

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**Le pendule simple****q 1**

```
Entrée[1]: import numpy as np          # pour la gestion des tableaux
import matplotlib.pyplot as plt # pour la gestion des graphiques
from math import *
```

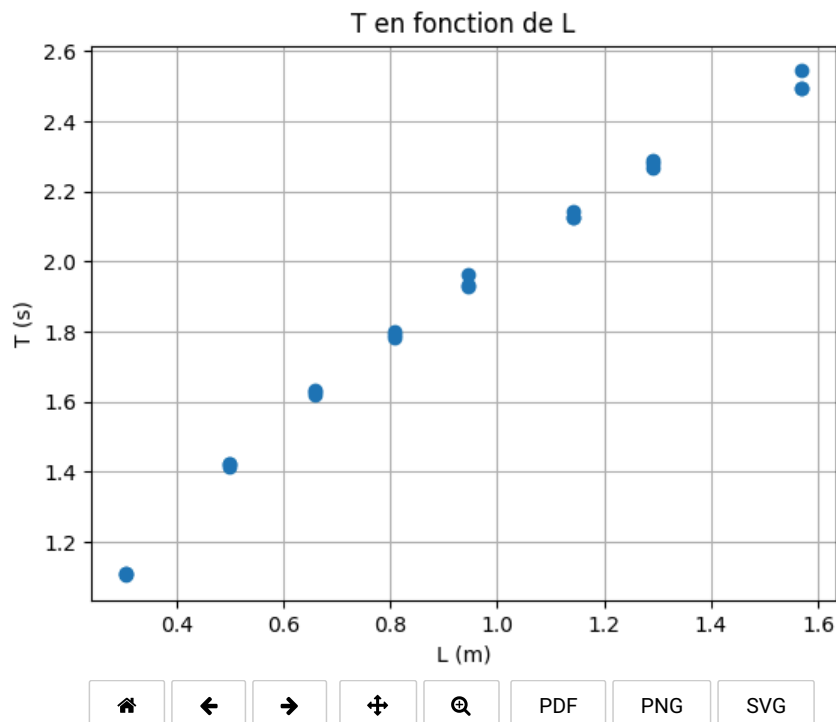
```
Entrée[2]: # Vos données :

# essai au lycée, masse 100g, ficelle épaisse
# la hauteur de la masse est de 26 mm.
# on mesure 10 périodes
#T = np.array([6.38, 7.6, 8.89, 10.04, 10.42, 11.01, 11.75, 12.18, 12.21, 12.52, 13.11]) / 10
#L = np.array([105 , 139, 201, 243, 264, 318, 338, 355, 368, 398, 433 ]) / 1e3

# essai ficelle rouge fine, masse 100g de hauteur 26mm
T = np.array([19.32/10, 19.27/10, 19.65/10, 32.62/20, 32.40/20, 32.54/20, 28.49/20, 28.42/20, 28.42/20, 28.42/20])
L = np.array([932+13, 932+13, 932+13, 645+13, 645+13, 645+13, 485+13, 485+13, 485+13, 291+13])

# Tracé du graphique
plt.figure()
plt.plot(L,T,'o')
plt.xlabel("L (m)")
plt.ylabel("T (s)")
plt.title('T en fonction de L')
plt.grid()
plt.show()
```

Figure 1



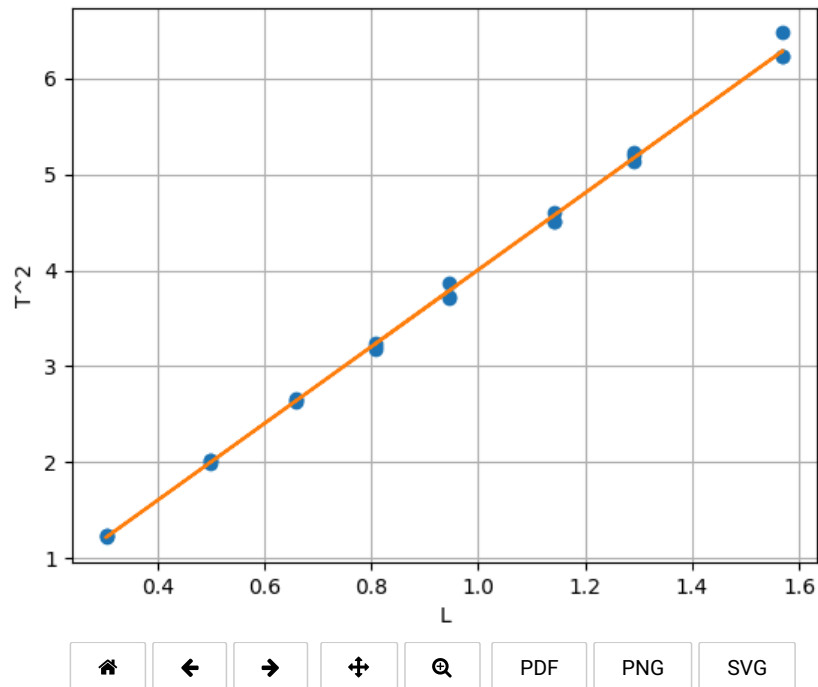
q2

```
Entrée[3]: # On effectue une régression linéaire sur les données, du type  $y = a*x + b$ 
x = L # on pose  $x = ?$ 
y = T**2 # on pose  $y = ?$ 
a,b = np.polyfit(x,y,1) # ceci donne les meilleurs a et b possibles
print("a=",a," b=",b) # affiche a et b
print('g à partir de la pente :', 4*pi**2/a)

# Tracé du graphique
plt.figure()
plt.plot(x,y,'o') # tracer les données (comme précédemment)
plt.plot(x,a*x+b,'-') # tracé de la régression linéaire
plt.xlabel("L")
plt.ylabel("T^2")
plt.grid()
plt.show()
```

a= 4.001444249548999 b= 0.003113648042278278
g à partir de la pente : 9.86604214436001

