Course apprenez à programmer en python

I.

## **Découvrez Python**

- Python est un langage de programmation interprété, à ne pas confondre avec un langage compilé.

- Il permet de créer toutes sortes de programmes, comme des jeux, des logiciels, des progiciels, etc.

- Il est possible d'associer des ****bibliothèques**** à Python afin d'étendre ses possibilités.

- Il est portable, c'est à dire qu'il peut fonctionner sous différents systèmes d'exploitation (Windows, Linux, Mac OS X,…).

## II. **Faites vos premiers pas avec l'interpréteur de commandes Python**

- L'interpréteur de commandes Python permet de tester du code au fur et à mesure qu'on l'écrit.

- L'interpréteur Python accepte des nombres et est capable d'effectuer des calculs.

- Un nombre décimal s'écrit avec un point et non une virgule.

- Les calculs impliquant des nombres décimaux donnent parfois des résultats approximatifs, c'est pourquoi on préfèrera, dans la mesure du possible, travailler avec des nombres entiers.

## III. **Entrez dans le monde merveilleux des variables**

- Les variables permettent de conserver dans le temps des données de votre programme.

- Vous pouvez vous servir de ces variables pour différentes choses : les afficher, faire des calculs avec, etc.

- Pour affecter une valeur à une variable, on utilise la syntaxe nom\_de\_variable = valeur.

- Il existe différents types de variables, en fonction de l'information que vous désirez conserver :int,float, chaîne de caractères etc.

- Pour afficher une donnée, comme la valeur d'une variable par exemple, on utilise la fonction print.

## IV. **Créez des structures conditionnelles**

- Les conditions permettent d'exécuter certaines instructions dans certains cas, d'autres instructions dans un autre cas.

- Les conditions sont marquées par les mot-clésif(« si »),elif(« sinon si ») et else(« sinon »).

- Les mot-clés if et elif doivent être suivis d'un test (appelé aussi prédicat).

- Les booléens sont des données soit vraies (True) soit fausses (False).

## V. **Apprenez à faire des boucles**

- Une boucle sert à répéter une portion de code en fonction d'un prédicat.

- On peut créer une boucle grâce au mot-clé while suivi d'un prédicat.

- On peut parcourir une séquence grâce à la syntaxe for element in sequence:

## VI. **Avancez pas à pas vers la modularité**

- Une fonction est une portion de code contenant des instructions, que l'on va pouvoir réutiliser facilement.

- Découper son programme en fonctions permet une meilleure organisation.

- Les fonctions peuvent recevoir des informations en entrée et renvoyer une information grâce au mot-clé return.

- Les fonctions se définissent de la façon suivante : def nom\_fonction(parametre1, parametre2, parametre N):

- On peut écrire les programmes Python dans des fichiers portant l'extension.py.

- On peut créer des fichiers contenant des ****modules**** pour séparer le code.

- On peut créer des répertoires contenant des ****packages**** pour hiérarchiser un programme.

## VII. **Gérez les exceptions**

- On peut intercepter les erreurs (ou exceptions) levées par notre code grâce aux blocs tryexcept.

- La syntaxe d'une assertion est assert test.

- Les assertions lèvent une exception AssertionError si le test échoue.

- On peut lever une exception grâce au mot-clé raise suivi du type de l'exception.