

IFT211/776
Programmation scientifique en Python
Introduction aux bibliothèques

Gabriel Girard

Département d'informatique



19 janvier 2017

Thème 7 — Introduction aux bibliothèques

- 1 Les fichiers
- 2 Exécution de programmes/commandes
- 3 Manipulation de chaînes de caractères
- 4 NumPy
- 5 Scipy
- 6 Mathplotlib
- 7 (Anaconda)

Les fichiers : ouverture/fermeture

```
nom = input("Entrez le nom du fichier : ")
# Ouverture
fo = open(nom, "rb")
print("Nom du fichier : ", fo.name)
print(" Fermé ou non : ", fo.closed)
print("Mode d'ouverture : ", fo.mode)

# fermeture
fo.close()
```

Modes d'ouverture : r, rb, r+, rb+, w, wb, w+, wb+, ab, a+, ab+

Les fichiers

```
# Création d'un nouveau fichier et
# écriture dans le fichier
fo = open("nouveau.txt", "w")
fo.write( "Ecriture dans le fichier\n")
fo.close()

-----

nom = input("Entrez le nom du fichier : ")
fo = open(nom, "r+")
# Lecture de 10 caractères dans le fichier
str = fo.read(10)
print("la chaine lue = ", str)
# Lecture du fichier en entier
str = fo.read()
print("la chaine lue = ", str)

fo.close()
```



Les fichiers

```
# Manipulation de fichiers
import os
import filecmp
import shutil

shutil.copy("textel", "textell")
os.rename("textell", "texte2")
os.mkdir("newdir")
os.chdir("newdir")
print(os.getcwd())
fichier = input("Entrez le nom d'un fichier : ")
print(os.path.exists(fichier))
os.chdir("..")
print(filecmp.cmp("textel", "texte2"))
os.rmdir("newdir")
os.remove("texte2")
```

Exécution de programmes/commandes

```
import os
import subprocess as sub
import sys

sub.call(['ls', '-l'], shell=True)
fin = sub.check_output("ls", shell=True)
sub.run(["ls", "-l"]) # >= python 3.5
sub.Popen("echo Hello World", shell=True)

commande = ["python3 " + programme + " < " + ent]
fin = sub.check_output(commande, shell=True)
-----
```

Il y a aussi la bibliothèque

- pexpect
- os, putil, circus

Chaînes de caractères

```
ph.split(' '); ph.count('l'); ph.find("H")
ph.startswith("H") ; ph.endswith("d")
ph.index("mot"); p.replace("bon", "mauvais")
ph.upper(), phe.lower()
ph.strip(); ph.lstrip(); ph.rstrip()
mot.isalnum(); mot.isalpha(); mot.isdigit();
    mot.isupper() ; ...
```

Il y a aussi la bibliothèque "re" (expression régulière)



NumPy

- NumPy est une bibliothèque qui permet de créer et manipuler des tableaux homogènes à plusieurs dimensions (vecteurs, matrices, ...)
- Dans numpy les dimensions sont appelés des axes et le nombre d'axes est appelé le rang (rank)
- Le type introduit pour les tableaux est `ndarray` ou `array`.

Numpy : création de tableaux

```
>>> np.zeros((2,2))
>>> np.ones((1,2))
>>> np.full((2,2), 7)
>>> np.empty( (2,3) )
>>> np.eye(2)
>>> np.random.random((2,2))
>>> np.arange(10)
>>> np.arange(2, 10, dtype=np.float)
>>> np.arange(2, 3, 0.1)
>>> np.indices((3,3))
```

NumPy

```
>>> a = np.arange(15).reshape(3, 5)
>>> a.shape
>>> a.ndim
>>> a.dtype
>>> a.size
>>> a.itemsize
```

Numpy

```
a = np.array( [20,30,40,50] )
b = np.arange(4) # [0 1 2 3]

c = a-b          # [20 29 38 47]
d = b**2         # [0 1 4 9]
e = 10*np.sin(a) # [9.1.. -9.8.. 7.4.. -2.6..]
f = a<35        # [True True False False]
```



Numpy

```
A = np.array( [[1,1], [0,1]] )
B = np.array( [[2,0], [3,4]] )

C = A*B           # [[2 0] [0 4]]
D = A.dot(B)      # [[5 4] [3 4]]
E = np.dot(A, B)  # [[5 4] [3 4]]
```

