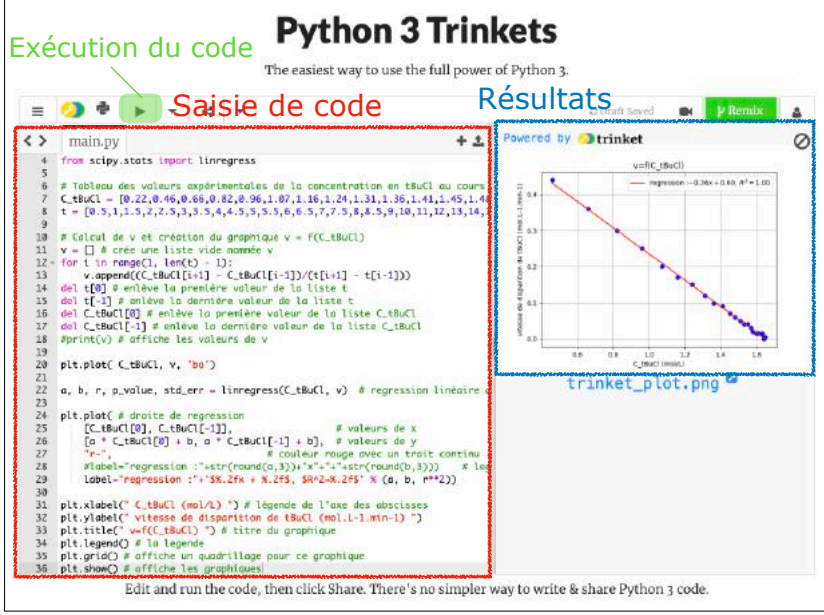
Introduction à Python
Physique-Chimie – Terminale

Objectif : utiliser Python comme outil de calcul et de représentation graphique

Fiche méthode « Python » sur sciencespartout.fr

Livre Hatier Rabat II et III (à la fin du livre)

https://trinket.io/python3



Objectifs

- Comprendre la logique de base d'un programme Python
- Manipuler des variables et des listes
- Utiliser des boucles et des conditions
- Tracer une trajectoire et calculer des vitesses à partir de données expérimentales

Variables

Une variable permet de stocker une valeur.

📘 Exemples

- x = 5
- t = 0.20
- message = "Physique"

Modifier une variable

On peut modifier une variable existante.

📘 Exemple

- a = 3
- a = a + 1 # a vaut maintenant 4
- a += 1 # écriture équivalente

Listes

Une liste stocke plusieurs valeurs.

📘 Exemples

- t = [0.0, 0.1, 0.2]
- x = [0.00, 0.12, 0.25]
- Vide=[]

❗ Important : le premier indice est 0.

Manipuler une liste

Accéder à un élément :

- x[0] → premier élément
- X[2] → troisième élément
- t[:-1] → toute la liste sans le dernier élément

Ajouter un élément (0.40) à la fin de la liste x :

- x.append(0.40)

Longueur de la liste :

- len(x)

Blocs d'instructions

En Python, les blocs sont définis par l'indentation et suivent le caractère « : »

Exemple

```
if v > 5:
    print("Rapide")
    print("et bien !")
```

bloc d'instructions exécuté lorsque la condition est vraie.

Indentation : décalage vers la droite

print() permet d'afficher à l'écran le contenu d'une variable ou un message.

Conditions : if / else

Permettent d'exécuter ou non un bloc.

Exemple

```
if v > 5:
    print("Rapide")
else:
    print("Lent")
```

bloc d'instructions exécuté lorsque la condition est vraie.

bloc d'instructions exécuté lorsque la condition est fausse.

Boucle for

Permet de répéter un bloc un nombre connu de fois.

Exemple

```
for i in range(5):
    print(i)
```

→ i prend les valeurs 0, 1, 2, 3, 4

La fonction range() : définir un intervalle

La fonction range() permet de définir une suite d'entiers, souvent utilisée dans une boucle for.

Syntaxe : range(début, fin, pas)

- début : valeur de départ (incluse)
- fin : valeur de fin (exclue)
- pas : incrément (optionnel, par défaut = 1)

Exemples

- range(5)
- range(1, 5)
- range(1, 10, 2)

Tracer un graphique

Avec matplotlib :

pyplot est un sous-module de la bibliothèque matplotlib, dédié au tracé de graphiques.

Importation du sous-module pyplot permettant de tracer des graphiques

- import matplotlib.pyplot as plt
- plt.plot(x, y)
- plt.xlabel("x (m)")
- plt.ylabel("y (m)")
- plt.show()

prépare le graphique , définit ce que l'on veut tracer(points, lignes, données) et ajoute une courbe à la figure

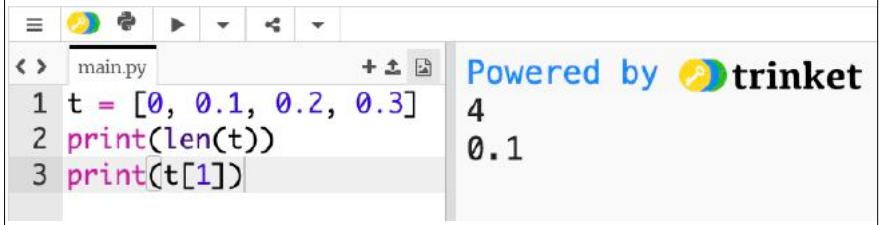
affiche les titres de l'axe des abscisses (axe x) et des ordonnées (axe y)

affiche la figure construite à l'écran, déclenche l'ouverture de la fenêtre graphique et termine le tracé

🔢 Application 1

- Créer une liste t contenant 0 ; 0,1 ; 0,2 ; 0,3
- Afficher la longueur de la liste
- Afficher le deuxième élément

Correction – Application 1



🚀 Application 2

- On dispose des listes t et x (chronophotographie) :
- t = [0, 0.1, 0.2]
- x = [0.00, 0.20, 0.54]

- Calculer les vitesses moyennes suivant x entre deux points.

Correction – Application 2

```
t = [0, 0.1, 0.2]
x = [0.00, 0.20, 0.54]
vx = []
for i in range(len(t)-1):
    vx.append((x[i+1]-x[i])/(t[i+1]-t[i]))
print(vx)
```

Powered by trinket

```
[2.0, 3.4]
```

📸 Exemple final : chronophotographie

- Données : t, x, y issus d'un pointage.
  - t = [0.0, 0.1, 0.2, 0.3]
  - x = [0.00, 0.12, 0.25, 0.40]
  - y = [1.20, 1.15, 1.05, 0.90]
- Calculer vx, vy puis la vitesse v.
- Tracer la trajectoire : plot(x, y)
- Afficher les vitesses v

Correction

```python
import matplotlib.pyplot as plt
import math

# Données
t = [0.0, 0.1, 0.2, 0.3]
x = [0.00, 0.12, 0.25, 0.40]
y = [1.20, 1.15, 1.05, 0.90]
v = []

# Calcul des vitesses
for i in range(len(t) - 1):
    vx_i = (x[i+1] - x[i]) / (t[i+1] - t[i])
    vy_i = (y[i+1] - y[i]) / (t[i+1] - t[i])
    v.append(math.sqrt(vx_i**2 + vy_i**2))

# Tracé de la trajectoire
plt.plot(x, y, 'r', marker='o')
plt.xlabel("x (m)")
plt.ylabel("y (m)")
plt.title("Trajectoire du mobile")
plt.show()

# Afficher les vitesses
print("les valeurs des vitesses sont : ")
# Arrondir à deux chiffres après la virgule à l'affichage
for val in v:
    print(f"{val:.2f}")
```