

Rappel Python



Données

- A saisir
 - Chaîne de caractères → **ch = input("message")** → type d'input toujours str
 - Nombre (ex : entier) → **n = int(input("n = "))**
 - Nécessite une conversion de type
 - **try** : bloc d'instructions non sûr **except** type d'exception : traitement en cas de problème
- A lire à partir d'un fichier
 - Ouverture de fichier → **f = open** (nom de fichier, mode lecture « r »)
 - Lecture des données → liste de lignes de texte ; **listeLignes = f.readlines()**
- A passer sous la forme de paramètres à une fonction → **f(n1, n2)**

Programme Python

1. Mettre les données dans des conteneurs (variables)
 - a. Choix des noms des variables → l'identificateur ne doit pas être un mot réservé (if, abs, break, ...)
 - b. Choix des types des variables
 - i. Données **simples** : une seule valeur → type simple : int, float, complex,
 - ii. Données **complexes** : plusieurs valeurs identifiées par une seule variable
 1. Données **non modifiables** tout au long d'exécution du programme
 - a. Valeurs : **caractères** → type : **str** ; ex : ch="abc" ; ch[0] → « a »
 - b. **Autres valeurs** → type : **tuple** ; ex : t = (1, 2.2) ; t[0] → 1
 2. Données **modifiables** durant l'exécution du programme
 - a. Type **indexé** : **list** ; ex : L=[1, 1, 2.2] → L[0]=5 → [5,1,2.2]
 - b. Type **Non indexé** :
 - i. **Avec répétitions** : **dict** {clé1 : val1, clé2 :val1, clé3 :val2 ...}
 - ii. **Sans répétitions** : **set** {val1, val2, ...}
2. Contrôler les données → if : elif : else : → ex: if 1 in L :
3. Faire des calculs ou autres traitements
 - a. Traitement simple exécuté une seule fois → suites d'instructions simples
 - b. Bloc d'instructions à exécuter plusieurs fois
 - i. Nombre d'itérations connu → boucles **for**
 - ii. Nombre d'itérations inconnu → boucle **while**
 - c. Bloc d'instructions à réécrire plusieurs fois
 - i. dans le même programme → définir une fonction : **def** et lui passer des paramètres pour bien fonctionner
 - ii. dans plusieurs programmes → créer une bibliothèque de fonctions/module Python .py → import

Résultats

- **Afficher** les résultat → **print**
- **Renvoyer** (retourner) le résultat d'une fonction vers le programme appelant → **return**
- Enregistrer le résultat dans un fichier → f = open en mode écriture « w »/Ajout« a » → f.write(resultat)