Numpy

```
a = np.ones((2,3), dtype=int) # [[1 1 1] [1 1 1]]
b = np.random.random((2,3))
  # [[ 0.71969048  0.01061456  0.33027147]
  # [ 0.95265034  0.42529348  0.72858577]]

a *= 3      # [[3 3 3] [3 3 3]]
b += a
  # [[ 3.71969048  3.01061456  3.33027147]
  # [ 3.95265034  3.42529348  3.72858577]]

a += b      # [[6 6 6] [6 6 6]]
```



Numpy

```
a = np.full((2,3),3) # [[ 3.  3.  3.] [ 3.  3.
  3.]]
```

```
a.sum() # 18.0
```

```
a=np.random.random((2,3))
```

```
# [[ 0.11071364  0.23442051  0.55624391]
```

```
# [ 0.30703067  0.66608416  0.44667706]]
```

```
a.min() # 0.110713639539
```

```
a.max() # 0.666084161428
```

```
b.sum(axis=0)
```

```
b.min(axis=1)
```



Numpy

```
B = np.arange(3)           # [0 1 2]
E = np.array([2., -1., 4.]) # [ 2. -1.  4.]

C = np.exp(B)             # [ 1.  2.71828183  7.3890561 ]
D = np.sqrt(B)            # [ 0.  1.  1.41421356]

F = np.add(B, E)          # [ 2.  0.  6.]
```

SciPy

- Collection d'algorithmes mathématiques bâties par dessus numPy.
- Organisé en sous-bibliothèques
 - constants : constantes mathématiques et physiques
 - fftpack : transformé de Fourier
 - linalg : algèbre linéaire
 - signal : traitement de signal
 - ...

Matplotlib

```
from matplotlib import pyplot as plt
```

```
x = [5, 8, 10]
```

```
y = [12, 16, 6]
```

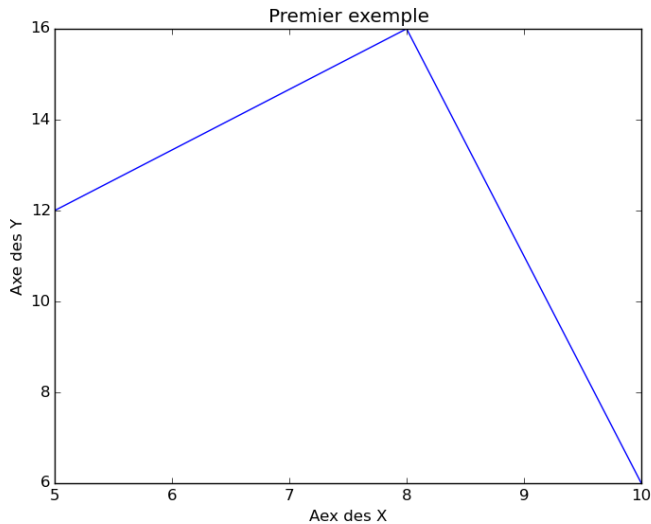
```
plt.plot(x, y)
```

```
plt.title('Epic Info')
```

```
plt.ylabel('Y axis')
```

```
plt.xlabel('X axis')
```

```
plt.show()
```



```
from matplotlib import pyplot as plt
from matplotlib import style

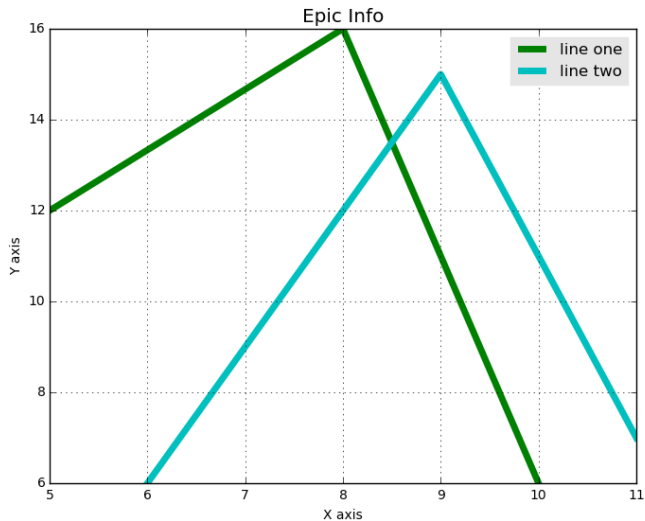
style.use('ggplot')

x = [5,8,10]
y = [12,16,6]
x2 = [6,9,11]
y2 = [6,15,7]
plt.plot(x,y,'g',label='ligne un', linewidth=5)
plt.plot(x2,y2,'c',label='ligne deux',linewidth=5)

plt.title('Second exemple')
plt.ylabel('Axe des Y')
plt.xlabel('Axe des X')

plt.legend()
plt.grid(True,color='k')
plt.show()
```





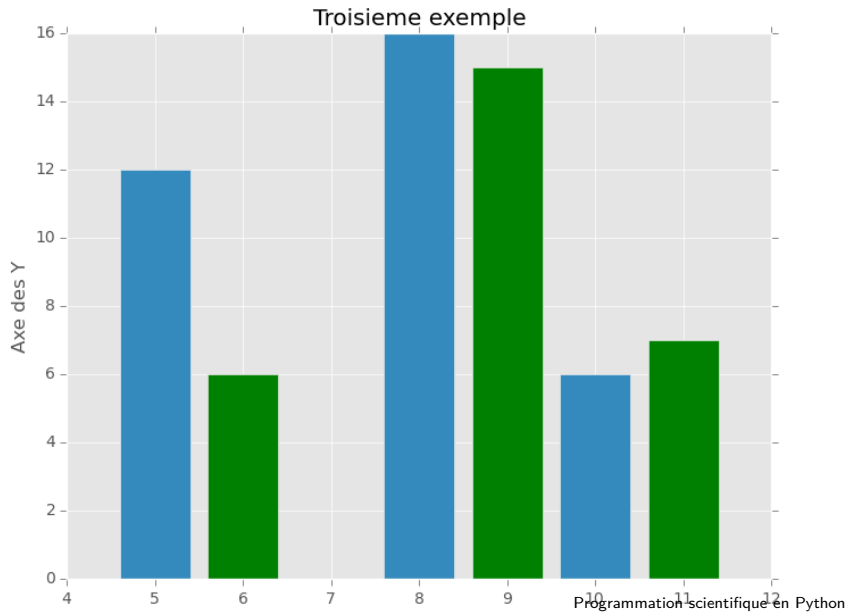
```
from matplotlib import pyplot as plt
from matplotlib import style

style.use('ggplot')
x = [5, 8, 10]
y = [12, 16, 6]
x2 = [6, 9, 11]
y2 = [6, 15, 7]

plt.bar(x, y, align='center')
plt.bar(x2, y2, color='g', align='center')

plt.title('Troisieme exemple')
plt.ylabel('Axe des Y')
plt.xlabel('Axe des X')
plt.show()
```





```
from matplotlib import pyplot as plt
from matplotlib import style
```

```
style.use('ggplot')
```

```
x = [5, 8, 10]
```

```
y = [12, 16, 6]
```

```
x2 = [6, 9, 11]
```

```
y2 = [6, 15, 7]
```

```
plt.scatter(x, y)
```

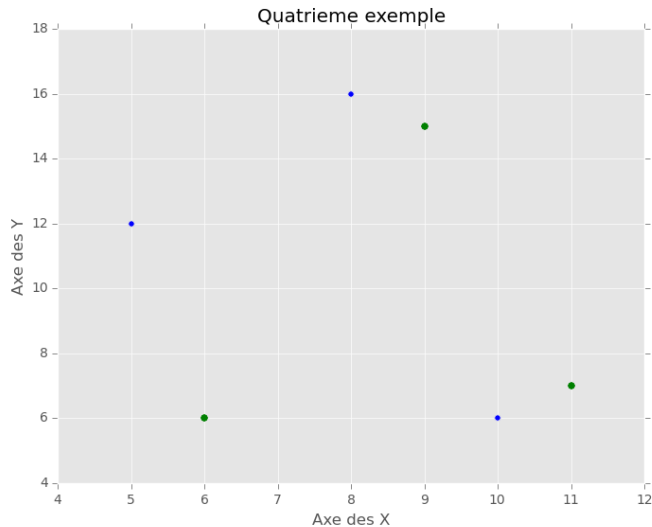
```
plt.scatter(x2, y2, color='g')
```

```
plt.title('Quatrieme exemple')
```

```
plt.ylabel('Axe des Y')
```

```
plt.xlabel('Axe des X')
```

```
plt.show()
```



Anaconda

- Anaconda une distribution de Python
- Elle inclut plus de 300 bibliothèques de sciences, de mathématiques d'ingénierie et d'analyse de données.