

RER- Python : les conditions & les booléens

Contexte :

Comprendre les structures de contrôle conditionnel dans l'exécution d'un code.
Comprendre les booléens et les manipuler.

Mots clés :

- **Bloc de codes** : en Python, c'est un groupe d'instructions délimité par une instruction unique suivie de ":"; on retrouve par exemple "if", "elif", "else", "for", "while", "try", "except", "def", "class", etc.
- **Structures de contrôle conditionnel** : un bloc de code qui s'exécute lorsqu'une ou plusieurs conditions déterminées sont vérifiées. Ce sont des structures qui permettent de contrôler l'ordonnancement des actions.
- **Alternatives** : c'est le bloc de conditions qui est exécuté lorsque la première condition n'est pas vérifiée. Il y a plusieurs façons de faire des alternatives.
- **Condition** : c'est une valeur booléenne qui répond à une question. On peut utiliser des variables booléennes ou des opérateurs de comparaison pour former une condition.
- **Tests à plusieurs cas "test imbriqué"** : ensemble de conditions qui s'enchainent si la valeur renvoyée précédente est "False". C'est l'équivalent des structures alternatives.
- **Tests multiples** : c'est le fait de tester plusieurs conditions en même temps en utilisant des opérateurs booléens : "or", "and", "^".
- **Tests de valeur sur des floats** : dans les bonnes pratiques on évite de faire des tests de valeur stricte. On utilise des intervalles avec une certaine précision.

- **Variables booléennes** : ce sont des variables qui retournent que deux valeurs, soit "True" ou "False" et "1" et "0".

- **Opérateurs booléens** :

- AND
- OR
- ^ (ou exclusif)
- NOT

X	Y	and	or	^ (xor/ ou exclusif)	not(x and y)	not(x or y)	not (x ^ y)
0	0	0	0	0	1	1	1
0	1	0	1	1	1	0	0
1	0	0	1	1	1	0	0
1	1	1	1	0	0	0	1

- **Opérateurs de comparaison** : opérateur qui compare la valeur de deux variables et qui renvoie une valeur booléenne.
 - < strictement inférieur
 - > strictement supérieur
 - <= inférieur ou égal
 - >= supérieur ou égal
 - == égal
 - != différent
 - X is Y : X et Y représentent le même objet.
 - X is not Y : X et Y ne représentent pas le même objet

Problématique(s) :

- Comment construire des tests conditionnels à l'aide d'opérateurs booléens et de conditions dans Python ?
- Comment modéliser les problèmes en termes de tests et d'alternatives ?

Hypothèses :

1. Les blocs de conditions peuvent en inclure d'autres. Solenn. **VRAI**
2. Il peut y avoir un état intermédiaire entre le vrai et le faux. Seydou. **FAUX**
3. Les blocs conditionnels peuvent aider à optimiser l'exécution du programme en mémoire. Etienne **FAUX**
4. Une succession de blocs conditionnels s'appelle une boucle. Loïc **FAUX**
5. Le codage des structures de contrôle conditionnel dans le langage Python est très spécifique. Adeline **FAUX**
6. Il n'y a pas de limite au nombre de blocs conditionnels successifs. Briand **VRAI**
7. Les booléens ont été inventés par Mr Boole. Adrien **VRAI**
8. Les structures conditionnelles permettent de tester un grand nombre de conditions. Aude **VRAI**
9. Le langage Python permet de mixer plusieurs principes algorithmiques dans un même bloc de code. Adeline **VRAI**
10. En Python chaque structure conditionnelle ne peut avoir qu'une seule condition à tester. Axel **FAUX**
11. Le *True* et le *False* existent sans les conditions. Jean Paul **VRAI**
12. Il y a des meilleures façons de tester une condition qu'un "if". Nicolas **FAUX**
13. Si les conditions ne sont pas remplies on obtient un message d'erreur. Tetyana **FAUX**
14. On peut construire un bloc "if" sans conditions. Solenn **VRAI**

Plan d'action :

1. Exploration des ressources
2. Définition des mots clefs
3. Réponses aux hypothèses
4. Utilisation du Jupyter Notebook pour tester toutes les notions abordées
5. Faire les exercices dans le fichier : Booléens et les conditions.ipynb
6. Livrable