

Langages de scripts

Frédéric Goualard

Laboratoire des Sciences du Numérique de Nantes, UMR CNRS 6004
Office #112-11

Le langage Python

- ▶ Langage *interprété* inventé par Guido van Rossum (1991)
- ▶ Priorité à la lisibilité du code (indentation pour les blocs)

```
for i in range(10):  
    print("Itération ",i)
```

- ▶ Deux branches :
 - ▶ Python 2.x (fin de vie : 2020)
 - ▶ **Python 3.x**
- ▶ Multiparadigme (fonctionnel, POO, ...)
- ▶ Typage dynamique

```
x = 2  
x = "Hello"  
x = int("3")
```
- ▶ Nombreuses bibliothèques (**Numpy**, **SciPy**, **Matplotlib**, **BioPython**, ...)
- ▶ Grande facilité pour encapsuler du code C/C++

Trois possibilités d'utilisation

▶ Appeler l'interprète **python** sur du code

```
% python3 -c 'age=input("age? "); print("Vous avez {}".format(age))'  
age? 22  
Vous avez 22 ans
```

▶ Coder directement dans l'interprète

```
% python3  
Python 3.5.2 (default, Nov 23 2017, 16:37:01)  
[GCC 5.4.0 20160609] on linux  
Type "help", "copyright", "credits" or "license" for more information.  
>>> for fruit in ["pomme", "poire"]:  
...     print(fruit)  
...  
pomme  
poire
```

▶ Écrire le code dans un fichier **“.py”** et l'exécuter en ligne de commande

Trois possibilités d'utilisation

- ▶ Appeler l'interprète **python** sur du code

```
% python3 -c 'age=input("age? "); print("Vous avez {}".format(age))'
age? 22
Vous avez 22 ans
```

- ▶ Coder directement dans l'interprète

```
% python3
Python 3.5.2 (default, Nov 23 2015, 15:34:58) on linux64
[GCC 5.4.0 20160609] on linux64
Type "help", "copyright", "credits() or help()"
>>> for fruit in ["pomme", "poire"]:
...     print(fruit)
...
pomme
poire
```

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

```
age = int(input("Quel est votre âge ? "))
if age >= 18:
    print("Vous êtes majeur(e)")
else:
    print("Vous êtes mineur(e)")
```

- ▶ Écrire le code dans un fichier **“.py”** et l'exécuter en ligne de commande

Trois possibilités d'utilisation

- ▶ Appeler l'interprète **python** sur du code

```
% python3 -c 'age=input("age? "); print("Vous avez {}".format(age))'
age? 22
Vous avez 22 ans
```

- ▶ Coder directement dans l'interprète

```
% python3
Python 3.5.2 (default, Nov 23 2015, 15:34:58) on linux64
Type "help", "copyright", "credits() or "help()" to get more help
>>> for fruit in ["pomme", "poire"]:
...     print(fruit)
...
pomme
poire
```

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

```
age = int(input("Quel est votre âge ? "))
if age >= 18:
    print("Vous êtes majeur(e)")
else:
    print("Vous êtes mineur(e)")
```

- ▶ Écrire le code dans un fichier **“.py”** et l'exécuter en ligne de commande

```
% python3 mineur.py
Quel est votre âge ? 17
Vous êtes mineur(e)
```

```
% ./mineur.py
Quel est votre âge ? 17
Vous êtes mineur(e)
```



Structures de contrôle (1)

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

import sys

if len(sys.argv) < 3:
    print("shout.py ---\nUsage: shout.py <what> <ntimes>", file=sys.stderr)
    exit(1)

for i in range(int(sys.argv[2])):
    print("We want {}".format(sys.argv[1]))

