

SAÉ 15 - Traitement des données

Projet n°1 - Aider le secrétariat à contacter un vacataire



Projet n°1 : Aider le secrétariat à contacter un vacataire

Le secrétariat peut être amené à rencontrer les enseignants vacataires pour compléter des documents administratifs. Ceux-ci n'étant pas toujours présents dans les locaux, cette tâche n'est pas aisée. L'objectif de ce projet est d'afficher, pour chaque enseignant vacataire du département, la date et l'heure de leur prochaine intervention suivant une date donnée, ainsi que le module pour lequel ils interviennent.

Cahier des charges :

- La date d'analyse doit être paramétrable
- Seuls les vacataires intervenant encore dans l'établissement après la date d'analyse seront affichés dans le résultat
- Le résultat sera affiché :
 - sous forme d'un tableau, au format csv ou xls ou pdf ;
 - sous forme d'une frise chronologique (format libre, par exemple .png).

Pour faciliter le codage et la lisibilité globale du projet, il a été divisé en plusieurs sous-programmes qui vont être détaillés ci-après.

Le code en lui même peut être divisé en trois parties :

- Pré-traitement (conversion du .ical en .csv)
- Traitement des données (création de la liste des prochaines interventions des vacataires)
- Affichage des résultats
- Restitution des résultats

Dans ce programme, nous utilisons plusieurs modules qui serviront dans les sous-programmes.

```
import csv
from matplotlib.pyplot import annotate, figure, plot, axis, savefig, arrow, title
from datetime import datetime, timedelta
```

Figure 1 : Importation des modules

Le module csv sert à exporter les données au format CSV.

Les fonctions du module matplotlib.pyplot servent à l'affichage de la frise.

Les fonctions du module datetime sont utilisées pour la comparaison des dates dans l'affichage de la frise.

1. Pré-traitement

a. importation

```
def importation(fichier):
    """Cette fonction importe les données d'un fichier spécifié et les organise dans une liste (une ligne = une liste)"""
    liste = []

    with open(fichier, newline='', encoding="UTF-8") as file:
        for ligne in file:
            # Vérifie si la ligne n'est pas vide et commence par un espace
            if ligne != "" and ligne[0] == " ":
                liste.append(" " + ligne.strip()) # Ajoute la ligne avec un espace au début à la liste et supprime les espaces inutiles à gauche et à droite de la ligne
            else:
                liste.append(ligne.strip()) # Ajoute la ligne sans espace au début à la liste

    return liste

print(importation("ADECal.ics"))
```

Figure 2 : Sous-programme importation

```
0220105T100449Z, 'SEQUENCE:-2094427407', 'END:VEVENT', 'BEGIN:VEVENT', 'DTSTAMP:20220105T100450Z', 'DTSTART:20220107T123000Z', 'DTEND:20220107T143000Z', 'SUMMARY:TP Maintenance de la disponibilité', 'LOCATION:RT-Salle-TD4', 'DESCRIPTION:\n\nLP-CyberSécurité\n\nEMERY FRANK\n(Exported :05/01/2022 11:04)\n', 'UID:ADE60323032312d32303232556e69766572736974e96465426f7572676676e652d37', '34333932d302d31', 'CREATED:19700101T000000Z', 'LAST-MODIFIED:20220105T100449Z', 'SEQUENCE:-2094427407', 'END:VEVENT', 'BEGIN:VEVENT', 'DTSTAMP:20220105T100450Z', 'DTSTART:20220106T115000Z', 'DTEND:20220106T115000Z', 'SUMMARY:Transmission large bande', 'LOCATION:RT-Labo Informatique 1', 'DESCRIPTION:\n\nRT2Dijkstra\n\nAZZOUNI SOUMAYA\n(Exported :05/01/2022 11:04)', 'UID:ADE60323032312d32303232556e69766572736974e96465426f7572676676e652d33', '31333232d302d30', 'CREATED:19700101T000000Z', 'LAST-MODIFIED:20220105T100449Z', 'SEQUENCE:-2094427407', 'END:VEVENT', 'BEGIN:VEVENT', 'DTSTAMP:20220105T100450Z', 'DTSTART:20220107T163000Z', 'DTEND:20220107T163000Z', 'SUMMARY:TP Maintenance de la disponibilité', 'LOCATION:RT-Salle-TD4', 'DESCRIPTION:\n\nLP-CyberSécurité\n\nEMERY FRANK\n(Exported :05/01/2022 11:04)\n', 'UID:ADE60323032312d32303232556e69766572736974e96465426f7572676676e652d37', '34333932d302d30', 'CREATED:19700101T000000Z', 'LAST-MODIFIED:20220105T100449Z', 'SEQUENCE:-2094427407', 'END:VEVENT', 'BEGIN:VEVENT', 'DTSTAMP:20220105T100450Z', 'DTSTART:20220109T130000Z', 'DTEND:20220109T160000Z', 'SUMMARY:Principes et architecture des réseaux', 'LOCATION:RT-Labo réseaux 2', 'DESCRIPTION:\n\nRT1Turing\n\nDEPREZ JEAN-LUC\n(Exported :05/01/2022 11:04)\n', 'UID:ADE60323032312d32303232556e69766572736974e96465426f7572676676e652d31', '32313132d302d33', 'CREATED:19700101T000000Z', 'LAST-MODIFIED:20220105T100449Z', 'SEQUENCE:-2094427407', 'END:VEVENT', 'BEGIN:VEVENT', 'DTSTAMP:20220105T100450Z', 'DTSTART:20220112T14T071500Z', 'DTEND:20220112T14T115000Z', 'SUMMARY:SAE12: Réseaux', 'LOCATION:RT-Labo Informatique 3', 'DESCRIPTION:\n\nRT1Shannon1\n\n(Exported :05/01/2022 11:04)\n', 'UID:ADE60323032312d32303232556e69766572736974e96465426f7572676676e652d34', '393232302d302d32', 'CREATED:19700101T000000Z', 'LAST-MODIFIED:20220105T100449Z', 'SEQUENCE:-2094427407', 'END:VEVENT', 'BEGIN:VEVENT', 'DTSTAMP:20220105T100450Z', 'DTSTART:20220109T15000Z', 'DTEND:20220109T15000Z', 'SUMMARY:Sources de données', 'LOCATION:RT-Salle Labo Visio', 'DESCRIPTION:\n\nRT1App\n\nVIDIX JEAN-BAPTISTE\n(Exported :05/01/2022 11:04)', 'UID:ADE60323032312d32303232556e69766572736974e96465426f7572676676e652d38', '37353337d302d30', 'CREATED:19700101T000000Z', 'LAST-MODIFIED:20220105T100449Z', 'SEQUENCE:-2094427407', 'END:VEVENT', 'BEGIN:VEVENT', 'DTSTAMP:20220105T100450Z', 'DTSTART:20210922T120000Z', 'DTEND:20210922T140000Z', 'SUMMARY:Réseaux opérateurs', 'LOCATION:RT-Amphi\n\nRT-Salle-TD4', 'DESCRIPTION:\n\nRT2App\n\nRT2Hamming\n\nRT2Dijkstra\n\nDEJUSSIIEU JEAN LOUIS\n(Exported :05/01/2022 11:04)\n', 'UID:ADE60323032312d32303232556e69766572736974e96465426f7572676676e652d35', '363538302d302d31', 'CREATED:19700101T000000Z', 'LAST-MODIFIED:20220105T100449Z', 'SEQUENCE:-2094427407', 'END:VEVENT', 'END:VCALENDAR']
```

Figure 3 : Extrait du résultat du sous programme importation

Ce sous-programme importe le fichier .ical dans une liste. Chaque ligne du fichier .ical devient un élément de la liste.

Dans ce fichier, certaines lignes sont coupées en deux (voir figure ci-dessous), la deuxième débute alors par un espace.

```
DESCRIPTION:\n\nRT1Shannon1\nRT1Turing\nTDFourier\nZIMMER CHRISTINE\n(Exported :05/01/2022 11:04)\n
```

Figure 4 : Exemple d'une ligne "coupée" en deux

Le programme supprime les retours à la ligne (\n) mais conserve l'espace qui permettra par la suite de "recoller" les lignes coupées en deux.

b. rassemblement

```
def rassemblement(liste):
    """Cette fonction rassemble les listes communes à un même événement (un événement = une liste de liste)"""
    liste_event = []
    evenement = []

    for ligne in liste:
        if "BEGIN:VEVENT" == ligne:
            evenement = [] # Commence un nouvel événement en initialisant une nouvelle liste
        elif "END:VEVENT" == ligne:
            liste_event.append(evenement) # Ajoute l'événement à la liste principale
            evenement.append(ligne) # Ajoute la ligne à l'événement en cours

    # Supprime la dernière ligne de la dernière sous-liste (correspondant à "END:VEVENT")
    del liste_event[-1][-1]

    return liste_event

print(rassemblement(importation("ADECal.ics")))
```

Figure 5 : Sous-programme rassemblement

```
20220107T123000Z', 'DTEND:20220107T143000Z', 'SUMMARY:TP Maintien de la disponibilité', 'LOCATION:RT-Salle-TD4', 'DESCRIPTION:\n\nLP-CyberSécurité\nEMERY FRANK\Exported :05/01/2022 11:04', 'UID:ADE60323032312d32303232556e69766572736974e96465426f7572676f676e652d37', '343339332d302d31', 'CREATED:19700101T000000Z', 'LAST-MODIFIED:20220105T100449Z', 'SEQUENCE:-2094427407', 'END:VEVENT'], ['BEGIN:VEVENT', 'DTSTAMP:20220105T100450Z', 'DTSTART:20220106T1081500Z', 'DTEND:20220106T11500Z', 'SUMMARY:Transmission large bande', 'LOCATION:RT-Labo Informatique 1', 'DESCRIPTION:\n\nRT201jksra\nhaZOUNI SOUMAVA\nExported :05/01/2022 11:04', 'UID:ADE60323032312d32303232556e69766572736974e96465426f7572676f676e652d37', '31332342d32d30', 'CREATED:19700101T000000Z', 'LAST-MODIFIED:20220105T100449Z', 'SEQUENCE:-2094427407', 'END:VEVENT'], ['BEGIN:VEVENT', 'DTSTAMP:20220105T100450Z', 'DTSTART:20220107T143000Z', 'DTEND:20220107T163000Z', 'SUMMARY:TP Maintien de la disponibilité', 'LOCATION:RT-Salle-TD4', 'DESCRIPTION:\n\nLP-CyberSécurité\nEMERY FRANK\Exported :05/01/2022 11:04', 'UID:ADE60323032312d32303232556e69766572736974e96465426f7572676f676e652d37', '343339332d302d30', 'CREATED:19700101T000000Z', 'LAST-MODIFIED:20220105T100449Z', 'SEQUENCE:-2094427407', 'END:VEVENT'], ['BEGIN:VEVENT', 'DTSTAMP:20220105T100450Z', 'DTSTART:20220105T100450Z', 'DTEND:20220105T100450Z', 'SUMMARY:Principes et architecture des réseaux', 'LOCATION:RT-Labo réseaux 2', 'DESCRIPTION:\n\nRT1Turing\nndEPREZ JEAN-LUC\nExported :05/01/2022 11:04', 'UID:ADE60323032312d32303232556e69766572736974e96465426f7572676f676e652d37', '323131322d302d33', 'CREATED:19700101T000000Z', 'LAST-MODIFIED:20220105T100449Z', 'SEQUENCE:-2094427407', 'END:VEVENT'], ['BEGIN:VEVENT', 'DTSTAMP:20220105T100450Z', 'DTSTART:20220105T100450Z', 'DTEND:20220105T100450Z', 'SUMMARY:Réseaux', 'LOCATION:RT-Labo Informatique 3', 'DESCRIPTION:\n\nRT1Shannon1\nRT1Shannon2\nExported :05/01/2022 11:04', 'UID:ADE60323032312d32303232556e69766572736974e96465426f7572676f676e652d34', '393232382d302d32', 'CREATED:19700101T000000Z', 'LAST-MODIFIED:20220105T100449Z', 'SEQUENCE:-2094427407', 'END:VEVENT'], ['BEGIN:VEVENT', 'DTSTAMP:20220105T100450Z', 'DTSTART:20220105T100450Z', 'DTEND:20220105T100450Z', 'SUMMARY:Sources de données', 'LOCATION:RT-Salle Labo Visio', 'DESCRIPTION:\n\nRT1App\nVIOIX JEAN-BAPTISTE\nExported :05/01/2022 11:04', 'UID:ADE60323032312d32303232556e69766572736974e96465426f7572676f676e652d38', '373533372d302d30', 'CREATED:19700101T000000Z', 'LAST-MODIFIED:20220105T100449Z', 'SEQUENCE:-2094427407', 'END:VEVENT'], ['BEGIN:VEVENT', 'DTSTAMP:20220105T100450Z', 'DTSTART:20220105T100450Z', 'DTEND:20220105T100450Z', 'SUMMARY:Réseaux opérateurs', 'LOCATION:RT-Amphi\RT-Salle-TD4', 'DESCRIPTION:\n\nRT2App\nRT201jksra\nDEJUSSIEU JEAN LOUIS\nExported :05/01/2022 11:04', 'UID:ADE60323032312d32303232556e69766572736974e96465426f7572676f676e652d35', '363538302d302d31', 'CREATED:19700101T000000Z', 'LAST-MODIFIED:20220105T100449Z', 'SEQUENCE:-2094427407', 'END:VEVENT']]]
```

