

Informatique 1
Introduction à la programmation

Mission 6 : introduction

Kim Mens Charles Pecheur Cristel Pelsser

Mission 6

- Objectifs
  - fichiers
  - exceptions

- Problème
  - Création d'un outil dans lequel l'utilisateur peut donner des commandes; doit continuer même si l'utilisateur donne des mauvaises commandes

Fichiers

Disque dur de l'ordinateur



128Go, 256Go, …

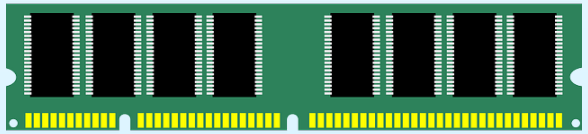
Plus lente

Contient de l'information même si l'ordinateur n'est pas allumé

Lire / Read

Écrire / Write

Mémoire de l'ordinateur



8Go, 16Go, …

Plus rapide

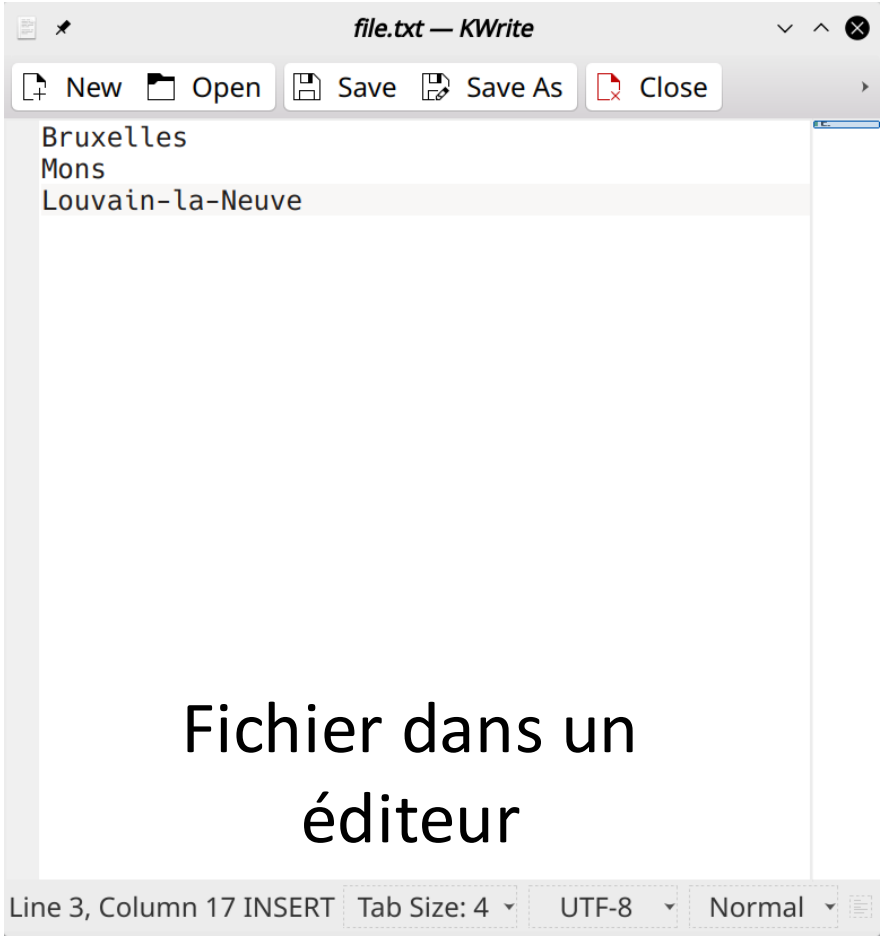
Contient de l'information seulement si l'ordinateur est allumé

Fichiers de Texte

B	r	u	x	e	l
l	e	s	\n	M	o
n	s	\n	L	o	u
v	a	i	n	-	l
a	-	N	e	u	v
e	\n	EOF			

`file.txt`

Disque dur de l'ordinateur



Fichier dans un éditeur

Lire Fichiers de Texte

approche simple

(1) Ouverture

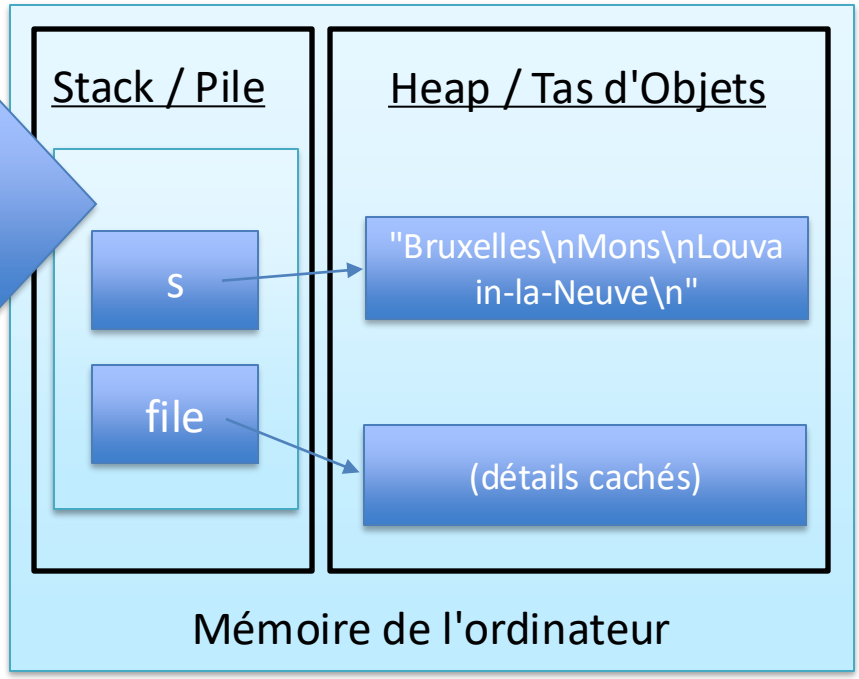
```
file = open ( "file.txt", "r" )
s = file.read ()
file.close ()
```

B	r	U	x	e	l
l	e	s	\n	M	o
n	s	\n	L	o	u
v	a	i	n	-	l
a	-	N	e	u	v
e	\n	EOF			

file.txt

Disque dur de l'ordinateur





Mémoire de l'ordinateur

Lire Fichiers de Texte

approche simple

(1) Ouverture

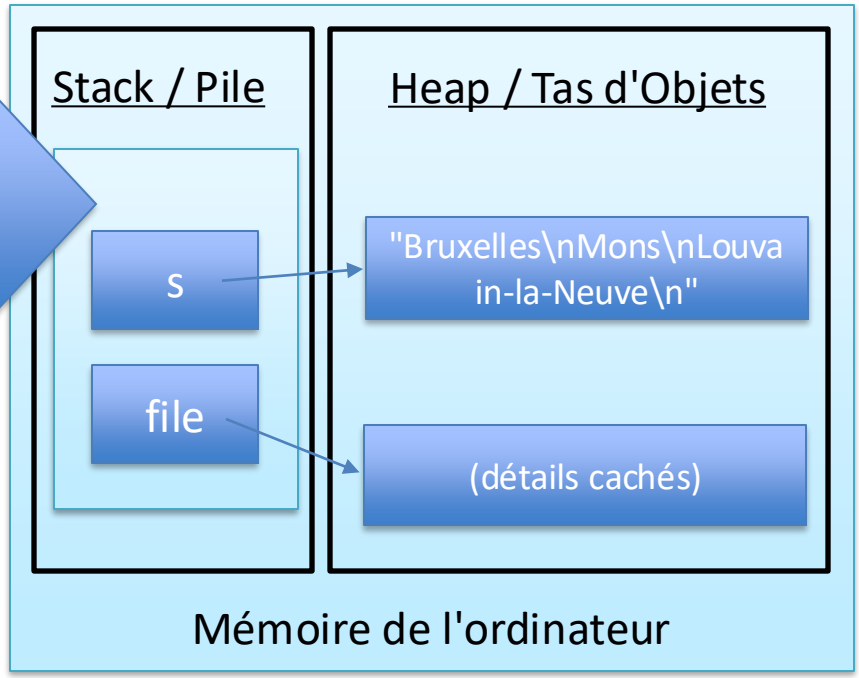
```
file = open ( "file.txt", "r" )
s = file.read ()  (2) Traitement
file.close ()
```

B	r	u	x	e	l
l	e	s	\n	M	o
n	s	\n	L	o	u
v	a	i	n	-	l
a	-	N	e	u	v
e	\n	EOF			

`file.txt`

Disque dur de l'ordinateur

Lire / Read



Lire Fichiers de Texte

approche simple

B	r	u	x	e	l
l	e	s	\n	M	o
n	s	\n	L	o	u
v	a	i	n	-	l
a	-	N	e	u	v
e	\n	EOF			

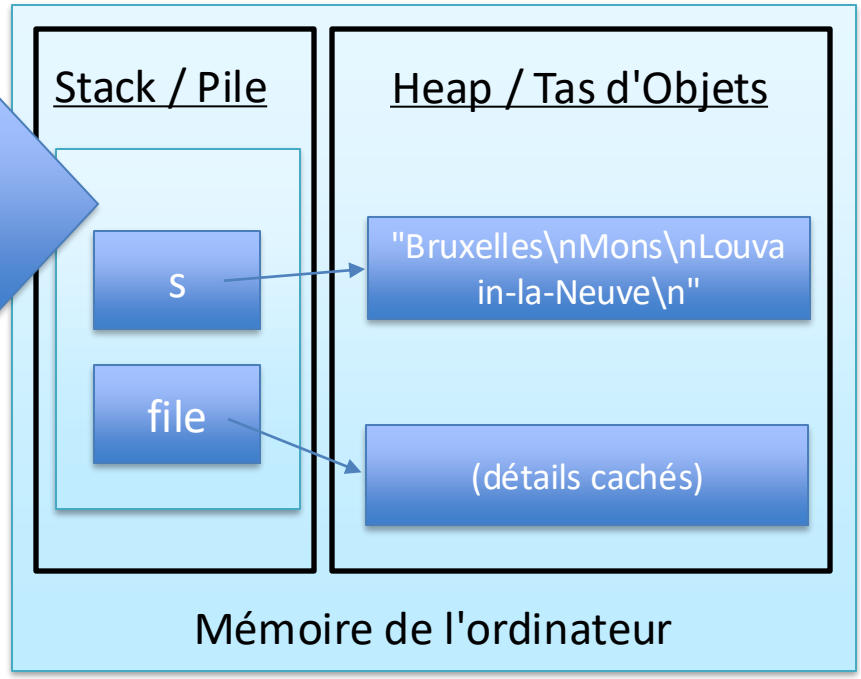
`file.txt`

Disque dur de l'ordinateur

(1) Ouverture

```
file = open ( "file.txt", "r" )
s = file.read ()  (2) Traitement
file.close ()  (3) Fermeture
```

