

Exercice type – Le plus grand, le plus petit

On donne deux entiers a et b. Ecrire un code qui calcule le plus petit et le plus grand des deux entiers a et b. Les valeurs calculées seront placées dans des variables mini et maxi.

Solution

Il suffit de comparer les deux nombres pour savoir qui est mini et qui est maxi, d'où le code :

```
1 a = 42
2 b = 17
3
4 if a < b:
5     mini=a
6     maxi=b
7 else:
8     mini=b
9     maxi=a
10
11 print(a, b)
12 print("mini =", mini, "maxi =", maxi)
```

```
13 42 17
14 mini = 17 maxi = 42
```

Il est intéressant de remarquer que des variables mini et maxi sont créées à la fin du code sans que pour autant on ait **en tout début de code** (comme pour a et b) une définition par affectation de ces variables : dans le corps d'une instruction **if** on peut placer n'importe quel type d'instruction qu'on aurait placée en début de code.

On pouvait aussi écrire le code suivant qui évite le **else** :

```
1 a = 42
2 b= 17
3
4 maxi=a
5 mini=b
6
7 if a < b:
8     mini=a
9     maxi=b
10
11 print(a, b)
12 print("mini =", mini, "maxi =", maxi)
```

Exercice type – Régler un montant avec des coupures de même valeur

On vous donne la valeur b d'un billet en euros, par exemple $b = 10$ et un montant m , nombre entier représentant un montant en euros à régler, par exemple, $m = 8181$. Déterminer le nombre minimum N de billets de b euros pour être en mesure de régler le montant m .

Par exemple,

— si $b = 10$ et $m = 8181$ alors $N = 819$

— si $b = 10$ et $m = 800$ alors $N = 80$

Solution

Il y a deux cas selon que le montant peut être réglé de manière exacte avec des billets de montant b . Par exemple, si $b=10$ et $m=8181$, le montant ne peut être réglé de manière exacte (sans rendu de monnaie). Le nombre exact de billets est clairement 819 car avec 818 billets on peut régler jusqu'à 8180 euros (et il manque 1 euro) et avec un billet de plus, on peut régler le montant (et il restera neuf euros); ce dernier nombre (819) est une unité de plus que le quotient entier de m par b . D'où le code :

```
1 b= 10
2 m=8181
3 N=m//b
4 r=m%b
5 if r!=0:
6     N=N +1
7 print("b = ", b, "m =", m, "->", N)
8 print("->", N)
9
10 print()
11
12 b= 10
13 m=800
14 N=m//b
15 r=m%b
16 if r!=0:
17     N=N +1
18 print("b = ", b, "m =", m, "->", N)
19 print("->", N)
```

— Lignes 16 et 17 : s'il est possible de payer *exactement* la somme m avec des billets b alors la condition ligne 16 est fausse et c'est N qui donne la réponse attendue.

On peut aussi utiliser `else` :

```
if m%b!=0:
    N=m//b +1
else:
    N=m//b
```