

## Calendrier

Écrire une fonction `calendrier(j,m,a)` qui affiche un petit calendrier de la semaine lorsqu'on lui fournit en argument la date du dimanche. Si on entre `calendrier(29,4,18)` (29 avril 2018), ce programme doit nous afficher : Lundi 30 avril 2018, Mardi 1 mai 2018, ..., Dimanche 6 mai 2018.

```
deg PYTHON
>>> from calendrier import *
Lundi 14 mai 2018
Mardi 15 mai 2018
Mercredi 16 mai 2018
Jeudi 17 mai 2018
Vendredi 18 mai 2018
Samedi 19 mai 2018
Dimanche 20 mai 2018
>>> |
```

Il faut penser à traiter tous les cas (semaine à cheval sur deux mois ou deux années, années bissextiles). Ce programme nécessite qu'on lui fournisse le nom et la longueur des mois ainsi que la règle concernant les années bissextiles. J'ai simplifié le problème en ne traitant que les années après le 01/01/2000. Dans cette version, il faut entrer la date du dimanche qui précède la semaine (ce qui était demandé) mais cela amoindrit l'intérêt d'un tel programme.

```
1 def calendrier(j,m,a):
2     nom_jours=['Lundi','Mardi','Mercredi','Jeudi','Vendredi','Samedi','Dimanche']
3     nom_mois=['janvier','fevrier','mars','avril','mai','juin','juillet','aout','septembre','octobre','novembre','decembre']
4     long=[31,28,31,30,31,30,31,31,30,31,30,31]
5     if a%4==0 and not a==0:
6         long[1]=29
7     semaine,jour,mois,annee=[],j,m,2000+a
8     while len(semaine)<7:
9         jour=jour+1
10        if jour>long[mois-1]:
11            mois=mois+1
12            jour=1
13            if mois==13:
14                mois=1
```

```
15     annee=annee+1
16     semaine.append([nom_jours[len(semaine)],str(jour),nom_mois[mois-1],str(
17         annee)])
18     return semaine
19 Cal=calendrier(13,05,2018)
20 for i in range(7):
21     print(" ".join(Cal[i]))
```

On peut imaginer qu'il serait plus utile si, en fournissant la date du jour, par exemple 27 décembre 2018 sous la forme du triplet (27,12,18), le programme se charge de trouver le jour de la semaine correspondant à cette date et affiche la semaine qui suit cette date.

Dans le programme qui suit, pour déterminer le jour de la semaine, le module `jourSemaine(j,m,a)` calcule le nombre de jours écoulés entre le 01/01/2000 qui était un samedi et la date du jour (j,m,a).

```
1 def calendrier(j,m,a):
2     if a%4==0:
3         long[1]=29
4     semaine,jour,mois,annee=[],j,m,2000+a
5     while len(semaine)<7:
6         semaine.append([nom_jours[(len(semaine)+JS)%7],str(jour),nom_mois[mois-1],
7             str(annee)])
8         jour=jour+1
9         if jour>long[mois-1]:
10            mois=mois+1
11            jour=1
12            if mois==13:
13                mois=1
14                annee=annee+1
15    return semaine
16 def jourSemaine(j,m,a):
17     an,mo,jo=0,1,5 #le 1er janvier 2000 était un samedi (j=5)
18     while an<a:
19         if an%4==0:
20             jo+=366
21         else:
22             jo+=365
23         an+=1
24     while mo<m:
25         if a%4==0 and mo==2:
```

```
26     jo+=long[mo-1]+1
27     else:
28         jo+=long[mo-1]
29         mo+=1
30     jo+=j-1
31     return jo%7
32
33 jourJ,moisM,anA=27,12,18 #entrer ici la date du jour désiré (après le
    01/01/00)
34 nom_jours=['Lundi','Mardi','Mercredi','Jeudi','Vendredi','Samedi','Dimanche']
35 nom_mois=['janvier','fevrier','mars','avril','mai','juin','juillet','aout','
    septembre','octobre','novembre','decembre']
36 long=[31,28,31,30,31,30,31,31,30,31,30,31]
37 JS=jourSemaine(jourJ,moisM,anA)
38 Cal=calendrier(jourJ,moisM,anA)
39 for i in range(7):
40     print(" ".join(Cal[i]))
```

```
deg PYTHON
>>> from calendrier import *
Jeudi 27 decembre 2018
Vendredi 28 decembre 2018
Samedi 29 decembre 2018
Dimanche 30 decembre 2018
Lundi 31 decembre 2018
Mardi 1 janvier 2019
Mercredi 2 janvier 2019
>>> |
```

Trois listes sont fixes ici et permettent de donner le nom des jours (**nom\_jours**), des mois (**nom\_mois**) et la longueur des mois d'une année ordinaire (**long**). La liste semaine, par contre, est variable, remplie progressivement avec les jours de la semaine cherchés. On peut prolonger ce type de programme pour obtenir l'affichage d'un mois ou d'une année-type (sans tenir compte des années bissextiles, il n'y a que 7 types d'années : celles qui commencent un lundi, celles qui commencent un mardi, etc.). On peut aussi déterminer la phase de la Lune...