

TD-TP n° 2

Cette feuille d'exercices suit le cours 2 et reprend aussi des notions du cours 1.

Selon les indications données, les exercices sont à traiter

- en mode TD (papier, crayon),
- en mode TP (Thonny, jupyter-notebook)

Exercice 1. (TD-TP)

Dans cet exercice, vous avez plusieurs expressions booléennes. Vous devez les évaluer. D'abord "à la main", puis en contrôlant dans la console de Thonny. Si une erreur vous échappe, n'hésitez pas à demander pour comprendre.

False or True	False or (True and False)	$1 < 2$ and $2 > 3$
False or False and not False	False and True or False and not True	
$3 // 2 == 1$ and $3 \% 2 != 1$	"ab" > "acd" or "12" < "3"	"3" == "1+2"
$23 <= 11 * 3$	"23" <= "11 * 3"	"23" <= 11 * "3"

Exercice 2. (essentiellement TP) or, argent, bronze et nickel

Dans un pays lointain, le système monétaire en cours est basé sur 4 types de pièces :

- les pièces de nickel qui sont la base du système,
- les pièces de bronze : 1 pièce de bronze vaut 12 pièces de nickel,
- les pièces d'argent : 1 pièce d'argent vaut 12 pièces de bronze.
- les pièces d'or : 1 pièce d'or vaut 8 pièces d'argent

Une variable **P** contient un nombre (supposé positif) de pièces de nickel. On cherche à exprimer cette quantité en utilisant au maximum les pièces d'or, puis d'argent, puis de bronze.

Quatre autres variables, nommées **N**, **B**, **A** et **O** contiendront à terme les valeurs correspondant à ce nombre total de pièces de nickel.

Par exemple, si **P** vaut 15807 alors **O** vaudra 13, **A** vaudra 5, **B** vaudra 9 et **N** vaudra 3.

1. Écrivez sur papier le calcul qui permet de vérifier que si $O = 13$, $A = 5$, $B = 9$ et $N = 3$ alors cela correspond bien à $P = 15807$.
2. Écrivez sur papier des instructions de type **affectation** qui permettent de calculer les valeurs des 4 variables **O**, **A**, **B** et **N** connaissant la valeur de la variable **P**.

3. Écrivez avec Thonny le programme correspondant (`monnaie1.py`), qui affecte à **P** une certaine valeur, et qui calcule les valeurs des 4 autres variables, puis les affiche. Testez le programme.
4. Donnez une nouvelle version `monnaie2.py` qui permet une interaction avec l'utilisateur, à qui le programme demande le nombre initial de pièces de nickel avant d'annoncer les résultats précédents, adaptés à l'entrée donnée par l'utilisateur.

Rappel : la fonction `input` renvoie toujours une chaîne.

Exercice 3. (TP) Onzième croissant gratuit

À programmer en Python. Vous choisirez d'ouvrir le notebook en ligne sur **basthon.fr** pour coder et tester votre code.

Le prix unitaire d'un croissant est 1.25 euro. La boulangerie fait une promotion : un croissant gratuit par lot de onze. Autrement dit, si vous achetez 10 croissants, le onzième croissant vous est offert.

Écrivez un programme qui demande à l'utilisateur le nombre de croissants achetés et affiche le prix total à payer. Attention, il ne s'agit pas d'une réduction en pourcentage, mais d'un croissant gratuit par lot de 11 (les 22e, 33e . . . sont aussi gratuits).

Voici un exemple d'exécution :

```
combien de croissants ? 27
- il y a 2 lots de 11 croissants
- en plus, il reste 5 croissants
=> prix à payer : 31.25 €
```

Exercice 4. (TD) mois de l'année

Les mois sont représentés par leur numéro (c.-à-d. 1 pour janvier, 2 pour février etc).

Dans cet exercice, le numéro du mois courant est représenté par la variable de nom **m**, qui est donc une variable entière, dont la valeur est entre 1 et 12.

Dans les questions suivantes, on vous demande d'écrire des expressions booléennes à partir de **m**. Par exemple, **m > 6 and m <= 9** est une expression booléenne.

1. Écrire une expression qui ne soit vraie (évaluée à **True**) que si l'on est en mars ou avril.
2. Écrire une expression qui ne soit vraie que si l'on est à un mois de 31 jours.
3. Écrire une expression qui ne soit vraie que si l'on n'est pas à un mois de 30 jours.
4. Écrire une expression qui ne soit vraie que si l'on est à un mois d'au plus 30 jours.

Exercice 5. (TD-TP) Durée maximale de 10 minutes et 30 secondes

On chronomètre une course d'endurance. On obtient un temps en minutes et en secondes. On note **m** le nombre de minutes et **s** le nombre de secondes.

Le programme qu'on doit écrire utilisera ces deux variables **m** et **s**.

1. Écrivez une expression pour tester si la durée est au plus de 10 minutes et 30 secondes. cette expression ne doit utiliser que des opérateurs arithmétiques, un unique comparateur et aucun opérateur booléen.
2. Écrivez une autre expression pour cela qui n'utilise aucun opérateur arithmétique (mais plusieurs opérateurs booléens).
3. Écrivez et testez les programmes qui correspondent aux deux questions. Le programme doit demander à l'utilisateur les valeurs de **m** et **s** et annoncer le résultat.

Exercice 6. (TD) Affichages conditionnels

On exécute le programme suivant. L'utilisateur doit saisir un nombre entier, qui est stocké dans une variable **n**.

```

1  n = int(input("entrez un entier n : "))
2
3  if n > 10 :
4      print("A", end = " ")
5      if n < 20 :
6          print("B", end = " ")
7      else :
8          print("C", end = " ")
9  else :
10     print("D", end = " ")

```

1. Que va afficher le programme si on entre la valeur 15 pour **n** ?
2. Est-il possible de ne voir en sortie que l'affichage "B" ? Et "D" ?
3. Énumérez tous les affichages possibles et les valeurs de **n** qui les occasionnent.

Exercice 7. (TD-TP) Signe d'un entier

Écrivez un programme python qui demande à l'utilisateur la valeur d'un entier, puis affiche l'un des mots **POSITIF**, **NEGATIF** ou **NUL** selon la valeur entrée.

Exercice 8. (TD-TP) Années bissextiles

Une année est bissextile si son numéro est divisible par 400 ou si c'est un multiple de 4 mais pas de 100.

1. Dites si les années suivantes sont ou non bissextiles, et pourquoi :

1952	1957	1900	1998
2000	2004	2007	2008
2100	2104	2200	2400

2. Écrivez un programme python qui

— demande à l'utilisateur une année (stockée dans une variable),

- calcule les valeurs de 3 variables booléennes nommées **divisiblePar4**, **divisiblePar100** et **divisiblePar400** (les noms sont explicites),
 - calcule la valeur d'une variable booléenne **estBissextile** qui dit si oui ou non l'année est bissextile,
 - affiche le résultat à l'utilisateur sous la forme "l'année ... est bissextile" ou bien "l'année ... n'est pas bissextile".
3. Testez le programme avec les 12 valeurs d'année de la question 1.

Exercice 9. (TP) Comparer des dates

Le programme suivant demande à saisir les éléments (année, mois, jour) de deux dates.

```

1 # saisie de la date 1
2 annee1 = int(input("entrez l'année de la date 1 : "))
3 mois1 = int(input("entrez le mois de la date 1 : "))
4 jour1 = int(input("entrez le jour de la date 1 : "))
5
6 # saisie de la date 2
7 annee2 = int(input("entrez l'année de la date 2 : "))
8 mois2 = int(input("entrez le mois de la date 2 : "))
9 jour2 = int(input("entrez le jour de la date 2 : "))

```

L'objectif est de compléter le programme pour qu'après la saisie des deux dates, il affiche un message disant que la date 1 est antérieure, ou postérieure, ou enfin égale à la date 2.

Créez les variables booléennes et les instructions conditionnelles qui vous semblent judicieuses pour arriver à ce résultat.

Testez votre programme.

Exercice 10. (TP) Retour sur le TP précédent

Reprenez les deux derniers exercices du TP précédent de façon à interagir avec l'utilisateur qui fixera lui-même :

- le taux, le capital initial et le nombre d'années pour l'exercice 9,
- les paramètres de pousse de chaque arbuste pour l'exercice 10.

Exercice 11. (TD, TP) Triangles (im)possibles

Quand on se donne trois entiers (par exemple 5, 7 et 9), il n'est pas toujours possible de dessiner un triangle dont les côtés ont pour longueur ces entiers.

- Pour 5, 7 et 9 c'est possible ;
- Pour 5, 7, 13 ce n'est pas possible.

1. Trouvez une condition qui permet d'affirmer que 3 entiers a , b et c permettent de dessiner un "vrai" triangle ;
2. En niant cette condition, écrivez une condition qui dit qu'un tel triangle est impossible ;
3. A quelle condition est-on à la situation limite d'un triangle "plat" ?

4. Ecrivez un programme python qui demande 3 entiers à l'utilisateur, et qui dit si un triangle est alors possible ou non. Vous distinguerez le cas "plat".
5. Perfectionnez votre programme pour qu'il précise le cas échéant que le triangle est rectangle.