

TP9 En Python

Représenter graphiquement deux vecteurs égaux avec Python

Allumer l'ordinateur et connectez-vous en utilisant votre login et votre mot de passe puis lancer « Python en ligne : <https://www.codabrainy.com/python-compiler/> ».

Introduction

Ce code Python sert à **représenter graphiquement deux vecteurs égaux**, appliqués en **deux points différents du plan**. Il montre que l'égalité de vecteurs ne dépend pas de leur position, mais uniquement de leurs composantes.

Le programme commence par importer deux bibliothèques. `numpy` est utilisée pour définir les vecteurs sous forme de tableaux numériques. `matplotlib.pyplot` sert à tracer les figures.

Les vecteurs `v1` et `v2` sont ensuite définis avec les mêmes composantes (2, 3). Ils sont donc mathématiquement égaux. Deux points d'application sont choisis : `origin1 = (0, 0)` et `origin2 = (1, 0)`. Cela permet de dessiner le même vecteur à deux endroits différents du plan.

Activité N°1 (Représenter graphiquement deux vecteurs égaux)

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

# Définition des vecteurs
v1 = np.array([2, 3])
v2 = np.array([2, 3])

# Origine commune
origin1 = np.array([0, 0])
origin2 = np.array([1, 0])
# Tracé
plt.quiver(*origin1, *v1, angles='xy', scale_units='xy', scale=1, color='blue', label='v1 = (2, 3)')
plt.quiver(*origin2, *v2, angles='xy', scale_units='xy', scale=1, color='red', label='v2 = (1, 4)')

# Mise en forme
plt.xlim(-1, 5)
plt.ylim(-1, 7)
plt.gca().set_aspect('equal')
plt.grid()
plt.legend()
plt.title("Graphiquement deux vecteurs égaux")

plt.show()
```

Activité N°2

Les vecteurs `v3` et `v4` possèdent les mêmes composantes (1, 4). Donner un algorithme qui représente ces deux vecteurs en utilisant deux couleurs différentes, avec pour points de départ respectifs `origin3=(2, 0)` et `origin4=(3, 0)`.

Correction Activité N°2

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

# Définition des vecteurs
v1 = np.array([2, 3])
v2 = np.array([2, 3])
v3 = np.array([1, 4])
v4 = np.array([1, 4])

# Origine commune
origin1 = np.array([0, 0])
origin2 = np.array([1, 0])
origin3 = np.array([2, 0])
origin4 = np.array([3, 0])
# Tracé
plt.quiver(*origin1, *v1, angles='xy', scale_units='xy', scale=1, color='blue', label='v1 = (2, 3)')
plt.quiver(*origin2, *v2, angles='xy', scale_units='xy', scale=1, color='red', label='v2 = (2, 3)')
plt.quiver(*origin3, *v3, angles='xy', scale_units='xy', scale=1, color='green', label='v3 = (1, 4)')
plt.quiver(*origin4, *v4, angles='xy', scale_units='xy', scale=1, color='brown', label='v4 = (1, 4)')

# Mise en forme
plt.xlim(-1, 5)
plt.ylim(-1, 7)
plt.gca().set_aspect('equal')
plt.grid()
plt.legend()
plt.title("Graphiquement deux vecteurs égaux")

plt.show()
```