% ./shout.py spam 4
We want spam!
We want spam!
We want spam!
We want spam!
```



Structures de contrôle (1)

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

import sys

if len(sys.argv) < 3:
    print("shout.py ---\nUsage: shout.py <what> <ntimes>", file=sys.stderr)
    exit(1)

for i in range(int(sys.argv[2])):
    print("We want {}".format(sys.argv[1]))

% ./shout.py spam a
Traceback (most recent call last):
  File "./shout.py", line 10, in <module>
    for i in range(int(sys.argv[2])):
ValueError: invalid literal for int() with base 10: 'a'
```




Structures de contrôle (1)

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

import sys

if len(sys.argv) < 3:
    print("shout.py ---\nUsage: shout.py <what> <ntimes>", file=sys.stderr)
    exit(1)

try:
    for i in range(int(sys.argv[2])):
        print("We want {}".format(sys.argv[1]))
except ValueError:
    print("Erreur: 2e paramètre non numérique ({}).format(sys.argv[2]))
    exit(2)

% ./shout.py spam a
Erreur: deuxième paramètre non numérique (a)
```

Réalisation d'un filtre sommant les 4 premiers nombres donnés sur l'entrée standard ou dans un fichier passé en paramètre.

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

import fileinput

nnums = 0
accu = 0.0
with fileinput.input() as f:
    while nnums < 4:
        try:
            accu += float(f.readline())
            nnums += 1
        except ValueError:
            pass
print(accu)
```

```
% cat afiltrer.txt | filtre.py
124.10000000000001
```

```
% ./filtre.py afiltrer.txt
124.10000000000001
```

► Chaînes de caractères (Unicode par défaut avec Python 3)

```
>>> print("å lære {} er min glede!".format("python"))
å lære python er min glede!
>>> s="We want"+" spam!"; s
'We want spam!'
>>> s[1:4]
'e w'
```

► Listes

► Tuples

► Dictionnaires

► Ensembles

- ▶ Chaînes de caractères (Unicode par défaut avec Python 3)

- ▶ Listes

```
>>> l=["a",34]+["pomme",[45,6]];l
['a', 34, 'pomme', [45, 6]]
>>> len(l); l.reverse(); l
4
[[45, 6], 'pomme', 34, 'a']
>>> l.append("lapin"); l
[[45, 6], 'pomme', 34, 'a', 'lapin']
```

- ▶ Tuples

- ▶ Dictionnaires

- ▶ Ensembles

- ▶ Chaînes de caractères (Unicode par défaut avec Python 3)
- ▶ Listes
- ▶ Tuples

```
>>> t=(12,34,56); t; t[2]
```

```
(12, 34, 56)
```

```
56
```

```
>>> t[1]=34
```

```
Traceback (most recent call last):
```

```
  File "<stdin>", line 1, in <module>
```

```
TypeError: 'tuple' object does not support item assignment
```

- ▶ Dictionnaires
- ▶ Ensembles

- ▶ Chaînes de caractères (Unicode par défaut avec Python 3)
- ▶ Listes
- ▶ Tuples
- ▶ Dictionnaires

```
>>> d={"janvier":31, "python":"good", 12:34}
>>> d[12]; d["python"]
34
'good'
>>> del d["janvier"]; d
{12: 34, 'python': 'good'}
```

- ▶ Ensembles

- ▶ Chaînes de caractères (Unicode par défaut avec Python 3)
- ▶ Listes
- ▶ Tuples
- ▶ Dictionnaires
- ▶ Ensembles

```
>>> s={"jean", "chloé", "anatole"}
>>> "jean" in s
True
>>> s |= {"chloé", "émilie"}; s
{'anatole', 'jean', 'chloé', 'émilie'}
```

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

```
# Fichier powers.py
```

```
def square(x):  
    return x*x
```

```
def cube(x):  
    return square(x)*x
```

```
if __name__ == "__main__":  
    print(square(2))  
    print(cube(2))
```

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

```
import powers  
from powers import cube
```

```
print(powers.square(3))  
print(cube(4))
```


- ▶ Trinket: an hour of Python
- ▶ The Python Tutorial
- ▶ The Python Language Reference

Fin du cours