

SAé 105 : Traiter des données

Projet n°6 : aider un étudiant à travailler pendant les vacances

Notre objectif est de traiter les données fournis et d'en extraire les informations souhaité avant de les afficher. Nous allons devoir afficher, pour un groupe d'étudiants donné, l'intitulé, la date et l'heure de leur premier cours après chaque période de vacances.

Pour cela on utilisera plusieurs librairie pour optimiser le code : csv, datetime, tkinter, tkcalendar.

csv, va nous permettre la lecture d'un fichier csv dans python. On aura un éléments sous forme de liste qui va contenir toutes les informations du fichier csv à la suite. On va ouvrir le fichier csv et lire chaque ligne dans une liste se nommant « table », chaque ligne sera délimité par l'argument « , » :

```
table=[]
with open("test.csv",newline="") as csvfile:
    reader=csv.reader(csvfile,delimiter=",")
    for row in reader:
        table.append(row)
```

Fichier CSV

[illegible]

Conversion via la librairie CSV dans python

```
des syst mes d'exploitation", "2021-10-29 11:30:00+00:00", "2021-10-29 15:30:00+00:00", "RTIShannon2nRTIShannon\NNECTOUX  
X ANTOINEIN(Exported: 05/01/2022 11:04)", "RT-Labo Informatique 2", "[Infrastructure sans fil", "2021-11-24 13:00:00+00:00",  
"2021-11-24 15:00:00+00:00", "RT2HammingnMARCEL SEVERINEIN(Exported: 05/01/2022 11:04)", "RT-Labo Telecoms 1", "[Fondamen  
taux de la programmation", "2021-10-28 06:15:00+00:00", "2021-10-28 10:15:00+00:00", "RTIShannon2nRTIShannon\NZIMMER CHRIS  
TINEIN(Exported: 05/01/2022 11:04)", "RT-Labo Informatique 2", "[Anglais g n ral", "2021-10-13 08:15:00+00:00", "2021-10-  
13 10:15:00+00:00", "RTIHuffman\nPETITPIRE KARENNIN(Exported: 05/01/2022 11:04)", "RT-Labo Informatique 2", "[Math matiques  
des transmissions", "2021-10-14 08:15:00+00:00", "2021-10-14 10:15:00+00:00", "RTIApp\nNECTOUX ANTOINEIN(Exported: 05/01/20  
22 11:04)", "RT-Salle Labo Visio", "[SAE13: T l communication", "2022-01-21 08:15:00+00:00", "2022-01-21 11:15:00+00:00",  
"RTIHuffman\nMARCEL SEVERINEIN(Exported: 05/01/2022 11:04)", "RT-Labo Telecoms 2", "[Technologies d'acc s", "2021-09-15 08:  
15:00+00:00", "2021-09-15 10:15:00+00:00", "RT2App\nMARCEL SEVERINEIN(Exported: 05/01/2022 11:04)", "RT-Salle-TD3", "[Archit  
ecture des syst mes num riques et informatiques", "2022-01-11 13:00:00+00:00", "2022-01-11 17:00:00+00:00", "RTIHuffman\nVIOIX JEAN-BAPTISTEIN(Exported: 05/01/2022 11:04)", "RT-Labo Informatique 1", "[Electromagn tisme", "2021-09-16 06:15:00+0  
0:00", "2021-09-17 08:15:00+00:00", "RT2App\nNECTOUX ANTOINEIN(Exported: 05/01/2022 11:04)", "RT-Salle-TD3", "[Matrices et  
graphes", "2021-09-17 08:15:00+00:00", "2021-09-17 10:15:00+00:00", "RT2Dijkstra\nnRT2Hamming\nNECTOUX ANTOINEIN(Exported: 05/  
01/2022 11:04)", "RT-Labo Informatique 1", "[Scripts avanc s", "2021-09-07 12:00:00+00:00", "2021-09-07 14:00:00+00:00",  
"RT2Dijkstra\nnRT2Hamming\nVIOIX JEAN-BAPTISTEIN(Exported: 05/01/2022 11:04)", "RT-Labo Informatique 1", "Services r seaux  
avanc s", "2021-01-05 13:00:00+00:00", "2022-01-05 17:00:00+00:00", "RT2Huffman\nPERNOT ANTOINEIN(Exported: 05/01/2022 11:  
04)", "RT-Labo Informatique 3", "[TP Hacking  thique", "2021-12-16 09:15:00+00:00", "2021-12-16 11:15:00+00:00", "LP-Cyber  
S curit \n ndREPREZ JEAN-LUCIN(Exported: 05/01/2022 11:04)", "RT-Salle-TD1", "[Technologie de l'internet", "2022-02-15 13:00  
:00+00:00", "2022-02-15 15:00:00+00:00", "RTIApp\nROY MICHAELIN(Exported: 05/01/2022 11:04)", "[Scripts avanc s", "202  
1-09-06 08:15:00+00:00", "2021-09-06 10:15:00+00:00", "RT2App\nVIOIX JEAN-BAPTISTEIN(Exported: 05/01/2022 11:04)", "RT-Salle  
-TD3", "[Electromagn tisme", "2021-10-18 14:00:00+00:00", "2021-10-18 16:00:00+00:00", "RT2App\nNECTOUX ANTOINEIN(Exported  
:05/01/2022 11:04)", "RT-Salle-TD3", "[Anglais g n ral", "2021-11-24 13:00:00+00:00", "2021-11-24 15:00:00+00:00", "RTIS  
hannon2nRTIShannon\nPETITPIRE KARENNIN(Exported: 05/01/2022 11:04)", "RT-Labo Informatique 1", "[Applications informatiques  
s", "2021-10-26 08:15:00+00:00", "2021-10-26 10:15:00+00:00", "RT2Hamming\nnRT2Dijkstra\nnVIOIX JEAN-BAPTISTEIN(Exported: 05/0/  
1/2022 11:04)", "RT-Labo Informatique 1", "[Anglais g n ral", "2021-09-29 12:00:00+00:00", "2021-09-29 14:00:00+00:00",  
"RTITurning\nPETITPIRE KARENNIN(Exported: 05/01/2022 11:04)", "RT-Labo Informatique 1", "Services r seaux avanc s", "2022-
```

