

I22 : TP5

Axel COEZARD

Lecture d'un fichier texte

1. Affichez les 12 premiers caractères du fichier `texte1` en utilisant au moins 2 instructions `read()`:

```
with open("texte1", "r+") as f:  
    print(f.read(11), end="\n\n")
```

```
kakesinfo@laptop:~/Workspace/I22$ python3 tp_5.py  
Le corbeau
```

2. Affichez la première et la troisième ligne du même fichier en utilisant l'instruction `readline()`:

```
with open("texte1", "r+") as f:  
    print(f.readline(), end="\n\n")  
    f.readline()  
    print(f.readline(), end="\n\n")
```

```
Le corbeau et le renard.
```

```
« Maître Corbeau, sur un arbre perché,
```

3. Affichez la seconde et quatrième ligne du même fichier en utilisant l'instruction `readlines()` (attention au `s` à la fin de `readlines()`)

```
with open("texte1", "r+") as f:  
    lines = f.readlines()  
    print(lines[1], lines[3])
```

```
Cette fable de Jean de La Fontaine commence ainsi :  
Tenait en son bec un fromage. »
```

1. Lisez et affichez les trois premières lignes du fichier `texte1`:

```
with open("texte1", "r+") as f:
    for i in range(3):
        print(f.readline())
```

```
kakesinfo@laptop:~/Workspace/I22$ python3 tp_5.py
Le corbeau et le renard.

Cette fable de Jean de La Fontaine commence ainsi :

« Maître Corbeau, sur un arbre perché,
```

2. Relisez et affichez les trois premiers caractères du même fichier:

```
with open("texte1", "r+") as f:
    print(f.read(3))
```

```
Le
```

Écriture d'un fichier texte

1. Créez le fichier nommé `fichier_test` dans votre répertoire de travail `I22` et écrivez dans celui-ci le texte suivant : « *La cigale et la fourmi* » .
2. À partir d'un autre terminal, affichez avec `cat` ou `less` le contenu du fichier que venez de créer. Tout s'est-il bien passé ?

```
kakesinfo@laptop:~/Workspace/I22$ cat fichier_test
La cigale et la fourmi
kakesinfo@laptop:~/Workspace/I22$
```

Oui, tout c'est bien passé.

3. Ajoutez à la suite de ce fichier, sur une nouvelle ligne, le texte « *Le scorpion et la grenouille* ».
4. Toujours à partir de l'autre terminal, vérifiez que le contenu du fichier correspond.

```
kakesinfo@laptop:~/Workspace/I22$ cat fichier_test
La cigale et la fourmi
Le scorpion et la grenouille
```

Le contenu du fichier correspond

Lecture/écriture binaire

1. Lisez et affichez les 4 premiers caractères du fichier `valeurs`. Retrouvez-vous bien les valeurs attendues, comme avec `od` ? Pourquoi ?

```
with open("valeurs", "rb+") as f:
    print(f.read(4))
```

Non, je n'obtiens pas les valeurs attendues qu'on avait avec `od`, mais cela:

```
b'\xc0\xff\xfe\xff'
```

car la fonction `read()` lit par défaut le texte encodé en utf-8.

2. Dans un nouveau fichier `fichier_test_bis` dans votre répertoire de travail, ajoutez les nombres 1337, -2 et 10.25.
3. À partir d'un autre terminal, vérifiez le contenu du fichier. Retrouvez-vous la même chose qu'en utilisant la commande `od` comme en début de sujet ?

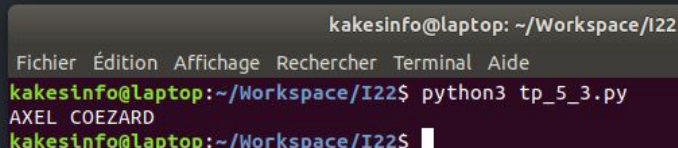
Non

Mise en pratique

1. Écrivez la fonction `majuscule()` qui prend en paramètre une chaîne de caractères et qui retourne la même chaîne mais en majuscule. Pour cela vous utiliserez les fonctions `ord()` et `chr()` et l'astuce additive pour passer des minuscules aux majuscules et inversement. Vous ne convertirez que les 26 lettres de l'alphabet, pas les lettres accentuées ou autres caractères spéciaux et évidemment vous n'utiliserez pas la fonction `upper()`:

```
def majuscule(chaine):
    whitelist = "abcdefghijklmnopqrstuvwxyz"
    chrs = [chr(ord(c) - 32) if c in whitelist else c for c in chaine]
    return "".join(chrs)

print(majuscule("axel coezard"))
```



The screenshot shows a terminal window titled "kakesinfo@laptop: ~/Workspace/I22". It contains a menu bar with "Fichier", "Édition", "Affichage", "Rechercher", "Terminal", and "Aide". The terminal shows the command `python3 tp_5_3.py` being executed, which outputs `AXEL COEZARD`. The prompt then returns to `kakesinfo@laptop:~/Workspace/I22$`.

2. Écrivez la fonction `val2ascii()` qui prend en paramètre un nombre entier positif et retourne sa représentation en chaîne de caractères à l'aide d'une astuce additive (évidemment sans utiliser la fonction `srt()`):

3. Le fichier `/home/partage/l22/data_1` contient une séquence de 512x512 nombres entiers codés sur un octet non signé. Écrivez la fonction `affiche2D()` qui prend en paramètre un nom de fichier et affiche son contenu sous forme d'image:

4. Le fichier `/home/partage/l22/data_2` contient une double séquence de nombres entiers signés codés sur 2 octets. Chaque nombre appartient alternativement à la séquence G ou à la séquence D. Écrivez la fonction `affiche_2_canaux()` qui prend en paramètre le nom d'un fichier et affiche séparément les deux séquences de nombres sous forme de deux graphiques: