Syntaxe

Fiche de synthèse

1 Variables et affectation

Définition 1 **(Variable)** Une **variable** est un nom qui désigne une donnée rangée en mémoire. On lui donne une valeur par une **affectation**, écrite avec le signe `=` : le nom à gauche, la valeur à droite.

```
age = 17
nom = "Alice"
taille = 1.65
```

L'affectation n'est pas une égalité mathématique : c'est une **instruction**, un ordre que Python exécute. Une **expression** est un calcul qui produit une valeur, comme `age + 1`. L'instruction `score = score + 10` calcule `score + 10`, puis donne ce résultat à `score`. Python exécute les lignes dans l'ordre ; la **trace d'exécution** note la valeur de chaque variable après chaque ligne, et une valeur déjà calculée ne se recalcule pas.

Choisir un nom de variable.

- Lettres, chiffres et tiret bas _ seulement, sans chiffre au début.
- Majuscules et minuscules comptent : `age` et `Age` sont deux variables différentes.
- Ni mot réservé (`if`, `for`, `def`…), ni nom de fonction comme `print`.
- Par convention, des mots en minuscules séparés par des tirets bas : `nombre_de_joueurs`.

2 Afficher : print

`print(...)` affiche une valeur : `print("Bonjour")`, `print(nom)`.

Pour mêler du texte et des valeurs, on écrit une **f-string** : la lettre `f` avant les guillemets, puis chaque variable ou calcul entre accolades.

```
prenom = "Alice"
age = 17
print(f"{prenom} a {age} ans")
print(f"Dans 10 ans : {age + 10} ans")
```

Ces lignes affichent `Alice a 17 ans`, puis `Dans 10 ans : 27 ans`.

3 Saisir : input

`input("question")` affiche la question et renvoie ce que tape l'utilisateur, **toujours** sous forme de chaîne de caractères. Pour calculer, on convertit avec `int` (entier) ou `float` (nombre à virgule).

```
nom = input("Quel est votre nom ? ")
age = int(input("Quel est votre âge ? "))
print(f"Bonjour {nom}, vous avez {age} ans")
```

Attention. Sans conversion, `input("Âge ? ") + 1` provoque l'erreur `TypeError` : Python n'additionne pas une chaîne et un nombre.

4 Types

Chaque donnée a un **type**, que donne la fonction `type` :

Valeur	Type	Nom
`42`	`int`	entier
`3.14`	`float`	flottant
`"Bonjour"`	`str`	chaîne de caractères
`True`	`bool`	booléen

C'est la donnée qui a un type, pas la variable : après `x = 42` puis `x = "quarante-deux"`, `x` désigne une chaîne. Attention, `42` est un entier, mais `"42"` est une chaîne.

5 Commentaires

Tout ce qui suit le caractère `#` sur une ligne est un **commentaire**, ignoré par Python : il explique le code à qui le lit.

6 Échanger deux variables

```
a = 5
b = 3
temp = a    # sauvegarde de a
a = b
b = temp
```

Écrire `a = b` puis `b = a` perdrait la valeur 5. Python permet aussi l'**affectation multiple** : `a, b = b, a`.

À retenir.

- `variable = valeur` affecte une valeur à une variable.
- `print(f"... {variable} ...")` affiche du texte et des valeurs.
- `input` renvoie toujours une chaîne : on la convertit avec `int` ou `float`.
- Les quatre types de base : `int`, `float`, `str`, `bool`.