Introduction to Python

ELC1035 - Prática em Sistemas Operacionais

João Vicente Ferreira Lima

Universidade Federal de Santa Maria
jvlima@inf.ufsm.br
http://www.inf.ufsm.br/~jvlima

2023/2

Outline

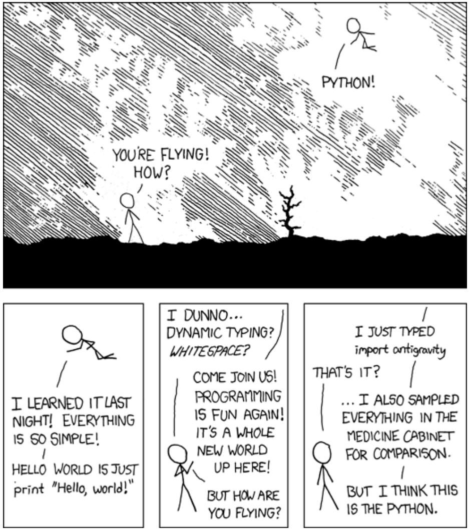
Outline

Monty Python

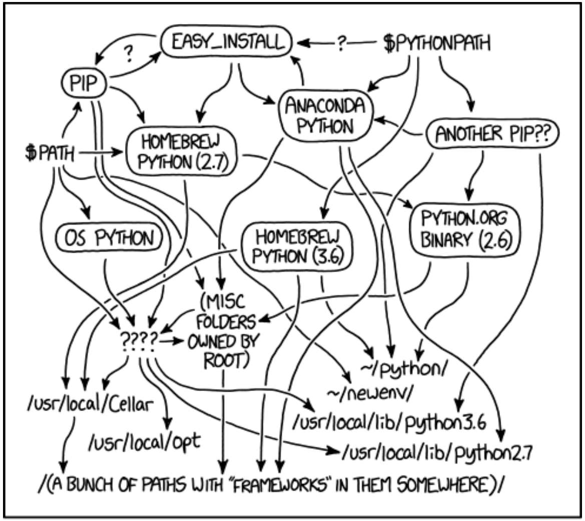
SPAM

SPAM SPAM SPAM SPAM SPAM SPAM SPAM SPAM
LOVELY SPAM, WONDERFUL SPAM!
imgflip.com

XKCD Python


YOU'RE FLYING! HOW?
PYTHON!
I LEARNED IT LAST NIGHT! EVERYTHING IS SO SIMPLE!
HELLO WORLD IS JUST print "Hello, world!"
I DUNNO... DYNAMIC TYPING? WHITESPACE?
COME JOIN US! PROGRAMMING IS FUN AGAIN! IT'S A WHOLE NEW WORLD UP HERE!
BUT HOW ARE YOU FLYING?
I JUST TYPED import antigravity
THAT'S IT?
... I ALSO SAMPLED EVERYTHING IN THE MEDICINE CABINET FOR COMPARISON.
BUT I THINK THIS IS THE PYTHON.

Python Environment



MY PYTHON ENVIRONMENT HAS BECOME SO DEGRADED THAT MY LAPTOP HAS BEEN DECLARED A SUPERFUND SITE.

Hello

```
#!/usr/bin/env python3

print('Hello world!')
print('What is your name?')
myName = input()
print('What is your age?')
myAge = input()
print('Your age is ' + str(int(myAge)))
```

If/else

```
#!/usr/bin/env python3

print('Type your password:')
word = input()

if word == "hello":
  print('Password correct!')
else:
  print('Wrong password!')
```

If/else

```
#!/usr/bin/env python3

print('Type your password:')
word = input()

if word == "hello":
  print('Password correct!')
else:
  print('Wrong password!')
```

Version note

In Python 3.X, `raw_input` was renamed to `input`. The equivalent of Python 2.X `input` is `eval(input())`.

While

```
#!/usr/bin/env python3

print('Type your password:')
word = input()
while word != "hello":
    print('Type your password:')
    word = input()

print('Thank you!')
```

While

```python
#!/usr/bin/env python3

while True:
    print('Type your password:')
    word = input()
    if word == "hello":
        break

print('Thank you!')
```

While

```
#!/usr/bin/env python3

while True:
    print('Type your login:')
    login = input()
    if login != "root":
        continue
    print('Type your password:')
    word = input()
    if word == "hello":
        break

print('Thank you!')
```

For

```
#!/usr/bin/env python3

for i in range(5):
    print(str(i)) # 0 1 2 3 4

for i in range(10,15):
    print(str(i)) # 10 11 12 13 14

for i in range(0, 10, 2):
    print(str(i)) # 0 2 4 6 8

for i in range(5, -1, -1):
    print(str(i)) # 5 4 3 2 1 0
```

For

```python
#!/usr/bin/env python3
import random

for i in range(5):
    print(random.randrange(1, 10)) # a <= N < b

for i in range(5):
    print(random.randint(1, 10)) # a <= N <= b
```

Exit

```
#!/usr/bin/env python3
import sys

while True:
    res = input('Type exit: ')
    if res == 'exit':
        sys.exit()
```

Functions

```
#!/usr/bin/env python3
import random
def getLevel(ph):
    if ph <= 6 and ph >= 8:
        return 'neutral'
    elif ph >= 8:
        return 'acid'
    else:
        return 'base'

    return 'invalid'

ph = input("Digit the pH level: ")
level = getLevel(ph)
print("pH level is: " + level)
```

Print

```
#!/usr/bin/env python3

print('Hello', end='')
print('World')

print('cats', 'dogs', 'mice')
print('cats', 'dogs', 'mice', sep=',')
```

Exceptions

```python
#!/usr/bin/env python3

def spam(divider):
    return 42/divider

try:
    print(spam(2))
    print(spam(12))
    print(spam(0))
    print(spam(1))
except ZeroDivisionError:
    print('Error: invalid argument')
```

```
21
3
Error: invalid argument
```

Outline

Lists

```
#!/usr/bin/env python3

spam = ['cat', 'bat', 'rat', 'elephant']
print(spam[0])

print(spam[-1])

print(spam[1:3])

del spam[1]
print(spam)
```

```
cat
elephant
['bat', 'rat']
['cat', 'rat', 'elephant']
```

Lists

```
#!/usr/bin/env python3
spam = ['cat', 'bat', 'rat', 'elephant']
if 'cat' in spam:
    print('There is a cat.')

print(spam.index('rat'))

spam.append('moose')
spam.insert(1, 'chicken')

print(spam)
```

```
There is a cat.
2
['cat', 'chicken', 'bat', 'rat', 'elephant', 'moose']
```

Lists

```
#!/usr/bin/env python3
spam = ['cat', 'bat', 'rat', 'moose',
        'chicken', 'elephant']

spam.sort()
print(spam)

spam.reverse()
print(spam)
```

```
['bat', 'cat', 'chicken', 'elephant', 'moose', 'rat']
['rat', 'moose', 'elephant', 'chicken', 'cat', 'bat']
```

Lists

```python
#!/usr/bin/env python3
import random
messages = ['It is certain',
            'It is decidedly so',
            'Yes definitely',
            'Reply hazy try again',
            'Ask again later',
            'Concentrate and ask again',
            'My reply is no',
            'Outlook not so good',
            'Very doubtful']

print(random.choice(messages))
```

```
Concentrate and ask again
Concentrate and ask again
Yes definitely
```

References

```python
#!/usr/bin/env python3

spam = 42
cheese = spam
spam = 100

print(spam)
print(cheese)
```

```
100
42
```

References

```python
#!/usr/bin/env python3

spam = [0, 1, 2, 3, 4]
cheese = spam
cheese[1] = 'Buenas'

print(spam)
print(cheese)
```

```
[0, 'Buenas', 2, 3, 4]
[0, 'Buenas', 2, 3, 4]
```

References

```python
#!/usr/bin/env python3

def eggs(something):
    something.append('Buenas')

spam = [1, 2, 3]
eggs(spam)
print(spam)
```

```
[1, 2, 3, 'Buenas']
```

Copy

```
#!/usr/bin/env python3
import copy

spam = [0, 1, 2, 3, 4]
cheese = copy.copy(spam)
cheese[1] = 'Buenas'

print(spam)
print(cheese)
```

```
[0, 1, 2, 3, 4]
[0, 'Buenas', 2, 3, 4]
```

Outline

Dictionaries

```
#!/usr/bin/env python3

myCat = {'size': 'fat', 'color': 'gray',
         'disposition': 'loud'}

print('My cat is my ' + myCat['size'])
print('It has ' + myCat['color'] + ' fur.')
```

```
My cat is my fat
It has gray fur.
```

Dictionaries

```python
#!/usr/bin/env python3
spam = {'color': 'red', 'age': 42}
for v in spam.values():
    print(v)

for k in spam.keys():
    print(k)

for i in spam.items():
    print(i)
```

```
red
42
color
age
('color', 'red')
('age', 42)
```

Dictionaries

```python
#!/usr/bin/env python3

spam = {'color': 'red', 'age': 42}

for k, v in spam.items():
    print('Key: ' + k + ' Value: ' + str(v))
```

```
Key: color Value: red
Key: age Value: 42
```

Dictionaries

```
#!/usr/bin/env python3

spam = {'name': 'Sophie', 'age': 7}

print( 'name' in spam.keys()      )
print( 'Sophie' in spam.values() )

print( 'color' in spam.keys() )
print( 'color' in spam.values() )
```

```
True
True
False
False
```

Dictionaries

```
#!/usr/bin/env python3

hwinfo = {'disk': 3, 'mem': 10, 'cpu': 2}
print('The PC has ' + str(hwinfo.get('disk', 0)) +
      ' disks.')

print('The PC has ' + str(hwinfo.get('tape', 0)) +
      ' data tapes.')
```

```
The PC has 3 disks.
The PC has 0 data tapes.
```

Outline

Strings

```python
#!/usr/bin/env python3

spam = "Hello one"
print(spam)

spam = "Hello two\nOther line"
print(spam)

spam = r"Hello three\nAnother line"
print(spam)
```

```
Hello one
Hello two
Other line
Hello three\nAnother line
```

Indexing and slicing

```python
#!/usr/bin/env python3

spam = 'Hello world!'

print(spam[0])
print(spam[-1])
print(spam[0:5])
```

```
H
!
Hello
```

In and not int

```
#!/usr/bin/env python3

spam= 'Hello World'
print( 'Hello' in spam )
print( 'HELLO' in spam )
print( 'World' not in spam )
```

```
True
False
False
```

Upper and lower

```python
#!/usr/bin/env python3

spam= 'Hello World'

print( spam.upper() )
print( spam.lower() )
```

```
HELLO WORLD
hello world
```

isX

- `isalpha()` only letters and not blank
- `isalnum()` only letters and numbers and not blank
- `isdecimal()` only numeric characters and not blank
- `isspace()` only spaces, tabs, and newlines and not blank
- `istitle()` only words that begin with an uppercase letter followed by only lowercase letters.

Star and end

```
#!/usr/bin/env python3

spam= 'Hello World'
print( spam.startswith('Hello') )
print( spam.endswith('World') )
```

```
True
True
```

Split and join

```python
#!/usr/bin/env python3

spam = ', '.join(['cats', 'rats', 'bats'])
print(spam)

spam = ' '.join(['My', 'name', 'is', 'Earl'])
print(spam)

print( spam.split() )
```

```
cats, rats, bats
My name is Earl
['My', 'name', 'is', 'Earl']
```

Justifying

```python
#!/usr/bin/env python3

spam = 'Hello'
print( spam.rjust(20) )
print( spam.ljust(20) )
print( spam.center(20) )

print( spam.center(20, '-') )
print( spam.rjust(20, '-') )
```

```
               Hello
Hello
       Hello
-------Hello--------
---------------Hello
```

Strip

```python
#!/usr/bin/env python3

spam = '      Hello world      '
print( spam.strip() )
print( spam.rstrip() )
print( spam.lstrip() )
```

```
Hello world
      Hello world
Hello world
```

Pyperclip

The package does not come installed. To install, run:

```
pip3 install pyperclip
```

```python
#!/usr/bin/env python3
import pyperclip

pyperclip.copy('Hello world')
spam = pyperclip.paste()
print( spam )
```

```
Hello world
```

Outline

Classes

```python
class Person:
    def __init__(self, name, age, pay=0, job=None):
        self.name = name
        self.age = age
        self.pay = pay
        self.job = job

    def lastName(self):
        return self.name.split()[-1]

    def giveRaise(self, percent):
        self.pay *= (1.0 + percent)

if __name__ == '__main__':
    bob = Person('Bob Smith', 42, 30000, 'software')
    print(bob.lastName())
```

Inheritance

```python
from person import Person

class Manager(Person):
    def giveRaise(self, percent, bonus=0.1):
        self.pay *= (1.0 + percent + bonus)

if __name__ == '__main__':
    tom = Manager(name='Tom Doe', age=50, pay=50000)
    print(tom.lastName())
    tom.giveRaise(.20)
    print(tom.pay)
```

Outline

Installation

virtualenv

`virtualenv` is a tool to create isolated Python environments. You can install libraries locally.

Install the `virtualenv` package.

```
sudo apt install virtualenv
```

Alternative:

```
pip install virtualenv
```

Creating

Creates a Python 2 environment.

```
virtualenv test
```

To use Python 3:

```
virtualenv -p python3 test
```

Usage

Enter the envrionment:

```
cd test
source bin/activate
(test) $
```

Exit:

```
(test) $ deactivate
```

Outline

Backslash and forward slash

```python
#!/usr/bin/env python3
import os
mypath = os.path.join('usr', 'local', 'bin')
print(mypath)

myprogs = ['git', 'gcc', 'ld']
for filename in myprogs:
    print(os.path.join(mypath, filename))
```

```
usr/local/bin
usr/local/bin/git
usr/local/bin/gcc
usr/local/bin/ld
```

Directories

```python
#!/usr/bin/env python3
import os

print( os.path.abspath('.') )
print( os.path.isabs('.') )
print( os.path.isabs(os.path.abspath('.')) )
```

```
/Users/jvlima/Source/disciplinas/pso/lectures
False
True
```

Directories

```python
#!/usr/bin/env python3
import os
print( os.path.relpath('/usr/local', '.') )
print( os.getcwd() )

path = '/usr/local/bin/git'
print( os.path.basename(path) )
print( os.path.dirname(path) )
print( path.split(os.path.sep) )
```

```
../../../../../../usr/local
/Users/jvlima/Source/disciplinas/pso/lectures
git
/usr/local/bin
['', 'usr', 'local', 'bin', 'git']
```

Directories

```
#!/usr/bin/env python3
import os

totalSize = 0
for filename in os.listdir('/usr/local/bin'):
    totalSize += os.path.getsize(
        os.path.join('/usr/local/bin', filename))

print(totalSize)
```

```
276636265
```

Check path

```python
#!/usr/bin/env python3
import os

print( os.path.exists('/usr/local') )
print( os.path.isdir('/usr/local') )
print( os.path.isfile('/usr/local') )
```

```
True
True
False
```

Outline

Reading files

```python
#!/usr/bin/env python3
import os

spamfile = open('/etc/networks')
contents = spamfile.read()
print(contents)
```

```
##
# Networks Database
##
loopback 127 loopback-net
```

Reading files

```python
#!/usr/bin/env python3
import os

spamfile = open('/etc/networks')
line = spamfile.readline()
while line != '':
    print(line, end='')
    line = spamfile.readline()
```

```
##
# Networks Database
##
loopback 127 loopback-net
```

Reading files

```
#!/usr/bin/env python3
import os

spamfile = open('/etc/networks')
for line in spamfile:
    print(line, end='')
```

```
##
# Networks Database
##
loopback 127 loopback-net
```

Reading files

```python
#!/usr/bin/env python3
import os

spamfile = open('/etc/networks')
contents = spamfile.readlines()
print(contents)
```

```
['##\n', '# Networks Database\n', '##\n', 'loopback\t127\t\tloopback-net\n']
```

Outline

Writing files

```
#!/usr/bin/env python3
import os

baconfile = open('bacon.txt', 'w')
baconfile.write("Hello world\n")
baconfile.write("Bacon is life\n")
baconfile.close()

baconfile = open('bacon.txt')
content = baconfile.read()
baconfile.close()
print(content)
```

```
Hello world
Bacon is life
```

Saving variables

The `shelf` module allows to save variables to binary shelf files.

```
#!/usr/bin/env python3
import shelve

datafile = shelve.open('mydata')
spam = ['cat', 'bat', 'rat', 'moose', 'chicken',
        'elephant']
datafile['zoo'] = spam
datafile.close()
```

Saving variables

```python
#!/usr/bin/env python3
import shelve

datafile = shelve.open('mydata')
print( list(datafile.keys()) )
print( list(datafile.values()) )
datafile.close()
```

```
['zoo']
[['cat', 'bat', 'rat', 'moose', 'chicken', 'elephant']]
```

Outline

Copying files and folders

```
#!/usr/bin/env python3
import shutil
import os

os.chdir('/Users/jvlima')
shutil.copy('a.txt', 'tmp')
if os.path.exists('/Users/jvlima/a.txt'):
    print('Created')
```

Copying files and folders

```
#!/usr/bin/env python3
import shutil
import os

os.chdir('/Users/jvlima')
shutil.copytree('tmp', 'tmp2')
if os.path.exists('/Users/jvlima/tmp2'):
    print('Ok')
```

Moving and renaming

```
#!/usr/bin/env python3
import shutil

shutil.move('/Users/jvlima/a.txt',
            '/Users/jvlima/tmp' )
```

Delete files and folders

- `os.unlink(path)` delete the file at *path.*
- `os.rmdir(path)` delete the folder at *path*.
- `shutil.rmtree(path)` remove the folder at *path* and all files/folders inside.

```
#!/usr/bin/env python3
import os

for filename in os.listdir():
    if filename.endswith('.txt'):
        os.unlink(filename)
```

Safe delete

```
pip3 install send2trash
```

```python
#!/usr/bin/env python3
import send2trash

baconFile = open('bacon.txt', 'a')
baconFile.write('Bacon is life')
baconFile.close()

send2trash.send2trash('bacon.txt')
```

Walking directories

```python
#!/usr/bin/env python3
import os

for name, subfolders, filenames in os.walk('tmp'):
    print('The current folder is ' + name)
    for subfolder in subfolders:
        print('SUBFOLDER OF ' + name + ': ' +
              subfolder)

    for filename in filenames:
        print('FILE INSIDE ' + name + ': ' +
              filename)
    print('')
```

Walking directories

```
The current folder is tmp
SUBFOLDER OF tmp: foo
FILE INSIDE tmp: a.txt
FILE INSIDE tmp: b.txt

The current folder is tmp/foo
FILE INSIDE tmp/foo: c.txt
FILE INSIDE tmp/foo: d.txt
```

Zip files

```
#!/usr/bin/env python3
import zipfile, os

filezip = zipfile.ZipFile('tmp.zip')
print( filezip.namelist() )

spaminfo = filezip.getinfo('spam.txt')
print( spaminfo.file_size )
print( spaminfo.compress_size )
filezip.close()
```

```
['b.txt', 'foo/', 'foo/c.txt', 'foo/d.txt', 'spam.txt']
15
15
```

Extract Zip files

```
#!/usr/bin/env python3
import zipfile, os

filezip = zipfile.ZipFile('tmp.zip')
filezip.extractall()
filezip.close()
```

Extract single Zip file

```python
#!/usr/bin/env python3
import zipfile, os

filezip = zipfile.ZipFile('tmp.zip')
filezip.extract('spam.txt', 'tmp1')
filezip.close()
```