TP1 : Affectations, fonctions, booléens, branchements conditionnels

Instructions introduites dans ce cours : `help, help(),max,min,int.`
Voici quelques conseils avant de débuter votre premier TP.

- Au début vous passerez beaucoup de temps à corriger vos erreurs de syntaxes, qui en général proviennent de l'oubli d'une parenthèse ou d'un espacement placé au mauvais endroit. Soyez donc vigilant !
- Si vous avez un doute concernant l'utilisation d'une commande, n'hésitez pas à utiliser l'aide du logiciel :
  - ▷ soit en entrant `help` dans la console : vous verrez alors apparaître l'URL (l'*adresse web* si vous préférez) de l'aide en ligne, qui vous permettra de vous déplacer dans le menu et de rechercher le sujet sur lequel vous avez des doutes ;
  - ▷ soit en tapant `help(machin)` dans la console, si vous avez des doutes concernant l'utilisation de la primitive `machin` : vous verrez alors apparaître dans la console la partie de l'aide relative celle-ci.
- Habituez-vous à utiliser les raccourcis clavier au maximum pour aller vite en TP (un certain nombre de ces raccourcis sont mentionnés dans le cours).
- Enregistrez dès le début du TP votre travail avec un nom de la forme `VOTRENOM_TPX.py`.

Exercice TP1.1

On considère la fonction définie par $f(x,y) = \begin{cases} 1 & \text{si } x = y \\ 0 & \text{sinon.} \end{cases}$

1. Implémentez f, puis testez votre programme avec $x = 1$ et $y = 1$, puis $x = 1$ et $y = 2$.
2. Que se passe-t-il si vous appliquez f à $x =$`True` et $y =$`False` ? Qu'en pensez-vous ?

Exercice TP1.2

On considère le problème de l'interversion de deux variables : partant de deux variables `a,b` possédant chacune une valeur (par exemple `a=1` et `b=2`), on cherche à échanger les valeurs contenues par celles-ci (on veut obtenir `a=2` et `b=1`).

1. Entrer `a=1` et `b=2` dans la console. Entrer ensuite successivement les lignes de code `a=b` et `b=a`. Est-ce que cela permet d'intervertir les valeurs de `a` et `b` ? Pourquoi ?
2. Entrer de nouveau `a=1` et `b=2` dans la console. En utilisant une troisième variable `c`, intervertissez les valeurs de `a` et de `b` (on s'abstiendra d'entrer des lignes tels que `a=2` et `b=1`, sinon l'exercice n'a aucun intérêt !).

Exercice TP1.3

On souhaite implémenter la fonction définie par $f(a,b) = \begin{cases} (a,b) & \text{si } a > 0 \\ (a,0) & \text{si } a \leqslant 0. \end{cases}$. On considère pour cela les deux déclarations ci-dessous :

```
def f1(a,b):
    if a>0:
        x=a
        y=b
    else:
        x=a
    y=0
    return(x,y)
```

```
def f2(a,b):
    if a>0:
        x=a
        y=b
    else:
        x=a
        y=0
    return(x,y)
```

1. L'une des deux fonctions `f1`, `f2` est une implémentation de `f`. À votre avis, de laquelle s'agit-il ?
2. Vérifiez votre réponse à la question précédente en déclarant `f1` et `f2` dans l'éditeur de texte, puis en évaluant ces fonctions en $(a,b) = (1,2)$ et $(a,b) = (-1,2)$.
3. On considère celle des deux fonctions `f1`, `f2` qui n'est pas une implémentation de `f`. Pouvez-vous expliquer (en moins d'une ligne) ce que fait cette fonction ?

Exercice TP1.4

Étant donnés trois réels a, b, c, on note $P_{a,b,c}$ la fonction définie par $P_{a,b,c}(x) = ax^2 + bx + c$.

