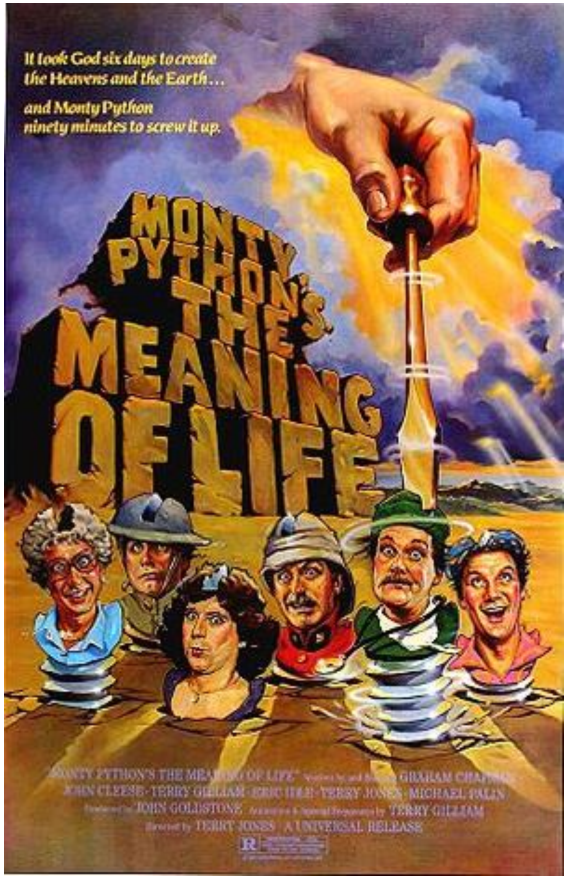
Introduction to Python

Origins

Nature of Python

Importance of Python

Example


It took God six days to create the Heavens and the Earth…
and Monty Python ninety minutes to screw it up.
MONTY PYTHON'S THE MEANING OF LIFE
GRAHAM CHAPMAN
JOHN CLEESE · TERRY GILLIAM · ERIC IDLE · TERRY JONES · MICHAEL PALIN
JOHN GOLDSTONE
TERRY GILLIAM
Directed by TERRY JONES · A UNIVERSAL RELEASE
R

What is Python?

Invented by Guido van Rossum in late 1980s in Netherlands.

General purpose, high level language intended to be straightforward to read and write.

Supports OOP, imperative *and* functional programming.

Dynamic type system and automatic memory management

Has large, comprehensive standard library

Python is interpreted

- Can type a little instruction and process it immediately at the command (or shell) prompt

History

Early Python – descendant of ABC that would appeal to UNIX/C programmers

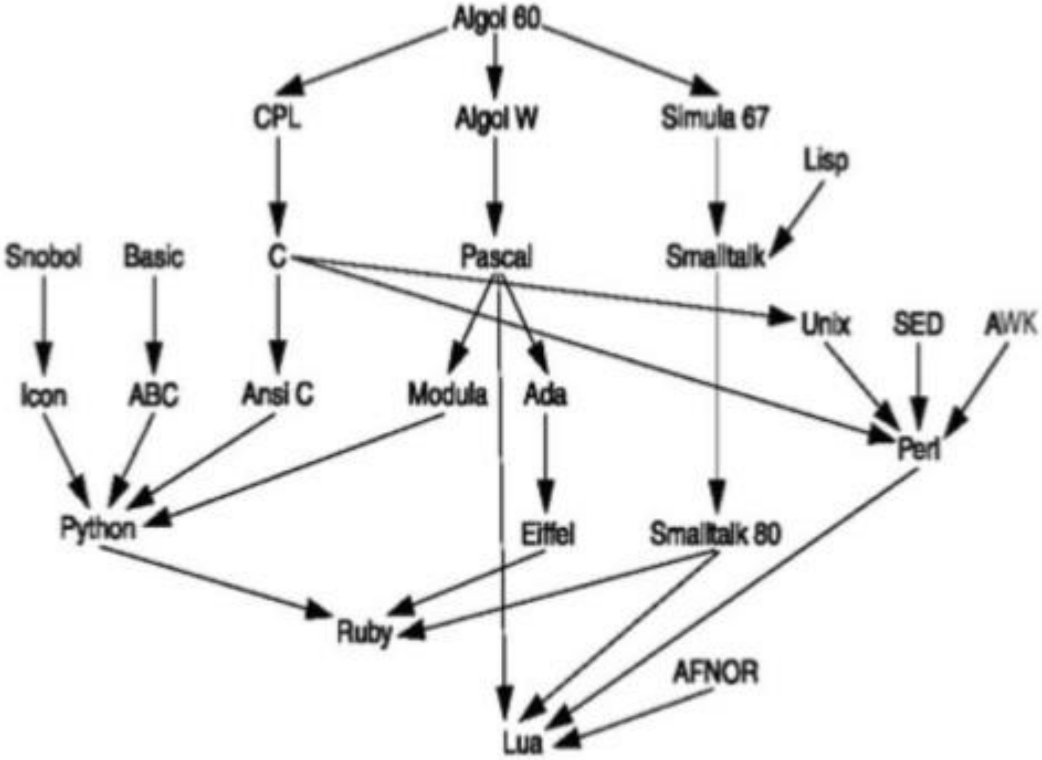
Python 2.0 released in 2000 had major new features

- Garbage collector
- Support for Unicode

Python 3.0 – major, backwards-incompatible release in 2008

Now Python – many major features have been backported to the backwards compatible 2.6 and 2.7

Python/Ruby/Lua Family Tree



Language and Programs

Simple language

- a beginner can write small useful programs (unlike Perl – only one way to do things)
- Sparser, less-cluttered grammar

Extensible language

- focuses on mechanisms to extend the programming language, compiler and runtime environment
- Over 65K packages
  - GUIs, web frameworks, multimedia, DBS, networking, test frameworks, system admin, scientific computing, text processing and image processing, etc.

Central Characteristics

Environment variables - defined by language implementation, not defined by user, but accessible

Program variables are implicitly declared

- type is inferred by compiler based on name syntax or content
- Starts with letter and followed by letters, numbers or _

Numeric types- `int`, `long`, `float` and `complex`, `Booleans` are a subtype of int

`str`- popular type in Python, easily converts to/from float

`float ('3.14')` → 3.14   and `str(3.14)` → '3.14'

A first Script

```
>>> print 'Hello world!'

Hello world!
```

A Second Script

```
>>> x = 6
>>> print x
6
>>> y = x * 7
>>> print y
42
>>>
```

Programs

Typing commands into the Python interpreter is great way to experiment, but not recommended for solving complex problems

Use a text editor to write Python instructions and save to a file.

Convention is to use the extension .py

To execute, at the command prompt your can view and run the program.

Example

```
csev$ cat hello.py

print ('Hello world!’)

csev$ python hello.py

Hello world!

csev$
```

Another example

```
name = input('Enter file:')
handle = open(name, 'r')
text = handle.read()
words = text.split()
counts = dict()

for word in words:
    counts[word] = counts.get(word,0) + 1

bigcount = None
bigword = None
for word,count in counts.items():
    if bigcount is None or count > bigcount:
       bigword = word
       bigcount = count

print (bigword, bigcount)
```

Getting Python

- https://www.continuum.io/downloads
- http://docs.continuum.io/anaconda/install
- http://docs.continuum.io/anaconda/index

Learning Resources

Python wiki (home page) Python is a interpretive programming language that is about as easy to learn as any.   It is heavily used by computer scientist as well as researchers in the other STEM areas. Biology, Physics and Engineering are popular areas where this is applied.

- PythonLearn Tutorial (includes videos, powerpoints etc)
- Book:Python for Informatics by Charles Severance.
- Book:Think Python by Allen Downey
- **Python Tutorial by UDEMY**

Python Lectures by Pattis (These are really a nice collection of lectures that are worth reading)

Python is in Visual Studio 2013. You can use Ubuntu (see ubuntu notes)