

Introduction à Python

Seconde

qkzk

2024-11-17

Introduction à la programmation avec Python

Nous allons apprendre à écrire nos premiers programmes pour résoudre des problèmes mathématiques avec votre calculatrice NumWorks ou son émulateur.

Python est un langage de programmation adapté aux novices. Il est utilisé durant tout le lycée dans différentes disciplines pour s'initier à la programmation et s'intéresser à l'informatique.

C'est un *vrai* langage de programmation avec lequel on peut *tout* programmer. Par exemple, Instagram est principalement écrit en Python.

1. Qu'est-ce qu'un programme ?

Un **programme** est une suite d'**instructions** que l'ordinateur (ou la calculatrice) exécute une par une. Par exemple, une instruction peut être une commande pour afficher un message ou pour effectuer un calcul. Regardons un exemple simple :

```
print("bonjour Python")
```

Cette instruction dit à l'ordinateur d'afficher le texte *Bonjour, Python !*. Essayez-la sur votre calculatrice pour voir le résultat.

1. Aller dans le menu Python
2. Tout en bas, "Ajouter un script". L'appeler `exemple.py`
3. Écrire le code :

```
print("bonjour python")
```

4. Quitter avec la touche "retour" à côté de OK
5. Aller sur les trois points ..., appuyer sur OK et "*Exécuter le script*"

2. Notion d'affectation

Une **affectation** permet de stocker une valeur dans une variable. Une **variable** est comme une boîte qui contient une valeur. Voici un exemple :

```
a = 5  
b = 3
```

Ici, nous avons deux variables, `a` et `b`, qui contiennent respectivement les valeurs 5 et 3. Une fois qu'une valeur est dans une variable, nous pouvons l'utiliser dans des calculs :

```
somme = a + b  
print(somme)
```

Essayez ce code et modifiez les valeurs de **a** et **b** pour observer comment le résultat change.

1. Aller dans le menu Python
2. Tout en bas, "Ajouter un script". L'appeler `somme.py`
3. Écrire le code :

```
a = 3
b = 5
somme = a + b
print(somme)
```

4. Sortir, exécuter le script

Autre exemple d'affectation

```
a = 4
b = 5
a = a + b
b = a + b
```

1. Faire dérouler le code suivant à la main sur une feuille et prédire la valeur de **b**
2. Programmer cet exemple, l'exécuter et vérifier.

3. Les fonctions : des outils pour automatiser des calculs

Une **fonction** est un bloc de code qui effectue une tâche spécifique. Elle ressemble beaucoup à une fonction en mathématiques : elle prend des **paramètres** (les données d'entrée), effectue un calcul, puis renvoie un **résultat**.

En Python, une fonction se définit ainsi :

```
def ma_fonction(x):
    return x * 2
```

Ici, la fonction `ma_fonction` prend un paramètre `x`, multiplie sa valeur par 2, et renvoie le résultat. Essayons :

```
resultat = ma_fonction(4)
print(resultat) # Affiche 8
```

Voici le code complet :

```
def ma_fonction(x):
    return x * 2

resultat = ma_fonction(4)
print(resultat)
```

1. Programmer cet exemple dans `fonction.py`
2. L'exécuter

Exemple avec les mathématiques

En géométrie, nous avons étudié la distance entre deux points dans un plan.

Si les coordonnées de deux points $A(x_A, y_A)$ et $B(x_B, y_B)$ sont données, la distance se calcule avec la formule :

$$AB = \sqrt{(x_B - x_A)^2 + (y_B - y_A)^2}$$

Nous allons créer une fonction Python pour calculer cette distance automatiquement.

4. Fonctions liées au chapitre *Repérage et vecteurs*

1. `distance.py` : Calculer la distance entre deux points

La fonction `distance` prend en entrée 4 paramètres : `xa`, `ya`, `xb`, et `yb`. Elle calcule et renvoie la distance entre les points *A* et *B*.

```
from math import *  
  
def distance(xa, ya, xb, yb):  
    return sqrt((xb - xa)**2 + (yb - ya)**2)
```

1. Programmer cet exemple
2. Calculer à la main la distance entre les points *A*(3;2) et *B*(7;5).
3. Vérifier le résultat.
 - Programmez la fonction,
 - Exécutez le script.
 - Testez avec `distance(3, 2, 7, 5)`

2. `milieu.py` : Calculer les coordonnées du milieu d'un segment

La fonction `milieu` calcule le milieu du segment entre *A*(x_a, y_a) et *B*(x_b, y_b).

Elle est incomplète !

1. Complétez la sur le cahier,
2. Programmez la,
3. Testez la sur un exemple.

```
def milieu(xa, ya, xb, yb):  
    xm = (xa + xb) / 2  
    ym = ...  
    return xm, ym
```

3. `vecteur.py` : Calculer les coordonnées d'un vecteur

La fonction `vecteur` calcule les coordonnées du vecteur allant de *A* à *B*.

Écrire la fonction `vecteur` et programmez la.

```
def vecteur(xa, ya, xb, yb):  
    ...  
    ...  
    return vx, vy
```