Figure 6 : Extrait du résultat du sous-programme rassemblement

Ce sous-programme permet de rassembler chaque événement dans une liste.

Nous avons donc une liste de listes. La liste globale contient chaque événement sous forme d'une liste contenant les informations du fichier .ical. Sur la capture d'écran, on visualise en bleu les informations (sous la forme d'une liste) d'un événement.

d. exportation

```
def exportation(nom, champs, liste):
    """Cette fonction exporte une liste de données dans un fichier CSV avec les champs spécifiés"""
    with open(nom, 'w', newline='', encoding="UTF-8") as csvfile:
        csvwriter = csv.writer(csvfile)
        csvwriter.writerow(champs) # Écrit les en-têtes de colonnes dans le fichier CSV
        csvwriter.writerows(liste) # Écrit les données de la liste dans le fichier CSV

liste_event = organisation(rassemblement(importation("ADECal.ics")))
exportation("ADECal.csv", ["DTSTART", "DTEND", "SUMMARY", "LOCATION", "DESCRIPTION", "UID", "CREATED", "LAST-MODIFIED", "SEQUENCE"], liste_event)
print("\nLe fichier ADECal.csv a été généré")
```

Figure 9 : Sous-programme exportation

Figure 10 shows an excerpt of the result of the exportation sub-program. It displays a list of events in a CSV format. The columns are: DTSTART, DTEND, SUMMARY, LOCATION, DESCRIPTION, UID, CREATED, LAST-MODIFIED, SEQUENCE. The events are numbered 1 to 24. The data includes dates, times, and locations for various events.

Figure 10 : Extrait du résultat du sous-programme exportation

Ce sous-programme permet d'exporter les listes obtenues dans un fichier .csv. Il prend en entrée le nom du fichier souhaité, le nom des champs (sous forme d'une liste), et la liste des événements.

2. Traitement

a. creation liste cours

```
def creation_liste_cours(liste):
    """Cette fonction prend une liste de données de cours brut et crée une nouvelle liste de cours formatée"""
    liste_cours = []

    for ligne in liste:

        # Extraction de la date et de l'heure à partir de la chaîne de date
        jour = ligne[1][6:8] + "/" + ligne[1][4:6] + "/" + ligne[1][0:4]
        heure = ligne[1][9:11] + "h" + ligne[1][11:13]

        nom = ""
        ligne_prof = ligne[5][:ligne[5].find("(")].strip().split(" ")

        for element in ligne_prof:
            # Filtre les éléments indésirables et de longueur suffisante pour être considérés comme le nom du professeur
            if ("TD" not in element) and ("RT1" not in element) and ("RT2" not in element) and ("LP" not in element) and ("GSIE" not in element) and ("Apprentissage" not in element) and ("Réunion/Évènement"
                if nom == "":
                    nom = element
                else:
                    nom = nom + " " + element

        liste_cours.append([jour, heure, ligne[3], nom]) # Ajout de la liste du cours à la liste principale

    return liste_cours

print(creation_liste_cours(liste_event))
```