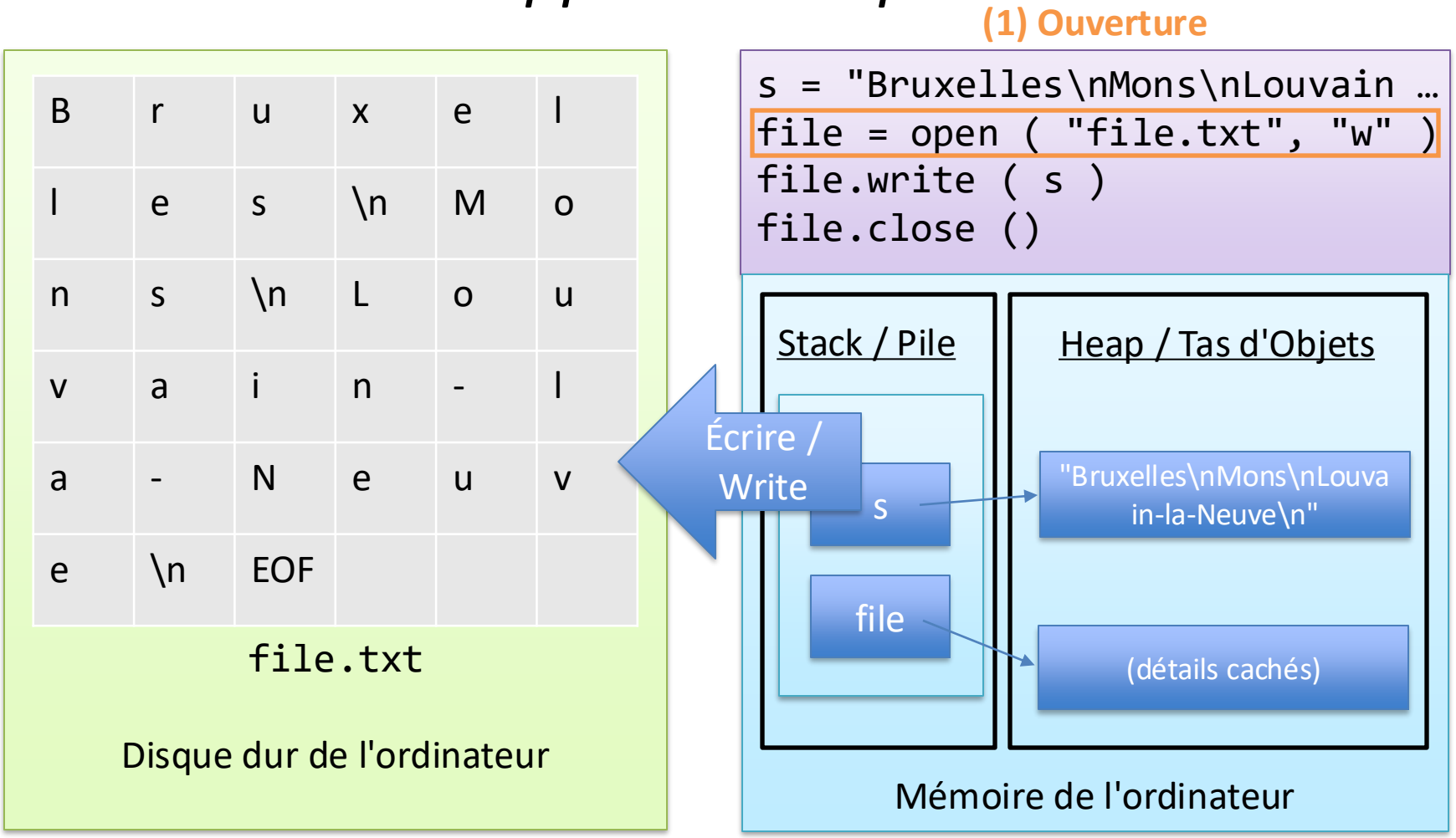
Lire / Read



Mémoire de l'ordinateur

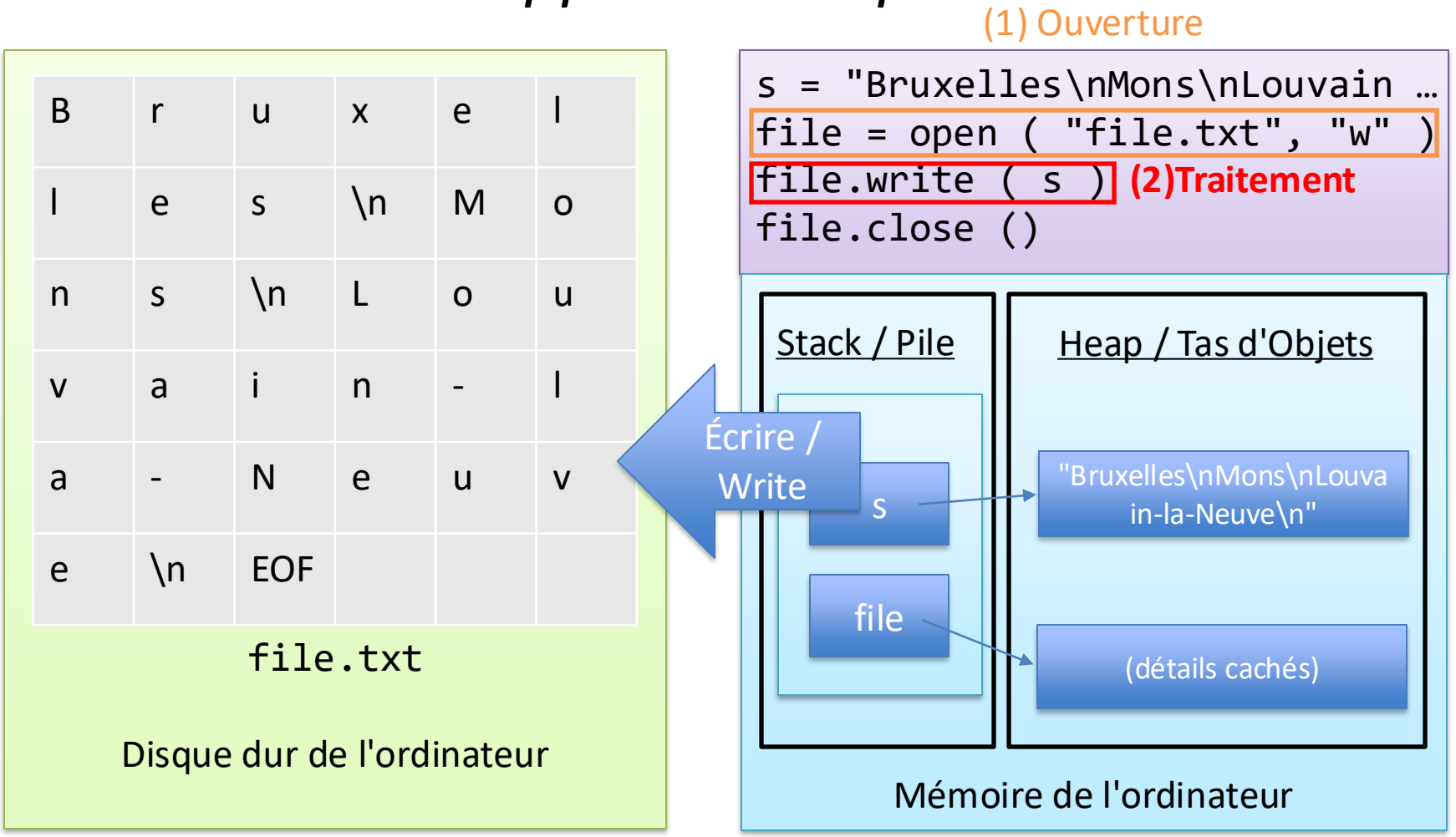
Écrire Fichiers de Texte

approche simple



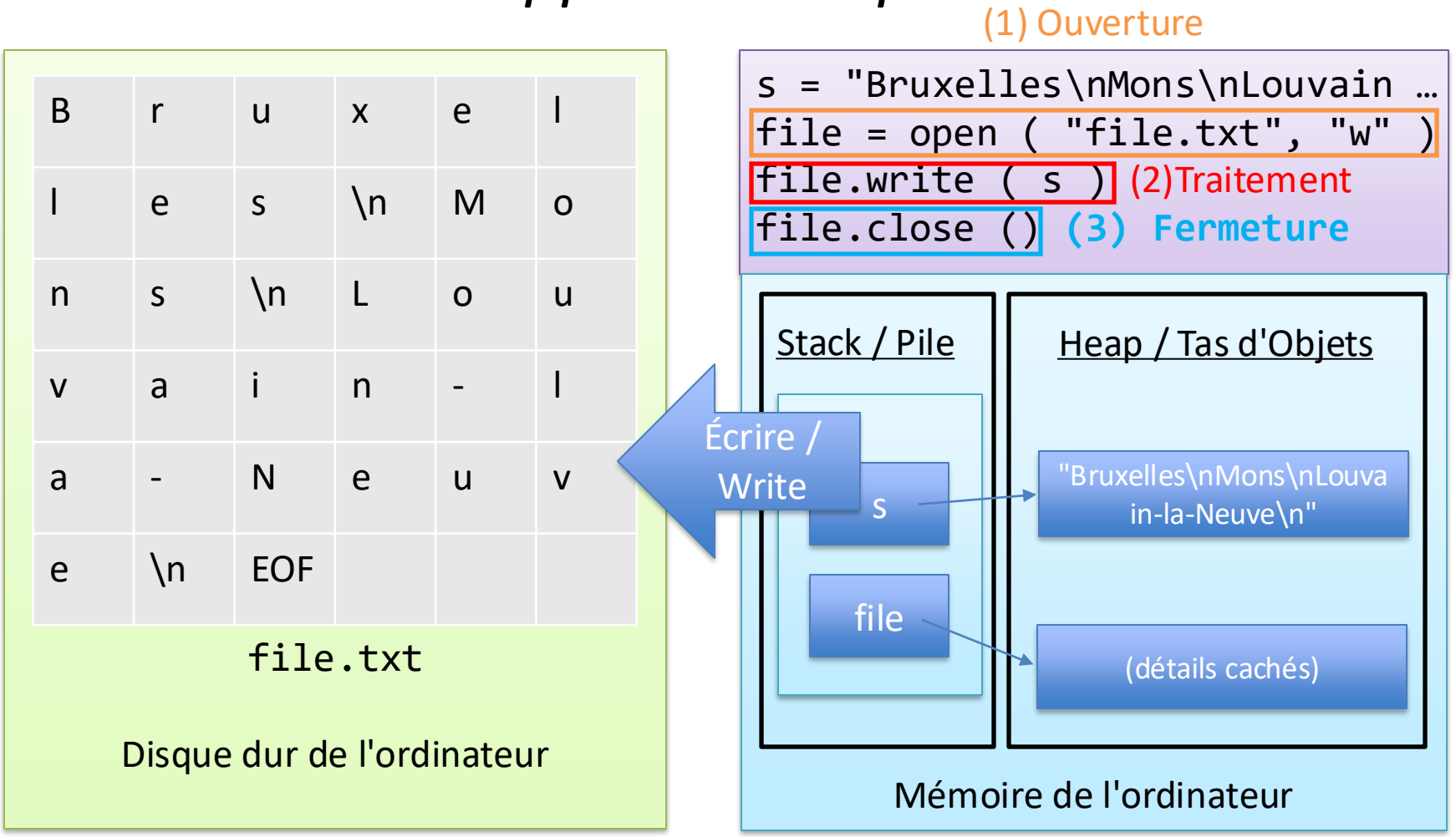
Écrire Fichiers de Texte

approche simple



Écrire Fichiers de Texte

approche simple



Lire Fichiers de Texte

boucle while

B	r	u	x	e	l
l	e	s	\n	M	o
n	s	\n	L	o	u
v	a	i	n	-	l
a	-	N	e	u	v
e	\n	EOF			

`file.txt`

Disque dur de l'ordinateur

Lire / Read

(1) Ouverture

```
file = open ( "file.txt", "r" )
s = file.readline ()
while s:
  print ( s )
  s = file.readline ()
file.close ()
```

(2) Traitement

(3) Fermeture

Mémoire de l'ordinateur

Écran de l'ordinateur

Lire Fichiers de Texte

boucle while

(1) Ouverture

```
file = open ( "file.txt", "r" )
s = file.readline ()
while s:
  print ( s )
  s = file.readline ()
file.close ()
```

B	r	u	x	e	l
l	e	s	\n	M	o
n	s	\n	L	e	u
v	a	i	n	-	l
a	-	N	e	u	v
e	\n	EOF			

file.txt

Disque dur de l'ordinateur

Lire / Read

file

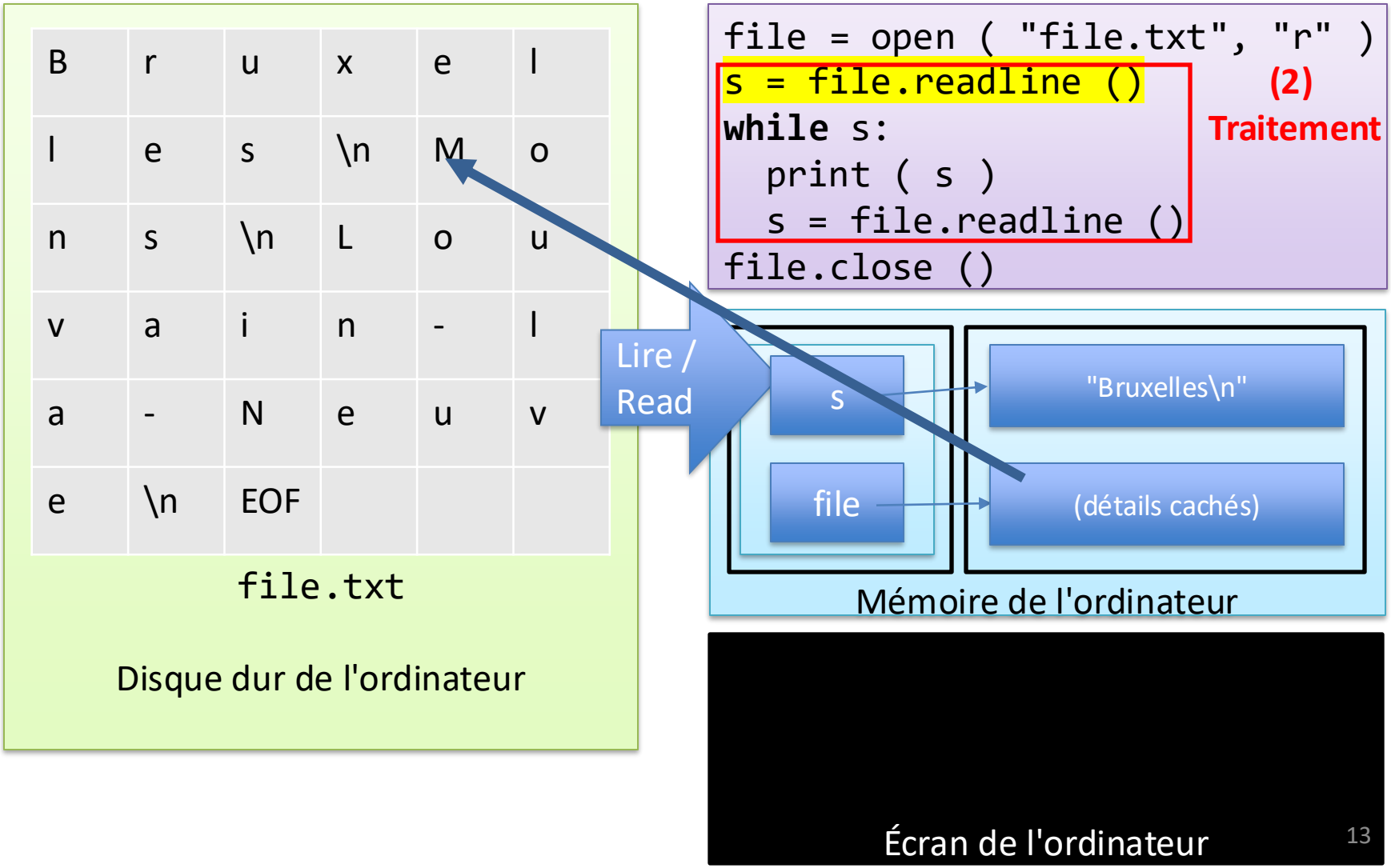
(détails cachés)

