

2DE SNT

PYTHON

Exemple : `print` est une instruction qui permet d'afficher un message, ici « Bonjour tout le monde ».

```
1 print("Bonjour tout le monde")
```

Exemple : Différents types de variables

```
1 entier1=2
2 flottant=3.147
3 texte="salut"
4 entier2=int(2.13)
```

1 • Le programme

Communiquer avec un ordinateur demande un langage particulier : un langage de programmation (Scratch, Python, etc.). Il utilise un ensemble de mots et de règles appelés « instructions » et « syntaxe » et forme des programmes informatiques.

Un **programme** est un ensemble d'opérations destinées à être effectuées par l'ordinateur. Une **instruction** correspond à une action.

2 • Les variables

Pour écrire un programme, il faut enregistrer les données nécessaires au traitement dans des espaces mémoires. Les variables désignent des emplacements de stockage. Dans un programme, elles sont repérées par des noms et prennent des valeurs qui peuvent évoluer au cours du temps.

Une **variable** stocke une valeur dans un espace de la mémoire de l'ordinateur. Elle est désignée par un nom.

Des valeurs sont affectées aux variables pendant l'exécution du programme.

En Python, on affecte une valeur à une variable à l'aide du signe « = ».

Une variable peut contenir des valeurs de différents types comme :

- des nombres entiers (**int**) ;
- des nombres décimaux, appelés aussi « flottants » (**float**) ;
- des textes, appelés « chaînes de caractères » (**str**).

Exemple : Quelques opérateurs en Python

```
1 a=4
2 b=2
3 c="Bonjour"
4 somme=a+b
5 produit=a*b
6 cube=a**3
7 division=a/b
8 longueur=len(c)
```

Exemple : Instruction permettant la saisie d'un entier

```
1 nombre=int(input("nombre?"))
```

3 • Les instructions élémentaires

A Les opérations sur les variables

À chaque variable est associé un ensemble d'opérations qui dépendent de son type (opérations mathématiques sur les nombres, longueur des chaînes de caractères, etc.).

Un **opérateur** permet d'effectuer des opérations ou d'agir sur les variables.

Remarque : Les opérations sur les variables respectent les règles de priorité mathématiques.

B Les instructions d'entrée et de sortie

Une instruction d'entrée permet à un programme de lire des valeurs saisies au clavier par l'utilisateur. Une instruction de sortie affiche les valeurs de variables à l'écran.

En Python, l'instruction d'entrée « **input** » permet d'affecter la valeur saisie dans une variable. L'instruction de sortie « **print** » permet d'afficher à l'écran la valeur des variables.

L'instruction « **input** » permet d'obtenir des chaînes de caractères. Si on veut saisir un entier, il faut alors utiliser en plus l'instruction « **int** ».

Exercice N°1

Pour passer d'un pixel couleur codé en RVB (mélange des trois couleurs rouge, vert, bleu) à un pixel en une nuance de gris, on utilise la formule suivante :

$$G = 0,11 \times R + 0,83 \times V + 0,06 \times B$$

Écrire le programme qui permet de passer les valeurs RVB d'un pixel en niveau de gris.

MÉTHODE

Pour écrire un programme :

- 1** Déterminer les informations à saisir ou à calculer. Nommer les variables correspondantes et déterminer leur type.
- 2** Exprimer le ou les traitements à réaliser.
- 3** Déterminer la ou les variables à afficher.
- 4** Écrire le programme en respectant la syntaxe Python.

Regarder cette vidéo

lienmini.fr/3389-802

Syntaxe Python

- Une condition doit être suivie par deux points.
- Les instructions à exécuter si la condition est réalisée sont **indentées**, c'est-à-dire décalées vers la droite.

Exemple : Condition qui vérifie qu'une distance est inférieure à 40 m et qu'un accès wifi est autorisé.

```
1 distance<40 and acces=="autorisé"
```

4 • Les conditions

Un programme doit souvent prendre en compte différents cas et s'adapter à des situations. Il est alors nécessaire d'écrire des conditions.

Une **condition** est une expression logique dont le résultat est soit « vrai » soit « faux ».

Une condition est construite à l'aide d'opérateurs de comparaison :

- l'opérateur « égal » noté `==` ;
- l'opérateur « différent de » noté `!=` ou `<>` ;
- les opérateurs « inférieur à » ou « supérieur à » notés `<` et `>` ;
- les opérateurs « inférieur ou égal à » ou « supérieur ou égal à » notés `<=` et `>=`.

Lorsque la situation à tester est plus compliquée, il est possible de combiner plusieurs conditions grâce aux opérateurs logiques :

- « **and** » qui signifie « et » ;
- « **or** » qui signifie « ou » ;
- « **not** » qui signifie « non ».

Exemple : Programme qui affiche l'autorisation d'accès d'un membre à un groupe privé.

```
1 if membre=="oui":  
2     print("Accès autorisé")
```

Exemple : Programme qui affiche l'autorisation ou l'interdiction d'accès d'un membre à un groupe privé.

```
1 if membre=="oui":  
2     print("Accès autorisé")  
3 else:  
4     print("Accès refusé")
```

5 • Les instructions conditionnelles

A La structure conditionnelle « if »

Suivant la valeur d'une condition (vraie ou fausse), le programme choisit les actions à réaliser. On parle de **structures conditionnelles**.

La structure conditionnelle « if » permet d'exécuter un bloc d'instructions lorsqu'une condition est vérifiée.

B La structure conditionnelle « if ... else »

La structure conditionnelle « if ... else » permet d'exécuter un bloc d'instructions lorsqu'une condition est vérifiée et un autre bloc lorsqu'elle ne l'est pas.

C La structure conditionnelle « if ... elif ... else »

La structure conditionnelle « if ... elif ... else » permet de gérer plusieurs conditions. Si une condition n'est pas validée, la suivante est étudiée. En Python, « elif » est la contraction de *else if* qui signifie « sinon si ».

Exemple : Programme qui affiche les préférences de confidentialité d'un groupe.

```
1 if statut=="secret":  
2     print("seuls les membres voient le groupe et les publications")  
3 elif statut=="Fermé":  
4     print("Tout le monde voit le groupe mais pas les publications")  
5 else:  
6     print("Tout le monde voit le groupe et les publications")
```

Exercice N°2

La loi interdit à un jeune de moins de 13 ans de s'inscrire sur les réseaux sociaux. Entre 13 et 15 ans, l'inscription nécessite l'autorisation d'un responsable légal. Au-delà de 15 ans, l'inscription est autorisée sans condition.

Écrire un programme qui affiche les modalités d'inscription en fonction de l'âge de l'utilisateur.

MÉTHODE

Pour écrire une instruction conditionnelle :

- 1** Déterminer le nombre de conditions.
- 2** Écrire les conditions.
- 3** Choisir la structure conditionnelle à mettre en place.
- 4** Écrire le programme en respectant la syntaxe Python.

Regarder cette vidéo

lienmini.fr/3389-803