

Vocabulaire langage de programmation

Les fondamentaux

Concept	Exemple	Description
Variable	<code>x = 5</code>	Espace de stockage en mémoire pour les données, modifiable au fil du temps.
Instruction	<code>a = b + c</code>	Action exécutée séquentiellement pour effectuer des tâches.
Boucle	<code>for i in range(10):</code>	Structure de contrôle pour répéter des instructions en fonction d'une condition.
Condition	<code>if x > 0:</code>	Structure de contrôle pour exécuter des instructions en fonction d'une condition.
Fonction	<code>def addition(a, b):</code>	Bloc de code réutilisable pour effectuer une tâche avec des paramètres.
Paramètre	<code>a et b dans addition(a, b)</code>	Variable de la signature d'une fonction qui va recevoir des valeurs lors de l'appel de cette même fonction.
Argument	<code>a et b dans addition(a, b)</code>	Variable extérieur à la fonction que l'on passe en paramètre à la fonction.
Signature de fonction	<code>def addition(a, b) -> float:</code>	L'ensemble des éléments d'une fonction avec son nom, ses paramètres avec leurs types et le type de retour de la fonction.

Les types et collections primitifs de python

Concept	Exemple	Description
int	<code>x = 5</code>	Les entiers représentent des nombres entiers sans décimales.
float	<code>y = 3.14</code>	Les nombres à virgule flottante représentent des nombres avec décimales.
str	<code>texte = "Bonjour"</code>	Les chaînes de caractères représentent des séquences de caractères, souvent utilisées pour du texte.
bool	<code>est_vrai = True</code>	Les valeurs booléennes représentent la vérité (True) ou la fausseté (False).
list	<code>liste = [1, 2, 3]</code>	Les listes sont des collections ordonnées et modifiables d'éléments.
tuple	<code>tup = (1, 2, 3)</code>	Les tuples sont des collections ordonnées immuables d'éléments.

Concept	Exemple	Description
dict	<code>dictionnaire = {'a': 1}</code>	Les dictionnaires associent des clés à des valeurs, formant des paires clé-valeur.
set	<code>ensemble = {1, 2, 3}</code>	Les ensembles sont des collections non ordonnées d'éléments uniques.
complex	<code>z = 3 + 4j</code>	Les nombres complexes sont utilisés pour représenter des nombres avec une partie imaginaire.
NoneType	<code>valeur = None</code>	Le type None est utilisé pour représenter l'absence de valeur ou une valeur nulle.