Mémoire de l'ordinateur

Écran de l'ordinateur

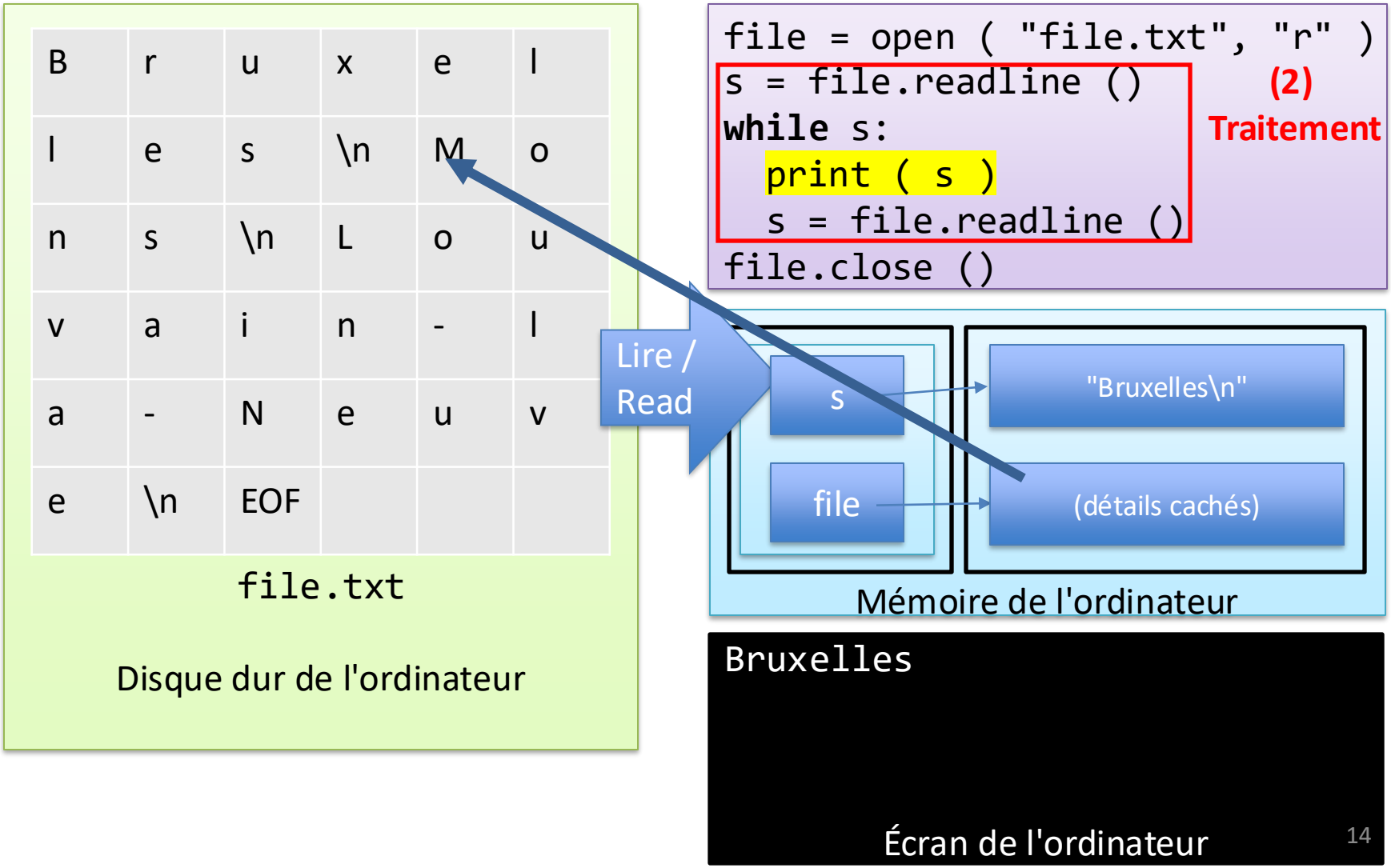
Lire Fichiers de Texte

boucle while



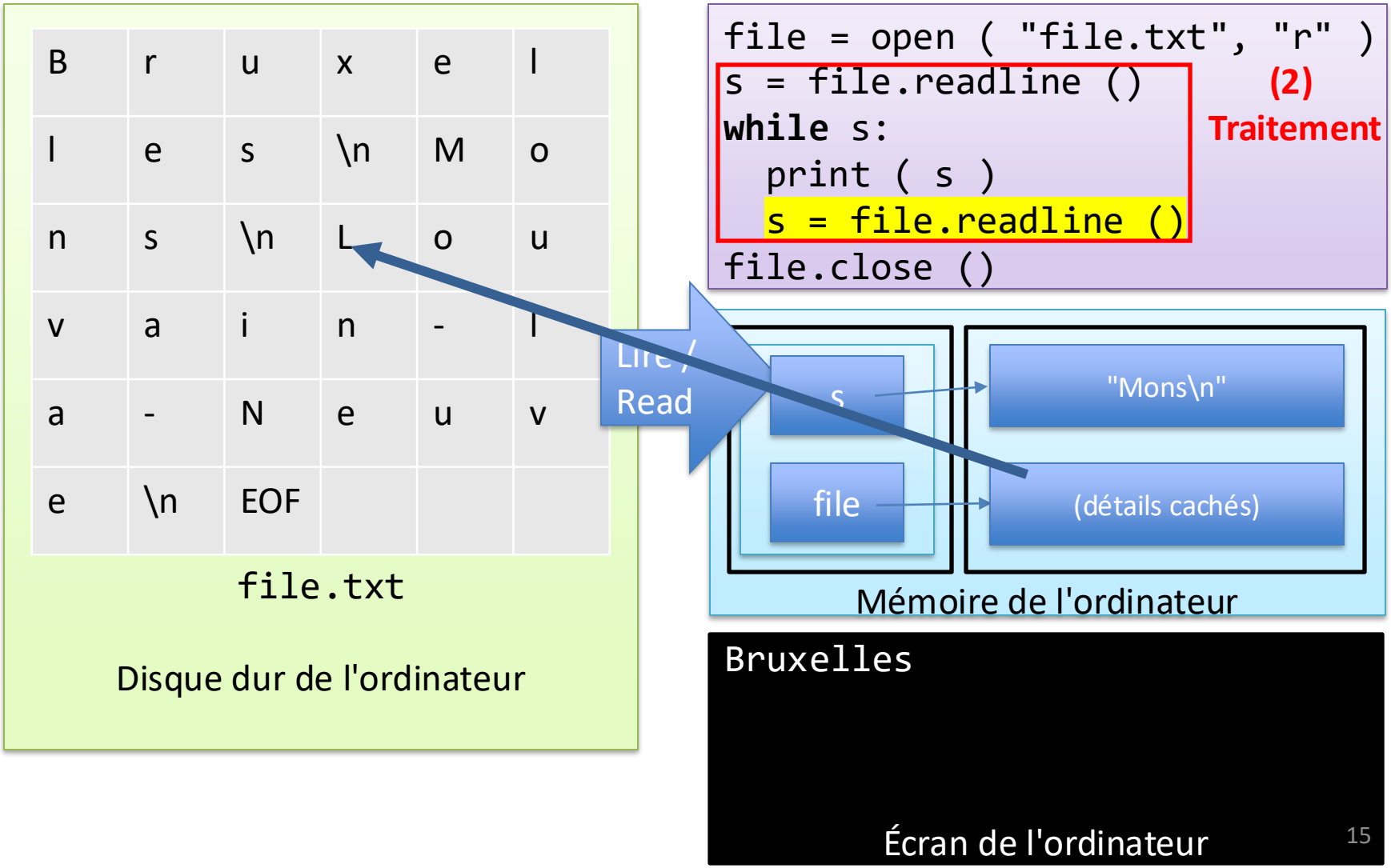
Lire Fichiers de Texte

boucle while



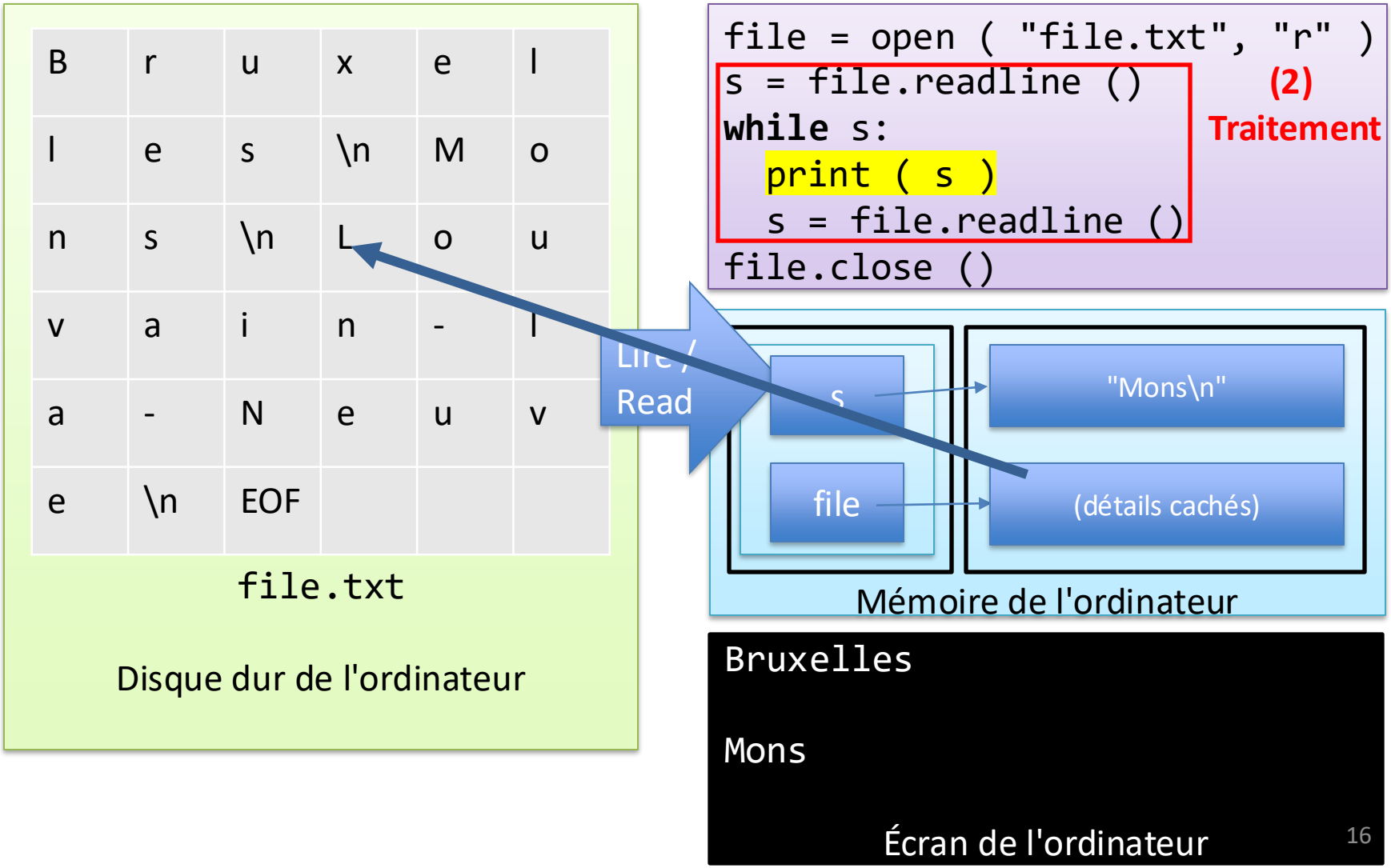
Lire Fichiers de Texte

boucle while



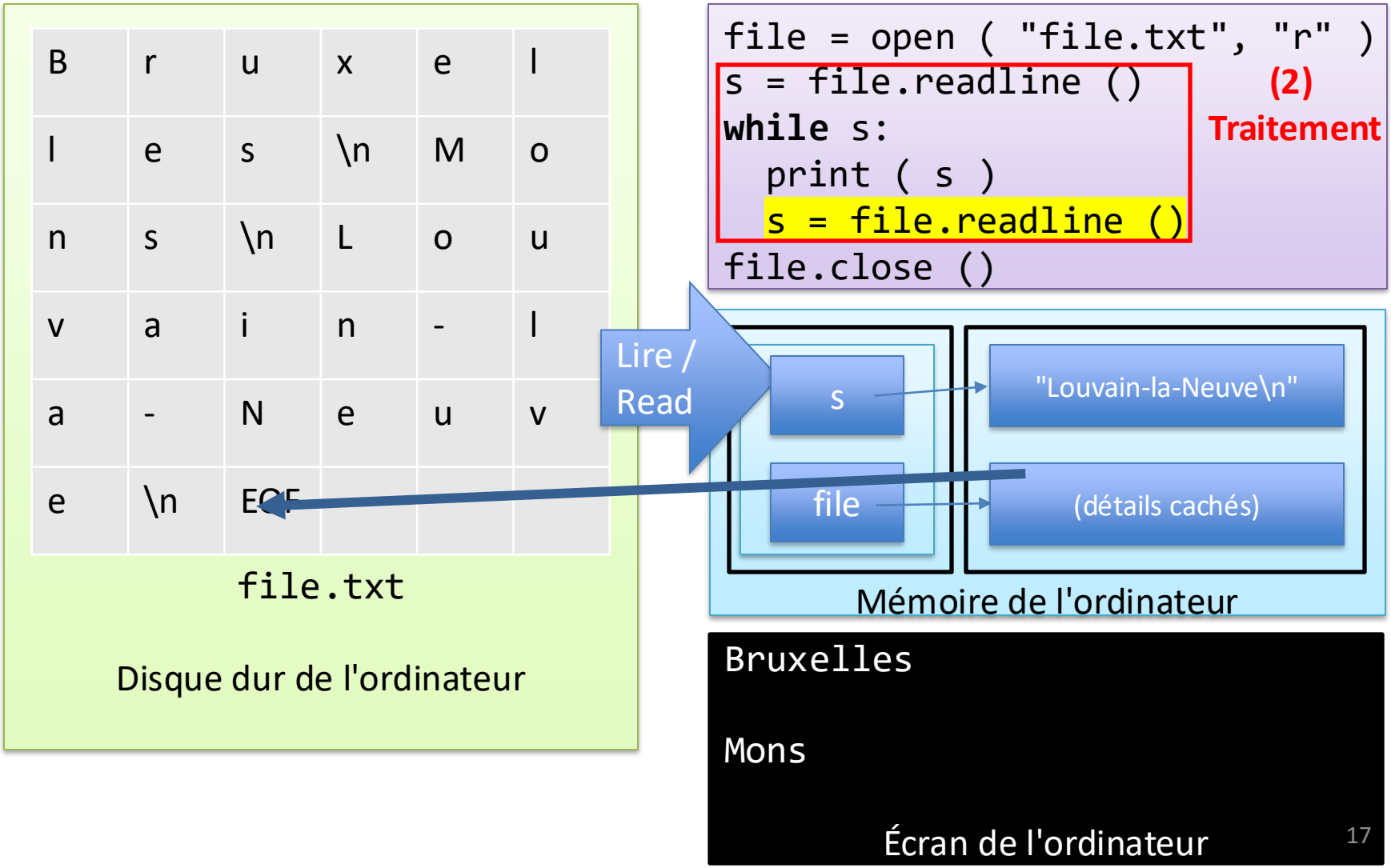
Lire Fichiers de Texte

boucle while



Lire Fichiers de Texte

boucle while



Lire Fichiers de Texte

boucle while

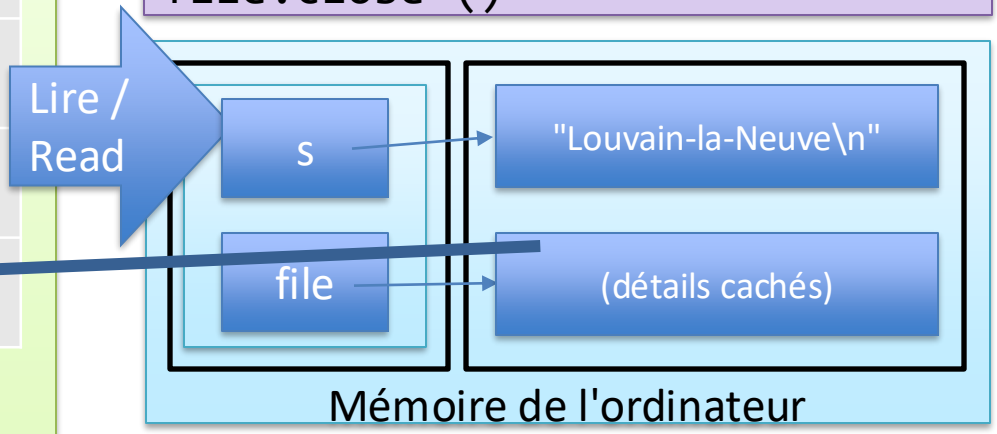
B	r	u	x	e	l
l	e	s	\n	M	o
n	s	\n	L	o	u
v	a	i	n	-	l
a	-	N	e	u	v
e	\n	EOF			

file.txt

Disque dur de l'ordinateur

```
file = open ( "file.txt", "r" )
s = file.readline ()
while s:
   print ( s )
   s = file.readline ()
file.close ()
```

(2) Traitement




```
Bruxelles

Mons

Louvain-la-Neuve
```

Lire Fichiers de Texte

boucle while

B	r	u	x	e	l
l	e	s	\n	M	o
n	s	\n	L	o	u
v	a	i	n	-	l
a	-	N	e	u	v
e	\n	EOF			

