

05- Programmes (Programmation)



I – Créer un script Python

1.1 Enregistrement

Dans Thonny, il faut taper le code une ligne à la fois dans la partie ne comportant pas les trois chevrons.

Il faut également veiller à enregistrer le script avec un nom qui n'est PAS le nom d'un module de Python : ni turtle, ni random, ni tkinter...

1.2 Déroulement séquentiel

Comme on peut le voir, le programme se déroule séquentiellement : une ligne à la fois, en partant de la ligne du haut et en se dirigeant vers le bas.

L'ordinateur va donc évaluer et exécuter une ligne à la fois.

II – Mode pas à pas et débogage

2.1 Débogage avec le mode pas à pas

Le mode pas à pas de Thonny est également nommé mode débogage. Il permet de suivre l'évolution des variables en fonction des lignes exécutées séquentiellement. On doit valider une ligne pour que l'interpréteur Python l'exécute.

2.2 Débogage avec la fonction native print()

En mode interactif, une ligne ne comportant qu'une évaluation provoque l'affichage de sa valeur dans la console.

```
>>> a = 5
>>> a + 2
7
```

Ce n'est pas le cas en mode programmation : si vous lancez ce programme, il n'affichera rien nulle part.

```
a = 5
b = a + 2
b
```

Si vous voulez voir un résultat s'afficher dans la console en utilisant un programme, il faut le demander gentiment à l'interpréteur en utilisant la fonction native `print()`.

```
a = 5
b = a + 2
print(b)
```

Cette fois, vous pourrez voir 7 apparaître dans la console.

Attention : cette fonction provoque un affichage du résultat mais absolument pas sa mise en mémoire : le `print()` est destiné à un humain lisant la console et pas à l'ordinateur. Il s'agit d'une fonction dite d'interface (IHM : Interface Homme Machine).

2.3 Débogage avec la console

Dans ce cas, on exécute le programme et on demande en mode interactif le contenu des variables via la console.

III – Commentaires et mise en forme

Les commentaires sont destinés à un lecteur humain. L'interpréteur Python ne tentera pas de les exécuter.

Pour rajouter un commentaire, on utilise simplement le caractère dièse (#) de façon adaptée :

```
1 # Toute cette ligne est un commentaire.
2 print("Bonjour tout le monde") # Ceci est également un commentaire
3 print("Cette ligne ne contient pas de # commentaire")
```

```
>>> %Run progcommentaires.py
Bonjour le monde
Cette ligne ne contient pas de # commentaire
```

IV- Structure d'un programme

1. On commence toujours par l'IMPORTATION des modules nécessaires au programme.
2. On déclare ensuite les CONSTANTES : en Python, on s'engage juste à ne jamais changer la valeur après la première initialisation. D'où le nom de constantes. Par convention, le nom des CONSTANTES est constitué uniquement de majuscules.
3. On déclare ensuite les FONCTIONS : nous verrons cette partie dans une autre activité, mais vous avez dû les rencontrer en seconde.
4. On finit avec les instructions du programme en lui-même : on nomme cette partie "PROGRAMME PRINCIPAL" parfois.