1. On se place dans le cas $a \neq 0$. Déclarer une fonction `nb_racines`, qui prend en entrée trois réels `a,b,c` tels que `a` contienne une valeur non nulle et qui renvoie :
   — `2` si le trinôme $P_{a,b,c}$ admet deux racines réelles ;
   — `1` si le trinôme $P_{a,b,c}$ admet une racine double réelle ;
   — `0` si le trinôme $P_{a,b,c}$ n'admet aucune racine réelle.

   Vérifier ensuite que le programme fonctionne, en considérant les trinômes $x^2 - 1$, $x^2 - 2x + 1$ et $x^2 + x + 1$.
2. On se place maintenant dans le cas général (c'est-à-dire que a peut être nul). Déclarer une fonction `nb_racines_general` qui possède la même spécification que la précédente, mais qui donne encore le bon résultat si $a = 0$ (on renverra -1 dans le cas où le polynôme possède une infinité de racines). Tester ensuite le programme avec les polynômes $x + 1$, 1 et 0.

Exercice TP1.5

1. Implémenter une fonction `maximum2`, qui prend en entrée deux réels x et y, puis renvoie en sortie le plus grand des deux.
2. Tester la fonction `maximum2` avec les couples $(1, 2)$, $(2, 1)$ et $(1, 1)$.
3. Implémenter une fonction `maximum3`, qui prend en entrée trois réels x, y et z, puis renvoie en sortie le plus grand des trois réels x, y, z.
4. Tester la fonction `maximum3` avec les triplets $(1, 2, 3)$, $(1, 3, 2)$, $(3, 1, 2)$ et $(1, 1, 2)$ pour vérifier que votre programme fonctionne correctement.
5. Regarder dans l'aide du logiciel à quoi servent les primitives `min` et `max`. À l'avenir on aura toujours le droit d'utiliser celles-ci.

Exercice TP1.6

Écrire trois fonctions qui calculent respectivement la négation d'une variable booléenne, le « et logique » puis le « ou logique » de deux variables booléennes, sans utiliser aucune des trois instructions `not,and,or`.

Exercice TP1.7

On appelle « partie entière », et l'on note $\lfloor\ \rfloor$, la fonction définie de la manière suivante : pour tout réel x, $\lfloor x \rfloor$ est le plus grand entier inférieur ou égal à x.
On a par exemple $\lfloor 2 \rfloor = 2$ et $\lfloor 1,23 \rfloor = 1$ (intuitivement il suffit, pour calculer la partie entière d'un nombre décimal positif, de supprimer les chiffres présents après la virgule).
On voit par exemple aisément que l'on a $\lfloor x \rfloor = x$ si et seulement si x est un entier.

1. La primitive `int` implémente la fonction partie entière pour les réels positifs. Allez chercher dans l'aide de Python la partie relative à `int` ; trouvez-vous les explications claires ?
2. Soit n un entier. À l'aide d'un stylo et d'une feuille de papier (mais sans utiliser Python), calculer $2\left\lfloor \dfrac{n}{2} \right\rfloor$ en distinguant les cas n pair et n impair.
3. En déduire une implémentation de la fonction pair définie sur $\mathbb{N}$ par :

$$\mathrm{pair}(n) = \begin{cases} 1 & \text{si } n \text{ est pair} \\ 0 & \text{sinon.} \end{cases}$$

   Testez enfin votre programme pour vérifier qu'il fonctionne.

Exercice TP1.8

Soit $n \in \mathbb{N}^*$. On appelle g_n la fonction définie sur $[0, 1[$ par $g_n(x) = k$ où k est l'unique entier tel que :

$$\frac{k}{n} \leqslant x < \frac{k+1}{n}.$$

1. À l'aide d'une feuille de papier et d'un stylo, tracer les graphes de g_1, g_2 et g_3. Vérifier que l'on obtient trois signaux en créneaux.
2. Implémenter la fonction h qui prend en entrée un réel `x` dans $[0, 1[$ et un entier `n`, puis renvoie $g_n(x)$.