

Aide-mémoire Python

1) Constructions de base

Commentaire	<code># Ceci est un commentaire</code>
Déclaration d'une variable	<code>nom = valeur</code>
Modification d'une variable	<code>nom = nouvelle_valeur</code>
Déclaration d'une fonction	<code>def f (x1, x2, ..., xN) :</code> <code> expr1</code> <code> ...</code> <code> exprN</code> <code> return result</code>
Séquence	<code>expr1</code> <code>expr2</code>
Test	<code>if (cond) :</code> <code> expr1</code> <code> ...</code> <code> exprN</code> <code>else:</code> <code> expr1</code> <code> ...</code> <code> exprM</code>
Boucle for	<code>for i in range(a_inclus, b_exclus) :</code> <code> expr1</code> <code> ...</code> <code> exprN</code>
Boucle for – Variante « départ de 0 »	<code>for i in range(b_exclus) :</code> <code> expr1</code> <code> ...</code> <code> exprN</code>
Boucle for – Variante « sélection du pas »	<code>for i in range(a_inclus, b_exclus, pas) :</code> <code> expr1</code> <code> ...</code> <code> exprN</code>
Boucle while	<code>while (cond) :</code> <code> expr1</code> <code> ...</code> <code> exprN</code>

2) Opérateurs arithmétiques

Addition	<code>+</code>
Soustraction	<code>-</code>
Multiplication	<code>*</code>
Division (quotient div. euc.)	<code>//</code>
Modulo (reste div. euc.)	<code>%</code>

3) Fonctions booléennes

Vrai	<code>True</code>
Faux	<code>False</code>
Et	<code>and</code>
Ou	<code>or</code>
Negation	<code>not</code>

4) Opérateurs de comparaison

Test d'égalité (valeur)	<code>==</code>
Test d'égalité (référence)	<code>is</code>
Test de non égalité (valeur)	<code>!=</code>
Supérieur (strictement)	<code>></code>
Inférieur (strictement)	<code><</code>
Supérieur ou égal	<code>>=</code>
Inférieur ou égal	<code><=</code>

5) Conversion

Conversion en entier	<code>int</code>
Conversion en nombre réel	<code>float</code>
Conversion en chaîne de caractères	<code>str</code>
Conversion en booléen	<code>bool</code>
Arrondi à l'inférieur	<code>math.floor</code>
Arrondi au supérieur	<code>math.ceil</code>
Arrondi au plus proche (bancaire)	<code>round</code>

6) Entrées et sorties

Afficher un texte (et passer à la ligne)	<code>print(texte_ou_valeur)</code>
- variante : plusieurs éléments	<code>print(elem1, ... , elemN)</code>
- variante : choix du dernier symbole	<code>print(texte_ou_valeur, end = "")</code>
- variante : passage à la ligne	<code>print()</code>
Lecture d'un texte	<code>input()</code>
- variante : stockage dans une variable	<code>nom = input()</code>
- variante : texte d'invite	<code>input(texte_d_invite)</code>

7) Tableaux et listes

Création d'un tableau	<code>nom = [expr_ini] * longueur</code>
Création d'un tableau (terme par terme)	<code>nom = [t0, t1, ... , tn]</code>
Accès à un élément	<code>nom[i]</code>
Modification d'un élément	<code>nom[i] = expr</code>
Longueur d'un tableau	<code>len(nom)</code>
Copie d'un tableau	<code>nom_copie = list(nom)</code>
Liste vide	<code>[]</code>

8) Erreurs et exceptions

Levée d'une exception quelconque	<code>raise e</code>
Gestion des exceptions	<pre>try: code # Fonctionnement "normal" except type_exception: code2 # Exception typée except type_exception, e: code3 # Exception typée et instancée except: code4 # Exception quelconque</pre>