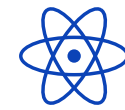




Le langage Python

Écrire, exécuter et comprendre un programme avec Spyder.

COURS



Prof-TC

SNT

LE LANGAGE PYTHON

Python est un langage lisible et polyvalent : calcul scientifique, traitement de données, simulations et graphiques.

OBJECTIF : PASSER D'UN ALGORITHME À UN PROGRAMME EXÉCUTABLE

1 — PYTHON ET SPYDER

Spyder est un environnement de développement adapté aux sciences. Il regroupe un éditeur, une console, un explorateur de variables et des graphiques.

1. ÉDITEUR

écrire et enregistrer le script .py

2. CONSOLE

exécuter et tester une instruction

3. VARIABLES

observer noms, types et valeurs

4. GRAPHIQUES

afficher les figures produites

MÉTHODE DE TRAVAIL

1. créer un dossier ; 2. placer script et données ensemble ; 3. enregistrer ; 4. exécuter ; 5. corriger puis recommencer.

BONNE PRATIQUE

Enregistrer avant d'exécuter avec le bouton ► ou la touche F5.

2 — PREMIER PROGRAMME

Un programme est une suite d'instructions exécutées de haut en bas. Le symbole # introduit un commentaire ignoré par Python.

```
# Mon premier programme
nom = "Ada"
age = 16
print("Bonjour", nom)
print("Âge :", age)
```

RÈGLES ESSENTIELLES

- respecter les majuscules et minuscules ;
- ne pas oublier guillemets et parenthèses ;
- choisir des noms explicites ;
- lire le message en cas d'erreur.

FICHIERS

Un script porte l'extension .py. Éviter espaces, accents et noms de modules dans le nom du fichier.

ERREUR ≠ ÉCHEC

Le message indique la ligne et la nature du problème : SyntaxError, NameError, TypeError...

3 — VARIABLES ET TYPES

Une variable associe un nom à une valeur. Son type détermine les opérations possibles.

int entier `n = 12`

float décimal `x = 2.5`

str texte `ville = "Paris"`

bool logique `ok = True`

```
a = float(input("a = "))
print(type(a))
```

CONVERSIONS

`str(3.2)` `int("12")` `float("2.5")`

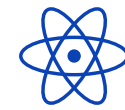
input() renvoie du texte. On convertit avec int() ou float() avant de calculer.



Calculer avec Python

Variables, opérateurs, fonctions mathématiques et précision.

COURS



Prof-TC

SNT

LE LANGAGE PYTHON

Python peut servir de calculatrice, mais surtout automatiser des calculs reproductibles avec des unités cohérentes.

1 — OPÉRATEURS

+ addition

- soustraction

***** multiplication

/ division

****** puissance

// quotient entier

% reste

```
E = m * c**2  
moyenne = (a + b + c) / 3
```

Priorité : () puis ** puis * / puis + -

2 — MODULE MATH

Un module apporte de nouvelles fonctions. On l'importe au début du script.

```
import math  
angle = math.radians(30)  
s = math.sin(angle)  
r = math.sqrt(2)  
p = math.pi
```

EXEMPLE : CHUTE LIBRE

```
g = 9.81      # m/s²  
h = 20        # m  
t = math.sqrt(2*h/g)  
print("t =", round(t, 2), "s")
```

ATTENTION AUX UNITÉS

Convertir les données avant le calcul : cm → m, km/h → m/s, degrés → radians.

```
v_m_s = v_km_h / 3.6
```

3 — CALCUL SCIENTIFIQUE

NumPy manipule efficacement des tableaux de nombres. Il est très utilisé en physique et en traitement de données.

```
import numpy as np  
x = np.array([0, 1, 2, 3])  
y = x**2 + 2*x - 1  
print(y)
```

FONCTIONS UTILES

`np.linspace(a,b,n)`
n valeurs de a à b

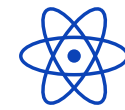
`np.mean(x)`
moyenne

`np.min(x), np.max(x)`
extrêmes

`np.exp(x), np.log(x)`
exp. et logarithme

`np.sin(x), np.cos(x)`
trigonométrie

Les nombres décimaux sont approchés : comparer avec une tolérance plutôt qu'avec ==.



LE LANGAGE PYTHON

L'indentation (décalage vers la droite) délimite les blocs. Dans Spyder, utiliser quatre espaces.

1 — TESTER UNE CONDITION

Une condition choisit les instructions à exécuter selon une expression vraie ou fausse.

```
T = 18
if T < 0:
    etat = "solide"
elif T < 100:
    etat = "liquide"
else:
    etat = "gaz"
print(etat)
```

COMPARAISONS

== égal ; != différent ; <, <=, >, >= ; and, or, not combinent des conditions.

Ne pas confondre = (affectation) et == (comparaison).

2 — RÉPÉTER : LES BOUCLES

for parcourt une suite connue ; while répète tant qu'une condition reste vraie.

```
# Carrés de 0 à 5
for n in range(6):
    print(n, n**2)

# Seuil
x = 1
while x < 100:
    x = 2*x
```

ACCUMULER

```
somme = 0
for k in range(1, 11):
    somme = somme + k
```

range(début, fin, pas) exclut toujours la valeur de fin.

3 — CRÉER UNE FONCTION

Une fonction regroupe un calcul réutilisable. Les paramètres sont les entrées ; return fournit le résultat.

```
def energie_cinetique(m, v):
    """Énergie en joules."""
    return 0.5 * m * v**2

E = energie_cinetique(0.145, 20)
print(round(E, 1), "J")
```

DÉCOMPOSER UN PROBLÈME

1. identifier entrées et sortie ;
2. écrire la formule ou méthode ;
3. tester avec un cas simple ;
4. vérifier unité et ordre de grandeur.

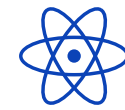
Une fonction doit accomplir une tâche précise et porter un nom explicite.



Tracer avec Python

Représenter des données et des modèles avec Matplotlib dans Spyder.

COURS



Prof-TC

SNT

LE LANGAGE PYTHON

Un graphique scientifique comporte des axes nommés, des unités, un titre, une légende et une échelle adaptée.

1 — TRACER UNE COURBE

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

t = np.linspace(0, 5, 101)
v = 9.81 * t

plt.plot(t, v, color="blue", label="v(t)")
plt.xlabel("Temps t (s)")
plt.ylabel("Vitesse v (m/s)")
plt.title("Chute libre sans frottement")
plt.grid()
plt.legend()
plt.show()
```

DANS SPYDER

Exécuter le script : la figure apparaît dans le panneau Graphiques. `plt.show()` force son affichage.

2 — DONNÉES EXPÉRIMENTALES

Pour des mesures, utiliser des points. Une incertitude peut être représentée par des barres d'erreur.

```
x = [0, 1, 2, 3]
y = [0.1, 2.0, 3.9, 6.2]
u = [0.2, 0.2, 0.3, 0.3]

plt.errorbar(x, y, yerr=u,
             fmt="o",
             capsize=4)
```

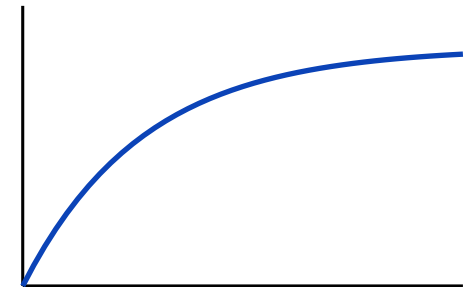
AJUSTER UNE DROITE

```
a, b = np.polyfit(x, y, 1)
modele = a*np.array(x) + b
plt.plot(x, modele)
```

Un ajustement ne prouve pas un modèle : il faut analyser les écarts et les incertitudes.

3 — LIRE ET AMÉLIORER

Grandeur y (unité)



Temps t (s)

CHECK-LIST

- ☐ axes et unités
- ☐ titre informatif
- ☐ légende utile
- ☐ points visibles
- ☐ échelle pertinente
- ☐ modèle distingué des mesures

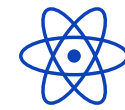
Toujours confronter la forme du graphe au phénomène étudié.



Données et fichiers

Listes, tableaux NumPy, lecture CSV et organisation des résultats.

COURS



Prof-TC

SNT

LE LANGAGE PYTHON

Un programme scientifique sépare les données, leur traitement, leur représentation et l'interprétation des résultats.

1 — LISTES ET TABLEAUX

Une liste Python peut contenir plusieurs valeurs. Un tableau NumPy permet des calculs terme à terme, plus rapides et plus lisibles.

