

Fiche mémo PYTHON

<code>nom=input("Quel est ton nom ? ")</code>	saisie d'un texte stocké dans la variable nom	SAISIE CLAVIER
<code>nombre=float(input("décimal ?"))</code>	saisie de caractères transformés en décimal	
<code>entier=int(input("entier ?"))</code>	saisie de caractères transformés en entier	
<code>str(12)</code>	renvoie la chaîne 12	
<code>age = 16</code>	la valeur 16 est stockée dans la variable age	AFFICHAGE
<code>prenom = "Antoine"</code>	la valeur <i>Antoine</i> est stockée dans la variable prenom	
<code>print("Bonjour")</code>	affiche <i>Bonjour</i>	
<code>print(prenom)</code>	affiche la valeur de prenom	
<code>print(f"Tu as {age} ans")</code>	affiche <i>Tu as 16 ans</i> (si 16 est la valeur de age)	
<code>print("Tu as " + age + " ans")</code>	affiche <i>Tu as 16 ans</i> (si 16 est la valeur de age)	
<code>print("C'est une \nphrase")</code>	affiche <i>C'est une</i> <i>phrase</i> (\n pour un saut de ligne)	
<code>print("C'est une \tphrase")</code>	affiche <i>C'est une</i> <i>phrase</i> (\t pour une tabulation)	
<code>4 + 5</code>	calcule la valeur 9	OPERATIONS DE BASE
<code>7 - 2</code>	calcule la valeur 5	
<code>3 * 5</code>	calcule la valeur 15	
<code>15 / 2</code>	calcule la valeur 7.5	
<code>15 // 2</code>	calcule le quotient de 15 par 2, donc 7	
<code>15 % 2</code>	calcule le reste de division de 15 par 2, donc 1	
<code>2 ** 3</code>	calcule 2 puissance 3, donc 8	
<code>"Antoine" + "Robert"</code>	renvoie <i>AntoineRobert</i> (concaténation)	
<code>2 * "Antoine"</code>	renvoie <i>AntoineAntoine</i>	
<code>True and True</code>	évalue la solution à True	
<code>True and False</code>	évalue la solution à False	
<code>False and False</code>	évalue la solution à False	
<code>True or True</code>	évalue la solution à True	
<code>True or False</code>	évalue la solution à True	
<code>False or False</code>	évalue la solution à False	
<code>a == b</code>	teste si a vaut b	
<code>a != b</code>	teste si a est différent de b	
<code>i += 1</code>	i est incrémenté de 1	
<code>if n > 10: print("Bien") print("Continue") elif n < 5: print("Insuffisant") else: print("Attention") print("Suite")</code>	si n est supérieur à 10 alors on affiche <i>Bien</i> puis <i>Continue</i> sinon si n est inférieur à 5 alors on affiche <i>Insuffisant</i> sinon (dans tous les autres cas) alors on affiche <i>Attention</i> dans tous les cas le programme continue et affiche <i>Suite</i>	CONDITIONS
<code>texte = "Bonjour"</code>	affecte la valeur <i>Bonjour</i> à texte	CHAINES DE CARACTERES
<code>texte[3]</code>	renvoie le caractère d'indice 3 de texte (1 ^{er} caractère a l'indice 0)	
<code>len(texte)</code>	renvoie la longueur de texte	
<code>texte = ",".join(liste)</code>	crée une chaîne séparée par des virgules à partir de liste	
<code>"boB".lower()</code>	renvoie <i>bob</i>	
<code>"boB".upper()</code>	renvoie <i>BOB</i>	
<code>"boB".capitalize()</code>	renvoie <i>Bob</i> , seule la 1 ^{ère} lettre est majuscule	
<code>"Aurelie".find('e')</code>	renvoie l'indice du 1 ^{er} 'e', renvoie -1 s'il n'y a pas de 'e'	
<code>"Aurelie".count('e')</code>	renvoie le nombre de 'e', renvoie -1 s'il n'y a pas de 'e'	
<code>"Aurelie".replace('e', 'o')</code>	renvoie <i>Aurolie</i> , mais ne change pas le texte initial	
<code>chr(97)</code>	renvoie le caractère correspondant à 97 dans la table ASCII	
<code>ord("a")</code>	renvoie la valeur correspondante à a dans la table ASCII	
<code>from math import *</code>	importe la bibliothèque math	FONCTIONS
<code>sqrt(4)</code>	renvoie la racine de 4, donc 2	

