

TP1: bases de programmation en python

Le document de référence doit être apporté en TP. Les exercices 1 à 9, incontournables, doivent être parfaitement traités en séance de TP. Les exercices suivants peuvent être traités en fin de séance ou à la maison, pour s'entraîner. Tous ces exercices sont susceptibles d'être posés en DS.

Créer un dossier "TP1" dans votre répertoire personnel. Pour l'exercice 1, dans Pyzo, enregistrer un fichier nommé "exo1.py" dans le dossier "TP1". Pour l'exercice 2, dans Pyzo, enregistrer un fichier nommé "exo2.py" dans le dossier "TP1"..... Pour chaque exercice, le fichier python sera régulièrement sauvegardé et exécuté intégralement avec la commande "Run file as script" du menu Run (Ctrl+Shift+E). Les instructions d'affichage seront saisies sur le fichier python. Au fur et à mesure de l'avancée de l'exercice, les instructions devenant inutiles et gênantes pourront être désactivées en utilisant le caractère #.

Exercice 1. Recopier et faire exécuter la suite d'instructions suivante et examiner les affichages afin de s'assurer d'avoir compris le fonctionnement.

<pre>print(2 + 3) print(2 - 3) print(2 * 3) print(2 * *3)</pre>	<pre>print(2 + 3, 2 - 3, 2 * 3, 2 * *3) print(28//5, 28%5) print(2.5+3.5, 2.5-3.5, 2.5*3.5, 2.5/3.5, 2.5**3.5)</pre>
---	--

Exercice 2. Recopier et faire exécuter la suite d'instructions suivante et examiner les affichages afin de s'assurer d'avoir compris le fonctionnement.

<pre>print(2 == 3) print(2! = 3) print(2 < 3) print(2 <= 3)</pre>	<pre>print(2 == 2, 2! = 2) print(3 <= 2 and 3 <= 5) print(3 <= 2 or 3 <= 5) print(not(3 <= 2))</pre>
---	---

Exercice 3. Recopier et faire exécuter la suite d'instructions suivante et examiner les affichages afin de s'assurer d'avoir compris le fonctionnement.

<pre>a = 0 print(a) a = 5 print(a) print(a + 2) b = a + 2 print(b)</pre>	<pre>c = 3 print(c) print(a, c) print(a < c) print(a + c) d = a + c print(d)</pre>
--	---

Ecrivons une fonction nommée "car" ayant pour argument x (nombre) et retournant x^2 .

```
def car(x) :
    return(x**2)
```

Exercice 4. Ecrire une fonction nommée "Fermat" ayant pour argument n (entier naturel) et retournant $2^{(2^n)} + 1$. Faire afficher Fermat(0), Fermat(1), Fermat(2).

Ecrivons une fonction nommée "inv" ayant pour argument x (nombre non nul) et retournant $\frac{1}{x}$. On contrôlera la précondition $x \neq 0$.

```
def inv(x) :  
    assert (x != 0)  
    return(1/x)
```

Exercice 5. Ecrire une fonction nommée "rac" ayant pour argument x (un nombre positif) et retournant $x^{\frac{1}{2}}$. On contrôlera la précondition $x \geq 0$. Faire afficher `rac(2)` et `rac(-1)`.

Ecrivons une fonction "moy2" ayant pour arguments a et b (nombres) et retournant la moyenne de a et b :

```
def moy2(a,b) :  
    return((a+b)/2)
```

Exercice 6. Un professeur a organisé 4 devoirs de coefficients respectifs 2.5, 3, 1, 0.5. La somme des coefficients vaut donc 7.0. Ecrire une fonction "moypond" ayant pour arguments $n1, n2, n3, n4$, les notes d'un étudiant aux 4 quatrers devoirs, et retournant la moyenne pondérée (avec coefficients) de l'étudiant. Faire afficher `moypond(12, 15, 9, 13)`.

Un client d'une banque fait ouvrir un livret A en y déposant une somme d'argent, puis, pendant deux ans, le client n'effectue aucune opération sur son livret et le taux d'intérêt annuel n'est pas modifié. Ecrivons une fonction nommée "bilan" ayant pour arguments t (taux d'intérêt annuel) et $s0$ (somme initialement déposée par la client) et retournant le couple $(i, s2)$ où i est le montant des interet percus au bout de 2 ans et $s2$ est le solde du compte au bout de 2 ans.

```
def bilan(t, s0) :  
    # on introduit i1 : intérêts obtenus pour la première année  
    i1 = s0 * t  
    # on introduit s1 : solde du compte à l'issue de la première année  
    s1 = s0 + i1  
    # on introduit i2 : intérêts obtenus pour la deuxième année  
    i2 = s1 * t  
    s2 = s1 + i2  
    i = i1 + i2  
    return(i, s2)
```

Exercice 7. Une caissière dispose d'une infinité de pièces de 1, de 5 et de 20 . Lorsqu'elle rend la monnaie, elle rend un maximum de pièces de 20, puis un maximum de pièces de 5, puis des pièces de 1. Ecrire une fonction nommée "monnaie" ayant pour argument s (somme à rendre par la caissière) et retournant $(n1, n5, n20)$ où $n1$ est le nombre de pièces de 1 rendues, $n5$ est le nombre de pièces de 5 rendues et $n20$ est le nombre de pièces de 20 rendues. Faire afficher `monnaie(73)`.

