



## PYTHON diapo 10

### Graphe d'une fonction

Roblet<sup>2</sup>

dernière MAJ le 24/10/19

## Diapo 10 Graphe d'une fonction

Dans ce diaporama vous apprendrez à créer sous Python le graphe d'une fonction donnée. Pour cela, on peut utiliser plusieurs bibliothèques. Nous nous concentrerons sur la bibliothèque **matplotlib.pyplot**.

La bibliothèque **matplotlib.pyplot** est une bibliothèque qui s'utilise souvent avec la bibliothèque **numpy**, qui nous permettra de subdiviser régulièrement un intervalle grâce à l'instruction « *linspace* ».

importation des bibliothèques

```
import matplotlib.pyplot as plt  
import numpy as np
```

alias souvent utilisés



## Diapo 10 Graphe d'une fonction

Voici un exemple de programme permettant de tracer la fonction  $f(x) = \frac{x^3}{6} - 2x + 1$  sur l'intervalle  $[-8 ; 3]$ .  
Cliquer sur une ligne de code pour avoir son explication !

Ajout de titres

Couleur et style des courbes

```
import matplotlib.pyplot as plt  
import numpy as np  
plt.axis([-8,3,-5,5])  
plt.grid()  
x=np.linspace(-7,2,50)  
y=x**3/6-2*x+1  
plt.plot(x,y)  
plt.show()
```

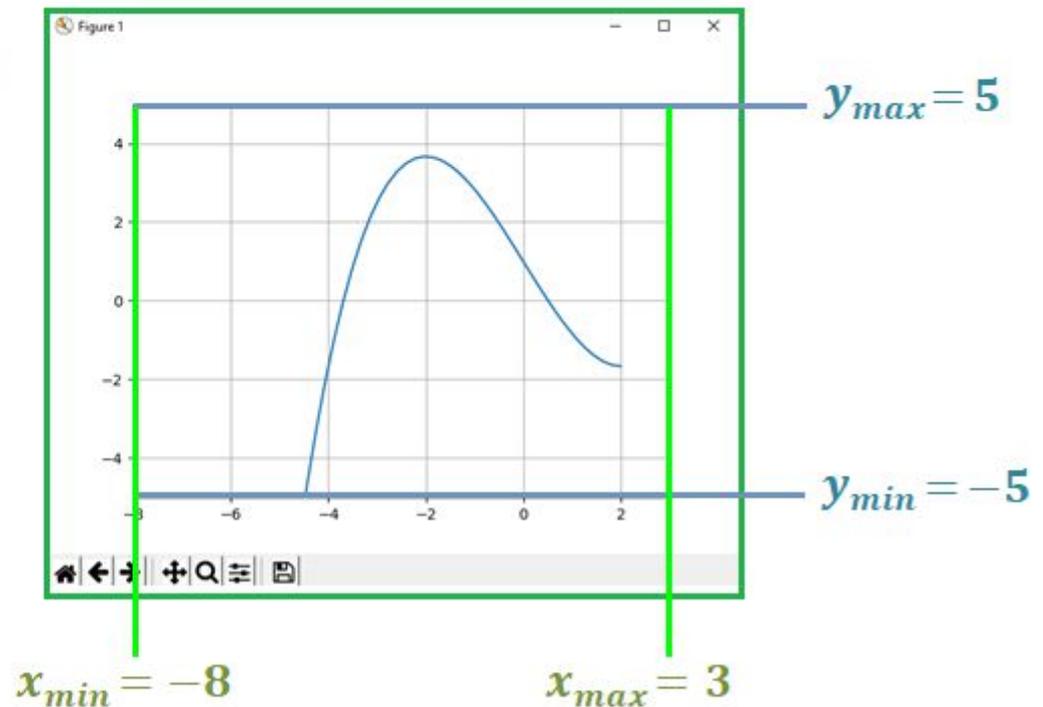
## Diapo 10 Graphe d'une fonction

```
plt.axis([ $x_{min}$ ,  $x_{max}$ ,  $y_{min}$ ,  $y_{max}$ ])
```

Cette instruction n'est pas obligatoire. On l'utilise si l'on veut imposer la taille de la fenêtre d'affichage.