```
mesures = [2.1, 2.0, 2.2, 1.9]
mesures.append(2.1)
print(len(mesures))
print(mesures[0])

import numpy as np
m = np.array(mesures)
print(np.mean(m), np.std(m))
```

INDICES ET TRANCHES

```
m[0]      # première valeur
m[-1]     # dernière valeur
m[1:3]    # indices 1 et 2
m[m > 2]  # filtrage
```

L'indice de la première valeur est 0. Une tranche exclut sa borne finale.

2 — LIRE UN FICHIER CSV

Un fichier CSV stocke une table en texte. En France, le séparateur est souvent le point-virgule et la virgule peut noter les décimales.

```
import numpy as np

data = np.loadtxt("mesures.csv",
                  delimiter=";",
                  skiprows=1)

t = data[:, 0]
U = data[:, 1]
```

CHEMIN DU FICHIER

Le plus simple est de placer le CSV dans le même dossier que le script. Sinon, préciser son chemin complet.

STRUCTURE ATTENDUE

```
temps; tension
0.0; 0.12
0.5; 1.08
1.0; 2.03
```

Toujours contrôler le nombre de lignes, les unités et les valeurs manquantes.

3 — EXPLOITER ET SAUVEGARDER

Après traitement, on peut calculer des indicateurs, créer une nouvelle table puis l'enregistrer sans modifier les données brutes.

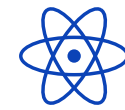
```
moy = np.mean(U)
Ucorr = U - moy
resultats = np.column_stack((t, Ucorr))

np.savetxt("resultats.csv", resultats,
           delimiter=";",
           header="temps;U_corrigee",
           comments="")
```

CONTRÔLES AVANT CONCLUSION

- afficher quelques valeurs ;
- rechercher minimum et maximum ;
- tracer les données ;
- vérifier unités et chiffres significatifs ;
- conserver les données originales.

Un résultat numérique doit être accompagné de son unité, de sa précision et d'une interprétation.



LE LANGAGE PYTHON

Une photographie matricielle est un tableau de pixels : chaque pixel possède généralement trois composantes Rouge, Verte et Bleue.

1 — OUVRIR ET AFFICHER

Avec Pillow, l'image est ouverte depuis le dossier du script. Matplotlib permet ensuite de l'afficher dans Spyder.

```
from PIL import Image
import matplotlib.pyplot as plt

img = Image.open("photo.jpg")
print(img.format, img.mode, img.size)
plt.imshow(img)
plt.axis("off")
plt.show()
```

DIMENSIONS ET MODE

`img.size` renvoie (largeur, hauteur). Le mode RGB utilise trois composantes de 0 à 255 ; RGBA ajoute la transparence.

REPÈRE DE L'IMAGE

L'origine (0, 0) est en haut à gauche. x augmente vers la droite et y vers le bas.

2 — PIXELS ET TRANSFORMATIONS

On convertit l'image en tableau NumPy pour accéder aux composantes et appliquer le même calcul à tous les pixels.

```
import numpy as np
tab = np.array(img)
print(tab.shape)
R = tab[:, :, 0]
V = tab[:, :, 1]
B = tab[:, :, 2]
```

PASSAGE EN NIVEAUX DE GRIS

```
gris = 0.299*R + 0.587*V + 0.114*B
gris = gris.astype("uint8")
plt.imshow(gris, cmap="gray")
```

MODIFIER UN PIXEL

```
tab[100, 200] = [255, 0, 0]
# ligne y=100, colonne x=200
```

Après un calcul, ramener les valeurs dans [0 ; 255] avec `np.clip`.

3 — RECADRER ET SAUVEGARDER

Pillow fournit des opérations pratiques qui conservent l'image originale si l'on crée un nouvel objet.

```
zone = img.crop((100, 50, 500, 350))
petite = img.resize((800, 600))
miroir = img.transpose(
    Image.Transpose.FLIP_LEFT_RIGHT)

zone.save("recadrage.png")
petite.save("photo_reduite.jpg", quality=90)
```

FILTRE SIMPLE : NÉGATIF

```
negatif = 255 - tab
Image.fromarray(negatif).save(
    "negatif.png")
```

PRÉCAUTIONS

- travailler sur une copie ;
- respecter le droit à l'image ;
- distinguer JPEG (compressé) et PNG ;
- vérifier dimensions et poids du fichier ;
- documenter chaque transformation.

Chaîne de traitement : charger → contrôler → transformer → afficher → sauvegarder.