Ecrivons une fonction "vabs" d'argument x (nombre) et retournant valeur absolue de x .

```
def vabs(x) :  
    if x >= 0 :  
        return(x)  
    else :  
        return(-x)
```

Exercice 8. Ecrire une fonction nommée `mini2` ayant pour arguments a et b (nombres) et retournant le minimum de a et b . Faire afficher `mini2(8, 4)`.

La distance entre deux nombres est la valeur absolue de leur différence. Ecrivons une fonction "dis" ayant pour arguments x et y (nombres) et retournant la distance entre x et y . La fonction *dis* devra faire appel à la fonction *vabs*.

```
def dis(x,y) :
    return(vabs(y-x))
```

Exercice 9. Coller la fonction *mini2* (après l'avoir copiée dans le fichier *exo8.py*). Ecrire une fonction nommée *mini3* ayant pour argument trois nombres a , b et c et retournant le minimum de a , b et c . La fonction *mini3* devra faire appel à la fonction *mini2* (deux fois) et ne pas utiliser de structure conditionnelle ("if"). Faire afficher *mini3*(8, 3, 4).

Exercice 10. Ecrire une fonction nommée "TVActprix" ayant pour arguments *taux* (taux de la TVA appliquée à un objet) et *PHT* (le prix hors taxes de l'objet) et retournant le couple (*taux*, *PTTC*) où *taux* est la TVA appliquée à l'objet et *PTTC* est le prix toutes taxes comprises de l'objet. Faire afficher *TVActprix*(20.0/100, 15).

Exercice 11.

1. Ecrire une fonction nommée *f* ayant pour argument un entier n et renvoyant $\frac{n}{2}$ si n est pair et $3n + 1$ si n est impair.
2. Faire afficher *f*(14) et *f*(9).

Exercice 12. Un magasin de reprographie facture 10 centimes chacune des dix premières photocopies, 9 centimes chacune des vingt suivantes et 8 centimes au-delà. Ecrire une fonction nommée *repro* ayant pour argument n (nombre de photocopies) et renvoyant le prix des photocopies. Faire afficher *repro*(7), *repro*(28), *repro*(35).

Exercice 13. Une montre à cristaux liquides affiche à chaque instant un numéro d'heure (compris entre 0 et 23) et un numéro de minute (compris entre 0 et 59). Ecrire une fonction nommée *heurmin* ayant pour arguments h et m les numéros d'heure et de minute affichés à un instant donné et qui retourne les numéros d'heure et de minute affichés une minute plus tard. Faire afficher *heurmin*(4, 56), *heurmin*(7, 59), *heurmin*(23, 59).

Exercice 14.

1. Un entier a est le numéro d'une année bissextile ssi (a est divisible par 4 mais pas par 100) ou (a est divisible par 400). Ecrire une fonction nommée *bis* ayant pour argument un entier a et retournant *True* si a est le numéro d'une année bissextile et *False* sinon. Faire afficher *bis*(1900), *bis*(2000), *bis*(2001), *bis*(2004).
2. Le mois de février possède 29 jours si l'année est bissextile et 28 jours sinon. Les mois possédant 30 jours sont avril, juin, septembre et novembre. Les autres mois possèdent 31 jours. Ecrire une fonction nommée *nb* prenant en argument deux entiers a et m avec $1 \leq m \leq 12$ et retournant le nombre de jours du mois m de l'année a . Faire afficher *nb*(2000, 1), *nb*(2000, 2), *nb*(2000, 3), *nb*(2000, 4).
3. Trois entiers a , m , j correspondent à une date valide ssi $1 \leq m \leq 12$ et $1 \leq j \leq \text{nb}(a, m)$. Ecrire une fonction nommée *datevalide* ayant pour argument trois entiers a , m , j en retournant *True* si a, m, j correspondent à une date valide et *False* sinon.

Exercice 15. Les taux d'imposition pour les revenus de 2018 sont les suivants :

jusqu'à 9807 euros	0%
de 9807 à 27086 euros	14%
de 27086 à 72617 euros	30%
de 72617 à 153783 euros	41%
plus de 153783 euros	45%

Par exemple, si un contribuable a gagné 32579 euros en 2018, le montant en euros de son impôt sur le revenu est $9807 \times 0.00 + (27086 - 9807) \times 0.14 + (32579 - 27086) \times 0.30$.

1. Ecrire une fonction nommée *impotrev* ayant pour argument r le montant en euros des revenus 2018 d'un contribuable et retournant le montant en euros de son impôt sur le revenu.
2. Faire afficher *impotrev*(5000), *impotrev*(10000), *impotrev*(20000), *impotrev*(30000), *impotrev*(100000), *impotrev*(1000000).