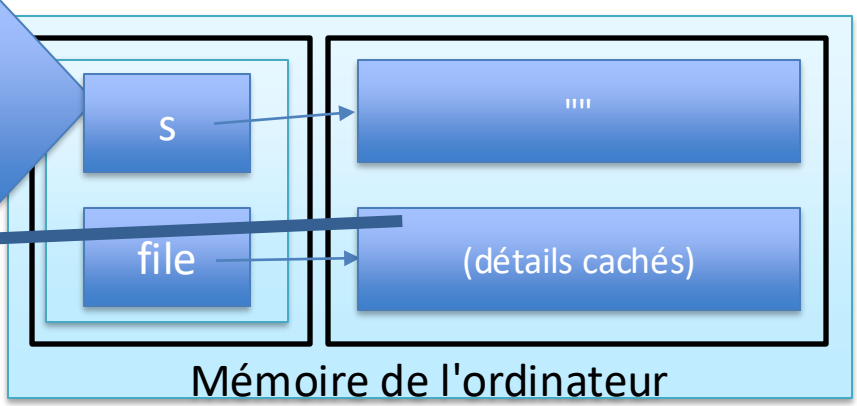
file.txt

Disque dur de l'ordinateur

```
file = open ( "file.txt", "r" )
s = file.readline ()
while s:
  print ( s )
  s = file.readline ()
file.close ()
```

(2) Traitement

Lire / Read




```
Bruxelles

Mons

Louvain-la-Neuve
```

Lire Fichiers de Texte

boucle while

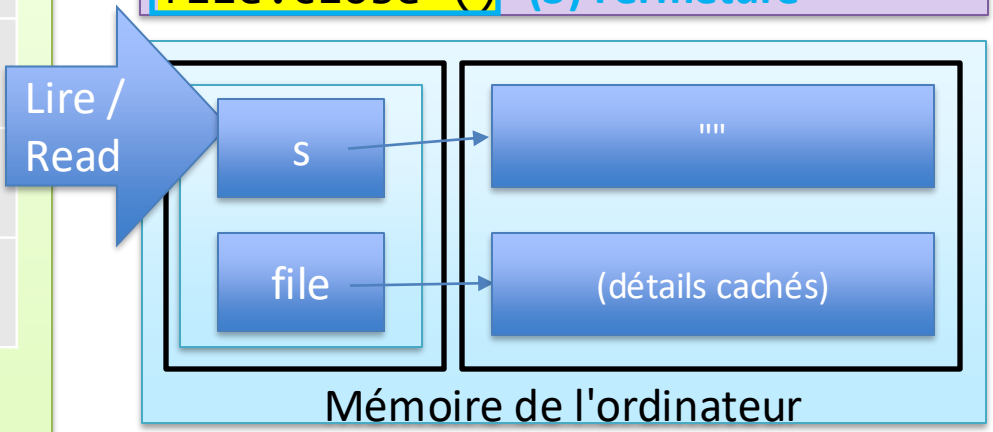
B	r	u	x	e	l
l	e	s	\n	M	o
n	s	\n	L	o	u
v	a	i	n	-	l
a	-	N	e	u	v
e	\n	EOF			

`file.txt`

Disque dur de l'ordinateur

```
file = open ( "file.txt", "r" )
s = file.readline ()
while s:
   print ( s )
   s = file.readline ()
file.close ()
```

(3) Fermeture



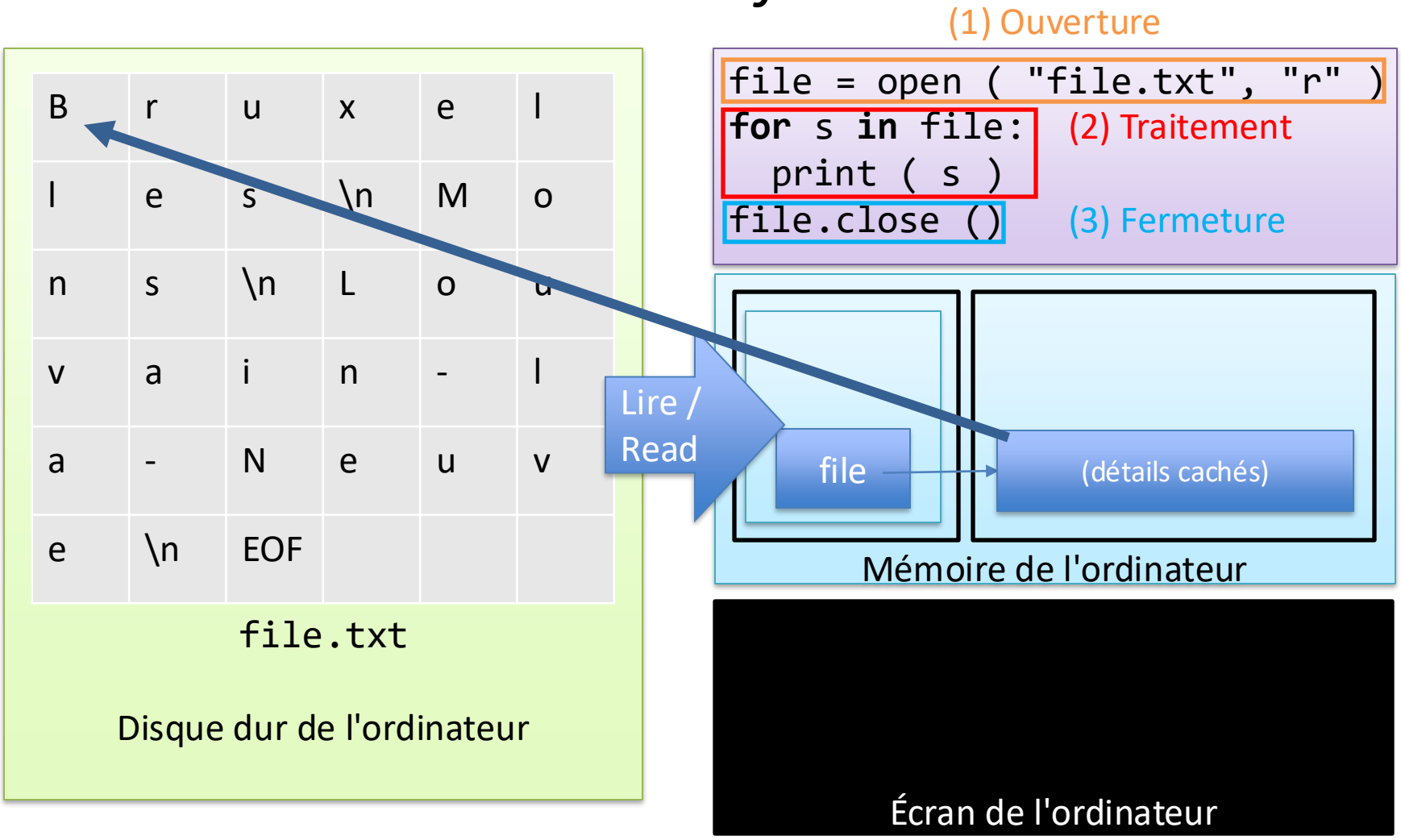

```
Bruxelles

Mons

Louvain-la-Neuve
```

Lire Fichiers de Texte

boucle for



Lire Fichiers de Texte

boucle for

(1) Ouverture

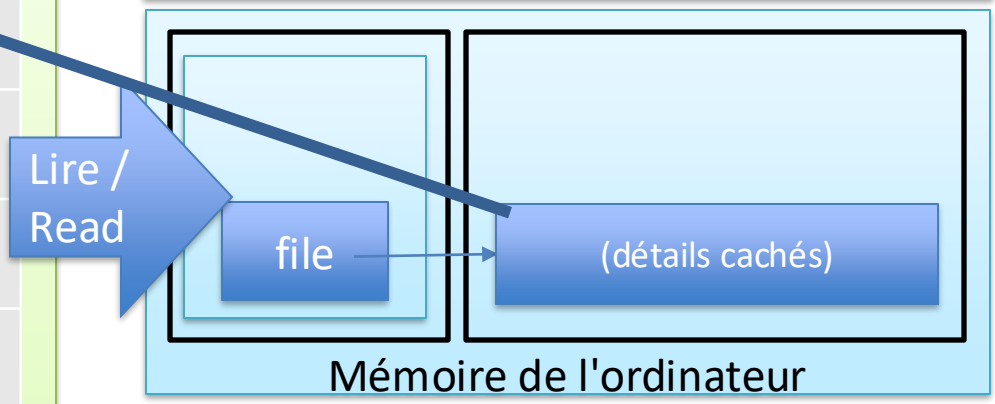
```
file = open ( "file.txt", "r" )
for s in file:
  print ( s )
file.close ()
```

B	r	u	x	e	l
l	e	s	\n	M	o
n	s	\n	L	o	u
v	a	i	n	-	l
a	-	N	e	u	v
e	\n	EOF			

file.txt

Disque dur de l'ordinateur

Lire / Read



Écran de l'ordinateur

Lire Fichiers de Texte

boucle for

B	r	u	x	e	l
l	e	s	\n	M	o
n	s	\n	L	o	u
v	a	i	n	-	l
a	-	N	e	u	v
e	\n	EOF			

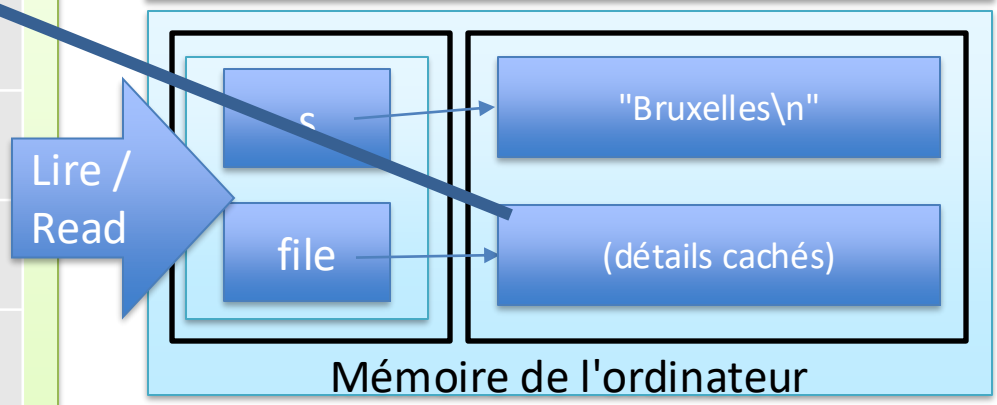
`file.txt`

Disque dur de l'ordinateur

```
file = open ( "file.txt", "r" )
for s in file:
  print ( s )
file.close ()
```

(2) Traitement

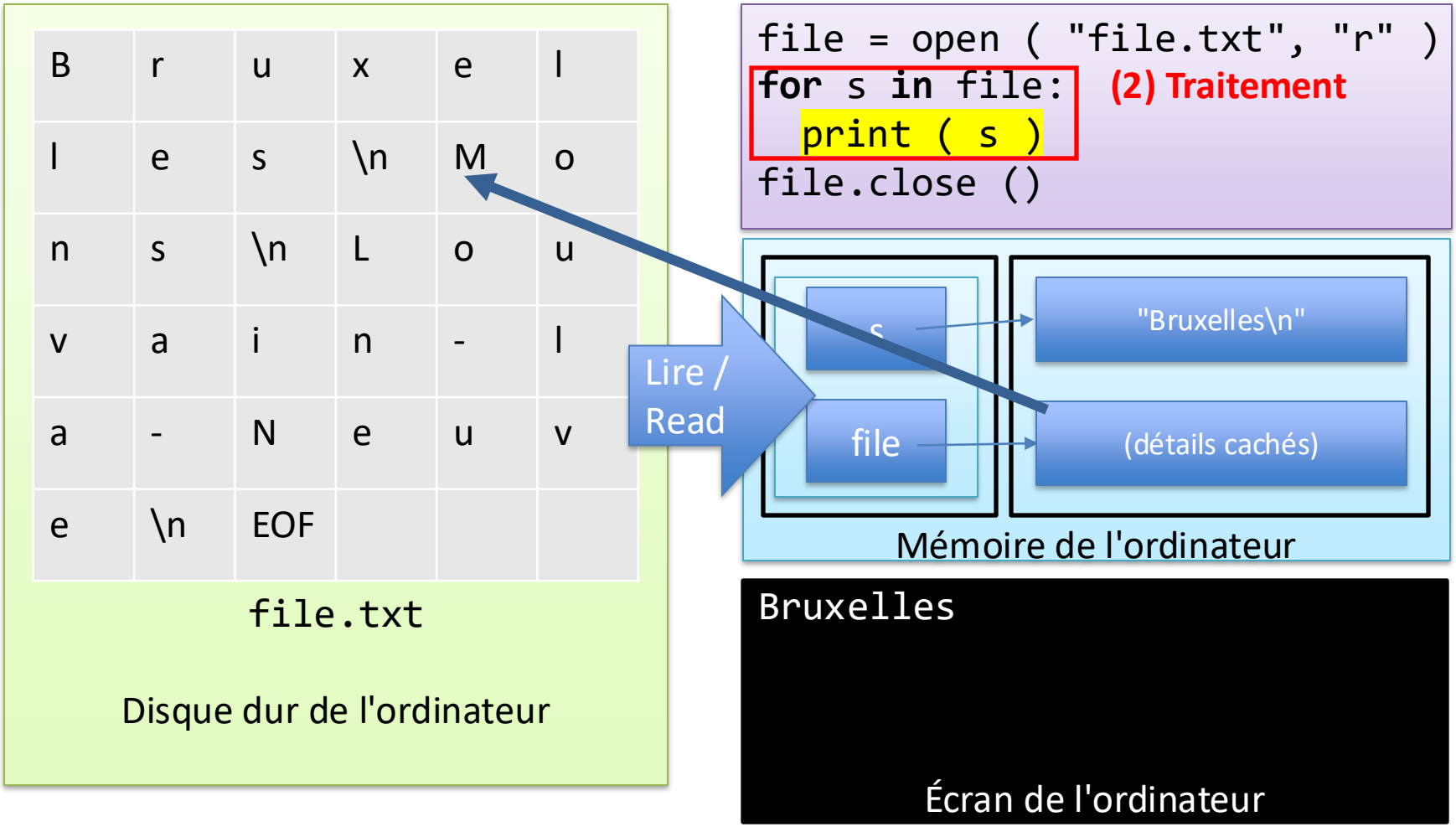
Lire / Read



Écran de l'ordinateur

Lire Fichiers de Texte

boucle for



Lire Fichiers de Texte

boucle for

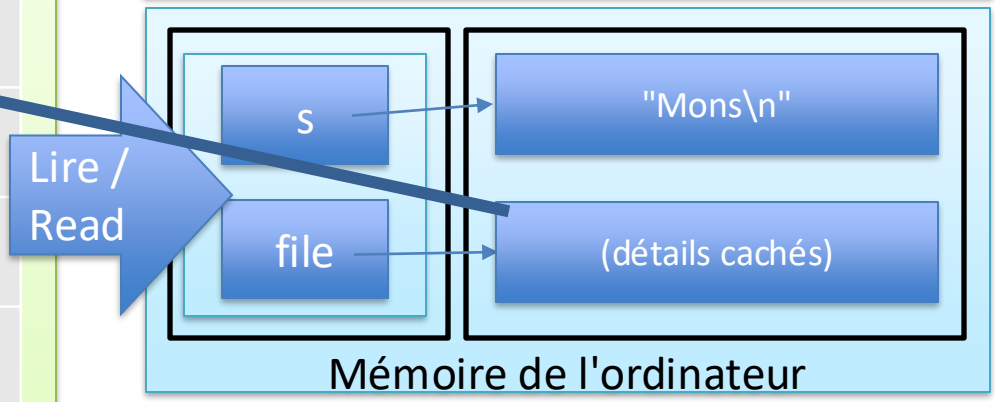
B	r	u	x	e	l
l	e	s	\n	M	o
n	s	\n	L	o	u
v	a	i	n	-	l
a	-	N	e	u	v
e	\n	EOF			

file.txt

Disque dur de l'ordinateur

```
file = open ( "file.txt", "r" )
for s in file:
  print ( s )
file.close ()
```

(2) Traitement




```
Bruxelles

Mons
```

Écran de l'ordinateur

Lire Fichiers de Texte

boucle for

B	r	u	x	e	l
l	e	s	\n	M	o
n	s	\n	L	o	u
v	a	i	n	-	l
a	-	N	e	u	v
e	\n	EOF			

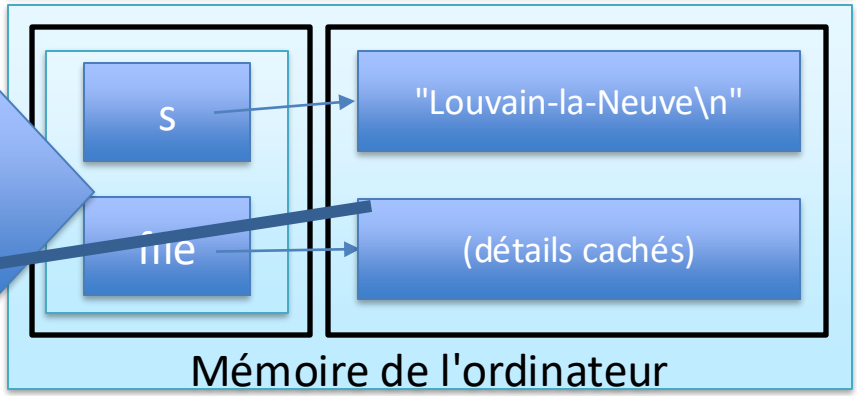
file.txt

Disque dur de l'ordinateur

Lire / Read

```
file = open ( "file.txt", "r" )
for s in file:
    print ( s )
file.close ()
```

(2) Traitement




```
Bruxelles

Mons

Louvain-la-Neuve
```

Lire Fichiers de Texte

boucle for

B	r	u	x	e	l
l	e	s	\n	M	o
n	s	\n	L	o	u
v	a	i	n	-	l
a	-	N	e	u	v
e	\n	EOF			

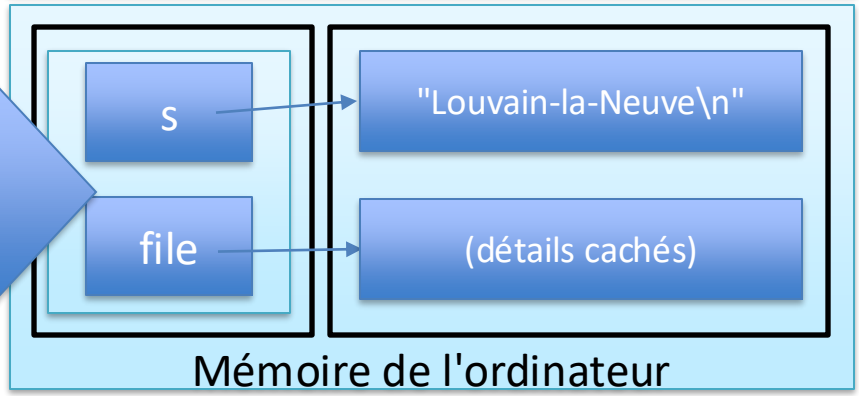
`file.txt`

Disque dur de l'ordinateur

Lire / Read

```
file = open ( "file.txt", "r" )
for s in file:
  print ( s )
file.close ()
```

(3) Fermeture




```
Bruxelles

Mons

Louvain-la-Neuve
```

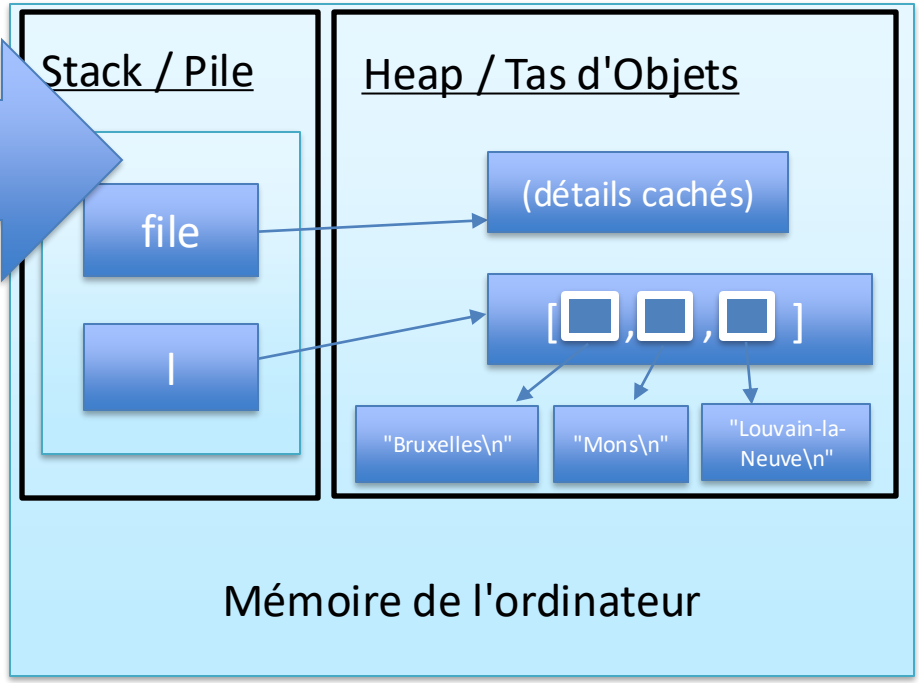
Lire Fichiers de Texte

B	r	u	x	e	l
l	e	s	\n	M	o
n	s	\n	L	o	u
v	a	i	n	-	l
a	-	N	e	u	v
e	\n	EOF			

file.txt

Disque dur de l'ordinateur

```
file = open ( "file.txt", "r" )
l = file.readlines ()
file.close ()
```



