

Aide-mémoire Python

1) Constructions de base

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

2) Opérateurs arithmétiques

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

3) Fonctions booléennes

| | |
|----------|-------|
| Vrai | True |
| Faux | False |
| Et | and |
| Ou | or |
| Negation | not |

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> |