**Tâche 1 : Obtenir la liste de toutes les fonctions permettant d’interagir avec des données externes.**

**À noter** : J’ai sélectionné seulement les paramètres que j’ai trouvés pertinents.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Librairie | Nom de la fonction | Principaux arguments | Exemple | Mode de chargement | Commentaires |
| native,  os, io | open (name, mode) | name : string indiquant le nom du fichier à ouvrir  mode : string indiquant comment le fichier sera ouvert | open(‘file.txt’, ‘r+’) | Lecture et écriture | La lecture ou l’écriture dépend des valeurs du mode.  ‘r’ : lecture seulement (défaut)  ‘w’ : écriture  ‘x’ : création du fichier s’il n’existe pas déjà  ‘a’ : ouvre en écriture et ajoute à la fin du fichier les éléments à ajouter  ‘b’ : mode binaire  ‘t‘ : mode texte (par défaut)  ‘+’ : ouvre en modification (lecture et écriture)  (The Python Software Foundation, 2020) |
| native, os, io | close () | - | fic.close()  (avec fic la variable qui reçoit l’ouverture d’un fichier : fic= open("fichier.txt", mode)) | Fermeture | On peut utiliser le mot clé with pour ne pas être obligé de vérifier la fermeture du fichier (with open (…) as fic :  -----)  (The Python Software Foundation, 2020)  Aucun effet si le fichier est déjà fermé. |
| numpy | np.loadtxt (fname, dtype, delimiter)  (Pour fichier texte) | fname (string) : fichier à lire  dtype : type du tableau résultant (le type par défaut est float). Ce paramètre est facultatif  delimiter (string) : utilisé pour séparer les valeurs. Le délimiteur par défaut est l’espace. | np.loadtxt("fichier.txt", dtype=float, delimeter=',')  ou  np.loadtxt("fichier.txt ") | Ouverture | Il retourne les données contenues dans le fichier texte qu’on a.  (The SciPy community, 2020) |
| numpy | np.savetxt (fname, array)  (Pour fichier texte) | fname (string) : nom du fichier dans lequel on veut sauvegarder notre tableau  array : tableau à mettre dans le tableau  (The SciPy community, 2020) | np.savetxt("fichier.txt", myArray) | Sauvegarde |  |
| numpy | numpy.load(file, mmap\_mode=None, allow\_pickle=False)  (Pour fichier binaire) | file (string) : fichier à lire  mmap\_mode (char) : le mode d’ouverture. Si rien n’est fourni, alors par défaut le mode est lecture. (facultatif)  allow\_pickle (bool) : permet de charger des tableaux d’objets dans des fichiers npy. (facultatif) | np.load ("fichier.npy", "w") | Ouverture | par défaut, allow\_pickle est à false pour des raisons de sécurité. Il faut changer sa valeur car s’il est désactivé, on ne pourra pas charger de tableaux.  L’instance retournée doit être fermée pour éviter les fuites de descripteurs de fichiers.  (The SciPy community, 2020) |
| numpy | numpy.save (file, arr, allow\_pickle=True)  (Pour fichier binaire) | file (string) : fichier dans lequel les données doivent être sauvegardées.  arr : le tableau de données à sauvegarder  allow\_pickle (bool) : par défaut prend la valeur true. | np.save (fic, x)  avec fic la variable qui reçoit l’ouverture du fichier, et x le tableau à sauvegarder | Sauvegarde | Si le fichier est une chaîne de caractères ou un chemin d’accès, une extension .npy sera ajoutée au nom du fichier s’il n’en a pas déjà.  (The SciPy communit, 2020) |
| csv | open (name, mode, newline=’ ’) | newline : est utilisé parce que le csvfile est un fichier (csvfile est la variable qui reçoit l’ouverture du fichier).  (Python Software Foundation, 2020) | open(‘file.csv’, ‘w’, newline=’ ’) | Ouverture |  |
| csv | csv.reader (csvfile, dialect='excel', \*\*fmtparams) | csvfile : un objet itérateur qui renvoie une string chaque fois que sa méthode next() est appelée.  dialect (string) : paramètre qui définit un ensemble de paramètres spécifiques à un dialecte csv. (facultatif)  fmtparams : arguments qui peuvent être spécifiés pour redéfinir des paramètres de formatage particuliers dans le dialecte courant. (facultatif) | csv.reader (csvfile, delimiter= ' ') |  | Si csvfile est un fichier, il doit être ouvert avec newline=' '  (Python Software Foundation, 2020) |
| csv | csv.writer (csvfile, dialect= ‘excel’, \*\*fmtparams) | csvfile : peut être n’importe quel objet avec une méthode write ().  Les autres paramètres sont similaires à ceux de la précédente fonction. | csv.reader (csvfile, delimiter = ‘ ’) |  | Renvoie un objet transcripteur responsable de convertir les données de l'utilisateur en chaînes délimitées sur l'objet fichier-compatible donné.  Si csvfile est un fichier, il doit être ouvert avec newline=' '.  (Python Software Foundation, 2020) |

# Références

* Python Software Foundation. (2020, 06 18). *7. Les entrées/sorties*. Retrieved from Python: https://docs.python.org/fr/3.6/tutorial/inputoutput.html
* Python Software Foundation. (2020, juin 24). *csv — Lecture et écriture de fichiers CSV*. Retrieved from Python: https://docs.python.org/fr/3/library/csv.html?highlight=csv#module-csv
* The Python Software Foundation. (2020, juin 24). *Fonctions natives*. Retrieved from Python: https://docs.python.org/fr/3/library/functions.html#open
* The SciPy communit. (2020, mai 24). *numpy.save*. Retrieved from Numpy: https://numpy.org/doc/stable/reference/generated/numpy.save.html#numpy.save
* The SciPy community. (2020, mai 24). *numpy.load*. Retrieved from Numpy: https://numpy.org/doc/stable/reference/generated/numpy.load.html#numpy.load
* The SciPy community. (2020, mai 24). *numpy.loadtxt*. Retrieved from Numpy: https://numpy.org/doc/stable/reference/generated/numpy.loadtxt.html#numpy.loadtxt
* The SciPy community. (2020, mai 24). *numpy.savetxt*. Retrieved from Numpy: https://numpy.org/doc/stable/reference/generated/numpy.savetxt.html#numpy.savetxt

///**Commentaires**/////

* Je ne comprends pas vraiment le principe de pickled files ou pickled data.
* Je me dis que ce serait peut-être bon de mentionner aussi les fonctions qui permettent de savoir si on peut manipuler de telle ou telle manière des fichiers.

Exemple : Librairie io :

closed () : retourne true si le fichier est bel et bien fermé.