Initiation à la Programmation Python

Programme de la classe de Seconde 207 AS: 2026-2027

1 1. Introduction et Variables

Un **programme informatif** est une suite d'instructions exécutées par l'ordinateur. En Python, une **variable** permet de stocker une information en mémoire.

1.1 Types de données de base

Il existe plusieurs types de données principaux en Seconde :

- **Entier (`int`)** : nombres sans virgule (ex : `15`, `-3`).
- **Flottant (`float`)** : nombres à virgule (ex : `3.14`, `-0.5`).
- **Chaîne de caractères (`str`)** : texte entoure de guillemets (ex : `"Bonjour"`).
- **Booleen (`bool`)** : prend la valeur `True` (Vrai) ou `False` (Faux).

1.2 Affectation et Affichage

On utilise l'opérateur `=` pour affecter une valeur à une variable, et la fonction `print()` pour l'afficher.

Affectation de variables age = 15 taille = 1.68 prenom = "Alice"

Affichage des resultats print("Je m'appelle", prenom) print("J'ai", age, "ans.")

1.3 Entrée utilisateur

La fonction `input()` permet de demander une saisie à l'utilisateur. Par défaut, elle renvoie une chaîne de caractères (`str`). Il faut la convertir si l'on attend un nombre.

nom = input("Quel est votre nom ? ") age = int(input("Quel est votre age ? ")) Conversion en entier print("L'annee prochaine, vous aurez", age + 1, "ans.")

2 2. Opérations Mathématiques

Python permet d'effectuer tous les calculs usuels :

Opération	Symbole	Exemple	Résultat
Addition / Soustraction	`+` , `-`	`5 + 3`	`8`
Multiplication	`*`	`4 * 2.5`	`10.0`
Division réelle	`/`	`7 / 2`	`3.5`
Division entière	`//`	`7 // 2`	`3`
Reste (Modulo)	`%`	`7 % 2`	`1`
Puissance	`**`	`2 ** 3`	`8`

Table 1 – Opérateurs arithmétiques principaux

3 3. Instructions Conditionnelles (Si ... Sinon)

Les conditions permettent d'exécuter des blocs d'instructions différents selon qu'une expression est vraie ou fausse.

3.1 Opérateurs de comparaison

- `==` : Égal à
- `!=` : Différent de
- `<` , `>` : Strictement inférieur / supérieur
- `<=` , `>=` : Inférieur ou égal / Supérieur ou égal

3.2 Structure `if / elif / else`

Attention à l'indentation (décalage de 4 espaces) qui structure le code en Python.

note = float(input("Entrez votre note sur 20 : "))

if note >= 10 : print("Moyenne atteinte : Admis !") elif note >= 8 : print("Rattrapage.") else : print("Non admis.")

4 4. Les Boucles (Répétitions)

Les boucles servent à répéter un bloc d'instructions plusieurs fois.

4.1 La boucle bornée : `for`

On l'utilise quand on connaît à l'avance le nombre de répétitions. La fonction `range(n)` génère les entiers de `0` à `n - 1`.

Repeter 5 fois for i in range(5) : print("Passage numero", i)

4.2 La boucle non bornée : `while`

On l'utilise tant qu'une condition reste vraie.

compteur = 1 while compteur <= 3 : print("Compteur :", compteur) compteur = compteur + 1

5 5. Les Fonctions

Une **fonction** est un bloc de code réutilisable. Elle peut prendre des paramètres en entrée et retourner un résultat avec le mot-clé `return`.

def calculer$_aire_rectangle(largeur, longueur)$: $aire = largeur * longueurreturnaire$

Appel de la fonction resultat = calculer$_aire_rectangle(4, 7)print("L'aireestde : ", resultat)Affiche28$

6 6. Exercices d'Application

Exercice 1 : Calcul de moyenne
Écrire une fonction Python `moyenne(a, b)` qui prend deux nombres en paramètres et renvoie leur moyenne arithmétique.

Exercice 2 : Test de majorité
Écrire un programme qui demande l'âge de l'utilisateur et affiche s'il est majeur (≥ 18 ans) ou mineur.

Exercice 3 : Somme des premiers entiers
Écrire une fonction `somme(n)` qui calcule la somme des entiers de 1 à n à l'aide d'une boucle `for`.