

Principe d'un notebook comme celui-ci :

- Les cellules de code sont toutes à exécuter, soit avec le bouton en haut, soit avec ctrl+entrée ou maj+entrée.
- Vous pouvez aussi éditer les textes (comme celui-ci) si vous en avez envie, ajouter des cellules, etc.
- Pensez à enregistrer pour pouvoir retrouver votre travail.

TP - Régime transitoire du circuit RLC série

Saisie et tracé des données :

```
Entrée[18]: import numpy as np          # pour La gestion des tableaux
import matplotlib.pyplot as plt # pour La gestion des graphiques

# Vos données (exemple données étudiants 2023) :
R = np.array([10,20,30,40,50,60,100,150,200,250,300,350,400])
Q = np.array([7.7,6.35,5.76,5.25,4.94,4.53,3.52,2.66,2.2,1.83,1.54,1.42,1.18])
a=1/Q
```

Régression linéaire :

```
Entrée[20]: # On effectue une régression linéaire sur Les données, du type y = a*x + b
x = R      # on pose x = ?
y = 1/Q    # on pose y = ?
a,b = np.polyfit(x,y,1)
print("a=",a," b=",b) # affiche a et b

atheo = (100e-9/40e-3)**.5
print("a_theo = ", ateo)

RbobetGBF = b/a
print("Rbob + RGBF (donné par b/a): ",RbobetGBF)
print("Rbob + RGBF - 50 = ",RbobetGBF-50)

# Tracé du graphique : tracer Les données (comme précédemment) et aussi La rég
# Remarque : La régression linéaire n'a un sens que si Les points sont alignés
plt.figure()
plt.plot(R,y,'o')
plt.plot(x,(a*x+b))
plt.xlabel("R")
plt.ylabel("1/Q")
plt.title('Régression linéaire de 1/Q en fonction de R')
plt.grid()
```

$a = 0.0017591373215161503$ $b = 0.11452047901495727$
 $a_{\text{theo}} = 0.0015811388300841897$
Rbob + RGBF (donné par b/a): 65.10036346466421
Rbob + RGBF - 50 = 15.100363464664213

Figure 17