Cet étape est peu digeste, on va devoir filtrer toutes ces informations en fonction des données à extraire et du cahier des charges.

Nous avons au préalable déterminé les périodes de vacances et ainsi de rentrée à travers le tableau csv et les périodes de vacances de la région. Ces dates vont nous permettre de faire un premier filtre d'information.

On les importe alors dans une liste pour chacune d'entre elle et on initialise trois listes de matières qui vont nous permettre de stocker les matières des rentrées dans différentes listes :

Initialisation des rentrées

```
rentrée1 = ["2021-09-01"]
rentrée2 = ["2021-11-08"]
rentrée3 = ["2022-01-03"]

matieres1 = []
matieres2 = []
matieres3 = []
```

On va donc utiliser une boucle for pour parcourir chaque élément de la liste table.

On initialise une valeur « a » qui est égale à l'élément de la liste qui indique la date e les horaires.

a = '2021-09-13 08:15:00+00:00' pour chaque élément de la liste.

On va ensuite comparé les dates des éléments de la liste avec les dates de rentrées et s'il y a correspondance, l'élément est ajouté à la liste « matières[nombre] » initialisé au préalable.

```
for i in table:
    a = i[2]

    if a[0:10] in rentrée1:
        matieres1.append(i)
    elif a[0:10] in rentrée2:
        matieres2.append(i)
    elif a[0:10] in rentrée3:
        matieres3.append(i)
```

On se retrouve avec 3 listes avec uniquement les matières des jours de rentrée.

On va donc appliquer un second filtre en vérifiant si les éléments correspondent bien au groupe demandé. Pour cela, on va créer une liste avec tous les groupes possibles dans celle-ci.

on va donc faire un « input » pour quel groupe on va devoir chercher les informations sur la rentrée.

On utilise la fonction « while » pour sécuriser la recherche et être sûr que l'éléments recherché existe vraiment. Tant que la valeur donnée n'est pas un groupe, alors on affiche un message d'erreur et on redemande le groupe :

```
groupe_possible = ["TDFourier", "TDBell", "RT1Turing", "RT1Huffman", "RT1App", "RT1-S1", "RT1Shannon1", "RT1Shannon2", "RT2Dijkstra", "RT2Hamming"]
groupe1 = input("Donner le groupe TD(ex :TDFourier, TDBell, RT2Hamming..) :")

while groupe1 not in groupe_possible:
    print("Ce groupe n'existe pas : ")
    groupe1 = input("Donner le groupe TD(ex :TDFourier, TDBell, RT2Hamming..) :")

groupe = []
groupe.append(groupe1)
```

On ajoute le groupe recherché dans une liste qui se nomme « groupe »

On va devoir filtrer notre liste matières[nombre] en prenant en compte le nouvel élément. Cette fois-ci on utilise la fonction « find » pour savoir si le groupe se trouve dans l'élément de la liste matière. Si c'est le cas on continue le programme, si ce n'est pas le cas on supprime l'élément de la liste.

```
for i in matieres1[:]:
    a = i[3]
    position = a.find(groupe[0])

    if position != -1:
        continue
    else:
        matieres1.remove(i)
```

On répète l'opération pour les différentes rentrées.

On remarque que pour certain groupe la première est vide, ce qui signifie que tous les groupes ne commence pas le 01 septembre 2021. On va donc remonter le calendrier pour voir quand le groupe apparaît.

```
jour_rentree = 1
mois = "2021-09-0"
compter = 9

while len(matieres1) == 0:

    rentrée1 = [mois + str(jour_rentree)]

    for i in table:
        a = i[2]

        if a[0:10] in rentrée1:
            matieres1.append(i)
    for i in matieres1[:]:
        a = i[3]
        position = a.find(groupe[0])

        if position != -1:
            continue
        else:
            matieres1.remove(i)

    jour_rentree = jour_rentree + 1

    if jour_rentree > 31:
        mois = "2021-" + str(compter+1) + "-0"
        compter +=1
        jour_rentree = 1
        if compter >= 12:
            compter = 0
```

On se retrouve avec des listes de matière uniquement avec le groupe voulu pour les 3 rentrées.

Il reste à savoir l'heure la plus tôt de la journée pour chaque jour pour afficher les informations de la première heure de cours.

On utilisera la librairie « datetime » pour comparer les heures entre elles principalement. On initialise une année, mois, jour, heure, minute, seconde dans « datetime » pour avoir un premier élément de comparaison. C'est pourquoi ces éléments doivent être le plus élevé possible.

```
min = datetime.datetime(2024,12,31,23,59,59)
```

On va alors pour chaque élément de la liste séparer les éléments de la date de la manière suivante :

```
matiere1_fin = []
for k in matieres1:

    année = int(k[1][0:4])
    mois = int(k[1][5:7])
    jour = int(k[1][9:10])
    heure = int(k[1][11:13])
    minutes = int(k[1][14:16])

    time = datetime.datetime(année,mois,jour,heure,minutes)

    if time < min :
        min = time
        matiere1_fin.clear()
        matiere1_fin.append(k)

for w in matiere1_fin:
    année = int(w[1][0:4])
    mois = int(w[1][5:7])
    jour = int(w[1][9:10])
    heure = int(w[1][11:13])
    minutes = int(w[1][14:16])
```

On détermine la heure minimum avec la variable « min » et on fini par avoir une liste matiere1_fin avec un seul élément : la matière la plus tôt dans la journée de la rentrée.

On répète l'opération pour les trois rentrées.

On va donc déterminer dans la liste les éléments qui nous intéresse comme la salle du cours, le nom de la matière, le prof présent, heure de fin etc..

On va aussi essayer d'apporter plus de précisions en donnant le jour de la semaine de la rentrée : on utilise « .weekday » pour savoir le numéro du jour de la semaine. 0 = lundi , 6 = dimanche.

On utilise donc une liste « week_day » avec les jours de la semaine pour déterminer le jour de la rentrée comme ci-dessous :

```
for valeur in matiere1_fin:
    salle = valeur[4]
    heure_fin = int(valeur[2][11:13])
    minutes_fin = int(valeur[2][14:16])
    day_num = datetime.date(année,mois,jour).weekday()
    prof = valeur[3]
    matiere = valeur[0]

week_day = ["Lundi","Mardi","Mercredi","Jeudi","Vendredi", "Samedi", "Dimanche"]
print(week_day[day_num], rentrée1,"de",heure,":", minutes," à ", heure_fin,":",minutes_fin, "dans la salle ", salle, "avec ", prof, "e
```

On obtient donc dans le terminal :

```
Donner le groupe TD(ex :TDFourier, TDBell, RT2Hamming..) :RT2Hamming
Pour la première rentrée

Mercredi ['2021-09-01'] de 7 : 0 à 8 : 0 dans la salle RT-Salle-TD4 avec RT2Dijkstra
RT2Hamming
RÃ©union/Evenement
(Exported :05/01/2022 11:04) en RentrÃ©e RT2

Pour la deuxieme rentrée

Lundi ['2021-11-08'] de 9 : 15 à 11 : 15 dans la salle RT-Salle-TD4 avec RT2Dijkstra
RT2Hamming
MANSOURI ALAMIN
(Exported :05/01/2022 11:04) en Gestion d'annuaires unifiÃ©s

Pour la troisième rentrée
Lundi ['2022-01-03'] de 8 : 15 à 11 : 15 dans la salle RT-Labo Electronique 1 avec RT2Hamming
ZIMMER CHRISTINE
(Exported :05/01/2022 11:04) en Fibres optiques
```

A partir de ces éléments , nous allons faire une représentation graphique à l'aide de la librairie tkclandar et tkinter.

On initialise une fenêtre nommé « window », de taille « 600x600 » qui se nomme « rentrée[numéro]

```
window = Tk()
window.geometry("600x600")
window.title("rentrée 1")

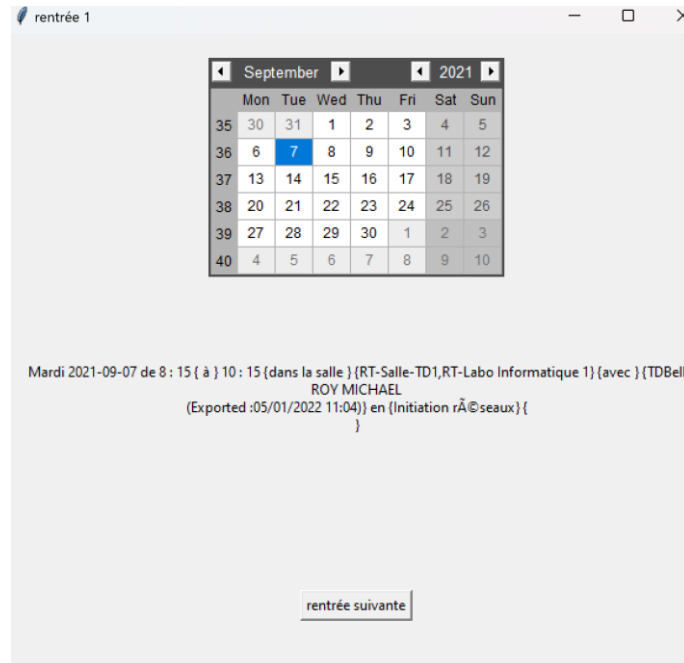
Cal = Calendar(window, selectmode = "day", year = année, month = mois, day = jour )
Cal.pack(pady = 20)
```

On initialise et on affiche le calendrier correspondant aux dates des rentrées.

On affiche également les précisions de la rentrée sur la fenêtre :

```
informations = Label(window2, text = (week_day[day_num], rentrée2,"de",heure,":", minutes," à ", heure_fin,":",minutes_fin, "  
informations.pack(pady = 50)
```

On obtient la fenêtre suivante :



On va créer un bouton pour donnée accès à une autre fenêtre qui correspondra à la rentrée suivante :

```
def Cal2():  
    window.quit()  
    windows2 = Tk()  
    windows2.geometry("600x600")  
    windows2.title("rentrée 2")  
  
    Cal = Calendar(windows2, selectmode = "day", year = année, month = mois, day = jour )  
    Cal.pack(pady = 20)  
    informations = Label(windows2, text = (week_day[day_num], rentrée2,"de",heure,":", minutes," à ", heure_fin,":",minutes_fin, "  
    informations.pack(pady = 50)  
  
    def Cal3():  
        windows2.quit()  
        windows3 = Tk()  
        windows3.geometry("600x600")  
        windows3.title("rentrée 3")  
  
        Cal3 = Calendar(windows3, selectmode = "day", year = année_r3, month = mois_r3, day = jour_r3)  
        Cal3.pack(pady = 50)  
        informations = Label(windows3, text = (week_day[day_num], rentrée3,"de",heure_r3,":", minutes_r3," à ", heure_fin_r3,":",minutes_r3, "  
        informations.pack(pady = 50)  
        windows3.mainloop()  
  
    rt_suivante = Button(windows2, text = "rentrée suivante", command = Cal3)  
    rt_suivante.pack(pady = 80)  
  
    windows2.mainloop()  
  
rt_suivante = Button(window, text = "rentrée suivante", command = Cal2)  
rt_suivante.pack(pady = 80)  
  
window.mainloop()
```

On obtient alors les trois fenêtres suivante une fois qu'on a appuyé deux fois sur le bouton « rentrée suivante » :

rentrée 1

September2021

	Mon	Tue	Wed	Thu	Fri	Sat	Sun
35	30	31	1	2	3	4	5
36	6	7	8	9	10	11	12
37	13	14	15	16	17	18	19
38	20	21	22	23	24	25	26
39	27	28	29	30	1	2	3
40	4	5	6	7	8	9	10

Mardi 2021-09-07 de 8 : 15 { à } 10 : 15 { dans la salle } { RT-Salle-TD1,RT-Labo Informatique 1 } { avec } { TDBell
ROY MICHAEL
(Exported :05/01/2022 11:04) en {Initiation rÃ©seaux} {
}

rentrée suivante

rentrée 2

November2021

	Mon	Tue	Wed	Thu	Fri	Sat	Sun
44	1	2	3	4	5	6	7
45	8	9	10	11	12	13	14
46	15	16	17	18	19	20	21
47	22	23	24	25	26	27	28
48	29	30	1	2	3	4	5
49	6	7	8	9	10	11	12

Lundi 2021-11-08 de 13 : 0 { à } 15 : 0 { dans la salle } RT-Salle-TD1 { avec } { TDBell
ANSEL DAVID
(Exported :05/01/2022 11:04) en {MathÃ©matiques du signal} {
}

rentrée suivante

rentrée 3

January2022

	Mon	Tue	Wed	Thu	Fri	Sat	Sun
52	27	28	29	30	31	1	2
1	3	4	5	6	7	8	9
2	10	11	12	13	14	15	16
3	17	18	19	20	21	22	23
4	24	25	26	27	28	29	30
5	31	1	2	3	4	5	6

Lundi 2022-01-03 de 7 : 15 { à } 9 : 15 { dans la salle } RT-Salle-TD1 { avec } { TDBell
NECTOUX ANTOINE
(Exported :05/01/2022 11:04) en {SAE15: Traiter des donnÃ©es} {
}