Type de document	Cours et exercices	Classe	1re	Durée	2h	Date	06/10/2022
Thème et contenu(s)	Langages et programmation – Représentation des données						
Capacités attendues	Bases de Python : création de scripts simples, fonctions conditionnelles						
Prérequis	Connaître les 4 types de base : Boolean, Integer, Float, String						
Description	Cours et exercices, idéalement en salle informatique devant les machines						

On travaillera désormais exclusivement avec des fichiers **.py**, mais plus avec la console

Jusqu'à présent, chaque ligne de code de notre programme était exécutée **sans exception**. Désormais nous verrons que, très souvent, toutes les lignes de code d'un programme ne s'exécutent pas à chaque fois. Le programme « prend des chemins » différents en fonction de certaines **conditions**. Cette notion est commune à la plupart des langages de programmation, mais sa syntaxe diffère légèrement.

I) Expressions conditionnelles

Une **expression conditionnelle** est une expression dont l'**évaluation** donne un résultat **booléen**, c'est-à-dire True ou False.
On utilise les **expressions conditionnelles** dans les **branchements logiques** (ou **structures conditionnelles**) : certaines parties du code ne sont exécutées que si une condition est vérifiée.

II) Principe

Pour qu'une expression soit conditionnelle, il faut qu'elle emploie soit des **comparateurs** (>,<,<=,>=,==, !=) soit des **opérateurs booléens** (not, and, or, ^), ou n'importe quelle combinaison des deux. L'idée est la suivante :

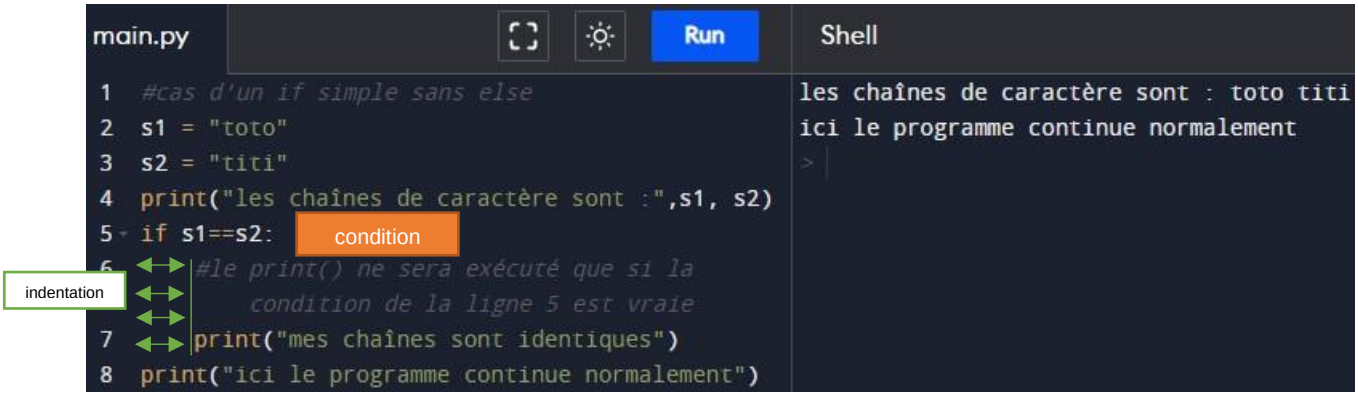
Si (expression) Alors
    Instruction**1**
    Instruction**2**
    …
    Instruction**n**

Les instructions ne s'exécutent que si l'expression est évaluée à **True**.
Optionnellement, cette structure peut être complétée par un « sinon » :
Sinon
    Instruction**1**
    Instruction**2**
    …
    Instruction**n**

dont l'exécution n'aura lieu que si l'évaluation de l'expression donne **False**.

III) Syntaxe Python

On introduit une structure conditionnelle en Python par le mot-clef `if` suivi d'une **expression conditionnelle**, elle-même suivie de « : ». Tout ce qui « fait partie du if », c'est-à-dire qui sera exécuté si la condition est vraie, doit être **indenté**. L'indentation consiste à écrire les instructions appartenant au `if` avec un décalage d'une tabulation (ou alors 4 espaces). Optionnellement, on peut ensuite mettre un `else`, au même niveau que le `if`, qui contient des instructions exécutées si l'expression logique du `if` est évaluée à False. Voici un exemple :

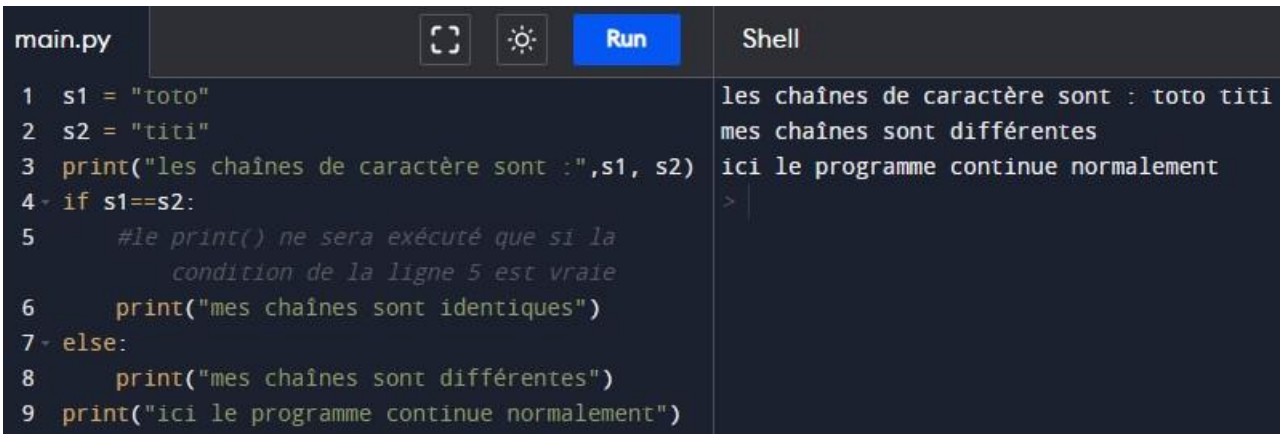


Les lignes 2,3,4,5,8 sont exécutées quoi qu'il arrive. Par contre, l'exécution de la ligne 7 dépend de ce qui a été évalué dans le `if` de la ligne 5 : dans le cas présent, elle n'est pas exécutée (comme vous pouvez le voir sur la console) car l'expression logique est évaluée à False (s1 étant différente de s2) donc la condition n'est pas remplie.
Vous devez comprendre ceci : la ligne 7 et la ligne 8 ne sont pas situées au même niveau : la ligne 7 dépend du `if` et son exécution n'est pas certaine (elle dépend de la condition), la ligne 8 est complètement indépendante de la condition, elle s'exécute toujours.

Exemple dans le cas où les variables sont identiques :

Enfin, dans l'exemple qui suit, on a rajouté un `else`. Il ne s'exécute que si l'évaluation de la condition est False. Donc quel que soit le cas de figure, soit la ligne 6 soit la ligne 8 ne sera pas exécutée.

En aucun cas, les instructions du `if` et celles du `else` ne peuvent être exécutées simultanément : cela signifierait que l'évaluation de la condition donne à la fois True et False ce qui est impossible



IV) Erreurs fréquentes

`if` et/ou `else` mal orthographiés, il manque les « : », indentation non respectée, l'expression n'est pas conditionnelle (par exemple, 2+3*2 n'est pas une expression conditionnelle)