

Programmation Python

Programmer l'exécution d'une séquence d'instructions exige une réflexion préliminaire quant à la structuration du code. L'utilisation de fonctions s'avère particulièrement bénéfique dans l'articulation des commandes.

Exemple de code Python, écrit en mode **Éditeur**.

```
# Détermination des racines d'un trinôme du second degré

from math import sqrt

# La fonction poly renvoie la liste des 3 coefficients d'un trinôme.
def poly(a,b,c):
    coeff = [a, b, c]
    return coeff

# La fonction delta renvoie le discriminant à partir de la liste
# de coefficients [a, b, c].
def delta(L):
    a, b, c = L[0], L[1], L[2]
    discriminant = b**2 - 4*a*c
    return discriminant

# La fonction racines renvoie la ou les racines d'un trinôme
def racine(L):
    if delta(L) < 0:
        return "Le discriminant est négatif. Aucune racine."
    elif delta(L) == 0:
        x_0 = -L[1]/(2*L[0])
        return f"Le discriminant est nul. Une racine double : {x_0}."
    else:
        x_1 = (-L[1] - sqrt(delta(L)))/(2*L[0])
        x_2 = (-L[1] + sqrt(delta(L)))/(2*L[0])
        return f"Le discriminant est égal à {delta(L)}. Le trinôme possède 2 racines : {x_1} et {x_2}."
```

Exécution du code Python dans l'**Interpréteur**.

```
Python 3.13.7 (v3.13.7:bcee1c32211, Aug 14 2025, 19:10:51) [Clang 16.0.0 (clang-1600.0.26.6)] on darwin
Enter "help" below or click "Help" above for more information.

>>> = RESTART: /Users/patrickjanc/Documents/lycée_2026_27/spe_prem/Functions/Trinôme
/Racines_Trinôme/Algo_Trinome/prgm_trinomes.py
>>> p = poly(1,5,6)
>>> p
[1, 5, 6]
>>> delta(p)
1
>>> racine(p)
'Le discriminant est égal à 1. Le trinôme possède 2 racines : -3.0 et -2.0.'
```

L'instruction `p = poly(1,5,6)` crée un trinôme dénommé `p`.

L'instruction `delta(p)` affiche la valeur calculée du discriminant du trinôme `p`.

L'instruction `racine(p)` affiche la valeur calculée du discriminant et les éventuelles racines du trinôme `p`.