Figure 11 : Sous-programme création liste cours

```
TISTE'], ['22/09/2021', '06h15', 'Scripts avancés', 'RT2App VIOIX JEAN-BAPTISTE'], ['13/10/2021', '14h00', 'Anglais général', 'RT1App PETITPRE KAREEN'], ['10/11/2021', '07h15', 'Applications informatiques', 'RT2Dijkstra VI
OIX JEAN-BAPTISTE'], ['17/11/2021', '13h00', 'Anglais général', 'RT1Turing PETITPRE KAREEN'], ['17/12/2021', '12h30', 'SAE12: Réseaux', 'RT1Huffman DEPREEZ JEAN-LUC'], ['14/02/2022', '15h00', 'Analyse et traitement de doné
es structurées', 'RT1App ZIMMER CHRISTINE'], ['15/10/2021', '06h15', 'Bases des systèmes d'exploitation', 'RT1Turing AZZOUNI SOUMAYA'], ['07/12/2021', '13h00', 'DS TP Infrastructure sans fil', 'RT2Dijkstra MARCEL SEVERINE'],
['26/11/2021', '12h30', 'Architecture des systèmes numériques et informatiques', 'RT1Turing VIOIX JEAN-BAPTISTE'], ['07/09/2021', '14h00', 'Expression Culture Comm pro - Outils info', 'RT1Huffman TROUTTET SYLVAIN'], ['1
2/11/2021', '07h15', 'Applications informatiques', 'RT2Hamming VIOIX JEAN-BAPTISTE'], ['06/01/2022', '12h30', 'TP Maintenance de la disponibilité', 'LP-CyberSécurité BARANEK OLIVIER'], ['28/09/2021', '12h00', 'Initiation rése
aux', 'RT1Shannon1 RT1Shannon2 AZZOUNI SOUMAYA'], ['05/10/2021', '06h15', 'Technologies d'accès', 'RT2Dijkstra DEJUSSEU JEAN LOUIS'], ['10/12/2021', '09h15', 'Réseaux locaux et équipements actifs', 'RT1App DEPREEZ JEAN-LUC
'], ['13/12/2021', '13h00', 'Fibres optiques', 'RT2Hamming RT2Dijkstra ZIMMER CHRISTINE'], ['12/10/2021', '14h00', 'Principes et architecture des réseaux', 'TDBe11 DEPREEZ JEAN-LUC'], ['17/11/2021', '15h00', 'Communication',
'RT2Dijkstra RT2Hamming CHANABEAU CHRISTINE'], ['12/11/2021', '12h30', 'Réseaux locaux et équipements actifs', 'RT1-S1 DEPREEZ JEAN-LUC'], ['25/11/2021', '09h15', 'Réseaux cellulaires', 'RT2Hamming RT2Dijkstra ZIMMER CHR
ISTINE'], ['13/09/2021', '06h15', 'Initiation réseaux', 'TDBe11 ROY MICHAEL'], ['07/10/2021', '06h15', 'Réseaux opérateurs SM', 'RT2Dijkstra RT2Hamming MARCEL SEVERINE'], ['08/09/2021', '06h15', 'Anglais général', 'RT1Shan
non1 RT1Turing PETITPRE KAREEN'], ['21/09/2021', '06h15', 'Anglais', 'RT2Dijkstra CARTIER ANNA'], ['19/10/2021', '06h15', 'Principes et architecture des réseaux', 'RT1App DEPREEZ JEAN-LUC'], ['08/09/2021', '12h00', 'Initiat
ion réseaux', 'TDBe11 ROY MICHAEL'], ['25/01/2022', '07h15', 'Administration système', 'RT1Huffman RT1Shannon2 NECTOUX ANTOINE'], ['23/09/2021', '06h15', 'Anglais général', 'RT1App PETITPRE KAREEN'], ['01/12/2021', '09h15',
'Anglais général', 'RT1Shannon2 RT1Shannon1 PETITPRE KAREEN'], ['19/01/2022', '13h00', 'SAE14: Se présenter sur Internet', 'RT1Shannon1 RT1Shannon2'], ['11/10/2021', '12h00', 'Technologies d'accès', 'RT2App MARCEL SEVERI
NE'], ['26/11/2021', '12h30', 'Gestion d'annuaires unifiées', 'RT2Dijkstra HANSOURT ALAMON'], ['29/09/2021', '12h00', 'Initiation réseaux', 'RT1Huffman NECTOUX ANTOINE'], ['09/12/2021', '07h30', 'DS TP Infrastructure sans f
il', 'RT2Hamming MARCEL SEVERINE'], ['09/02/2022', '15h00', 'Analyse mathématique des signaux', 'RT1App QUEVAL MATHIEU'], ['10/01/2022', '07h15', 'Réseaux opérateurs MR', 'RT2Dijkstra ROY MICHAEL'], ['05/01/2022', '12h30',
'Gestion - Analyse des coûts', 'LP-CyberSécurité LP-GSIE-ARE-1 HOUDIN ELISA'], ['26/10/2021', '06h15', 'Réseaux opérateurs SM', 'RT2Dijkstra RT2Hamming MARCEL SEVERINE'], ['15/12/2021', '13h00', 'SAE12: Réseaux', 'RT1Shan
non1 RT1Shannon2'], ['27/01/2022', '09h15', 'Anglais Technique', 'TDFourier TDFourier CARTIER ANNA'], ['02/12/2021', '12h30', 'Infrastructure sans fil', 'RT2App MARCEL SEVERINE'], ['12/01/2022', '14h30', 'DS Mathématiques
des transmissions', 'RT1App NECTOUX ANTOINE'], ['11/10/2021', '14h00', 'Mathématiques des transmissions', 'RT1-S1 NECTOUX ANTOINE'], ['09/09/2021', '06h15', 'Initiation réseaux', 'TDBe11 ROY MICHAEL'], ['07/01/2022', '12h3
0', 'TP Maintenance de la disponibilité', 'LP-CyberSécurité EMERY FRANCK'], ['06/01/2022', '06h15', 'Transmission large bande', 'RT2Dijkstra AZZOUNI SOUMAYA'], ['07/01/2022', '14h30', 'TP Maintenance de la disponibilité', 'LP-Cy
berSécurité EMERY FRANCK'], ['09/11/2021', '13h00', 'Principes et architecture des réseaux', 'RT1Turing DEPREEZ JEAN-LUC'], ['14/12/2021', '07h15', 'SAE12: Réseaux', 'RT1Shannon1 RT1Shannon2'], ['14/02/2022', '09h15', 'Sour
ces de données', 'RT1App VIOIX JEAN-BAPTISTE'], ['22/09/2021', '12h00', 'Réseaux opérateurs', 'RT2App RT2Hamming RT2Dijkstra DEJUSSEU JEAN LOUIS']]
```

Figure 12 : Extrait du résultat du sous-programme création liste cours

Ce sous programme permet de conserver uniquement les informations nécessaires au projet. On convertit la date en une deux éléments : la date au format JJ/MM/AAAA et l'heure au format 00h00. On conserve l'intitulé du cours en retirant les groupes concernés par le cours (RT1, RT2...). Et on conserve le nom du professeur.