### exemple

```
import matplotlib.pyplot as plt  
import numpy as np  
plt.axis([-8, 3, -5, 5])  
plt.grid()  
x=np.linspace(-7, 2, 50)  
y=x**3/6-2*x+1  
plt.plot(x,y)  
plt.show()
```



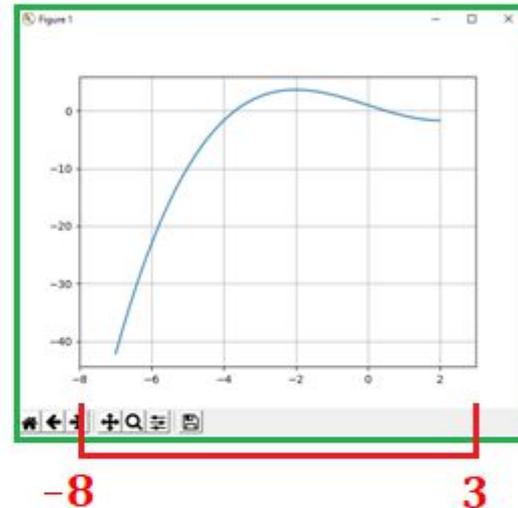
`plt.xlim( $x_{min}$ ,  $x_{max}$ )` et `plt.ylim( $y_{min}$ ,  $y_{max}$ )`

retour au  
début du  
diapo

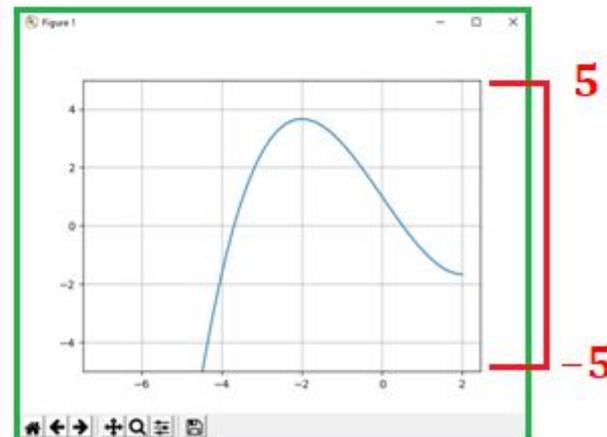
Il est aussi possible de fixer uniquement l'axe des abscisses ou uniquement l'axe des ordonnées avec « *xlim* » et « *ylim* ».

## exemple

```
import matplotlib.pyplot as plt
import numpy as np
plt.xlim(-8,3)
plt.grid()
x=np.linspace(-7,2,50)
y=x**3/6-2*x+1
plt.plot(x,y)
plt.show()
```



```
import matplotlib.pyplot as plt
import numpy as np
plt.ylim(-5,5)
plt.grid()
x=np.linspace(-7,2,50)
y=x**3/6-2*x+1
plt.plot(x,y)
plt.show()
```



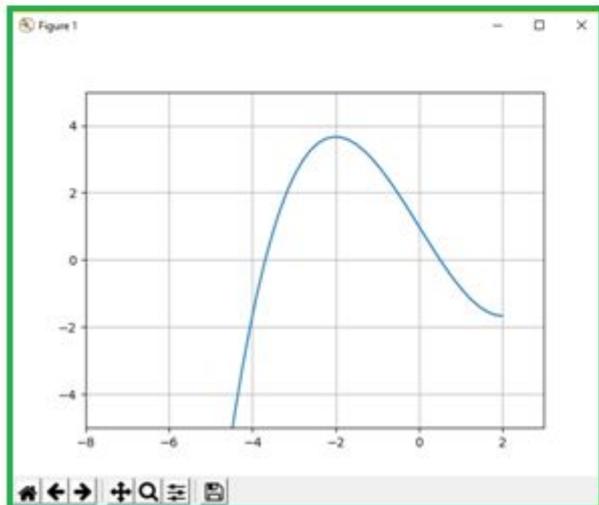
## plt.grid()

Cette instruction fait afficher un quadrillage.

### exemple

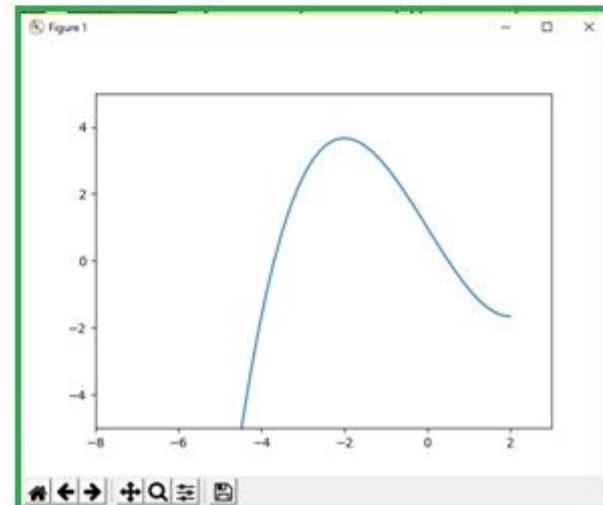
avec « grid »

```
import matplotlib.pyplot as plt
import numpy as np
plt.axis([-8,3,-5,5])
plt.grid()
x=np.linspace(-7,2,50)
y=x**3/6-2*x+1
plt.plot(x,y)
plt.show()
```



sans « grid »

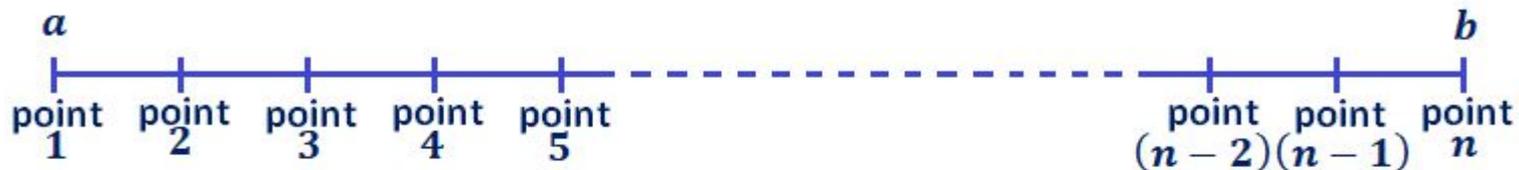
```
import matplotlib.pyplot as plt
import numpy as np
plt.axis([-8,3,-5,5])
x=np.linspace(-7,2,50)
y=x**3/6-2*x+1
plt.plot(x,y)
plt.show()
```



## `np.linspace(a, b, n)`

retour au  
début du  
diapo

Cette instruction crée une liste de  $n$  points répartis régulièrement sur l'intervalle  $[a; b]$ .



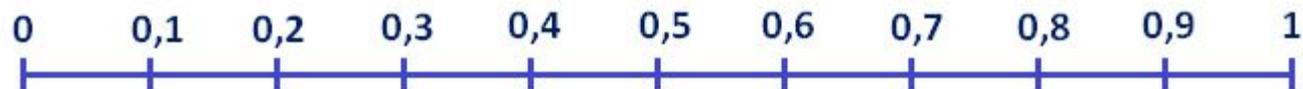
### exemple

```
import numpy as np
x=np.linspace(0,1,11)
```

dans la console

```
>>> x
array([0. , 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1. ])
```

On note ici que  $x$  est du type « array », c'est-à-dire une liste avec la condition supplémentaire que tous les éléments soient du même type (ici ce sont tous des réels).



## Diapo 10 Graphe d'une fonction

$y$  sera une liste dont chaque élément sera le résultat de l'opération avec chaque élément de la liste  $x$ .

### exemple

```
import numpy as np  
x=np.linspace(0,1,11)  
y=x+1
```

dans la console

```
>>> x  
array([0. , 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1. ])  
>>> y  
array([1. , 1.1, 1.2, 1.3, 1.4, 1.5, 1.6, 1.7, 1.8, 1.9, 2. ])
```

## Diapo 10 Graphe d'une fonction

### remarque importante concernant la définition de y

Avec **matplotlib.pyplot**, on ne peut pas utiliser les fonctions prédéfinies de la librairie **math** :

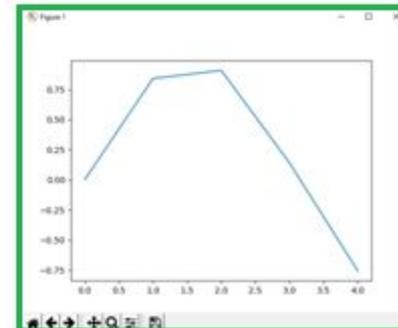
```
import matplotlib.pyplot as plt
from math import *
x=[0,1,2,3,4]
y=sin(x)
plt.plot(x,y)
plt.show()
```

← message d'erreur ici

Cependant les fonctions de **numpy** sont compatibles avec **matplotlib.pyplot**.

```
import matplotlib.pyplot as plt
from numpy import *
x=[0,1,2,3,4]
y=sin(x)
plt.plot(x,y)
plt.show()
```

```
import matplotlib.pyplot as plt
import numpy as np
x=[0,1,2,3,4]
y=np.sin(x)
plt.plot(x,y)
plt.show()
```



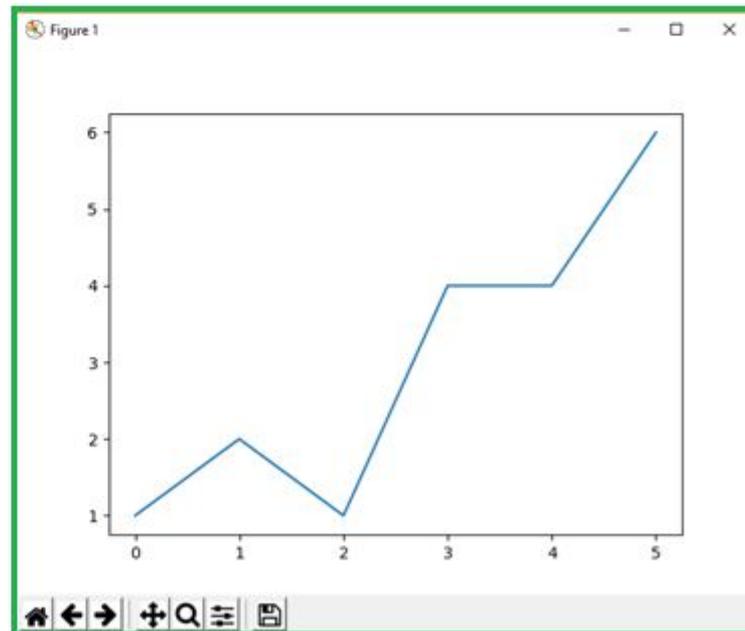
## Diapo 10 Graphe d'une fonction

`plt.plot(x,y)` 1/2

Si on a une liste  $x = [x_0, x_1, x_2, \dots, x_{n-1}]$  et une liste  $y = [y_0, y_1, y_2, \dots, y_{n-1}]$  alors l'instruction `plot` représentera les points  $M_i(x_i ; y_i)$  pour  $i \in \llbracket 0 ; n - 1 \rrbracket$  et les reliera par un segment.

### exemple

```
import matplotlib.pyplot as plt
x=[0,1,2,3,4,5]
y=[1,2,1,4,4,6]
plt.plot(x,y)
plt.show()
```



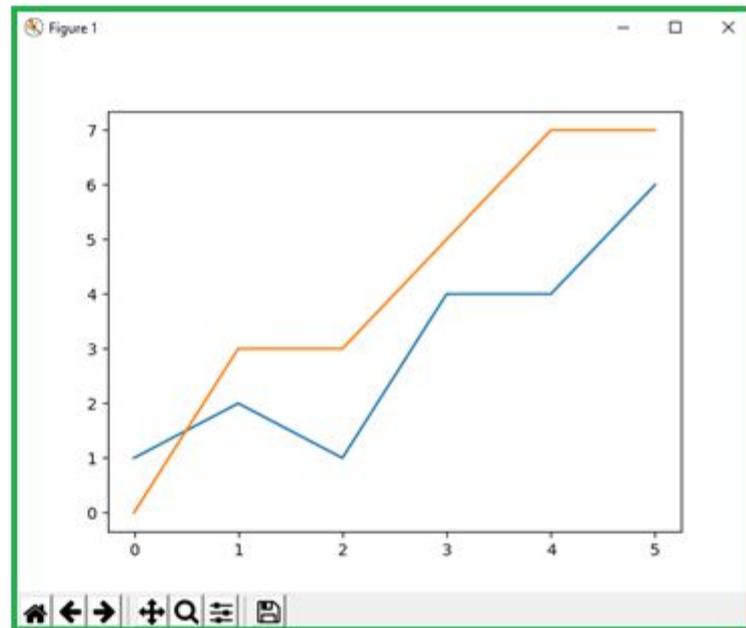
## Diapo 10 Graphe d'une fonction

`plt.plot(x,y)` 2/2

On peut aussi tracer plusieurs graphes dans un même repère.

### exemple

```
import matplotlib.pyplot as plt
x=[0,1,2,3,4,5]
y=[1,2,1,4,4,6]
z=[0,3,3,5,7,7]
plt.plot(x,y)
plt.plot(x,z)
plt.show()
```



## Diapo 10 Graphe d'une fonction

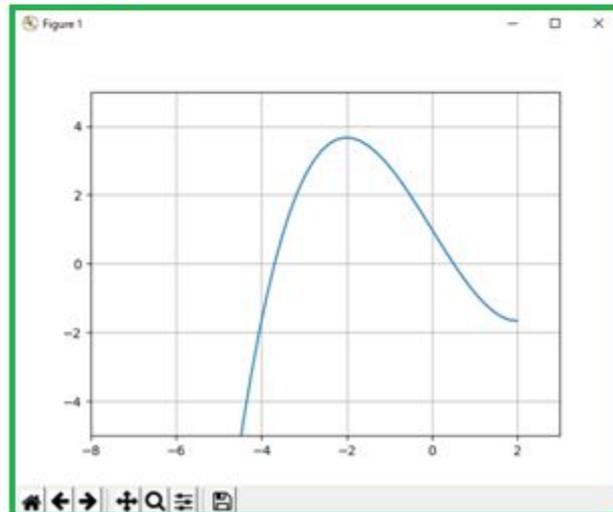
`plt.show()`

show permet d'afficher la courbe.

### exemple

avec « show »

```
import matplotlib.pyplot as plt
import numpy as np
plt.axis([-8,3,-5,5])
plt.grid()
x=np.linspace(-7,2,50)
y=x**3/6-2*x+1
plt.plot(x,y)
plt.show()
```



sans « show »

```
import matplotlib.pyplot as plt
import numpy as np
plt.axis([-8,3,-5,5])
plt.grid()
x=np.linspace(-7,2,50)
y=x**3/6-2*x+1
plt.plot(x,y)
```



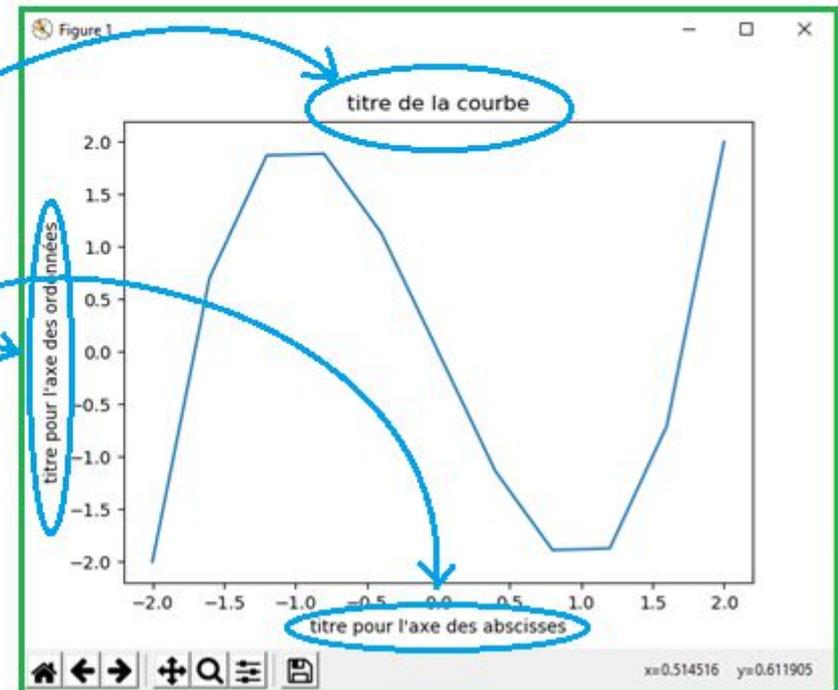
## Diapo 10 Graphe d'une fonction

### Ajout de titres

```
plt.title("...")  
plt.xlabel("...")  
plt.ylabel("...")
```

### exemple

```
import matplotlib.pyplot as plt  
import numpy as np  
x=np.linspace(-2,2,11)  
y=x**3-3*x  
plt.plot(x,y)  
plt.title("titre de la courbe")  
plt.xlabel("titre pour l'axe des abscisses")  
plt.ylabel("titre pour l'axe des ordonnées")  
plt.show()
```



# Diapo 10 Graphe d'une fonction

## Couleur et style des courbes

On peut changer le style des points ou de la courbe en ajoutant une chaîne de caractère dans `plot` :

```
plt.plot(x, y, "bs--")
```



ici la courbe sera bleue, les points de forme carrée, et la ligne en traits pointillés

style du trait	syntaxe
—	"_"
---	"_ _"
.....	".."
-.-.-	"_ ."

← style du trait par défaut

couleur	syntaxe
	"b" ou "blue"
	"c" ou "cyan"
	"g" ou "green"
	"k" ou "black"
	"m" ou "magenta"
	"r" ou "red"
	"y" ou "yellow"

style du point	syntaxe
+	"+"
x	"x"
●	"o"
■	"s"
★	"*"
▲	"^"
▼	"v"
▶	">"
◀	"<"