

# Introduction à python

---

M1 RÉSEAUX ET TELECOMS– RT0704

OLIVIER FLAUZAC



# Généralités

---

Langage de programmation objet

Langage multi-plate-forme

Faiblement typé

Doté d'un vaste API (modules)

Versions

- **Python 2.7** et Python 3
- Jython

# Éléments de Python

---

Retour à la ligne = fin d'instruction

- pas de ;

Indentation = fin de bloc

- pas de { ... }

# Premiers programmes

---

```
#!/usr/bin/python
```

```
print "Hello"
```

```
#!/usr/bin/python
```

```
'''
```

```
@author: oflauzac
```

```
'''
```

```
if __name__ == '__main__':
```

```
    i = 0
```

```
    while i < 10:
```

```
        print "Bonjour"
```

```
        i+=1
```

```
    for caractere in "salut les amis":
```

```
        print(caractere)
```

# Les données

---

## Numériques

```
#!/usr/bin/python
i = 12
j = 13
k = i + j
print "valeur" , i , " + " , j , " = " , k
```

## Chaînes

```
#!/usr/bin/python
# -*- coding: utf8 -*-

myStr = "salut les amis"
print "myStr[0] " , myStr[0]
print "myStr[1:5] " , myStr[1:5]
print "taille de la chaîne " , len(myStr)
```

# Les listes

---

Série d'éléments

Déclaré avec des []

Déclaration avec des ()

- tuples non modifiables

```
#!/usr/bin/python
# -*- coding: utf8 -*-
```

```
def affList(l):
    for elem in l:
        print elem
```

```
myList =[1,2,"salut","bonjour"]
affList(myList)
print "premier élément", myList[0]
myList.append("un de plus")
affList(myList)
v = myList.pop()
affList(myList)
print len(myList)
```

# Retour sur le for

---

```
for v in range (5,9):  
    print v
```

```
for v in range (5,9):  
    print v  
else:  
    print "v est égal à 9"
```

# Les objets

---

```
#!/usr/bin/python
# -*- coding: utf8 -*-
'''
@author: oflauzac
'''

class Personne:
    'définition de la classe personne'
    nbPersonne = 0

    def __init__(self,nom,prenom):
        self.nom = nom
        self.prenom = prenom
        Personne.nbPersonne += 1

    def aff(self):
        print "nom ", self.nom , " prenom " , self.prenom , " nb créés ", Personne.nbPersonne

if __name__ == '__main__':
    p1 = Personne("Flauzac","Olivier")
    p1.aff()
    print p1.nbPersonne
    print p1.nom
    print p1.prenom
```



# Modificateur d'accès

---

En python tout est public par défaut

Possibilité de rendre des élément protégé ou privé

Utilisation de `_` en préfixe du nom

- un `_` : élément protégé
- deux `_` : élément privé

# Architecture

---

Possibilité de donner une architecture au projet

- séparation des classes du programme
- inclusion des éléments (classes voulues)

```
from fichier import classe
```

# Modules

---

Éléments d'extension

Similaires au packages Java

Définition de types et des opérations associées

Possibilité de modules utilisateurs

# Module HTTP

---

```
#!/usr/bin/python
'''
@author: oflauzac
'''

import httplib
if __name__ == '__main__':
    conn = httplib.HTTPConnection("flauzac.eu")
    conn.request("GET", "/")
    r1 = conn.getresponse()
    print r1.status, r1.reason
    data = r1.read()
    print data
```

# Client TCP

---

```
#!/usr/bin/python
'''
@author: oflauzac
'''
import socket
if __name__ == '__main__':
    TCP_IP = '127.0.0.1'
    TCP_PORT = 5005
    BUFFER_SIZE = 1024
    MESSAGE = " Salut les amis!"

    s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    s.connect((TCP_IP, TCP_PORT))
    s.send(MESSAGE)
    data = s.recv(BUFFER_SIZE)
    s.close()

    print "received data:", data
```

# Serveur TCP

---

```
#!/usr/bin/python
'''
@author: oflauzac
'''
import socket

if __name__ == '__main__':
    TCP_IP = '127.0.0.1'
    TCP_PORT = 5005
    BUFFER_SIZE = 1024

    s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    s.bind((TCP_IP, TCP_PORT))
    s.listen(1)

    conn, addr = s.accept()
    print 'Connection address:', addr
    data = conn.recv(BUFFER_SIZE)
    print "received data:", data
    conn.send(data)
    conn.close()
```

# Client UDP

---

```
#!/usr/bin/python
'''
@author: oflauzac
'''

import socket

if __name__ == '__main__':
    UDP_IP = "127.0.0.1"
    UDP_PORT = 5005
    MESSAGE = "Hello, World!"

    print "UDP target IP:", UDP_IP
    print "UDP target port:", UDP_PORT
    print "message:", MESSAGE

    sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
    sock.sendto(MESSAGE, (UDP_IP, UDP_PORT))
```

# Serveur UDP

---

```
#!/usr/bin/python
'''
@author: oflauzac
'''
import socket
if __name__ == '__main__':
    UDP_IP = "127.0.0.1"
    UDP_PORT = 5005

    sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM) # UDP
    sock.bind((UDP_IP, UDP_PORT))
    while True:
        data, addr = sock.recvfrom(1024) # buffer size is 1024 bytes
        print "received message:", data
```



# Threads

---

```
#!/usr/bin/python
# -*- coding: utf8 -*-
import thread
import time

def print_time( threadName, delay):
    count = 0
    while count < 5:
        time.sleep(delay)
        count += 1
        print "%s: %s" % ( threadName, time.ctime(time.time()) )

if __name__ == '__main__':
    try:
        t1=thread.start_new_thread( print_time, ("Thread-1", 2, ) )
        t2=thread.start_new_thread( print_time, ("Thread-2", 4, ) )
    except:
        print "Error: unable to start thread"
    while 1:
        pass
```