Nous avons maintenant une liste avec uniquement les éléments nécessaires au programme.

b. filtre

```
def filtre(liste):
    """Cette fonction filtre une liste de cours en fonction des titulaires spécifiés"""
    titulaires = ["CARTIER ANNA", "DEPREZ JEAN-LUC", "MANSOURI ALAMIN", "MARCEL SEVERINE", "NECTOUX ANTOINE", "PETITPRE KAREEN", "ROY MICHAEL", "VIOIX JEAN-BAPTISTE", "ZIMMER CHRISTINE", ""]
    nv1_liste = []

    for ligne in liste:
        if ligne[3] not in titulaires: # Vérifie si le titulaire n'est pas dans la liste spécifiée
            if ligne[3].count(" ") > 2: # Vérifie si le nom du titulaire contient plus de deux espaces (au moins trois parties)
                nom = ligne[3].split()
                for i in range(len(nom) - 1):
                    # Vérifie si la combinaison de deux parties du nom n'est pas dans la liste des titulaires
                    if i % 2 == 0 and (nom[i] + " " + nom[i + 1]) not in titulaires:
                        nv1_liste.append(ligne[0:2] + [nom[i] + " " + nom[i + 1]])
            else:
                nv1_liste.append(ligne) # Ajoute la ligne à la nouvelle liste si les conditions ne sont pas remplies

    return sorted(nv1_liste, key=lambda ligne: ligne[0][6:] + ligne[0][3:5] + ligne[0][1:2] + ligne[1][1:2] + ligne[1][2:])

print(filtre(creation_liste_cours(liste_event)))
```

Figure 13 : Sous-programme filtre

```
15, 'Programmation appareils CISCO', 'BACONNIER RICHARD'], ['13/12/2021', '09h15', 'DS Mathématiques du signal', 'ANSEL DAVID'], ['13/12/2021', '12h30', 'Programmation appareils CISCO', 'BACONNIER RICHARD'], ['13/12/2021', '15h00', 'PPP', 'CHABOT ROBERT'], ['14/12/2021', '07h15', 'BOD : ADMINISTRATION', 'PRESLES BENOIT'], ['14/12/2021', '12h30', 'BOD : ADMINISTRATION', 'PRESLES BENOIT'], ['15/12/2021', '07h15', 'Communication', 'CHANAMBEAU CHRISTINE'], ['15/12/2021', '07h15', 'Réseaux WLAN', 'PRESLES BENOIT'], ['15/12/2021', '12h30', 'BOD : NOUVEAU RELATIONNEL', 'PRESLES BENOIT'], ['16/12/2021', '08h15', 'Réseaux cellulaires', 'AZZOUNI SOUMAYA'], ['17/12/2021', '07h15', 'BOD : ADMINISTRATION', 'PRESLES BENOIT'], ['17/12/2021', '12h30', 'Administration Active Directory', 'CABEZAS NICOLAS'], ['03/01/2022', '07h15', 'Services réseaux avancés', 'PERNOT ANTOINE'], ['03/01/2022', '12h30', 'Administration Active Directory', 'CABEZAS NICOLAS'], ['03/01/2022', '13h00', 'Réseaux cellulaires', 'AZZOUNI SOUMAYA'], ['03/01/2022', '15h00', 'Services réseaux avancés', 'PERNOT ANTOINE'], ['04/01/2022', '07h15', 'Services réseaux avancés', 'PERNOT ANTOINE'], ['04/01/2022', '07h15', 'Programmation appareils CISCO', 'BACONNIER RICHARD'], ['04/01/2022', '08h15', 'Transmission large bande', 'AZZOUNI SOUMAYA'], ['04/01/2022', '13h00', 'Services réseaux avancés', 'PERNOT ANTOINE'], ['05/01/2022', '07h15', 'Linux', 'BOUCHER ARNAUD'], ['05/01/2022', '07h15', 'Communication', 'CHANAMBEAU CHRISTINE'], ['05/01/2022', '09h15', 'DS Services réseaux avancés', 'PERNOT ANTOINE'], ['05/01/2022', '12h30', 'Gestion - Analyse des coûts', 'HOUDIN ELISA'], ['05/01/2022', '13h00', 'Services réseaux avancés', 'PERNOT ANTOINE'], ['05/01/2022', '13h00', 'Transmission large bande', 'AZZOUNI SOUMAYA'], ['06/01/2022', '07h15', 'CH Maintenance de la disponibilité', 'BARANEK OLIVIER'], ['06/01/2022', '12h30', 'TP Maintenance de la disponibilité', 'BARANEK OLIVIER'], ['06/01/2022', '14h30', 'TP Maintenance de la disponibilité', 'BARANEK OLIVIER'], ['07/01/2022', '07h15', 'CH Maintenance de la disponibilité', 'EMERY FRANCK'], ['07/01/2022', '09h15', 'CH Maintenance de la disponibilité', 'EMERY FRANCK'], ['07/01/2022', '12h30', 'Programmation appareils CISCO', 'BACONNIER RICHARD'], ['07/01/2022', '12h30', 'TP Maintenance de la disponibilité', 'EMERY FRANCK'], ['07/01/2022', '14h30', 'TP Maintenance de la disponibilité', 'EMERY FRANCK'], ['10/01/2022', '07h15', 'Linux', 'BOUCHER ARNAUD'], ['10/01/2022', '08h15', 'Transmission large bande', 'AZZOUNI SOUMAYA'], ['10/01/2022', '12h30', 'Linux', 'BOUCHER ARNAUD'], ['11/01/2022', '07h15', 'Programmation appareils CISCO', 'BACONNIER RICHARD'], ['11/01/2022', '15h00', 'PPP', 'CHABOT ROBERT'], ['12/01/2022', '07h15', 'Linux', 'BOUCHER ARNAUD'], ['12/01/2022', '07h15', 'Communication', 'CHANAMBEAU CHRISTINE'], ['12/01/2022', '12h30', 'Gestion - Analyse des coûts', 'HOUDIN ELISA'], ['13/01/2022', '07h15', 'Management de projet (Etude de Cas)', 'BACONNIER RICHARD'], ['14/01/2022', '07h15', 'Administration Active Directory', 'CABEZAS NICOLAS'], ['14/01/2022', '08h15', 'Réseaux cellulaires', 'AZZOUNI SOUMAYA'], ['14/01/2022', '12h30', 'Administration Active Directory', 'CABEZAS NICOLAS'], ['19/01/2022', '07h15', 'DS Communication', 'CHANAMBEAU CHRISTINE'], ['02/02/2022', '13h00', 'Administration système', 'AZZOUNI SOUMAYA'], ['04/02/2022', '07h15', 'Administration système', 'AZZOUNI SOUMAYA'], ['07/02/2022', '07h15', 'CH Solutions métier', 'CABEZAS NICOLAS'], ['07/02/2022', '09h15', 'CH Solutions métier', 'CABEZAS NICOLAS'], ['07/02/2022', '12h30', 'TP Solutions métier', 'CABEZAS NICOLAS'], ['07/02/2022', '13h00', 'Administration système', 'AZZOUNI SOUMAYA'], ['07/02/2022', '14h30', 'TP Solutions métier', 'CABEZAS NICOLAS'], ['08/02/2022', '07h15', 'Linux', 'BOUCHER ARNAUD'], ['08/02/2022', '12h30', 'Gestion - Analyse des coûts', 'HOUDIN ELISA'], ['09/02/2022', '07h15', 'Management de projet', 'HENNEBELLE FRANCOIS'], ['09/02/2022', '12h30', 'Gestion - Analyse des coûts', 'HOUDIN ELISA'], ['09/02/2022', '13h00', 'Analyse mathématique des signaux', 'QUEVAL MATHIEU'], ['09/02/2022', '15h00', 'Analyse mathématique des signaux', 'QUEVAL MATHIEU'], ['10/02/2022', '07h15', 'Administration système', 'AZZOUNI SOUMAYA'], ['10/02/2022', '07h15', 'CH Solutions métier', 'CABEZAS NICOLAS'], ['10/02/2022', '09h15', 'CH Solutions métier', 'CABEZAS NICOLAS'], ['10/02/2022', '12h30', 'TP Solutions métier', 'CABEZAS NICOLAS'], ['10/02/2022', '14h30', 'TP Solutions métier', 'CABEZAS NICOLAS'], ['11/02/2022', '07h15', 'Linux', 'BOUCHER ARNAUD'], ['11/02/2022', '12h30', 'Linux', 'BOUCHER ARNAUD'], ['14/02/2022', '07h15', 'Administration système', 'AZZOUNI SOUMAYA'], ['14/02/2022', '13h00', 'Signaux et systèmes pour les transmissions', 'AZZOUNI SOUMAYA'], ['15/02/2022', '07h15', 'Administration système', 'AZZOUNI SOUMAYA'], ['16/02/2022', '07h15', 'Administration système', 'AZZOUNI SOUMAYA'], ['16/02/2022', '07h15', 'Analyse et traitement de données structurées', 'AZZOUNI SOUMAYA'], ['17/02/2022', '07h15', 'Analyse et traitement de données structurées', 'AZZOUNI SOUMAYA'], ['17/02/2022', '12h30', 'Analyse et traitement de données structurées', 'AZZOUNI SOUMAYA'], ['18/02/2022', '12h30', 'Administration système', 'AZZOUNI SOUMAYA'] ]
```

