

```

Entrée [13]: from scapy.all import *

print('Lecture contenue fichie Texte.txt, fichier au format text : \

fichier= open("Texte.txt",'rt')
texte= fichier.read()
print('READ :')
print(texte)
fichier.close()
#afficher le contenu complet du text

fichier= open("Texte.txt",'rt')
text= fichier.read(15)
print('READ (15):')
print(texte)
fichier.close()
#affiche seulement les 15 permier carractère

fichier= open("Texte.txt",'rt')
lignes= fichier.readline()
print('\nREADLINE :')
print(lignes)
fichier.close()
#affiche seulement la première ligne

fichier= open("Texte.txt",'rt')
lignes= fichier.readlines()
print('READLINE :')
print(lignes)
fichier.close()
#affiche sous forme de programme avec '' pour le texte \n pour les i

#com du fichier prof :
#r, (par défaut)ouverture en lecture seule (READ).
#w, ouverture en écriture (WRITE), à chaque ouverture le contenu du
#si fichier n'existe pas, Python le crée.

#- a, ouverture en écriture avec ajout (APPEND) à la fin du fichier,
#fichier n'existe pas, Python le crée.
#- x, ouverture en écriture seulement si le fichier n'existe pas.
#- b, pour une ouverture en mode binaire: les octets seront traités
#- t, (par défaut) pour une ouverture en mode texte: les octets sero
#caractères affichables.
#Une fois le fichier en lecture ouvert, on p

#read() qui retourne une chaîne d'octets contenant l'intégralité du
# read(x) qui retourne une chaîne contenant les x prochains octets c
# readline() qui retourne une chaîne contenant la prochaine chaîne c
# readlines() qui retourne une liste de chaque ligne du fichier. Cha
#\n" qui représente un retour à la ligne. Cette fonction ne concerne

```

Lecture contenue fichie Texte.txt, fichier au format text :

READ :

Ceci est un exemple de fichier texte.
 Son objectif est de démontrer les différentes possibilités offertes par Python pour accéder à son contenu.
 Un fichier texte ne contient par définition QUE des caractères affichables.
 Les caractères affichables incluent également les caractères de tabulation, de fin de ligne, ou autres caractères spéciaux considérés néanmoins comme affichables.

READ (15):
 Ceci est un exemple de fichier texte.
 Son objectif est de démontrer les différentes possibilités offertes par Python pour accéder à son contenu.
 Un fichier texte ne contient par définition QUE des caractères affichables.
 Les caractères affichables incluent également les caractères de tabulation, de fin de ligne, ou autres caractères spéciaux considérés néanmoins comme affichables.

READLINE :
 Ceci est un exemple de fichier texte.

READLINE :
 ['Ceci est un exemple de fichier texte.\n', 'Son objectif est de démontrer les différentes possibilités offertes par Python\n', 'pour accéder à son contenu.\n', 'Un fichier texte ne contient par définition QUE des caractères affichables.\n', 'Les caractères affichables incluent également les caractères de tabulation, de\n', 'fin de ligne, ou autres caractères spéciaux considérés néanmoins comme\n', 'affichables ']

Entrée [14]:

```
print('Lecture du contenu du fichier Texte.pdf (pdf au format binaire) et affichage des 30 premières lignes :')
fichier = open("Texte.pdf", 'rb')
texte = fichier.readlines() # affichage des lignes
print(texte[0:30]) # de 0 à 30
# dans le résultat du programme le début est "b" pour binaire
```

Lecture du contenu du fichier Texte.pdf (pdf au format binaire) et affichage des 30 premières lignes :

```
-----
-----
FileNotFoundError                                Traceback (most recent call last)
/tmp/ipykernel_3609/4024652281.py in <module>
      1 print('Lecture du contenu du fichier Texte.pdf (pdf au format binaire) et affichage des 30 premières lignes :')
----> 2 fichier = open("Texte.pdf", 'rb')
      3 texte = fichier.readlines()
      4 print(texte[0:30])
```

FileNotFoundError: [Errno 2] No such file or directory: 'Texte.pdf'

```
Entrée [15]: fichier = open("Texte2.txt",'at')

fichier.write(" \nAvec le mode 'a' le contenu du fichier n'est pas
fichier.close()
fichier = open("Texte2.txt",'rt')
texte = fichier.read()
print(texte)
#- a, ouverture en écriture avec ajout (APPEND) à la fin du fichier,

#important ne pas oublier le close() si pour fermer le fichier et qu

\nAvec le mode 'a' le contenu du fichier n'est pas effacé ...
```

Entrée []: