

Fiche récapitulatif - Partie 1

Les types natifs

Les types natifs sont la base de tout programme. C'est avec ces objets que tu vas représenter des données ou des conditions qui vont assurer le bon déroulement de ton script.

Les `chaînes de caractères` te permettront d'afficher du texte ou de définir des données (le prénom d'un utilisateur, le nom d'un pays, l'adresse d'un client...)

Les `nombres` (entier et décimaux) te permettront de faire des calculs et autres opérations mathématiques afin de valider des conditions ou gérer des données (calculer le nombre d'employés d'une entreprise, faire la somme des prix de différents items etc).

Finalement, les `booléens` te permettront d'ajouter de la logique dans ton script. Ce sont les 0 et les 1 qui sont à la base de l'informatique et qui permettent d'utiliser les structures conditionnelles et faire de l'algorithmie.

Le formatage des chaînes de caractères

Avec les chaînes de caractères, on peut afficher des messages à l'écran ou gérer des données de type texte. Et bien souvent, on a besoin que ce texte soit dynamique. On peut par exemple vouloir afficher le nom d'utilisateur de la personne qui utilise le script, insérer le nom d'un fichier, du salaire d'un employé, etc.

On réalise ces opérations grâce aux `f-string` et à la méthode `format` qui nous permettent d'insérer des variables dans une chaîne de caractères.

La manipulation des objets natifs

Un des aspects les plus importants d'un script est de gérer les entrées de l'utilisateur et les données qui parcourent le script. Parfois les données ne sont pas du bon type, ou on a besoin d'arrondir un nombre décimal au nombre entier supérieur ou convertir un objet d'un type à un autre pour ne pas avoir d'erreur.

En apprenant à vérifier le type d'un objet et à le changer, on peut ainsi manipuler les données comme bon nous semble pour arriver au résultat escompté.

Les opérateurs mathématiques

Un ordinateur, ça n'est rien de plus qu'une grosse calculatrice. On a très souvent besoin d'utiliser des nombres pour faire fonctionner notre script. Et avec les nombres viennent les opérations. Rien de bien compliqué dans cette partie, c'est vraiment comme la calculatrice que vous avez utilisé au collège.

Il y a bien sûr quelques exceptions et je t'invite à re-vérifier l'opérateur modulo (%) ou la division entière (//). Cependant, pas de panique, ce ne sont pas des opérateurs que l'on utilise tous les jours, si tu n'es pas sûr de comprendre à quoi peuvent bien servir ces opérateurs ou comment les utiliser, tu peux quand même continuer la formation et revenir sur ces notions quand l'occasion de les utiliser se présentera.

Interagir avec l'utilisateur

Il arrive que certains scripts n'aient besoin d'aucune interaction pour fonctionner. Mais bien souvent, on a besoin de demander des informations à l'utilisateur d'un script. Il est possible de créer des programmes avec interface graphique mais forcément c'est un peu plus complexe et vous n'êtes pas encore rendu là.

Pour l'instant, on a vu comment avec la fonction `input` on pouvait tout simplement demander des informations à un utilisateur pour les utiliser par la suite dans notre script.

Cela permet de créer des programmes en ligne de commande qui vont être interactif et ne produiront pas le même résultat en fonction de chaque utilisateur.

Les variables

À la base de n'importe quel langage de programmation, les variables permettent de stocker des informations dans la mémoire de notre ordinateur pour les ré-utiliser plus tard.

Cela permet notamment de ne pas répéter inutilement certains morceaux de code. Si jamais une chaîne de caractères change, on change son contenu lors de

la déclaration de la variable, et les changements seront répercutés à tous les endroits où cette variable est utilisée dans le script. Efficace !

Une variable, c'est un concept simple à première vue mais qui peut poser certains problèmes si on reste à la surface. N'hésite pas à revoir les notions par rapport à la place en mémoire des variables (avec la fonction `id`).

Il n'est pas primordial de parfaitement maîtriser sur le bout des doigts ces notions pour le moment.

Si c'est encore un peu flou, pas de panique. Python a l'avantage de ne pas trop nous embêter avec ça et on peut très facilement créer des scripts sans comprendre à 200% tout ce qu'il se passe en arrière-plan.

Mais n'oublie pas que ces concepts existent. Si jamais plus tard dans la formation tu vois des comportements bizarres, si tu modifies une variable et qu'une autre se retrouve affectée et que tu ne comprends pas pourquoi, ça peut être une bonne idée de revenir jeter un coup d'œil au cours sur les variables 😊

Les structures conditionnelles

Les structures conditionnelles sont à la base des algorithmes qui vont constituer vos programmes.

C'est grâce à elles que l'on va pouvoir modifier le comportement de notre script en fonction de différentes conditions.

Avec Python, on utilise les mots `if`, `elif` et `else` pour déterminer les différents cas de figure possibles.

Grâce à sa syntaxe très épurée, une structure conditionnelle avec Python se lit de façon presque transparente.

Couplées aux **opérateurs de comparaison** (`>`, `<`, `>=`, `<=` et `==`) et aux **opérateurs logiques** (`and`, `or`, `not`), on peut vérifier n'importe quel type de condition avec une structure conditionnelle.

Les listes

Parfois, on a besoin de stocker plusieurs éléments dans un seul objet. On peut stocker ces différents objets dans une liste.

Les listes sont très versatiles avec Python. On peut insérer n'importe quel type d'objet, imbriquer des listes les unes dans les autres et les modifier à guise. Pour bien maîtriser cet objet, n'hésitez pas à refaire les quiz et exercices de code pour bien maîtriser et retenir les différentes méthodes (`append` , `remove` , `extend` ...).

Les boucles

Les boucles sont à la base de l'automatisation de nombreuses tâches. Là où un humain peut passer des heures à renommer des centaines de fichiers, il est très simple avec Python d'écrire quelques lignes de code et de les répéter des centaines de fois sur des objets différents avec un simple boucle.

La boucle `for` et la boucle `while` sont les deux seules boucles disponibles avec Python. On utilise plus souvent la boucle `for` mais la boucle `while` a également son utilité.

N'hésitez pas à revoir les parties théoriques et à faire les différents exercices de code sur les boucles. Ce n'est pas forcément un concept évident au début mais une fois qu'on a compris son intérêt et les quelques règles à appliquer, ce n'est pas très compliqué !