Traitement de corpus

Cours n°2 - Les bases de Python

Hee-Soo Choi

UFR Sciences Humaines et Sociales
Université de Lorraine

13 octobre 2021

Table de matières

Python

Python est un langage de programmation :

- Version 3.10
- Typage dynamique
- Syntaxe épurée
- Interface graphique
- Portable (Windows/MacOS/Linux)

Exemple d'utilisation

Serveurs Youtube, serveurs Instagram, Dropbox, etc.

Téléchargement et installation

À télécharger ici :

https://www.python.org/downloads/windows/

Pycharm

À télécharger ici :

https://www.jetbrains.com/fr-fr/pycharm/download/section=windows

Interpréteur vs. Scripts

Interpréteur

Définition : interpréteur

Programme permettant d'entrer et d'exécuter des commandes une à une.

Exemple avec l'interpréteur

```
Python 3.8.10 (default, Sep 28 2021, 16:10:42)
[GCC 9.3.0] on linux
Type "help", "copyright", "credits" or "license"
for more information.
>>>
7
>>>
```

Script

Définition : script

Fichier regroupant un ensemble d'instructions et pouvant être exécuté

Exemple de script : `main.py`

```
#!/bin/env python3

a = 3
b = 4
c = a + b

print(c)
```

Premier script

Exemple de script

```
#!/bin/env python3

# Cette ligne est un commentaire, elle n'a pas d'effet

prenom = "Hee-Soo"
nom = "Choi"
age = 25

print(f"Je suis {prenom} {nom}, et j'ai {age} ans")
```

Au moment de l'exécution, le script affiche la ligne suivante :

```
Je suis Hee-Soo Choi, et j'ai 25 ans
```

Opérateurs arithmétiques

Les opérateurs arithmétiques en Python :

Nom	**Opérateur**
Addition	+
Soustraction	-
Division	/
Multiplication	*
Modulo	%
Puissance	**

Nom	**Opérateur**
Égalité	==
Différence	!=
Strictement supérieur	>
Supérieur ou égal	>=
Strictement inférieur	<
Inférieur ou égal	<=

Opérateurs logiques

Les opérateurs logiques en Python :

Nom	Opérateur
Ou	`or`
Et	`and`
Non	`not`

Exemple

```
if c1 == c2 and c2 == c3:
    return True
else:
    return False
```

Structures de contrôle : la condition

Bloc `SINON` / `SI`

Écriture

Entrée : données d'entrée
Sortie : résultat souhaité
si condition 1 **alors**
    Instruction 1
sinon si condition 2 **alors**
    Instruction 2
sinon
    Instruction 3

Exemple en python

```
if n < 0:
  print(f'Le nombre {n} est négatif')
elif n > 0:
  print(f'Le nombre {n} est positif')
else:
  print(f'Le nombre est nul')
```

Structures de contrôle : les boucles

Écriture

Entrée : données d'entrée
Sortie : résultat souhaité
tant que condition **faire**
    Instruction 1

Exemple en python

```
i = 0

while i < 10:
  print(i)
  i = i + 1
```

Structures de contrôle : les boucles

Écriture

Entrée : données d'entrée
Sortie : résultat souhaité
pour chaque élément **dans** E **faire**
    Instruction 1

Exemple en python

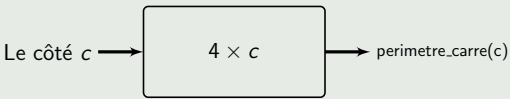
```
for lettre in mot:
  print(lettre.upper())
```

Les fonctions

Définition : fonction

Ensemble d'instructions qui renvoient une valeur à la fin de leur traitement

Exemple : fonction perimetre_carre(c)



`perimetre_carre(3)` vaut 12
`perimetre_carre(1)` vaut 4

Les fonctions, en python

Définition de la fonction :

```
def perimetre_carre(c)
  resultat = 4*c

  return resultat
```

Appel de la fonction :

```
p = perimetre_carre(2.3)
```

Terminologie

Le résultat d'une fonction est appelée **valeur de retour**.
Les entrées d'une fonction sont appelés **paramètres** ou **arguments**.