Figure 14 : Extrait du résultat du sous-programme filtre

Ce sous-programme permet de conserver uniquement les cours des enseignants vacataires.

Il permet aussi de prendre en compte les cours ayant plusieurs professeurs différents. Si il y a plus de 3 noms, on vérifie que chaque nom n'est pas titulaire.

Il renvoie la liste d'événements triés par date et par heure.

c. traitement

```
date_entree = input("Entrer une date (JJ/MM/AAAA) : ")
while date_entree[6:] + date_entree[3:5] + date_entree[:2] > filtre(creation_liste_cours(liste_event))[-1][0][6:] + filtre(creation_liste_cours(liste_event))[-1][0][3:5] + filtre(creation_liste_cours(liste_event))[-1][0][:2]:
    print("Il n'y a pas de cours après cette date. La date doit être antérieure au", filtre(creation_liste_cours(liste_event))[-1][0])
    date_entree = input("Entrer une date (JJ/MM/AAAA) : ")
```

Figure 15 : Demande de la date à l'utilisateur

Dans le programme principal, on demande à l'utilisateur de rentrer une date. Si cette date est après le dernier cours du fichier .ical, il est invité à en rentrer une autre.

```
def traitement(liste, date_entree):
    """Cette fonction effectue un traitement sur une liste de cours en fonction de la date d'entrée fournie"""
    post_event = [] # Liste des cours postérieurs à la date d'entrée
    liste_nom = [] # Liste des noms de titulaires déjà traités
    liste_finale = [] # Liste finale des cours traités

    # Filtre les cours postérieurs à la date d'entrée
    for ligne in liste:
        if ligne[0][6:] + ligne[0][3:5] + ligne[0][:2] > date_entree[6:] + date_entree[3:5] + date_entree[:2]:
            post_event.append(ligne)

    # Trie les cours filtrés par date et sélectionne uniquement le premier cours de chaque titulaire
    for event in post_event:
        if event[3] not in liste_nom:
            liste_nom.append(event[3])
            liste_finale.append(event)

    return liste_finale

liste_finale = traitement(filtre(creation_liste_cours(liste_event)), date_entree)
print(liste_finale)
```