Figure 2



q3

```
Entrée[11]: g = 4*pi**2*L/T**2 # calcul de g à partir des tableaux L et T

N = len(g)
gmoy = np.mean(g)
ug = np.std(g,ddof=1)/N**.5
z = np.abs(np.mean(g)-9.81)/ug

print('N =', N)
print('g =', round(gmoy,3))
print('u(gmoy) =', round(ug,4))
print('z =', round(z,2))
print('u(gmoy)/gmoy =', round(ug/np.mean(g)*100,2), '%')
```

N = 24
g = 9.847
u(gmoy) = 0.0261
z = 1.43
u(gmoy)/gmoy = 0.27 %

Bilan des résultats des PTSI

```
Entrée[10]: # valeurs moyennes obtenues par chacun des groupes :
g = np.array([9.82,9.98,9.86,9.81,9.69,9.73]) # valeurs exclues car z>3 : 9.52,10.34])

N = len(g)
gmoy = np.mean(g)
ug = np.std(g,ddof=1)/N**.5
z = np.abs(np.mean(g)-9.81)/ug

print('N =', N)
print('g =', round(gmoy,3))
print('u(gmoy) =', round(ug,4))
print('z =', round(z,2))
print('u(gmoy)/gmoy =', round(ug/np.mean(g)*100,2), '%')

N = 6
g = 9.815
u(gmoy) = 0.0417
z = 0.12
u(gmoy)/gmoy = 0.42 %
```

Bilan des résultats des PCSI

```
Entrée[12]: # valeurs moyennes obtenues par chacun des groupes :
g = np.array([10.03,9.62,9.91,9.68,9.51,9.75,9.94,9.61,10.2, 9.81,9.83,9.64,10,9.94,9.52])

N = len(g)
gmoy = np.mean(g)
ug = np.std(g,ddof=1)/N**.5
z = np.abs(np.mean(g)-9.81)/ug

print('N =', N)
print('g =', round(gmoy,3))
print('u(gmoy) =', round(ug,4))
print('z =', round(z,2))
print('u(gmoy)/gmoy =', round(ug/np.mean(g)*100,2), '%')

N = 15
g = 9.799
u(gmoy) = 0.0525
z = 0.2
u(gmoy)/gmoy = 0.54 %
```

Bilan des résultats des PTSI et des PCSI

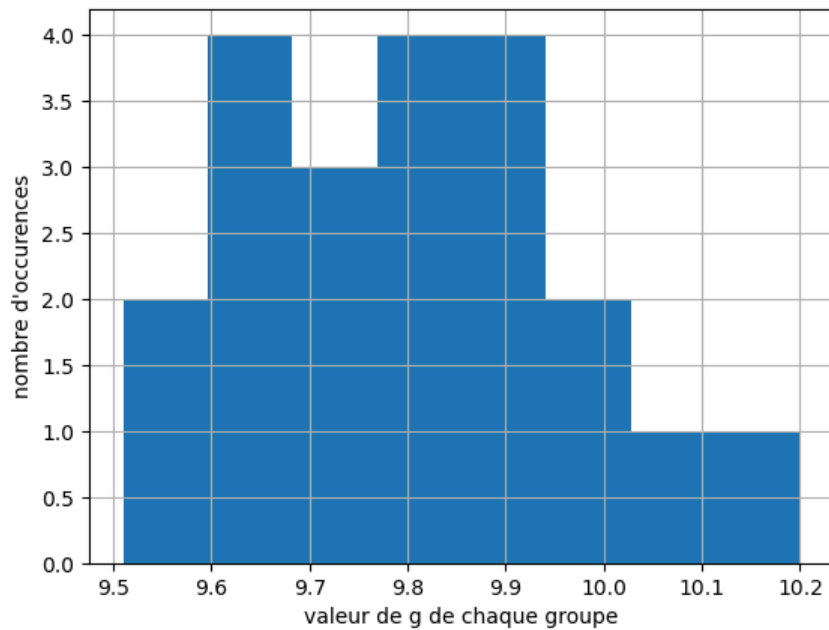
```
Entrée[14]: # valeurs moyennes obtenues par chacun des groupes :
g = np.array([9.82,9.98,9.86,9.81,9.69,9.73, 10.03,9.62,9.91,9.68,9.51,9.75,9.94,9.61,10.2,
N = len(g)
gmoy = np.mean(g)
ug = np.std(g,ddof=1)/N**.5
z = np.abs(np.mean(g)-9.81)/ug

print('N =', N)
print('g =', round(gmoy,3))
print('u(gmoy) =',round(ug,4))
print('z =', round(z,2))
print('u(gmoy)/gmoy =', round(ug/np.mean(g)*100,2), '%')

plt.figure()
plt.hist(g,bins=8)
plt.xlabel('valeur de g de chaque groupe')
plt.ylabel("nombre d'occurrences")
plt.grid()
plt.show()
```

N = 21
g = 9.804
u(gmoy) = 0.0388
z = 0.16
u(gmoy)/gmoy = 0.4 %

Figure 6



Entrée[]: