

<p style="text-align: center;">Informatique en CPGE (2018-2019) TP 2 : calculs, instructions conditionnelles</p>
--

1 Données, variables et calculs

Exercice 1

On étudie le mouvement vertical d'une balle lancée en l'air. Supposons que la position verticale de la balle à l'instant t est donnée par $y(t) = v_0 t - \frac{1}{2} g t^2$, où v_0 est la vitesse initiale et g l'accélération due à la gravité.

Si $v_0 = 4$ m/s, $g = 9,81$ m/s² et nous voulons connaître la position de la balle à $t = 0,3$ s, il suffit d'écrire : `print(4*0.3-0.5*9.81*0.3**2)`.

Si nous souhaitons utiliser plusieurs fois cette formule, il est préférable d'avoir recours à des variables en choisissant des noms explicites et à la fonction `input` ; écrire un programme (nommé "tp2ex1.py") qui demande à l'utilisateur de saisir les valeurs de v_0 et t puis calcule et affiche la position y .

Exercice 2

Ecrire un programme (nommé "tp2ex2.py") qui calcule et affiche la somme et la moyenne de deux nombres.

Exercice 3

Ecrire un programme qui demande à l'utilisateur les coordonnées de deux points dans le plan puis affiche la distance entre ces deux points. La formule est $d = ((x_2 - x_1)^2 + (y_2 - y_1)^2)^{0,5}$.

Exercice 4

Ecrire un programme qui demande à l'utilisateur d'entrer son nom puis son prénom et affiche les initiales.

Exercice 5

En Python, l'échange de deux variables nommées a et b peut se programmer grâce à l'affectation multiple : il suffit d'écrire l'instruction `a, b = b, a`. Mais ce n'est pas le cas dans la plupart des langages de programmation.

Ecrire un algorithme permettant d'échanger les valeurs de deux variables a et b , sans utiliser l'affectation multiple mais en utilisant une troisième variable.

Comment faire, si a et b sont des nombres, sans utiliser de variable supplémentaire, et toujours sans utiliser l'instruction `a, b = b, a` ?

Exercice 6

Pour représenter un caractère dans la machine, on attribue un nombre à chaque caractère par exemple à l'aide du code ASCII binaire (American Standard Code for Information Interchange). Ce code utilise un octet par caractère, avec le premier bit toujours égal à 0, et permet donc de représenter $2^7 = 128$ caractères. Ce sont les caractères que l'on trouve sur les touches d'un clavier "qwerty" d'ordinateur : par exemple, on attribue les nombres de 97 à 122 aux 26 lettres de l'alphabet a, b, c, \dots, z ; les nombres de 48 à 57 aux dix chiffres $0, 1, 2, \dots, 9$; l'espace est codé par le nombre 32, le point par le nombre 46, etc. (voir une table complète sur le site www.table-ascii.com)

Attention à ne pas confondre un chiffre avec le caractère qui le représente. Les caractères du pavé numérique $0, 1, 2, 3, \dots$ sont codés en ASCII par les nombres 48, 49, 50, 51, ...

L'instruction `chr(97)` donne le caractère "a" et l'instruction `ord("a")` donne le nombre 97.

Décoder la suite de nombres : 98 114 97 118 111

Exercice 7

Un temps est donné en secondes.

Ecrire un programme qui convertit ce temps en jours, heures, minutes, secondes.

Par exemple si le temps est $t = 286534$ secondes, le programme affiche le message :

286534 secondes = 3 jours 7 heures 35 minutes 34 secondes.

2 Instructions conditionnelles

Exercice 8

Ecrire un programme qui résout dans \mathbb{R} , (de manière approchée), l'équation du second degré : $ax^2 + bx + c = 0$ où les coefficients réels a , b et c sont donnés. Le programme affiche le nombre de solutions et leurs valeurs éventuelles.

Exercice 9

Les instructions suivantes permettent de trouver le plus petit des trois nombres x , y et z :

```
# x, y et z sont des flottants
if x<y and x<z:
    print('x est le plus petit')
elif y<z:
    print('y est le plus petit')
else:
    print('z est le plus petit')
```

Ecrire des instructions qui permettent de trouver, parmi trois nombres entiers naturels a , b et c , le plus grand nombre impair. Si aucun des trois nombres n'est impair, le programme affiche un message en ce sens.