

**1DEV1A – 1DEVL1A – Laboratoires Python****TD 06 – Mise en pratique : un calendrier**

Dans ce TD vous utiliserez les notions vues précédemment afin de réaliser un petit script permettant d'afficher sur le terminal le calendrier d'un mois et d'une année donnés.

**Table des matières**

<b>1</b>	<b>Un calendrier</b>	<b>2</b>
<b>2</b>	<b>Afficher le titre</b>	<b>2</b>
<b>3</b>	<b>Afficher l'entête</b>	<b>3</b>
<b>4</b>	<b>La suite des jours du mois</b>	<b>3</b>
<b>5</b>	<b>Calculer le premier jour du mois</b>	<b>4</b>
<b>6</b>	<b>Afficher le calendrier</b>	<b>5</b>

## Rappel

Vous devez absolument avoir fait et compris les TD précédents *avant* d'aborder celui-ci.

## 1 Un calendrier

Votre programme demandera à l'utilisateur d'entrer au clavier un mois et une année au format MM AAA ou M AAAA (par exemple 6 2018 ou 12 2015) et affichera le calendrier de ce mois.

Par exemple, pour le mois de juin 2018, le programme affichera :

```
=====
          Juin 2018
=====
Lu   Ma   Me   Je   Ve   Sa   Di
                    01   02   03
04   05   06   07   08   09   10
11   12   13   14   15   16   17
18   19   20   21   22   23   24
25   26   27   28   29   30
=====
```

Nous allons vous guider tout au long de ce TD afin de développer les différentes fonctions nécessaires. Créez tout d'abord :

1. un module `calendrier.py` destiné à contenir les fonctions du calendrier ;
2. un programme principal `main.py` qui fera appel aux différentes fonctions.

Tout le code que nous vous demanderons d'écrire dans la suite sera à placer dans ces fichiers.

## 2 Afficher le titre

Écrivons tout d'abord le code permettant de demander un numéro de mois et une année à l'utilisateur et d'afficher le titre correspondant à l'écran.

### Exercice 1 Le mois

Dans le module `calendrier.py`, écrivez la fonction `nom_du_mois(index)` qui reçoit le numéro d'un mois et retourne son nom en français.

Par exemple, si la fonction reçoit 1, elle retourne "Janvier" et si elle reçoit 6, elle retourne "Juin".

Nous supposons que la fonction reçoit toujours un nombre entre 1 et 12.

### Exercice 2 Le titre

Dans le module `calendrier.py`, écrivez la fonction `afficher_titre(mois, année)` qui reçoit un index de mois et une année et affiche le titre.

Par exemple, si la fonction reçoit 6 et 2018, elle affiche :

```
=====
Juin 2018
=====
```

Cette fonction fera un appel à la fonction `nom_du_mois` de l'exercice précédent.

### Exercice 3 Afficher le titre

Complétez votre script `main.py` de façon à importer le module `calendrier.py`. Demandez ensuite à l'utilisateur un mois et une année et affichez le titre correspondant. Vous ferez naturellement appel à la fonction `afficher_titre(mois, annee)` du module `calendrier.py`.

Par exemple, si l'utilisateur entre 6 et 2018, votre programme affiche :

```
=====
Juin 2018
=====
```

## 3 Afficher l'entête

### Exercice 4 Entête

Écrivez la fonction `afficher_entete()` qui affiche l'entête :

Lu Ma Me Je Ve Sa Di

Complétez votre script `main.py` afin qu'il affiche l'entête juste après le titre.

## 4 La suite des jours du mois

Nous allons construire une chaîne de caractères contenant la suite de tous les numéros des jours du mois. Cette chaîne commencera donc par 01 et se terminera par 28 ou 29 (si le mois est février), 30 ou 31.

Par exemple, pour le mois de juin 2018, cette suite sera :

```
"01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22  
23 24 25 26 27 28 20 30"
```

Remarque : nous nous contentons ici de renvoyer la suite des numéros, séparés par des espaces. Les retour à la ligne, qui seront nécessaires pour avoir un affichage correct du calendrier, seront pris en compte plus tard.

### Exercice 5 Années bissextiles

Pour connaître le nombre de jour dans un mois, il est nécessaire de savoir si l'année est bissextille ou non<sup>1</sup> pour traiter correctement le cas du mois de février.

La règle est simple, une année est bissextille dès que *l'une des deux* conditions suivantes est vraie :

- ▷ elle est divisible par 4 et non divisible par 100 ;
- ▷ elle est divisible par 400.

1. [http://en.wikipedia.org/wiki/Leap\\_year](http://en.wikipedia.org/wiki/Leap_year)

Par exemple, les années 2000 (car divisible par 400) et 2008 (car divisible par 4 mais pas par 100) sont bissextiles. Par contre, les années 1900, 2018 ou encore 2003 ne le sont pas.

Écrivez la fonction `est_bissextile(année)` qui reçoit une année en paramètre et retourne True si cette année est une année bissextile et False sinon.

## Exercice 6 Suite des numéros de jours dans un mois

Écrivez la fonction `suite_numeros_jours(mois, annee)` qui reçoit un index de mois et une année et retourne une chaîne de caractères contenant la suite des numéros de jours.

Par exemple, si elle reçoit 2 et 2018, elle retourne la chaîne de caractères :

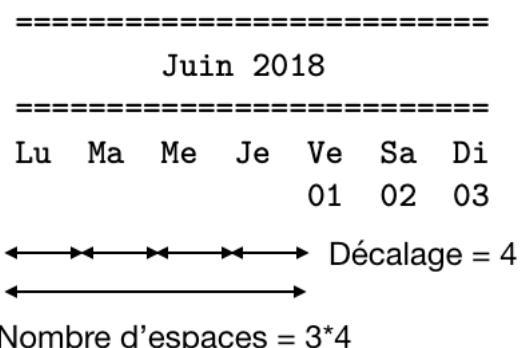
"01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22  
23 24 25 26 27 28"

car février 2018 a 28 jours puisque l'année 2018 n'est pas bissextile.

## 5 Calculer le premier jour du mois

Pour afficher correctement notre calendrier à l'écran, avoir la suite des numéros des jours n'est pas suffisant. Il nous faut également connaître à quel jour de la semaine correspond le premier jour du mois : lundi, mardi... Si le premier jour du mois est un lundi, il n'y aura pas de décalage à appliquer lors de l'affichage. Si c'est un mardi, il y aura un décalage de 1, etc.

Une fois le décalage connu, il suffira d'ajouter un nombre d'espace égal à  $4^2$  fois la valeur de ce décalage au début de la chaîne de caractères construite à l'étape précédente de façon à afficher le premier jour du mois au bon endroit, comme illustré ci-dessous<sup>3</sup> :



## Exercice 7 Le premier jour du mois

Il existe un formule permettant de calculer le jour correspondant à une date (jour, mois, année) donnée. Cette formule s'appelle la congruence de Zeller<sup>4</sup>.

$$h = \left( q + \left\lceil \frac{13(m+1)}{5} \right\rceil + K + \left\lfloor \frac{K}{4} \right\rfloor + \left\lceil \frac{J}{4} \right\rceil + 5J + 5 \right) \bmod 7$$

2. Ce nombre correspond à la largeur utilisée pour indiquer le nom du jour suivi de 2 espaces, comme dans « Lu », par exemple.

3. Remarquez qu'il y a une erreur dans la valeur du nombre d'espaces renseigné sur la figure. Il est égal à  $4^*4$ , et non  $3^*4$  comme indiqué.

4. Voir [http://en.wikipedia.org/wiki/Zeller's\\_congruence](http://en.wikipedia.org/wiki/Zeller's_congruence). L'expression donnée ici est légèrement différente, mais est équivalente.

où :

- ▷  $\lfloor x/y \rfloor$  représente le résultat de division entière de  $x$  par  $y$ ;
- ▷  $h$  est un entier représentant le jour de la semaine ( $0 =$  lundi,  $1 =$  mardi,  $2 =$  mercredi...);
- ▷  $q$  est un entier représentant le jour du mois (de 1 à 31);
- ▷  $m$  est un entier représentant le numéro du mois ( $3 =$  mars,  $4 =$  avril..., janvier et février étant considérés comme les mois 13 et 14 de l'année *précédente*<sup>5</sup>, donc janvier 2008 sera considéré comme 13 2007);
- ▷  $K$  est un entier représentant l'année dans le siècle, c'est à dire  $year \bmod 100$ <sup>6</sup> (par exemple 8 pour l'année ajustée 2008);
- ▷  $J$  est un entier représentant  $\lfloor year/100 \rfloor$  (par exemple 20 pour l'année ajustée 2008).

Ainsi, pour le 3 mars 2008,  $q = 3$ ,  $m = 3$ ,  $J = 20$  et  $K = 8$ . Le résultat  $h$  de la congruence de Zeller est  $((3 + 10 + 8 + 2 + 5 + 100 + 5) \bmod 7) = (133 \bmod 7) = 0$  et donc lundi.

Écrivez la fonction `numero_jour(jour, mois, annee)` qui retourne la valeur de la congruence de Zeller correspondant au jour reçu en paramètre.

#### **Exercice 8 Appliquer le décalage**

Dans votre script `main.py`, faites appel aux fonctions `numero_jour(jour, mois, annee)` et `suite_numeros_jours(mois, annee)` pour construire la chaîne de caractères `suite_numeros` contenant la suite des numéros des jours, précédée du bon nombre d'espaces.

## **6 Afficher le calendrier**

Pour afficher le calendrier, il n'y a plus qu'à afficher la valeur de la variable `suite_numeros` en passant à la ligne quand c'est nécessaire.

Astuce : utilisez pour cela le module `textwrap` de la bibliothèque standard Python.

#### **Exercice 9 Le module textwrap**

Ajoutez le module `textwrap` à votre script Python.

Ce module contient de nombreuses fonctionnalités permettant de mettre du texte en forme, n'hésitez pas à l'explorer. Ici, nous nous contenterons d'utiliser la fonction `fill` pour produire une nouvelle chaîne de caractères contenant un retour à la ligne quand c'est nécessaire.

#### **Exercice 10 Afficher le calendrier**

Utilisez la fonction `fill` pour afficher la suite des numéros de votre calendrier avec les retour à la ligne. Ceci finalise l'écriture de votre script.

Testez dans votre script que le calendrier s'affiche correctement.

---

5. Pour désigner cette année éventuellement diminuée de un, on parle d'*année ajustée* et on la note *year*.

6. Où *year* est l'année ajustée comme pour le calcul de *m*.