جامعة هواري بومدين
للعلوم والتكنولوجيا
U S T H B

Programmation PYTHON

Cours 4

Nassim ZELLAL

2020/2021

Les fichiers

- Les fichiers se trouvent dans tous les langages de programmation, mais la manière très simple et très puissante de les manipuler, fait des fichiers, une facilité de **Python**.

Ouverture d'un fichier en écriture - mode « w » (write)

- var= open('fic.txt','w')#on obtient un objet de type « file »
- var.write("Je suis devant l'école!\n")
- var.close()
- #----------#
- var= open('fic.txt','w') #on obtient un objet de type « file »
- var.write('Je suis devant l\'école!\n')
- var.close()

Ouverture d'un fichier en écriture - mode « a » (append)

- var= open('fic.txt', 'a') #on obtient un objet de type « file »
- var.write("Et je ne sais pas où me garer.\n")
- var.close()
- #--#
- var= open('fic.txt', 'a') #on obtient un objet de type « file »
- var.write('Et je ne sais pas où me garer pour l\'appeler.\n')
- var.close()

Ouverture d'un fichier en lecture - mode « r » (read)

- var = open('fic.txt','r') #on obtient un objet de type « file »
- res=var.read()
- print(res,end="")
- var.close()
-
- #La méthode read() renvoie une « chaîne de caractères » contenant l'intégralité du fichier.
- #Le paramètre **end=""** supprime le saut de ligne généré par défaut par la fonction print()

Ouverture d'un fichier en lecture - readlines()

- var = open('fic.txt','r') #on obtient un objet de type « file »
- print(var.readlines())
- var.close()

- #La méthode readlines() renvoie une « liste » contenant l'intégralité du fichier.

Ouverture d'un fichier en lecture - readline()

- var = open('fic.txt','r') #on obtient un objet de type « file »
- print(var.readline())
- var.close()

- #La méthode readline() renvoie une chaîne de caractères « str » contenant une seule ligne du fichier.

Parcourir un fichier ligne par ligne - readlines()

- var = open('fic.txt','r') #on obtient un objet de type « file »
- result=var.readlines()
- x=1
- for i in result:
- 	print(x,i) #ou bien print(str(x)+i)
- 	x+=1
- var.close()

Parcourir un fichier ligne par ligne - readlines()

- var = open('fic.txt','r') #on obtient un objet de type « file »
- res=var.readlines()
- x=1
- for i in range(len(res)):#ou bien for i in res:
- print(x,res[i]) #ou bien print(str(x)+res[i])#ou bien print(x,i)
- x+=1
- var.close()

Parcourir un fichier ligne par ligne - readline()

- var = open('fic.txt','r') #on obtient un objet de type « file »
- x=1
- while True: #ou bien while 1:
- texte=var.readline()
- if texte ==''': #ou bien **if not texte**#chaîne vide avec guillemets simples ou doubles
- break #expression/mot-clé « break »
- else:
- print(x,texte)
- x+=1
- var.close()

Boucler sur un objet « file »

- file = open('fic.txt', 'r')
- for a in file:
  - print(a)

#----------Ou bien----------#

- for a in open('fic.txt', 'r'):
  - print(a)

Les exceptions - try/except/finally

```
try:
    var = open('fics.txt','r')
    print(var.read())
    var.close()
except FileNotFoundError:
    print("Fichier introuvable!")
finally:
    print("Il faut toujours vérifier le chemin du fichier!")
```

#Une clause « finally » est toujours exécutée avant de quitter l'instruction « try » qu'une exception ait été déclenchée ou non.

Les exceptions - try/except/finally

```
try:
    var = open('fic.txt','r')
except FileNotFoundError:
    print("Fichier introuvable!")
else:
    print(var.read())
    var.close()
finally:
    print("Il faut toujours vérifier le chemin du fichier!")
```

#Une clause « finally » est toujours exécutée avant de quitter l'instruction « try » qu'une exception ait été déclenchée ou non.

Lecture depuis l'entrée standard - sys.stdin

- import sys
- var=sys.stdin.readline()
- print(var.rstrip(),end='')#rstrip() supprime le saut de ligne généré lors de la saisie et le paramètre end=" celui qui est généré par défaut par print()
- #ou bien print(sys.stdin.readline().rstrip(),end='')

- **#stdin (standard input): entrée standard/flux d'entrée.**

Lecture depuis l'entrée standard - sys.stdin - for

- import sys
- for line in sys.stdin:
- print(len(line.rstrip()))

- #afficher la longueur de la chaîne saisie grâce à la fonction len()
- **#stdin : entrée standard.**

Lecture depuis l'entrée standard - sys.stdin - while

```
import sys
x=1
while 1:
    texte=sys.stdin.readline()
    if texte == "\n":
        break
    else:
        print(x,texte)
        x+=1
```

#stdin : entrée standard.

Lecture depuis l’entrée standard - input()

- a = input()
- print(a)
- ----------
- a = input("Veuillez saisir quelque chose: ")
- print(a)

Sortie standard - sys.stdout

- import sys
- sys.stdout.write("test")
- #sys.stdout ne génère pas un saut de ligne par défaut comme la fonction print()
- ----------
- import sys
- for line in sys.stdin:
- sys.stdout.write(line)
- **#stdout (standard output) : sortie standard/flux de sortie.**

Exercice 1

- **À faire :**
- ***Demander à l'utilisateur de donner un verbe du premier groupe.**
- ***Afficher sur la console les formes conjuguées au présent de l'indicatif (avec les pronoms associés)**
- **Ex :**
- **Je donne**
- **Tu donnes**
- **Il/Elle donne**
- **Nous donnons**
- **Vous donnez**
- **Ils/Elles donnent**

Exercice 2

- **Créer un fichier « f.txt », contenant la chaîne de caractères suivante :**
- **The concept of morality lies in thinking good and doing good**

- **Écrire un script Python qui :**
  - **prend en argument « f.txt » ;**
  - **affiche dans un autre fichier « res.txt » :**
  - **token ➔ fréquence d'occurrence**