

I Manipulation de fichiers

Exercice1 : Ouverture et écriture dans un fichier

Modifier le répertoire courant de l'interpréteur

```
>>>from os import chdir
```

```
>>>chdir("/home/login/exercices") #Mettre ici le répertoire où sont stockés vos exercices
```

La première ligne permet d'importer la commande `chdir()` du module `os` contenant une série de fonctions permettant de dialoguer avec le système d'exploitation (*os = operating system*).

La deuxième ligne permet de spécifier le répertoire courant de l'interpréteur python, i.e. le répertoire où il va récupérer les fichiers.

Tapez le code suivant et observez le résultat.

```
>>> f=open('test.txt','w') # Pour ouvrir un fichier en mode écriture
>>> f.write("Bonjour\n") # Pour écrire dans le fichier
>>> f.write("Ceci est un test d'écriture dans un fichier")
>>> f.close() # Pour fermer le fichier - vérifier le fichier dans un éditeur de texte
>>> f=open('test.txt','r') # # Pour ouvrir un fichier en mode lecture
>>> b=f.read() # Pour lire tout le fichier
>>> print b
>>> f.close() # Pour fermer le fichier
>>> f=open('test.txt','r') # # Pour ouvrir un fichier en mode lecture
>>> b=f.read(3) # Pour lire 3 caractères à partir de la position courante du curseur
>>> print b
>>> f.close()
>>> f=open('test.txt','r')
>>> b = f.readline() # Pour lire une ligne dans le fichier
>>> print b
>>> b = f.readlines() # Pour lire toutes les lignes et les stocker dans une liste
>>> print b
>>> f.close()
>>> f=open('test.txt','a') # Pour ouvrir le fichier en mode ajout
>>> f.write("\n Fin")
>>> f.close()
>>> f=open('test.txt','r')
>>> b = f.readlines()
>>> print len(b)
>>> print b
>>> f.close()
>>> f=open('test.txt','r')
>>> print f.read() # Pour lire tout le fichier - déplace le curseur à la fin du fichier
>>> print f.read()
>>> f.close()
>>> f=open('test.txt','r')
>>> print f.read()
>>> f.seek(0,0) # Déplace le curseur de 0 caractère depuis le début du fichier (codé par 0)
>>> print f.read()
>>> f.seek(-10,2) # Déplace le curseur de 10 caractères en arrière
                    depuis la fin du fichier (codée par 2)
```

```
>>> print f.read()
>>> f.seek(-10,2)
>>> f.seek(-5,1) # Déplace le curseur de 5 caractères en arrière
                  depuis la position courante (codée par 1)
>>> print f.read()
>>> f.close()
```

Exercice 2 : Copie de fichiers

Récupérez le fichier CopieFichier.py

Etudiez le code en ouvrant le fichier dans un éditeur de texte et exécutez le.

Exercice 3 : Copie des variables dans un fichier

Tapez le code suivant et observez le résultat.

```
>>> a = 5
>>> b = 2.83
>>> c = 67
>>> f = open('test.txt', 'w')
>>> f.write(str(a)) # Pour écrire un entier converti en string
>>> f.write(str(b))
>>> f.write(str(c))
>>> f.close()
>>> f = open('test.txt', 'r')
>>> print f.read()
>>> f.close()

>>> import pickle # Pour importer un module permettant de conserver le type des variables
>>> f = open('Monfichier', 'w')
>>> pickle.dump(a, f) # Pour enregistrer dans le fichier
>>> pickle.dump(b, f) # équivalent à write mais en conservant le type de b
>>> pickle.dump(c, f)
>>> f.close()
>>> f = open('Monfichier', 'r')
>>> print f.read()
>>> f.close()
>>> f = open('Monfichier', 'r')
>>> t = pickle.load(f) # Opération inverse de dump
>>> print t, type(t)
>>> t = pickle.load(f) # Equivalent à read mais en conservant le type de b
>>> print t, type(t)
>>> t = pickle.load(f)
>>> print t, type(t)
>>> f.close()
```

II Programmation Orientée Objet

Exercice 1 : Première classe

Récupérez le fichier `ExempleClasse1.py`

étudiez le code en ouvrant le fichier et exécutez le.

Tapez le code suivant et observez le résultat.

```
>>> import ExempleClasse1
>>> c1 = ExempleClasse1.CompteBancaire('Toto', 1000)
>>> c1.depot(350)
>>> c1.retrait(200)
>>> c1.affiche()
>>> print c1.nom
>>> print c1
```

Exercice 2 : Accessibilité

Reprendre la classe de l'exercice précédent et remplacer `nom` par `__nom`. Exécutez à nouveau l'exemple de l'exercice précédent. Observez le résultat.

Exercice 3 : Objet complexe

Récupérez le fichier `rect_carre.py` étudiez le code en ouvrant le fichier et exécutez le.

Exercice 4 : Héritage

Récupérez le fichier `formes.py` étudiez le code en ouvrant le fichier et exécutez le.

NB : L'héritage multiple est possible en Python.