TP 1 : Extraction de Données à l'aide du Langage Python

Ne vous inquiétez pas si vous n'avez pas fini ce TP à la fin de la séance.
Ce TP a pour objectif principal de vous familiariser avec certains outils de Python qui est très souvent utilisé pour extraire et/ou mettre en forme des données.

1 Le Format ics

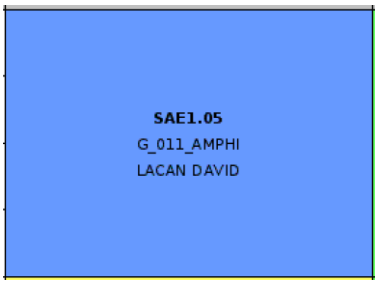
Le format `.ics` désigne la structure servant à décrire les informations/événements d'un calendrier. Ce format est compatible avec le standard `iCalendar` et permet une représentation textuelle des événements, souvent stockée dans un fichier d'extension `.ics`.

Il utilise un ensemble d'**identificateurs** textuels pour décrire différentes **propriétés d'un événement** (date, heure, lieu, ...).

1.1 Format ics d'un Événement

La représentation d"un événement `ics` (ici le CM d'introduction de cette SAÉ) est, par exemple (CM de M. Lacan de l'année dernière), la suivante (à droite, vous avez sa représentation sur ADE) :

```
BEGIN:VEVENT
DTSTAMP:20240110T053220Z
DTSTART:20240110T080000Z
DTEND:20240110T100000Z
SUMMARY:SAE1.05
LOCATION:G_011_AMPHI
DESCRIPTION:\n\nRT1-S1\nLACAN DAVID\n(Exporté le:10/01/2024 06:32)\n
UID:ADE60323032332d3230323453542d455449454e4e452d32323839322d302d30
CREATED:19700101T000000Z
LAST-MODIFIED:20240110T053220Z
SEQUENCE:2141064552
END:VEVENT
```



1.2 Identificateurs principaux

Les principaux identificateurs textuels que nous utiliserons s'interprètent de la sorte :

- `BEGIN` et `END` marquent le début et la fin de la description d'un événement de type, dans notre exemple, `VEVENT` et forme un nœud d'informations.
- `DTSTART` et `DTEND` précisent la date de début et la date de fin de l'événement (`DTSTAMP` est la date de création d'un message en lien avec l'événement et ne sera pas utilisé ici), dans un format que nous verrons plus tard.
- `SUMMARY` déclare l"intitulé de l'événement, dans notre exemple, `SAE1.05`.
- `UID` déclare la valeur d'un identifiant unique associé à cet événement, dans notre exemple, `ADE6032303 2332d3230323453542d455449454e4e452d32323839322d302d30` ; deux événements différents ne pourront ainsi **jamais** avoir le **même identifiant**.
- `DESCRIPTION` donne les groupes de TD/TP pour lesquels l'événement a lieu (ici) `RT1-S1` et/ou les enseignants qui interviennent sur l'événement (ici) `David LACAN`.
- `LOCATION` précise la ou les salles dans lesquelles se tiennent l'événement (ici) en `G_0111_AMPHI`.

Remarques :
Le caractère `:` sépare systématiquement l'identificateur de la description de la propriété.
Les propriétés peuvent être optionnelles, auquel cas l'identificateur n'apparaît pas dans la description `.ics`.
Les propriétés ne sont pas ordonnées (l'ordre de déclaration peut changer d'un évènement à un autre).

1.3 Format d'un événement pseudo-csv

Un événement au format pseudo-csv est décrit par une chaine de caractères de la forme :

`uid;date;heure;duree;modalite;intitule;salle1|salle2|...;prof1|prof2|...;groupe1|groupe2|...`

représentant les différentes propriétés d'un événement, séparées les unes des autres par des ';' avec :

- `uid` l'identifiant unique de l'événement.
- `date` la date au format `JJ-MM-AAAA` à laquelle débute l'événement.
- `heure` l'heure au format `HH:MM` marquant le début (horaire) de l'événement avec mention de l'heure `HH` et des minutes `MM` chacun sur 2 chiffres (les secondes étant omises).
- `duree` la durée de l'événement traduite au format `HH:MM` avec mention du nombre d'heures pleines `HH` et du nombre de minutes restantes `MM` chacun sur 2 chiffres (les secondes étant omises).
- `modalite` la modalité (CM/TD/TP/Proj/DS) de l'événement
- `intitule` l'intitulé de l'événement (tel qu'il apparait dans ADE)
- `salle1|salle2|...` les salles réservées pour le créneau, séparées par des |
- `prof1|prof2|...` le(s) professeur(s) qui encadre(nt) le créneau, séparés par des |
- `groupe1|groupe2|...` le(s) groupe(s) de TD/YP qui suivent le créneau, séparés par des |

L'événement de l'exemple précédent a donc pour description au format pseudo-csv la chaîne de caractères suivante :

`"ADE60323032332d3230323453542d455449454e4e452d32323839322d302d30;10-01-2024;08:00;02:00;CM;SAE1.5;G_011_AMPHI;LACAN DAVID;S1"`

1.4 Format des évènements temporels en .ics

Les évènements temporels d'un contenu `.ics` (rattachés notamment aux identificateurs `DTSTART` et `DTEND`) donnent en même temps la date et l'heure, en utilisant le format `AAAAMMDDThhmmssZ` où :

- `AAAA` donne la valeur de l'année (de 0000 à 9999),
- `MM` donne le mois (de 00 à 12),
- `DD` le jour (de 00 à 31),
- `hh` l'heure (de 00 à 23),
- `mm` les minutes (de 00 à 59)
- `ss` les secondes (de 00 à 59)

Le caractère `T` sépare la date de l'heure ; le caractère `Z` optionnel donne une indication sur le fuseau horaire.

En reprenant l'exemple de l'événement précédent, la date de début de l'événement donnée avec `DTSTART:20240110T080000Z` indique que l'année `AAAA` est 2024, le mois `MM` est 01, le jour `JJ` est 10 puis (après le `T`) l'heure `hh` est 08, les minutes `mm` sont 00 et les secondes 00.

La date obtenue est donc le 10-01-2024 à 08 :00.

1.5 Premier travail à faire

Nous vous demandons comme premier travail d'établir un programme `Programme1.py` en Python, qui permet de représenter le contenu d'un fichier ics représentant seulement une activité, par exemple `evenementSAE_15GroupeA1.ics` téléchargeable sur Moodle, sous la forme d'un pseudo-code CSV défini précédemment.

Vous devrez être capable de lire dans le fichier voulu, récupérer son contenu dans une variable, puis de la traiter afin d'obtenir la chaîne de caractères voulue.

2 Format ics d'un calendrier

2.1 Description

Un calendrier au format .ics liste les événements qui le composent, les uns après les autres (de leur identificateur `BEGIN` à leur identificateur `END`) en :

- les encapsulant dans un **nœud** de type `VCALENDAR`.
- décrivant des propriétés du calendrier grâce à des identificateurs textuels spécifiques.

Prenons l'exemple du calendrier `ADE_RT1_Septembre2023_Decembre2023` (téléchargeable sur Moodle) qui représente l'emploi du temps de tous les groupes de la formation `RT1` du semestre 1 de vos camarades de l'année dernière :

```
BEGIN:VCALENDAR
METHOD:REQUEST
PRODID:-//ADE/version 6.0
VERSION:2.0
CALSCALE:GREGORIAN
BEGIN:VEVENT
....
END:VEVENT
END:VCALENDAR
```

Dans lequel les identificateurs textuels s'interprètent de la sorte :

- Le `BEGIN` initial (ligne 1) et le `END` final (dernière ligne) marquent le début et la fin de la description du calendrier de type (ici) `VCALENDAR`
- `VERSION` et `PRODID` donnent des identifications sur le mode de production du calendrier `.ics`

La description d'un événement ne peut pas être commencée (`BEGIN`) avant que celle du précédent soit terminée (`END`), et il n'y a jamais de ligne vide.

Les données `.ics` sur lesquelles vous allez travailler ont été pré-traitées par les enseignants (notamment le champ `Description`) pour les simplifier et se libérer de certaines contraintes de la norme.

2.2 Deuxième travail à faire

Nous vous demandons comme deuxième travail d'établir un programme `Programme2.py` en Python, en vous inspirant de votre premier travail, qui permet de représenter le contenu d'un fichier ics contenant **plusieurs activités**, par exemple `ADE_RT1_Septembre2023_Decembre2023` téléchargeable sur Moodle, sous la forme d'un pseudo-code CSV.

Votre résultat sera sous la forme d'un tableau où chaque élément est une chaîne de caractères contenant le pseudo-code d'un événement.

Attention, pour certaines ressources ou SAE, il peut y avoir plusieurs salles, plusieurs professeurs ou aucun professeur. Par rapport à la norme que nous avons vue précédemment, si une valeur fait défaut, vous mettrez la chaîne de caractères `"vide"`.

3 Analyse des résultats

3.1 Troisième travail à faire

Nous souhaitons extraire de ce tableau **toutes les séances** de la ressource **R1.07** (Informatique), associées à votre groupe de TP.

Pour cela, vous récupérez comme sortie de votre programme Python, par exemple `Programme3.py`, un tableau contenant la date de la séance, la durée, le type de la séance (CM/TD/TP).

3.2 Quatrième travail à faire

Il est souvent plus parlant de représenter des données issues d'un tableau via des graphes.

Ils peuvent prendre différentes formes : diagramme circulaire (camembert), diagramme en bâtons, diagramme en bâtons empilés ... Vous pourrez vous aider du fichier `SAE15 - Python - Graphes`.

Reprendre votre travail précédent en créant un nouveau programme `Programme4.py` permettant d'afficher le nombre de séances de TP du groupe A1 en septembre, en octobre, en novembre et en décembre 2023.

Une fois le graphe conforme à vos attentes, en utilisant la fonction `export_png()` exporter votre graphe en format PNG.

3.3 Cinquième travail à faire

En utilisant le module `Markdown` (aidez-vous du fichier `SAE15 - Python - Markdown`), générer un fichier HTML permettant d'afficher vos travaux précédents (tableau des séances de R1.07 (travail 3) et diagramme circulaire (travail 4)) directement sur un navigateur Web.