

Construire un programme en Python

Alexandre Roulois (Université Paris Cité, LLF, CNRS)

Table of contents

Squelette d'un programme	1
Le préambule	2
Les modules	2
Les fonctions utilisateur	2
La procédure principale	3
Exemple de programme	3
Exécuter un script Python	4

Python est un langage informatique qui permet d'écrire des scripts en *.py* exécutables depuis une ligne de commande. Ils répondent à quelques exigences formelles que nous parcourons dans cette introduction.

Squelette d'un programme

Un programme classique en Python est constitué de quatre parties :

- le préambule comprenant le *shebang* et l'encodage
- la liste des modules à importer pour le bon fonctionnement du programme
- la déclaration des fonctions utilisateur
- la procédure principale

```
#!/usr/bin/env python
#-*- coding: utf-8 -*-

#
# Load modules
#

#
```

```

# User functions
#

#
# Main procedure
#
if __name__ == "__main__":

    #
    # program
    #

    # a single instruction
    print('Hello World')

```

Le préambule

Constitué de deux lignes :

- *shebang* (`#!` + chemin vers l'exécutable) précise au système qu'il n'est pas en présence d'un fichier binaire mais bien d'un script ;
- déclaration de l'encodage.

```

#!/usr/bin/env python
#-*- coding: utf-8 -*-

```

Les modules

Le langage Python est livré avec des utilitaires de base pour répondre aux problèmes courants. Lorsque ceux-ci ne suffisent plus, il est possible d'étendre le langage en ajoutant des *bibliothèques* (ou *modules*). Ces bibliothèques sont de trois types différents :

- soit elles sont incluses dans le langage mais non activées ;
- soit elles sont définies par l'utilisateur ;
- soit elles sont extérieures au langage et doivent être installées au préalable.

Les fonctions utilisateur

L'utilisateur programme souvent ses propres fonctions pour des besoins spécifiques ou par souci d'optimisation. Soit ces fonctions font partie d'une bibliothèque à part (à inclure dans la section

supérieure avec les modules), soit elles sont déclarées en tête du script, avant la procédure principale.

La procédure principale

Il s'agit du corps du programme. La procédure principale mobilise les ressources intégrées dans l'en-tête du script pour lancer des instructions et afficher un résultat. Notre programme de démonstration est simplement chargé d'afficher un message de bienvenue.

```
if __name__ == "__main__":
    #
    # program
    #

    # a single instruction
    print('Hello World')
```

Dès lors que le programme prend de l'ampleur, il devient intéressant de protéger son script d'effets de bord en encapsulant le code de la procédure principale dans une fonction.

```
# procedure 'main'
def main():
    """A welcoming message"""
    print("Hello World!")

if __name__ == "__main__":

    # call to 'main' procedure
    main()
```

Exemple de programme

```
#!/usr/bin/env python
#-*- coding: utf-8 -*-

"""
    This script is a minimal example that uses the prerequisites
    expected of any program written in Python.
"""
```

```

import math

#
# User functions
#

def to_square_root(x):
    """Get the square root of a given number."""

    return math.sqrt(x)

def main():

    nb = input("Saisissez un nombre :\n")
    nb = int(nb)

    sq = to_square_root(nb)

    print(f"La racine carrée de {nb} est {sq}")

#
# Main procedure
#
if __name__ == "__main__":

    # call to 'main'
    main()

```

Exécuter un script Python

Pour exécuter un script Python, il suffit d'appeler l'utilitaire `python` depuis une interface en ligne de commande (ILC) en lui indiquant le chemin pour l'atteindre :

```
$ python path/to/script.py
```