<code>import math as m</code>	importe la bibliothèque math
<code>m.sqrt(4)</code>	renvoie la racine de 4, donc 2
<code>import math</code>	importe la bibliothèque math
<code>math.sqrt(4)</code>	renvoie la racine de 4, donc 2
<code>def f(x):</code> <code>return 2 * x + 1</code>	définit la fonction f
<code>print(f(3))</code>	exécute la fonction f pour x valant 3 et affiche le résultat
<code>for _ in range(3):</code> <code>print("Bonjour")</code>	affiche 3 fois <i>Bonjour</i> BOUCLES FOR
<code>for i in range(3, 9, 2):</code> <code>print(i)</code>	affiche 3, 5, 7 (saut de 2, le dernier indice 9 n'est pas affiché)
<code>for i in range(7, 2, -1):</code> <code>print(i)</code>	affiche 7, 6, 5, 4, 3 (dans l'ordre décroissant)
<code>n = 0</code> <code>while n < 3:</code> <code>print(n)</code> <code>n += 1</code> <code>print("Suite")</code>	tant que n est inférieur à 3 BOUCLES WHILE on affiche n (donc 0 1 2) on augmente n de 1 on affiche <i>Suite</i> quand la boucle while est terminée
<code>liste = []</code>	crée une liste vide appelée liste LISTES
<code>liste = "Hello world".split(" ")</code>	crée la liste <code>["Hello", "world"]</code> à partir de la chaîne "Hello world"
<code>liste = ["A", "B", "C", "D"]</code>	crée une liste composée des éléments A, B, C, D
<code>liste[1]</code>	renvoie l'élément d'indice 1 (le 1 ^{er} caractère a pour indice 0)
<code>liste[2:]</code>	renvoie les éléments de l'indice 2 à la fin
<code>liste[:3]</code>	renvoie les éléments du début jusqu'à l'indice 3 exclu
<code>liste[2: 4]</code>	renvoie les éléments de l'indice 2 à l'indice 4 exclu
<code>liste[2] = "X"</code>	modifie l'élément d'indice 2 qui devient X
<code>len(liste)</code>	renvoie le nombre d'éléments de la liste
<code>del(liste[1])</code>	supprime l'élément d'indice 1, qui est remplacé par le prochain élément de la liste, ce qui décale tous les éléments suivants
<code>liste.append("X")</code>	rajoute l'élément X
<code>liste.remove("X")</code>	supprime le 1 ^{er} X de liste ou crée une erreur si X n'est pas dans liste
<code>liste.index("X")</code>	renvoie l'indice du 1 ^{er} X de liste ou renvoie -1 si X n'est pas dans liste
<code>liste.sort()</code>	trie les éléments de liste par ordre croissant, ne renvoie rien
<code>liste.sort(reverse=True)</code>	trie les éléments de liste par ordre décroissant, ne renvoie rien
<code>liste.sort(key=maFonction)</code>	trie les éléments de liste en respectant l'ordre donné par maFonction (fonction à créer) et ne renvoie rien
<code>sorted(liste)</code>	renvoie liste triée par ordre croissant mais ne modifie pas liste
<code>liste.pop(1)</code>	renvoie l'élément d'indice 1 et le supprime de liste
<code>liste.pop()</code>	renvoie le dernier élément de liste et le supprime de liste
<code>liste.reverse()</code>	inverse les éléments de liste mais ne renvoie rien
<code>for elt in liste:</code> <code>print(elt)</code>	parcourt les éléments de liste pour les afficher
<code>for i in range(len(liste)):</code> <code>print(liste[i])</code>	parcourt les indices de liste pour en afficher les éléments
<code>"B" in liste</code>	teste si B se trouve dans liste et renvoie True ou False
<code>liste.count("X")</code>	renvoie le nombre de X dans liste , ou -1 s'il est absent
<code>mat = [[1,2,3],[4,5,6],[7,8,9]]</code>	crée une matrice de 9 éléments
<code>dico = {}</code>	crée un dictionnaire vide DICTIONNAIRES
<code>dico['nom'] = "Hervé"</code>	crée un système clé-valeur "nom - Hervé" ou modifie la valeur "Hervé" pour la clé "nom"
<code>for elt in dico.items():</code> <code>print(dico.keys() + " : " + dico.values())</code>	parcourt le dictionnaire dico pour afficher les clés et valeurs