

Python (cours 1)

1) Bases de la syntaxe Python

Le python utilise une programmation et exécution séquentielle.

Il a pas de compilateur mais un interpréteur.

Tout code python est dans un fichier .py

2) Variable et tapage

Variable : « var = 15 » ou « a, b, c = 13, 3.4, 'toto' »

Afficher le contenu d'une variable « print(variable) »

3) Structures conditionnelles

IF :

if (a > 100) :

 consequence

elif (conditionB)

 consequence

else

 consequence

a et b —> (a and b) :

a ou b —> (a or b) :

non a —> (not(a)) :

4) Fonctions

Procédure : Nom, argument, instructions

Sous-programme : Nom, instructions, type de retour

Prototype : Nom, type de retour, arguments

Type de retour : Chaque fonction est une portion de code isolée dédiée à la réalisation d'une tâche. Cette tâche pouvant être la réalisation d'un calcul ou d'une opération, le type de retour permet de renvoyer la valeur calculée.

Argument : Les arguments ou paramètres de fonctions sont les valeurs demandées en entrée. Ce sont des variables locales à la fonction. Chaque variable doit avoir une valeur définie lors de l'appel de la fonction.

Prototype : La version primitive d'une fonction s'appelle son prototype. Il s'agit d'une pré-définition de la fonction. Un prototype renseigne sur les entrées et sorties de la fonction, sans en donner les instructions. C'est une méthode de représentation par boîte noire.

Programmation impérative : La programmation impérative est un modèle de programmation représentant un programme comme un ensemble d'opérations consécutivement exécutées par la machine. Le modèle consiste à morceler la structure d'un programme en tâches et sous-tâches décrivant le découpage fonctionnel d'un problème (Ne pas confondre programmation impérative avec programmation fonctionnelle)

5) Implémenter des fonctions en Python

Fonction : Une fonction est définie par le mot-clé **def** en Python.

```
def doSomething() :  
    instructions
```

Return : **return** permet de renvoyer la valeur d'une variable donnée en sortie d'une fonction. Cette valeur pourra être stockée dans une variable pour utilisation ultérieure.

```
def doStruff() :  
    a = 20  
    return a
```

Arguments en Python : Les arguments en Python sont définis à la volée dans le prototype de fonction. Il n'est pas nécessaire de les typer.

```
def doMult(a, b) :  
    return a*b
```

6) Récursivité

Une fonction récursive contient au moins un appel à elle-même.

```
def doAdd(a) :  
    if (a == 1) :  
        return a  
    else :  
        return 1+doAdd(a-1)
```