

### Exercice type – 14 juillet

Une date est codée avec deux variables  $j$  et  $m$  où  $j$  représente le jour du mois et  $m$  le numéro du mois. Par exemple, le 2 novembre est codé avec  $j = 2$  et  $m = 11$ . On vous donne deux variables  $j$  et  $m$  représentant une date valide et vous devez construire une variable booléenne notée `avant` et qui vaut `True` si la date donnée tombe le 14 juillet ou les jours d'avant et `False` si elle tombe à partir du 15 juillet. Ci-dessous, quelques exemples d'exécution du programme demandé :

```
(10, 4) -> True
(15, 6) -> True
(10, 7) -> True
(14, 7) -> True
(20, 7) -> False
(2, 11) -> False
(25, 12) -> False
```

### Solution

Pour savoir si une date ( $j$ ,  $m$ ) est antérieure au 14 juillet, on commence par regarder si le mois  $m$  vérifie  $m < 7$  et si c'est le cas, la date est bien antérieure ; sinon, on regarde si  $m$  vaut 7 et dans ce cas, la date sera antérieure selon que le jour sera ou pas, antérieur au 14. Dans tous les autres cas, la date n'est pas antérieure. D'où le code :

```
1 j = 10
2 m = 4
3 avant = m < 7 or m == 7 and j <= 14
4 print((j,m), "->", avant)
5
6 j = 15
7 m = 6
8 avant = m < 7 or m == 7 and j <= 14
9 print((j,m), "->", avant)
10
11 j = 10
12 m = 7
13 avant = m < 7 or m == 7 and j <= 14
14 print((j,m), "->", avant)
15
```

### Exercice type – Mois de 31 jours

On donne un numéro de mois `m` entre 1 et 12. Créer une variable booléenne `est_mois_31` (prenant comme valeur `True` ou `False`) qui teste si `m` est le numéro d'un mois ayant 31 jours <sup>a</sup> comme janvier (numéro 1) ou juillet (numéro 7) mais pas février (numéro 2).

a. Les mois ayant 31 jours sont : janvier (1), mars (3), mai (5), juillet (7), août (8), octobre (10), décembre (12).

### Solution

On dispose de la liste des numéros de mois ayant 31 jours, donc il suffit de tester les uns après les autres le mois donné `m` avec les mois possibles :

```
1 m = 4
2 est_mois_31 = (m == 1 or m == 3 or m == 5 or
3               m == 7 or m == 8 or m == 10 or m == 12)
4
5 print(est_mois_31)
```

```
6 False
```

— Lignes 2-3 : une expression placée entre des parenthèses peut être exceptionnellement écrite sur plusieurs lignes, cela évite d'écrire de longues lignes.

On peut aussi essayer de donner une condition plus synthétique. Un mois de 31 jours placé avant août est de numéro 1, 3, 5 ou 7 et donc impair ; à l'inverse, un mois de 31 jours placé à partir d'août est de numéro 8, 10 ou 12 et donc pair. D'où le code suivant :

```
1 m = 9
2 est_mois_31 = ((m <= 7 and m % 2 == 1) or (m >= 8 and m % 2 == 0))
3 print(est_mois_31)
```

```
4 False
```

On pouvait aussi remarquer qu'il y a plus de mois à 31 jours et donc il est (légèrement) plus court d'écrire de passer par la négation :

```
1 m = 4
2 est_mois_31 = not (m==2 or m==4 or m==6 or m==9 or m==11)
3 print(est_mois_31)
```

```
4 False
```