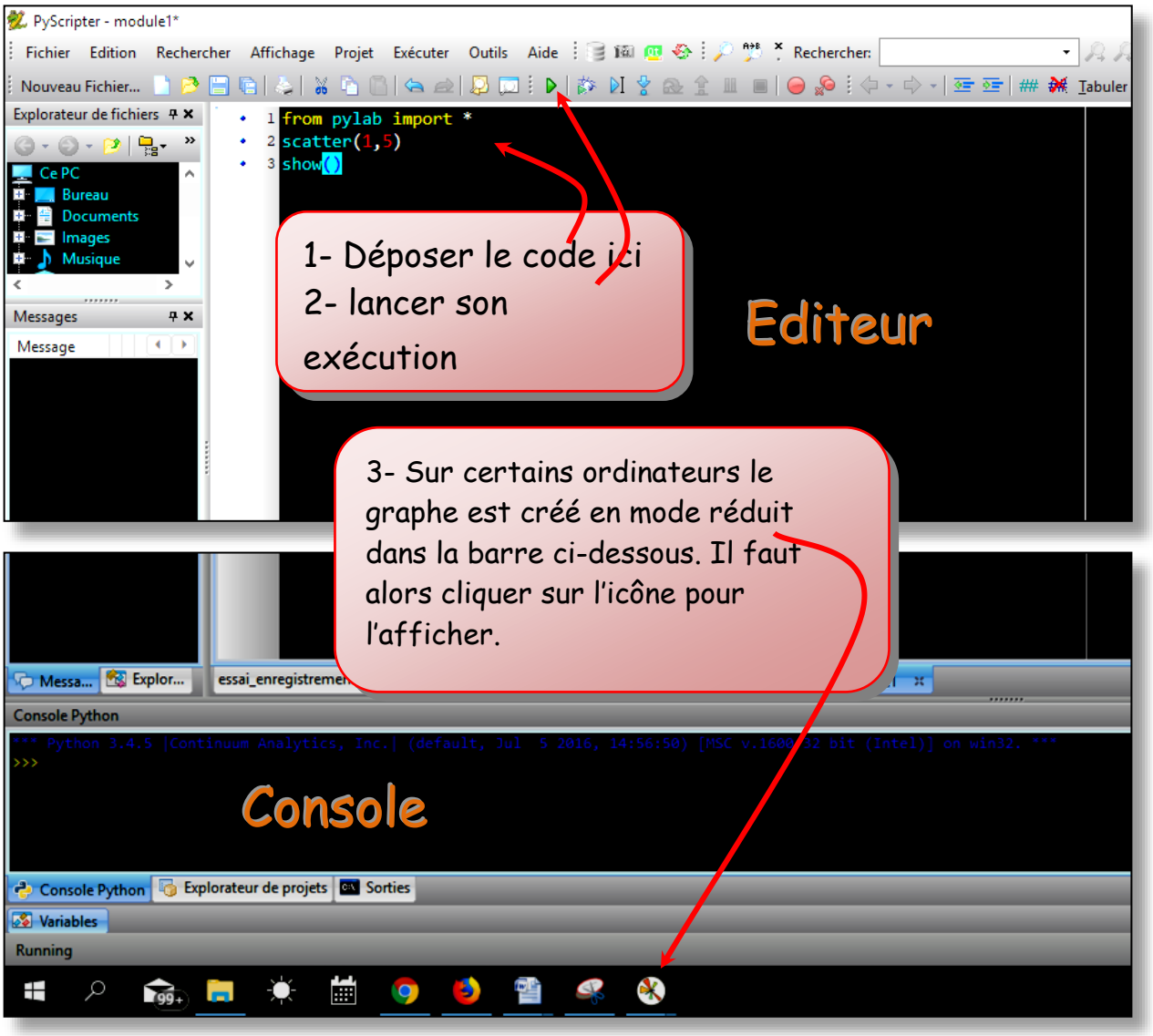
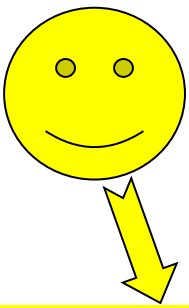
I - Afficher des Graphes de plus en plus « étoffés »

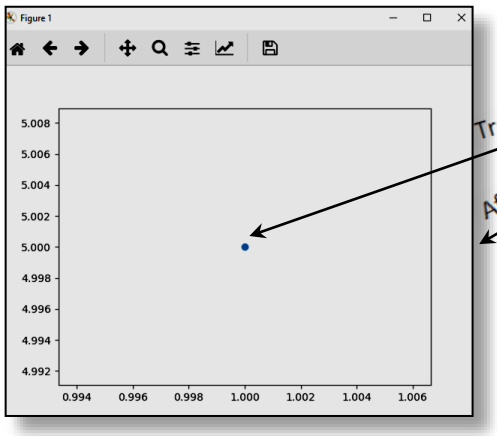
A- Introduction





Sous Edupython, penser à fermer le graphe précédent pour faire le suivant !

B- Un simple point avec 'scatter()'



Trace le point (1,5)

Affiche ce graphe

```
from pylab import *
scatter(1,5)
show()
```

Ouverture de la bibliothèque « pylab » contenant la plupart des instructions (*comme ici* `scatter()`) nécessaires.
Si besoin, il faudra « importer » d'autres bibliothèques.
L'astérisque « * » demande l'importation totale d'une bibliothèque.

Code à copier-coller

```
from pylab import *
scatter(1,5)
show()
```

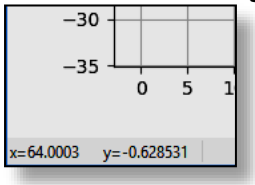
Si ce code (dit en « syntaxe pylab ») provoque une erreur cela peut provenir de l'absence de « `plt.` » devant `scatter()` et `show()`. Copier-coller alors le code suivant (dit en « syntaxe standard » , un peu plus lourd comme on le constate) :

```
import numpy as np
import matplotlib.pyplot as plt
plt.scatter(1,5)
```

De la même façon les quelques lignes ci-contre afficheront un second point (2,-2).

```
from pylab import *
scatter(1,5)
scatter(2,-2)
```

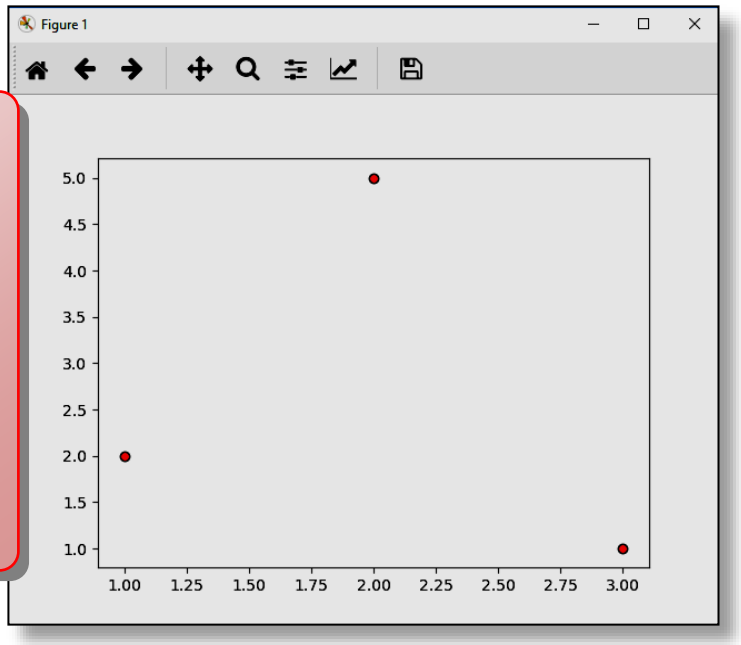
Remarque : En déplaçant la souris sur le graphe, ses coordonnées s'affichent en bas à gauche de la fenêtre du graphe.



C- Plusieurs points avec un seul 'scatter()'

```
from pylab import *
scatter([1,2,3],[2,5,1],color='#ff0000',edgecolor='k')
show()
```

[1,2,3] sont les abscisses et [2,5,1] les ordonnées

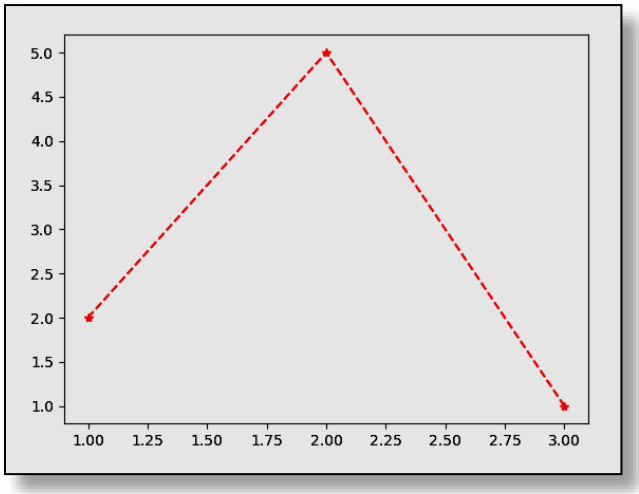


- 'color'est la couleur de fond des points (marqueurs) et 'edgecolor' celle de leur contour.
- 'k' signifie black (car 'b' est le bleu).
- '#FF0000' est la couleur rouge car le code en hexadécimal est : '#RougeVertBleu' chaque couleur allant de 0 à 255 (de 0 à FF).

Les couleurs :

'b'	blue
'g'	green
'r'	red
'c'	cyan
'm'	magenta
'y'	yellow
'k'	black
'w'	white

D- Une ligne reliant les points avec `plot()`




```
from pylab import *
plot([1,2,3], [2,5,1], 'r--*')
show()
```

Code à copier-coller

En pylab

```
from pylab import *
plot([1,2,3], [2,5,1], 'r--*')
show()
```

ou *en standard*

```
import numpy as np
import matplotlib.pyplot as plt
plt.plot([1,2,3], [2,5,1], 'r--*')
plt.show()
```

- Même principe que pour `'scatter'` : Les abscisses dans les premiers crochets et les ordonnées correspondantes dans les seconds.
- La propriété `'r--*'` signifie : marqueurs '*' reliés par des lignes pointillées (--) le tout en rouge (r).
- L'ordre n'a pas d'importance. Essayer : `'--b^'` ; `':yv'` etc avec les marqueurs et lignes des tableaux ci-contre.

Les lignes:

-	ligne continue
--	tirets
:	ligne en pointillé
-.	tirets points

Les marqueurs:

'.'	point marker
','	pixel marker
'o'	circle marker
'v'	triangle_down marker
'^'	triangle_up marker
'<'	triangle_left marker
'>'	triangle_right marker
'1'	tri_down marker
'2'	tri_up marker
'3'	tri_left marker
'4'	tri_right marker
's'	square marker
'p'	pentagon marker
'*'	star marker
'h'	hexagon1 marker
'H'	hexagon2 marker
'+'	plus marker
'x'	x marker
'D'	diamond marker
'd'	thin_diamond marker
'|'	vline marker
'_'	hline marker

E- Graphe d'une fonction mathématique

La variable t variera de 0 à 10 avec un pas de 0,5

```
from pylab import *
t=arange(0,10,.5)
plot( t, t**2, ':bs')
show()
```

La fonction t^2 (t**2 en python).

Ligne pointillée bleue et marqueurs carrés (square).

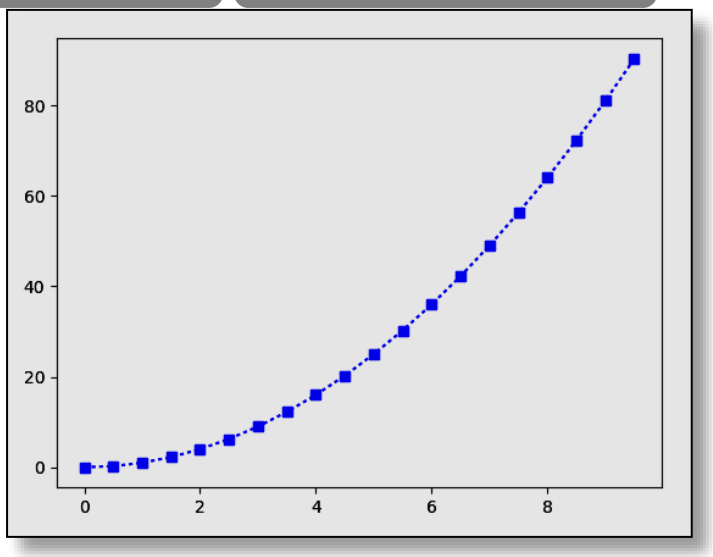
Code à copier-coller

En pylab

```
from pylab import *
t=arange(0,10,.5)
plot( t, t**2, ':bs')
show()
```

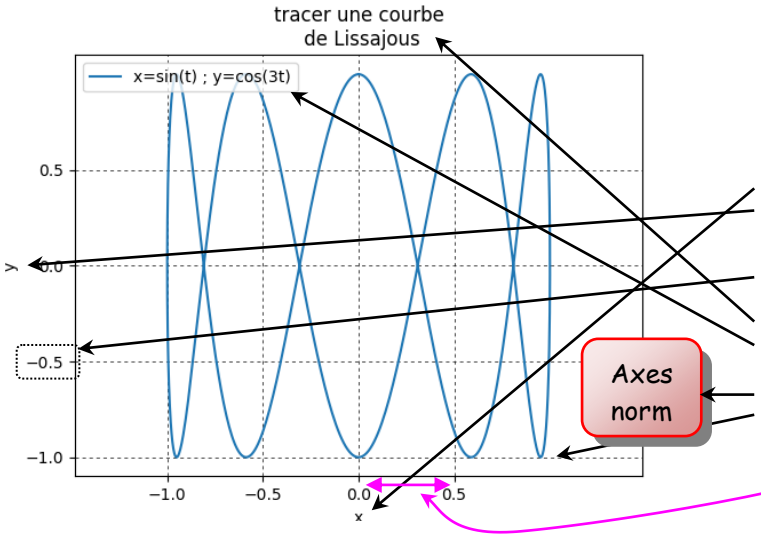
ou *en standard*

```
import numpy as np
import matplotlib.pyplot as plt
t=np.arange(0,10,.5)
plt.plot( t, t**2, ':bs')
plt.show()
```



F- Légender les graphes

Ce paragraphe peut être sauté dans un premier temps, en passant directement suivant, car il concerne quelques « détails » graphiques secondaires.




```
from pylab import *
t=arange(0,2*pi,.01)
plot(sin(t),cos(5*t),label='x=sin(t) ; y=cos(3t)')

xlabel('x')
ylabel('y')

xticks(arange(-1,1,.5)) # de -1 à +1 avec pas = 0.5 sur Ox
yticks(arange(-1,1,.5)) # de -1 à +1 avec pas = 0.5 sur Oy

title('tracer une courbe'+'\n de Lissajous')
legend(loc='upper left')

axis('equal')
grid('major',color='k',ls=':',lw=.5)

show()
```

Retour à la ligne

ls = line style

lw = line width

Les positions de la légende :

'best'	0
'upper right'	1
'upper left'	2
'lower left'	3
'lower right'	4
'right'	5
'center left'	6
'center right'	7
'lower center'	8
'upper center'	9
'center'	10

Code à copier-coller

En pylab

```
from pylab import *
t=arange(0,2*pi,.01)
plot(sin(t),cos(5*t),label='x=sin(t) ; y=cos(3t)')
xlabel('x')
ylabel('y')
xticks(arange(-1,1,.5)) # de -1 à +1 avec pas = 0.5 sur Ox
yticks(arange(-1,1,.5)) # de -1 à +1 avec pas = 0.5 sur Oy
title('tracer une courbe'+'\n de Lissajous')
legend(loc='upper left')
axis('equal')
grid('major',color='k',ls=':',lw=.5)
show()
```

ou *en standard*

```
import numpy as np
import matplotlib.pyplot as plt
t=np.arange(0,2*pi,.01)
plt.plot(np.sin(t),np.cos(3*t),label='x=sin(t) ;
y=cos(3t)')
plt.xlabel('x')
plt.ylabel('y')
plt.xticks(np.arange(-10,10,.5))
plt.yticks(np.arange(-10,10,.5))
plt.title('tracer une courbe de Lissajous')
plt.legend(loc='upper left')
plt.axis('equal')
plt.grid('major',color='k',ls=':',lw=.5)
plt.show()
```