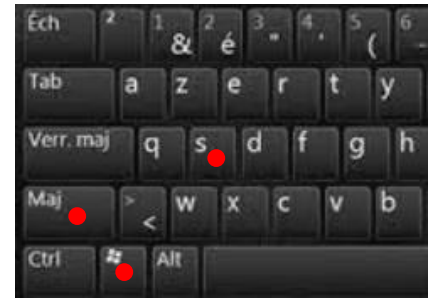


Info 10 - Accès fichiers avec python

Vous allez, dans ce Tp, créer 3 codes python dans lesquels vous allez écrire et lire dans un fichier. Utilisez l'interface de développement dans laquelle vous êtes le plus à l'aise, mais on **conseille d'utiliser Pyzo**, c'est plus simple.

Contrairement au fonctionnement habituel, on demande ici d'uploader en fin de tp, avec le code **tp10**, votre fichier `.py` qui contiendra uniquement les codes pythons des fonctions à créer. Ici, pas de programme principal, les 3 fonctions créées seront exécutées directement dans la console.

Il s'agira bien sûr, avant de pouvoir les exécuter dans la console, d'exécuter le fichier entier (touche F5) afin que vos fonctions soient mémorisées.



1- PREPARATION :

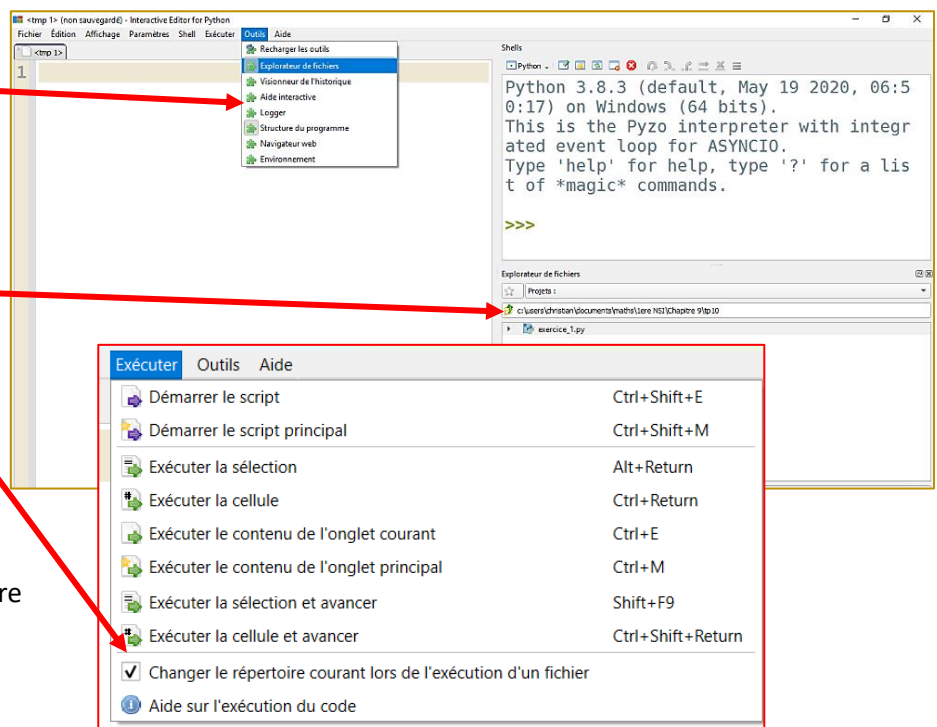
⇒ Créer un dossier nommé `tp10` dans votre espace de travail sur `U:\`

⇒ Dans Pyzo, dans l'onglet *Outils*, afficher la fenêtre « *Explorateur de fichier* ».

⇒ Dans cet explorateur qui s'affiche, se positionner dans ce répertoire `tp10` qui se trouve sur `U:\`

⇒ Dans l'onglet *Exécuter* de Pyzo, ne pas oublier de cocher la case « *changer le répertoire courant lors de l'exécution d'un fichier* »

⇒ Enregistrer votre fichier `.py`, ... vide pour l'instant, dans ce répertoire `tp10`.



Avertissement : On vous donne dans ce tp peu d'indications pour réaliser les codes attendus. Vous avez ainsi une certaine liberté pour coder de telle ou telle façon. Il n'y a pas forcément de « *meilleures solutions* ». Les fonctions à créer se construisent petit à petit, morceau par morceau, en ayant toujours le souci de tester sans cesse ce que vous faites. N'hésitez pas à placer des `print()` ici et là pour contrôler le contenu des variables.

N'hésitez pas à vous aider en écrivant des lignes sur un brouillon. N'hésitez pas à revoir vos polys de cours papiers ou en ligne pour savoir comment on procède pour accéder à un fichier, comment le lire ou y écrire des lignes, comment créer un liste, comment remplir une liste, comment transformer un string en entier, etc Lisez attentivement le message renvoyé par les erreurs d'exécution, Bien sûr ça ne marche pratiquement jamais du 1^{er} coup



2- ECRITURE DANS UN FICHER :

Une fonction nommée *saisirNotes()* est exécutée dans la console. On obtient alors l'affichage suivant et le code attend une saisie :

```
>>> saisirNotes()  
Prénom (ou rien écrire):
```

Après saisie d'un prénom (ici *albert*), on obtient l'affichage suivant et le code attend une autre saisie :

```
>>> saisirNotes()  
Prénom (ou rien écrire): albert  
Note sur 20 : |
```

Après saisi d'une note (ici *12*), on obtient l'affichage suivant et le code attend à nouveau une saisie :

```
>>> saisirNotes()  
Prénom (ou rien écrire): albert  
Note sur 20 : 12  
-----  
Prénom (ou rien écrire):
```

... ces mêmes saisies se répètent indéfiniment tant que la saisie du Prénom n'est pas vide : si à ce moment, on tape sur la touche Entrée sans rien écrire, la boucle des saisies s'arrête et on affiche « *fin de saisie* ». L'exécution du code est terminée.

En réalisant l'exécution ci-contre, cette fonction permet d'écrire dans un fichier texte, nommé « *notes.txt* », le contenu ci-dessous :

```
1 albert,12  
2 capucine,8  
3 gertrude,17  
4
```

Si ce fichier « *notes.txt* » n'existe pas, il est créé automatiquement. S'il existe déjà, les données saisies sont ajoutées en fin de fichier (mode append).

Par exemple, en exécutant une seconde fois la fonction avec les saisies ci-contre :

```
>>> saisirNotes()  
Prénom (ou rien écrire): albert  
Note sur 20 : 12  
-----  
Prénom (ou rien écrire): capucine  
Note sur 20 : 8  
-----  
Prénom (ou rien écrire): gertrude  
Note sur 20 : 17  
-----  
Prénom (ou rien écrire):  
fin saisie  
  
>>>
```

```
>>> saisirNotes()  
Prénom (ou rien écrire): théophile  
Note sur 20 : 19  
-----  
Prénom (ou rien écrire):  
fin saisie
```

Le contenu du fichier sera le suivant :

```
1 albert,12  
2 capucine,8  
3 gertrude,17  
4 théophile,19  
5
```

⇒ Ecrire le script python de cette fonction *saisirNotes()*

Aide : cette fonction comporte une boucle *while* avec des *input()* à l'intérieur. Si la saisie est vide on sort de cette boucle. Mais avant de quitter la fonction, ne pas oublier de fermer l'accès fichier.

```
def saisirNotes():  
    fichier = open("notes.txt","w",encoding='utf-8')  
    prenom = input("Prénom (ou rien écrire): ")  
    ligne = prenom + ", "  
    fichier.write(ligne)
```

Pour ceux qui n'arrivent pas à démarrer, essayer, commencer avec ce code-là :

3- LIRE DES DONNEES DANS UN FICHIER :

Une fonction nommée *lireNotes()* est exécutée dans la console. On obtient alors l'affichage suivant :

```
>>> lireNotes()  
La moyenne des notes est : 14.0
```

Cette fonction calcule la moyenne des notes qui sont écrites dans le fichier *notes.txt*, créé et complété par la fonction *saisirNotes()* précédente.

Le script de cette fonction permet ainsi d'ouvrir le fichier *notes.txt*, de lire toutes les lignes qui y sont écrites, d'en extraire les valeurs des notes, pour pouvoir calculer en même temps la moyenne.

⇒ Ecrire le script python de cette fonction *lireNotes()*

4- CHANGER LES DONNEES DANS UN FICHIER :

Le contenu du fichier *notes.txt* étant celui donné ci-contre, une fonction nommée *changerNotes()* est exécutée dans la console.

On obtient alors l'affichage suivant :

```
>>> changerNotes("albert",18)
```

Cette exécution change la note d'albert dans le fichier. Son contenu devient celui-ci-contre :

Si la fonction est exécutée avec un prénom qui n'est pas présent dans le fichier, le code s'arrête normalement sans provoquer d'erreurs. Le message suivant est écrit :

```
>>> changerNotes("gérard",18)  
gérard n'est pas dans ce fichier
```

```
1 albert,12  
2 capucine,8  
3 gertrude,17  
4 théophile,19  
5
```

```
1 albert,18  
2 capucine,8  
3 gertrude,17  
4 théophile,19  
5
```

⇒ Ecrire le script python de cette fonction *changerNotes()*

Aide : Dans cette fonction il s'agit dans un premier temps d'accéder au fichier en mode 'r'. On ferme ensuite l'accès et on y accède à nouveau en mode 'w' ..., si le prénom mis en argument est présent dans le fichier.

5- BONUS :

Modifier le script de la fonction *saisirNotes()* écrite dans le paragraphe 2 afin que les saisies des prénom et nom entraînent l'ajout d'une nouvelle ligne dans le fichier, uniquement si le prénom n'y est pas déjà présent. Dans le cas contraire, la note de la personne sera uniquement mise à jour.