Écrire Fichiers de Texte

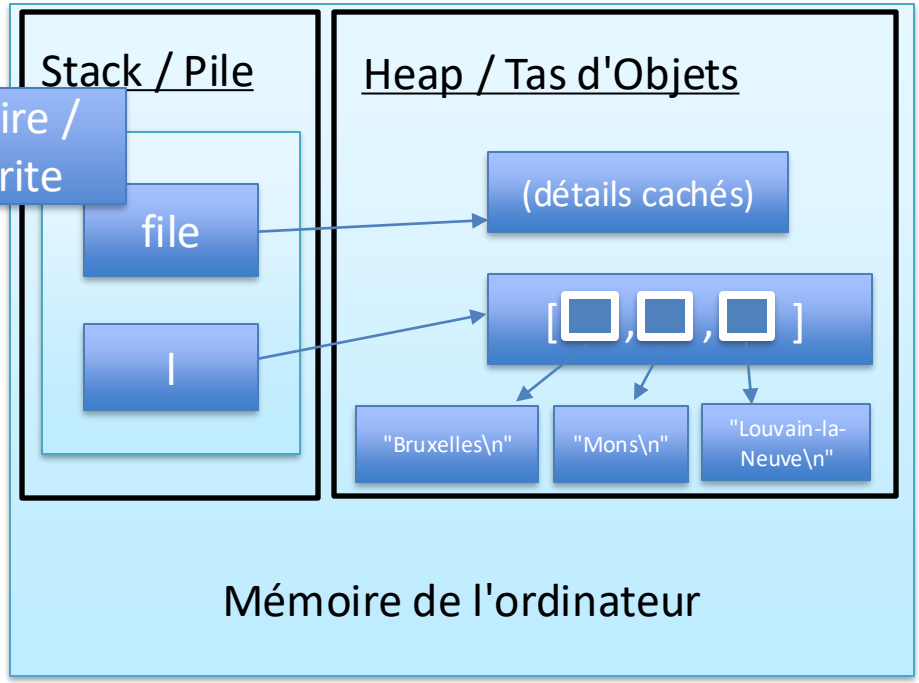
B	r	u	x	e	l
l	e	s	\n	M	o
n	s	\n	L	o	u
v	a	i	n	-	l
a	-	N	e	u	v
e	\n	EOF			

file.txt

Disque dur de l'ordinateur

```
file = open ( "file.txt", "w" )
file.writelines ( l )
file.close ()
```

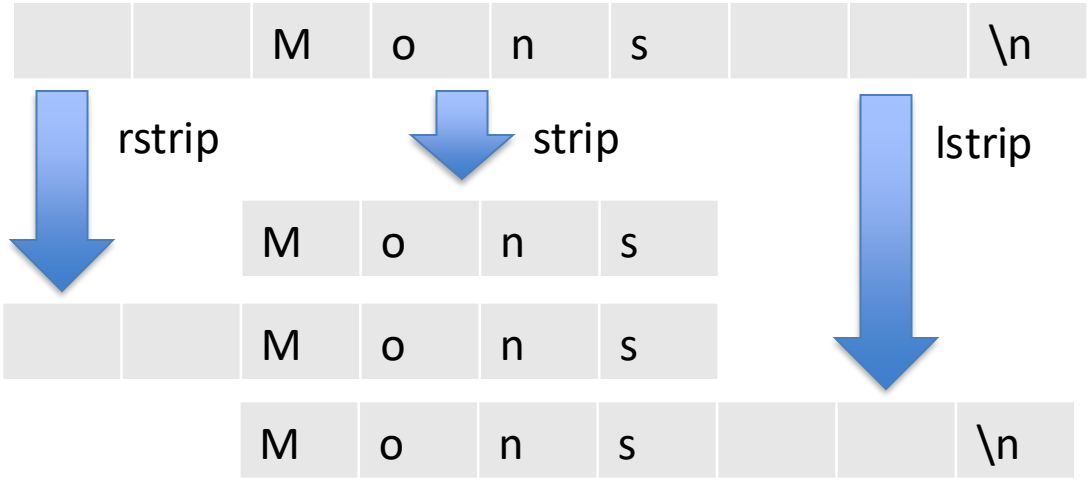
Écrire / Write



Mémoire de l'ordinateur

Méthodes sur Chaînes

- Des méthodes importantes sur chaînes de caractères dans le contexte de l'étape (2) **traitement** de fichiers:
  - `strip ()`: enlever les espaces à gauche et à droite
  - `lstrip ()`: enlever les espaces à gauche
  - `rstrip ()`: enlever les espaces à droite



Lire Fichiers de Texte

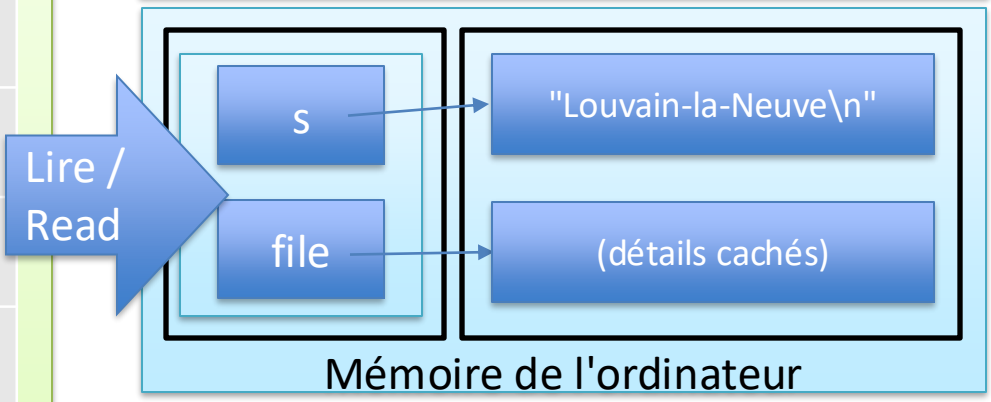
boucle for

B	r	u	x	e	l
l	e	s	\n	M	o
n	s	\n	L	o	u
v	a	i	n	-	l
a	-	N	e	u	v
e	\n	EOF			

`file.txt`

Disque dur de l'ordinateur

```
file = open ( "file.txt", "r" )
for s in file:
  print ( s )
file.close ()
```




```
Bruxelles

Mons

Louvain-la-Neuve
```

Lire Fichiers de Texte

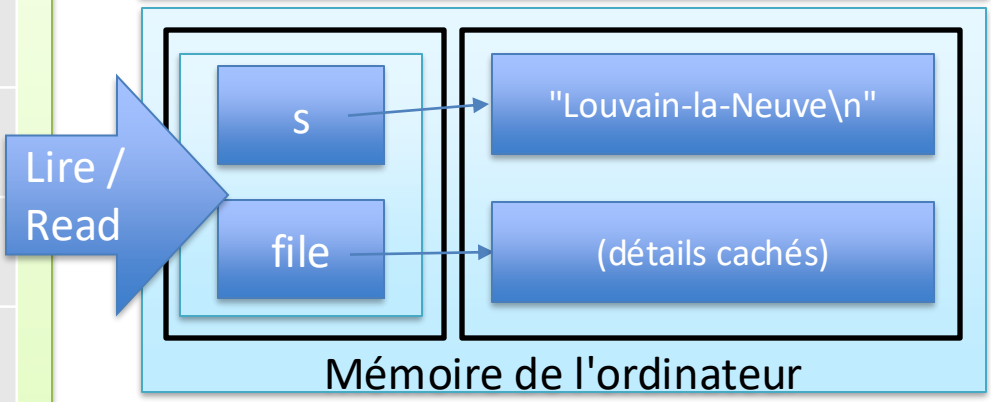
boucle for

B	r	u	x	e	l
l	e	s	\n	M	o
n	s	\n	L	o	u
v	a	i	n	-	l
a	-	N	e	u	v
e	\n	EOF			

`file.txt`

Disque dur de l'ordinateur

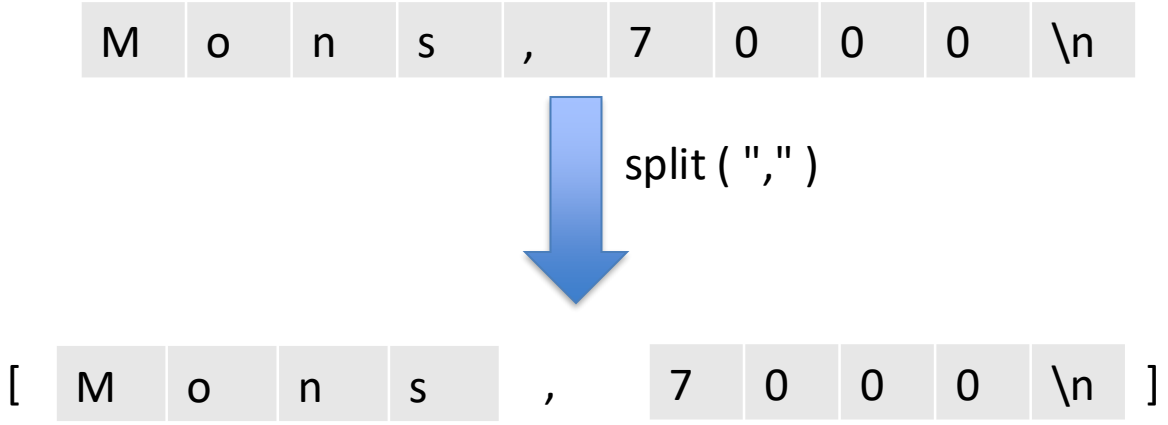
```
file = open ( "file.txt", "r" )
for s in file:
  print ( s.strip () )
file.close ()
```




```
Bruxelles
Mons
Louvain-la-Neuve
```

Méthodes sur Chaînes

- Des méthodes importantes sur chaînes de caractères dans le contexte de fichiers:
  - `split ()`: séparer des champs



Erreurs

- Qu'est-ce qui peut se passer à l'exécution de ce programme?

```
name = input ( "Provide a file name: " )
file = open ( name, "r" )
for line in file:
  print(line)
file.close ()
```

```
Provide a file name:
```

Erreurs

- Qu'est-ce qui peut se passer à l'exécution de ce programme?

```
name = input ( "Provide a file name: " )
file = open ( name, "r" )
for line in file:
  print(line)
file.close ()
```

```
Provide a file name: nexistepas.txt
```

Erreurs

- Qu'est-ce qui peut se passer à l'exécution de ce programme?

```
name = input ( "Provide a file name: " )
file = open ( name, "r" )
for line in file:
  print(line)
file.close ()
```

```
Provide a file name: nexistepas.txt
Traceback (most recent call last):
  File "test.py", line 2, in <module>
    file = open ( name, "r" )
FileNotFoundError: [Errno 2] No such
file or directory: 'nexistepas.txt'
```

Erreur

- Comment d'éviter que le programme s'arrête?

```
name = input ( "Provide a file name: " )
try:
  file = open ( name, "r" )
  for line in file:
    print(line)
  file.close ()
except:
  print ( "Error reading file" )
```

Gestion des Exceptions

- Python **lève** une **exception** s'il y a une erreur
- On peut **intercepter** des exceptions en utilisant un bloc `try … except`

```
name = input ( "Provide a file name: " )
try:
   file = open ( name, "r" )
   for line in file:
     print(line)
   file.close ()
except:
   print ( "Error reading file" )
```

(1) Ouverture
(2) Traitement
(3) Fermeture
(4) Exceptions

Gestion des Exceptions

- Il y a beaucoup d'erreurs différentes possibles:
  - Fichier n'existe pas
  - Assert error
  - Division par zéro
  - Valeur incorrecte (`int("Bonjour")`)
  - Index incorrect
- Toutes sont interceptées par **`except`**

Gestion des Exceptions

- Deux erreurs sont possibles ici:

```
num = input ( "Provide a number: " )
try:
  print ( 1 / int(num) )
except:
  print ( "Error doing calculation" )
```

- Le même message pour une division par 0 ou une valeur incorrecte

Gestion des Exceptions

- Intercepter des erreurs spécifiques :

```
num = input ( "Provide a number: " )
try:
  print ( 1 / int(num) )
except ValueError:
  print ( "No integer provided" )
except ZeroDivisionError:
  print ( "Division by zero" )
```

Gestion des Exceptions

- Python montre des messages plus détaillés

```
Provide a file name: nexistepas.txt
Traceback (most recent call last):
  File "test.py", line 2, in <module>
    file = open ( name, "r" )
FileNotFoundError: [Errno 2] No such
file or directory: 'nexistepas.txt'
```

Gestion des Exceptions

- Récupérer le message qui serait montré par Python même

```
num = input ( "Provide a number: " )
try:
  print ( 1 / int(num) )
except ValueError as error:
  print ( "ValueError: " + str ( error ) )
except ZeroDivisionError:
  print ( "Division by zero" )
```

```
Provide a number: age
ValueError: invalid literal for int()
with base 10: 'age'
```

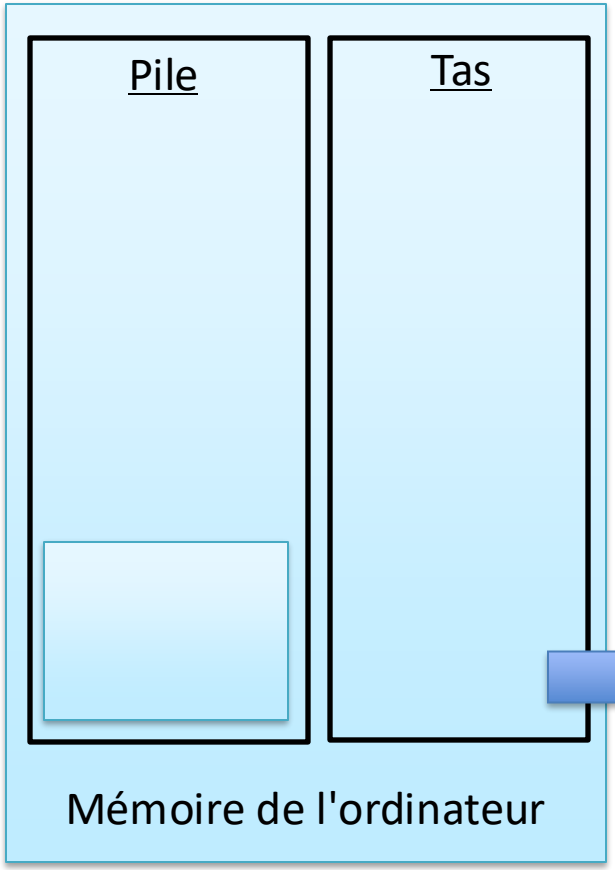
Gestion des Exceptions

- Lever des exceptions

```
try:
  age = int(input( "Provide your age: " ))
  if age < 0 or age >= 150:
    raise ValueError ( "Age out of range" )
  print ( "Your age is registered" )
except ValueError as error:
  print ( "Invalid age: " + str ( error ) )
```

```
Provide your age: 160
Invalid age: Age out of range
```

Flux d'Exécution



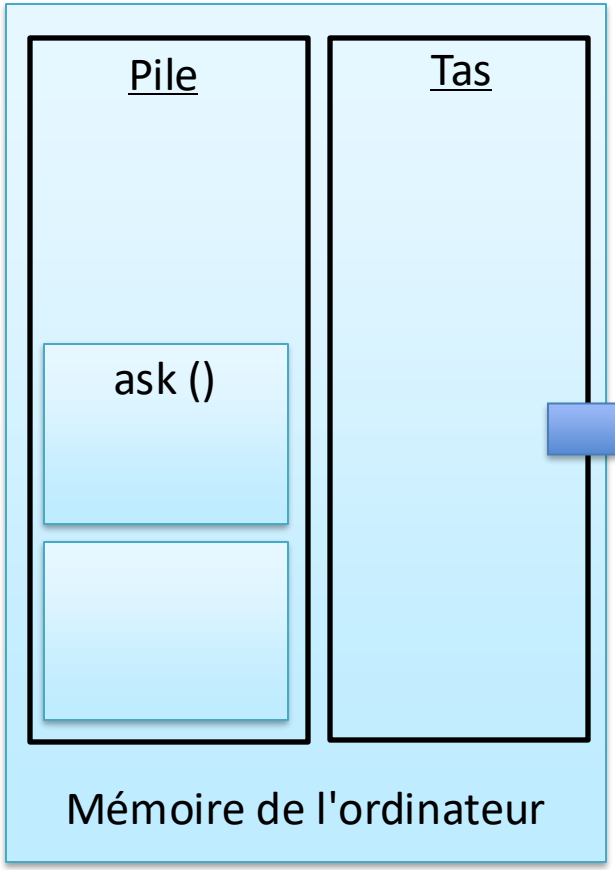

```
def askAge ():
  age = int(input( "Provide your age: " ))
  if age < 0 or age >= 150:
    raise ValueError ( "Age out of range")
  return age

def ask ():
  val = askAge ()
  print ( "Age entered: " + str(val) )
  return val

try:
  age = ask ()
except ValueError as error:
  print ( "Invalid age: " + str ( error ) )
print ( "Program finished" )
```



Flux d'Exécution



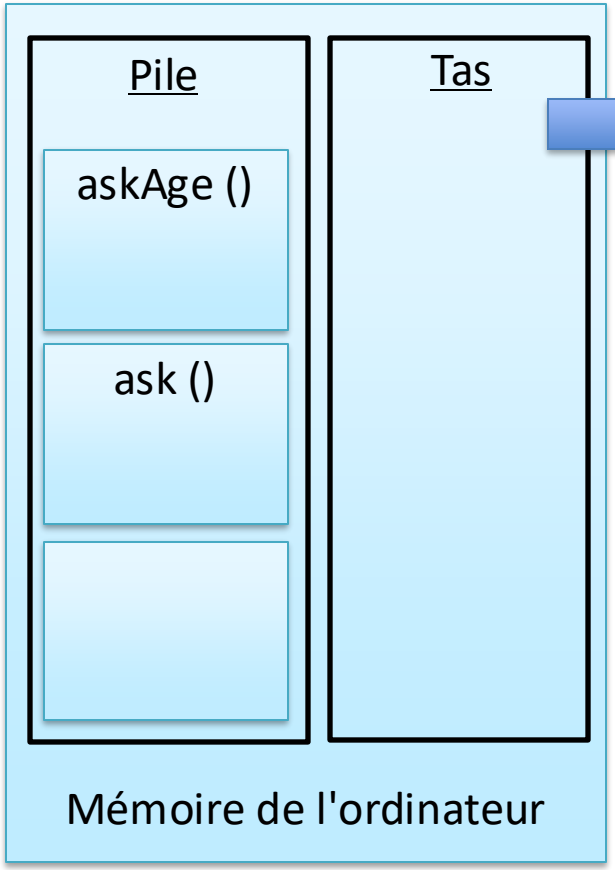

```
def askAge ():
  age = int(input( "Provide your age: " ))
  if age < 0 or age >= 150:
    raise ValueError ( "Age out of range")
  return age

def ask ():
  val = askAge ()
  print ( "Age entered: " + str(val) )
  return val

try:
  age = ask ()
except ValueError as error:
  print ( "Invalid age: " + str ( error ) )
print ( "Program finished" )
```



Flux d'Exécution



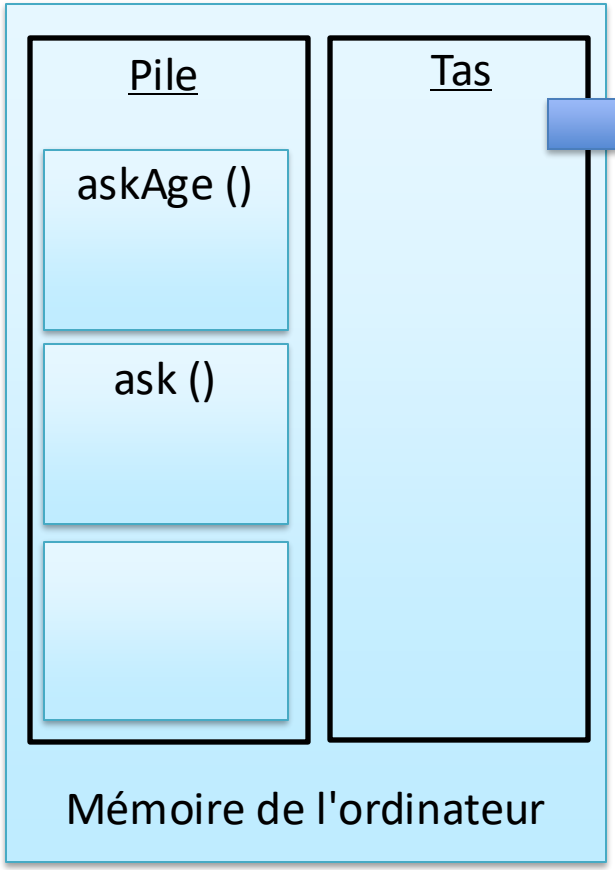

```
def askAge ():
  age = int(input( "Provide your age: " ))
  if age < 0 or age >= 150:
    raise ValueError ( "Age out of range")
  return age

def ask ():
  val = askAge ()
  print ( "Age entered: " + str(val) )
  return val

try:
  age = ask ()
except ValueError as error:
  print ( "Invalid age: " + str ( error ) )
print ( "Program finished" )
```



Flux d'Exécution



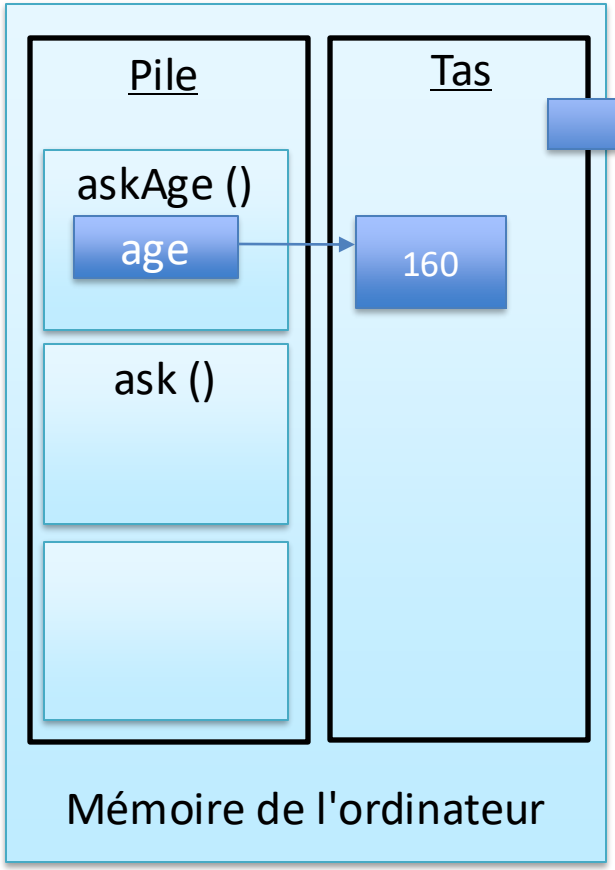

```
def askAge ():
  age = int(input( "Provide your age: " ))
  if age < 0 or age >= 150:
    raise ValueError ( "Age out of range")
  return age

def ask ():
  val = askAge ()
  print ( "Age entered: " + str(val) )
  return val

try:
  age = ask ()
except ValueError as error:
  print ( "Invalid age: " + str ( error ) )
print ( "Program finished" )
```

```
Provide your age:
```

Flux d'Exécution



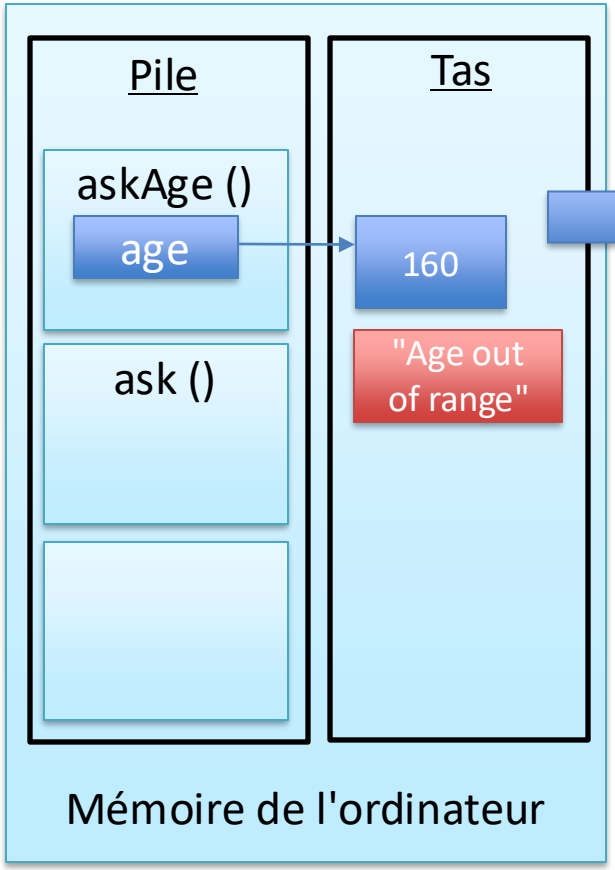

```
def askAge ():
  age = int(input( "Provide your age: " ))
  if age < 0 or age >= 150:
    raise ValueError ( "Age out of range")
  return age

def ask ():
  val = askAge ()
  print ( "Age entered: " + str(val) )
  return val

try:
  age = ask ()
except ValueError as error:
  print ( "Invalid age: " + str ( error ) )
print ( "Program finished" )
```

```
Provide your age: 160
```

Flux d'Exécution



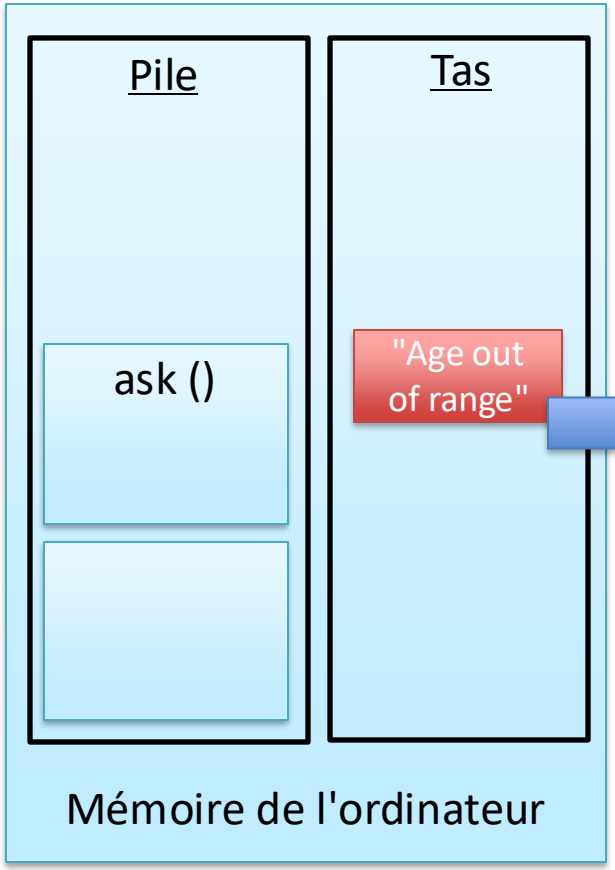

```
def askAge ():
  age = int(input( "Provide your age: " ))
  if age < 0 or age >= 150:
    raise ValueError ( "Age out of range")
  return age

def ask ():
  val = askAge ()
  print ( "Age entered: " + str(val) )
  return val

try:
  age = ask ()
except ValueError as error:
  print ( "Invalid age: " + str ( error ) )
print ( "Program finished" )
```

```
Provide your age: 160
```

Flux d'Exécution



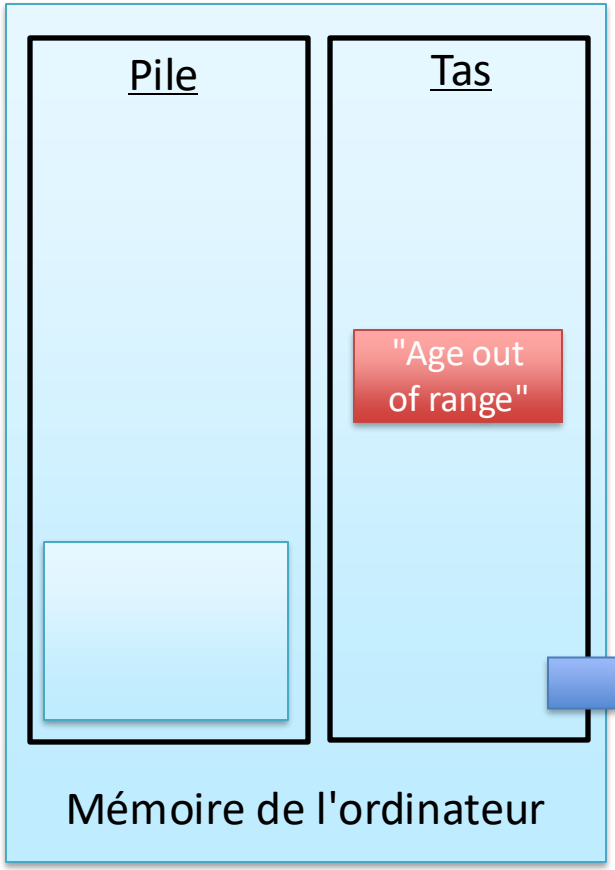

```
def askAge ():
  age = int(input( "Provide your age: " ))
  if age < 0 or age >= 150:
    raise ValueError ( "Age out of range")
  return age

def ask ():
  val = askAge ()
  print ( "Age entered: " + str(val) )
  return val

try:
  age = ask ()
except ValueError as error:
  print ( "Invalid age: " + str ( error ) )
print ( "Program finished" )
```

```
Provide your age: 160
```

Flux d'Exécution



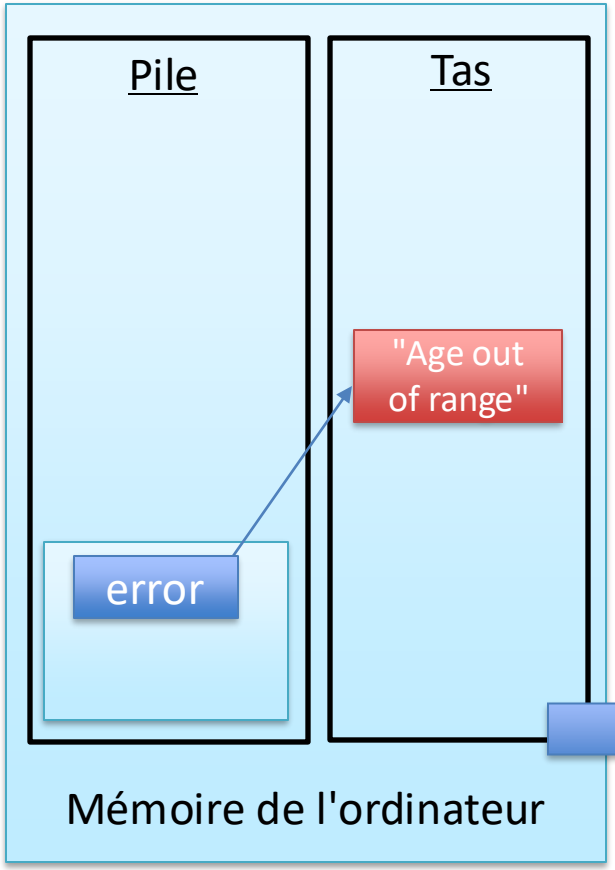

```
def askAge ():
  age = int(input( "Provide your age: " ))
  if age < 0 or age >= 150:
    raise ValueError ( "Age out of range")
  return age

def ask ():
  val = askAge ()
  print ( "Age entered: " + str(val) )
  return val

try:
  age = ask ()
except ValueError as error:
  print ( "Invalid age: " + str ( error ) )
print ( "Program finished" )
```

```
Provide your age: 160
```

Flux d'Exécution



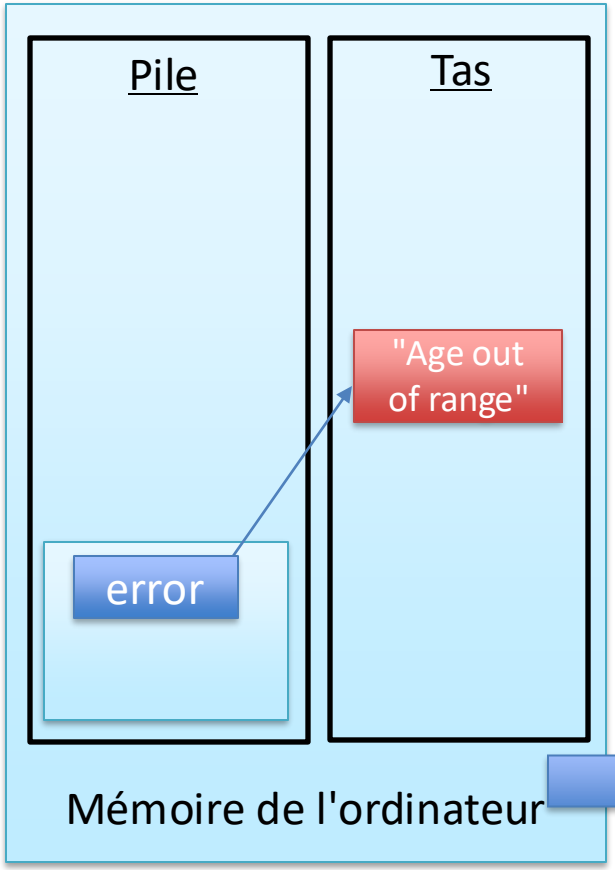

```
def askAge ():
  age = int(input( "Provide your age: " ))
  if age < 0 or age >= 150:
    raise ValueError ( "Age out of range")
  return age

def ask ():
  val = askAge ()
  print ( "Age entered: " + str(val) )
  return val

try:
  age = ask ()
except ValueError as error:
  print ( "Invalid age: " + str ( error ) )
print ( "Program finished" )
```

```
Provide your age: 160
```

Flux d'Exécution



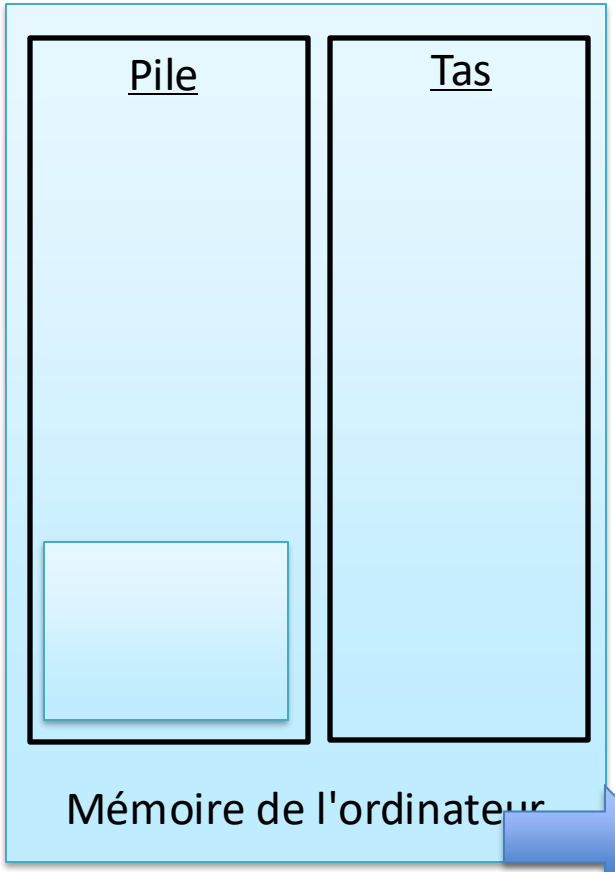

```
def askAge ():
  age = int(input( "Provide your age: " ))
  if age < 0 or age >= 150:
    raise ValueError ( "Age out of range")
  return age

def ask ():
  val = askAge ()
  print ( "Age entered: " + str(val) )
  return val

try:
  age = ask ()
except ValueError as error:
  print ( "Invalid age: " + str ( error ) )
print ( "Program finished" )
```

```
Provide your age: 160
Invalid age: Age out of range
```

Flux d'Exécution




```
def askAge ():
  age = int(input( "Provide your age: " ))
  if age < 0 or age >= 150:
    raise ValueError ( "Age out of range")
  return age

def ask ():
  val = askAge ()
  print ( "Age entered: " + str(val) )
  return val

try:
  age = ask ()
except ValueError as error:
  print ( "Invalid age: " + str ( error ) )
print ( "Program finished" )
```

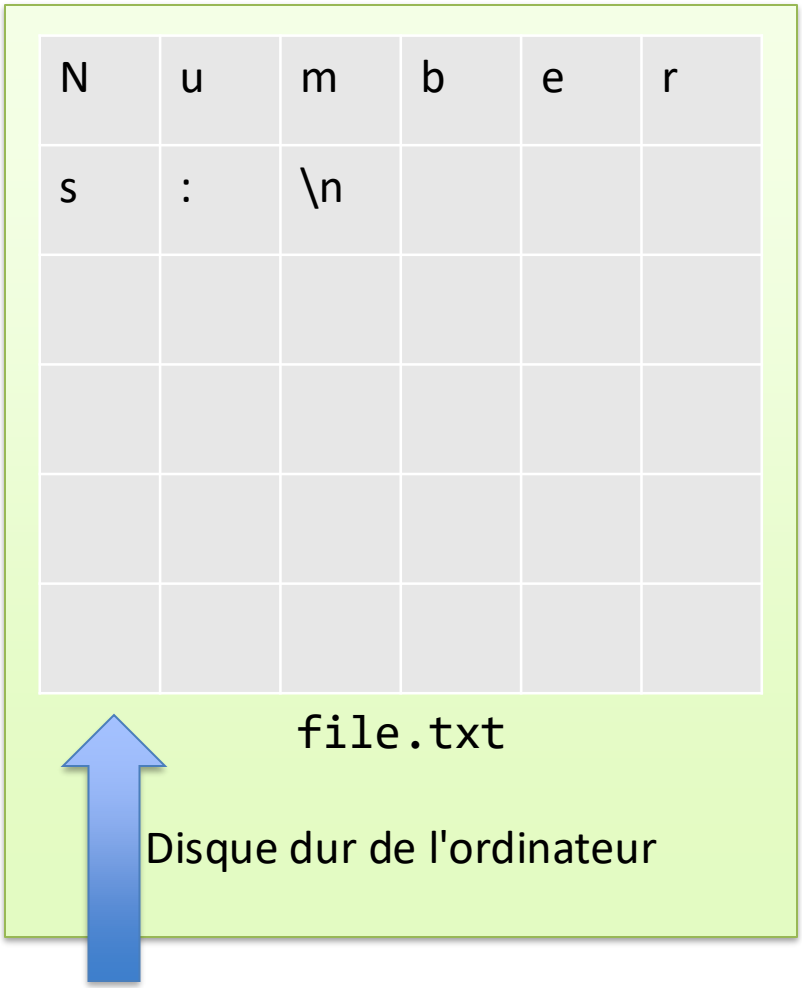
```
Provide your age: 160
Invalid age: Age out of range
Program finished
```

Fermer des Fichiers

```
name = input ( "Provide a file name: " )
try:
  file = open ( name, "w" )
  file.write ( "Numbers:\n" )
  for i in range(10):
    file.write ( str(1/i) )
  file.close ()
except:
  print ( "Error writing file" )
input ( "Press enter to continue" )
```

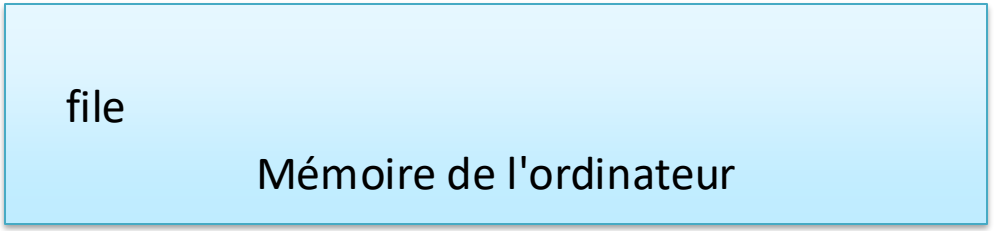
- Qu'est-ce qui se passe pendant l'exécution de ce programme?

Fermer des Fichiers




```
name = input ( "Provide a file name: " )
try:
  with open ( name, "w" ) as file:
    file.write ( "Numbers:\n" )
    for i in range(10):
      file.write ( str(1/i) )
except:
  print ( "Error writing file" )
input ( "Press enter to continue" )
```

```
Provide a file name: file.txt
Error writing file
Press enter to continue
```



File est automatiquement fermé si l'exécution sort du bloc 'with'.

Fichiers

Approche recommandée

```
name = input ( "Provide a file name: " )
try:
   with open( name, "r" ) as file:
        for line in file:
             print(line)

except:
   print ( "Error reading file" )
```

(1) Ouverture
(3) Fermeture
(2) Traitement
(4)Exceptions