Figure 16 : Sous-programme traitement

```
-----
Entrer une date (JJ/MM/AAAA) : 10/10/2030
Il n'y a pas de cours après cette date. La date doit être antérieure au 18/02/2022
Entrer une date (JJ/MM/AAAA) : 10/10/2021
[[['11/10/2021', '06h15', 'Projet Personnel et Professionnel', 'CHAROT ROBERT'], ['11/10/2021', '06h15', 'Mathématiques du signal', 'ANSEL DAVID'], ['12/10/2021', '06h15', 'Administration Windows Server', 'CABEZAS NICOLAS'],
['12/10/2021', '07h15', 'Visite Orange - Fibre Optique', 'DEJUSSEU JEAN LOUIS'], ['13/10/2021', '12h00', 'Expression Culture Communications professionnelles', 'CHANAMBEAU CHRISTINE'], ['14/10/2021', '06h15', 'CAHIER DES
CHARGES', 'HEINBELLE FRANCOIS'], ['15/10/2021', '06h15', 'ITIL', 'BOUCHER ARNAUD'], ['15/10/2021', '06h15', 'Bases des systèmes d'exploitation', 'AZZOUNI SOUMAYA'], ['18/10/2021', '12h00', 'Expression Culture Comm pro -
Outils Info', 'TROUETTE SYLVAIN'], ['15/11/2021', '07h15', 'Réseaux WLAN', 'PRESLES BENOIT'], ['16/11/2021', '07h15', 'Programmation appareils CISCO', 'BACONNIER RICHARD'], ['03/12/2021', '07h15', 'Services réseaux avancés',
'PERNOT ANTOINE'], ['05/01/2022', '12h30', 'Gestion - Analyse des coûts', 'HOLDEN ELISA'], ['06/01/2022', '07h15', 'CM Maintenance de la disponibilité', 'BARANEK OLIVIER'], ['07/01/2022', '07h15', 'CM Maintenance de la dispon
ibilité', 'EMERY FRANCK'], ['09/02/2022', '13h00', 'Analyse mathématique des signaux', 'QUEVAL MATHIEU]]
```

Figure 17 : Résultat du sous programme traitement

Ce sous-programme permet de conserver le premier cours de chaque vacataire après la date entrée précédemment.

Pour comparer les dates, on compare les nombre AAAAMMJJ (23/08/2022 devient 20220823), cela permet de donner le plus d'importance à l'année puis au mois. Les événements ayant été triés par date et heure précédemment, on prend donc le premier cours de la journée de chaque vacataire.

3. Affichage des résultats

a. Affichage_liste

```
def affichage_liste(liste):
    """Cette fonction affiche les informations sur les cours traités dans la liste"""
    for event in liste:
        print("Le prochain cours de l'enseignant", event[3], "a lieu le", event[0], "à", event[1], "pour un cours intitulé", event[2])

affichage_liste(liste_finale)
```

Figure 18 : Sous programme affichage liste

```
Entrer une date (JJ/MM/AAAA) : 10/01/2022
Le prochain cours de l'enseignant BACONNIER RICHARD a lieu le 11/01/2022 à 07h15 pour un cours intitulé Programmation appareils CISCO
Le prochain cours de l'enseignant CHABOT ROBERT a lieu le 11/01/2022 à 15h00 pour un cours intitulé PPP
Le prochain cours de l'enseignant BOUCHER ARNAUD a lieu le 12/01/2022 à 07h15 pour un cours intitulé LINUX
Le prochain cours de l'enseignant CHANAMBEAU CHRISTINE a lieu le 12/01/2022 à 07h15 pour un cours intitulé Communication
Le prochain cours de l'enseignant HOUDIN ELISA a lieu le 12/01/2022 à 12h30 pour un cours intitulé Gestion - Analyse des coûts
Le prochain cours de l'enseignant CABEZAS NICOLAS a lieu le 14/01/2022 à 07h15 pour un cours intitulé Administration Active Directory
Le prochain cours de l'enseignant AZZOUNI SOUMAYA a lieu le 14/01/2022 à 08h15 pour un cours intitulé Réseaux cellulaires
Le prochain cours de l'enseignant HENNEBELLE FRANCOIS a lieu le 09/02/2022 à 07h15 pour un cours intitulé MANAGEMENT DE PROJET
Le prochain cours de l'enseignant QUEVAL MATHIEU a lieu le 09/02/2022 à 13h00 pour un cours intitulé Analyse mathématique des signaux
```

Figure 19 : Résultat du sous-programme affichage_liste pour le 10/01/2022

Ce sous programme permet d'afficher dans la console les prochains cours des vacataires après la date choisie. On affiche leur nom, la date et l'heure du cours et l'intitulé du cours.

Ce sous programme n'est pas utilisé dans le programme, car ce format d'affichage n'est pas demandé. Il permet de tester les programmes plus simplement et visuellement que les exports.

b. exportation_frise

```
def exportation_frise(liste, date_entree):
    """Cette fonction génère une frise chronologique illustrant les cours programmés pour chaque enseignant"""
    liste_dates = []
    liste_noms = []

    # Crée une liste de dates uniques et une liste correspondante de noms d'enseignants avec les cours associés
    for ligne in liste:
        if datetime.strptime(ligne[0], "%d/%m/%Y") not in liste_dates:
            liste_dates.append(datetime.strptime(ligne[0], "%d/%m/%Y"))
            liste_noms.append(ligne[3] + " (" + ligne[2] + " à " + ligne[1] + ")")
        else:
            liste_noms[-1] = liste_noms[-1] + "\n" + ligne[3] + " (" + ligne[2] + " à " + ligne[1] + ")"

    ordonnee = [len(liste_dates) - 1 for i in range(len(liste_dates))]

    figure(figsize=(16, 9))

    # Trace une ligne horizontale représentant le temps
    arrow([liste_dates[0], 0, (liste_dates[-1] - liste_dates[0] + timedelta(days=3)).days*1.5, 0], head_width=len(liste_dates)/10, head_length=(liste_dates[-1] - liste_dates[0] + timedelta(days=1)).days*0.05, linewidth=20, color="c")

    couleurs = ["b", "g", "r", "c", "m", "y"] * 3

    # Trace des lignes pour chaque date avec des cercles pour chaque enseignant et ajoute des annotations pour chaque enseignant avec les cours associés et les dates
    for i, date in enumerate(liste_dates):
        plot([liste_dates[i] * 3, [ordonnee[i], 0, -(ordonnee[i] % 5) - 1], couleurs[ordonnee[i] - 1] * "-o")
        x = date + max([timedelta((liste_dates[-1] - liste_dates[0]).days*0.01), timedelta(1)*0.05])
        annotate(liste_noms[i], xy=[x, ordonnee[i]], fontsize=7)
        annotate(str(liste_dates[i].date()[8:10] + "/" + str(liste_dates[i].date()[5:7] + "/" + str(liste_dates[i].date()[0:4],
            xy=[x, -(ordonnee[i] % 5) - 1], verticalalignment="top", fontsize=7)

    # Désactive l'affichage des axes sur la figure
    axis("off")

    title("Date de la prochaines intervention des vacataires à compté du " + date_entree)
    # Sauvegarde la frise chronologique en tant qu'image PNG
    savefig("vacataire_" + date_entree.replace("/", "-") + ".png", dpi=300)
```

Figure 20 : Sous-programme exportation frise

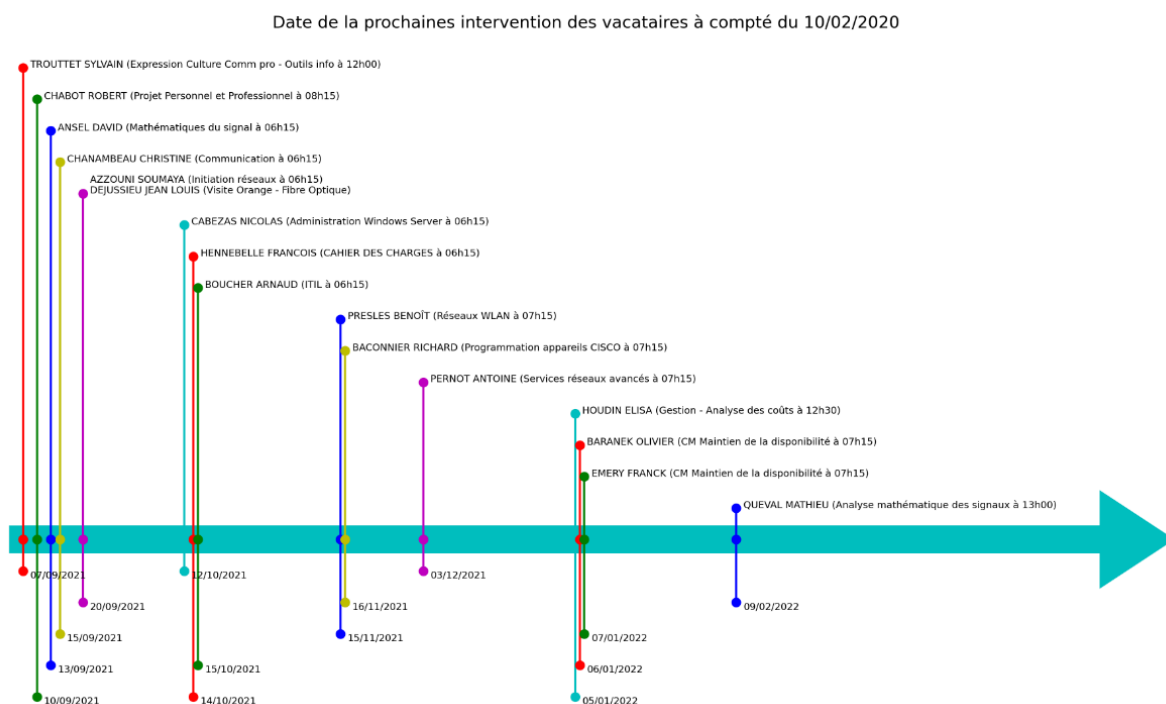


Figure 21 : Résultat du sous programme exportation frise pour le 10/02/2020

On commence par modifier la liste finale en combinant les événements ayant lieu à la même date. On dessine ensuite la flèche. Enfin, on trace les lignes coloré de chaque événement avec 3 points. On affiche en haut le professeur, l'intitulé du cours et l'heure et en bas la date. On finit par masquer les axes, ajouter un titre et sauvegarder l'image au format .png.

Restitution des résultats

Le résultat doit être affiché :

- sous forme d'un tableau, au format csv ou xls ou pdf ;
- sous forme d'une frise chronologique (format libre, par exemple .png).

```
# Affiche un message indiquant que les fichiers ont été générés avec succès
print("\nLes enseignants vacataires intervenant encore dans l'établissement après la date saisie sont donnés dans les fichiers générés suivants :\n")
# Exporte les résultats dans un fichier CSV
exportation("vacataire_" + date_entree.replace("/", "-") + ".csv", [{"Date", "Heure", "Cours", "Vacataire"}, liste_finale])
print("  - vacataire_" + date_entree.replace('/', '-') + ".csv (Tableau)")
exportation_frise(liste_finale, date_entree)
print("  - vacataire_" + date_entree.replace('/', '-') + ".png (Frise chronologique)")
```

Figure 22 : Fin du programme

```
Le fichier ADECal.csv a été généré

Entrer une date (JJ/MM/AAAA) : 10/10/2021

Les enseignants vacataires intervenant encore dans l'établissement après la date saisie sont donnés dans les fichiers générés suivants :

- vacataire_10-10-2021.csv (Tableau)
- vacataire_10-10-2021.png (Frise chronologique)
```

Figure 23 : Affichage final du programme

Le programme affiche donc :

- La bonne création du fichier ADECal.csv (une conversion du .ical en .csv), la demande
- La demande de la date à l'utilisateur
- L'affichage de la réponse final avec :
 - La création du fichier .csv
 - La création de la frise au format .png