Aide-mémoire Python

Variables	
x=1+2	déclare une variable x, initialisée avec 1+2
x = x + 1	incrémente la variable x
x += 1	une autre façon de l'écrire
text='chaise'	déclare une variable nommée text et lui affecte la chaîne de caractères 'chaise'
Entrée-Sorties	
print(2 * x)	affiche la valeur de 2 * x
print('2 * x')	affiche la chaîne 2 * x
s = input('nom :')	affiche 'nom' et attend une saisie au clavier, stockée dans la variable s au format chaîne de caractères
n = int(input('entrez n :'))	attend une saisie au clavier et l'interprète comme un entier, stocké dans la variable n
Entiers	
n // 3	le quotient de la division euclidienne de n par 3
n % 3	le reste de la division euclidienne de n par 3
n == 17	est-ce que n est égal à 17 ? (booléen)
n != 17	est-ce que n est différent de 17 ? (booléen)
Conditions	
if n > 100 :	si n est plus grand que 100
n = n − 10	alors lui retrancher 10
elif n >= 140 :	sinon si n ⩾ 140 :
n=n**2	l'élever au carré
else :	sinon
n = n + 11	lui ajouter 11
try : essaie d'exécuter le bloc 1	
bloc 1	et en cas d'erreur
except erreur :	traite cette erreur dans le bloc 2
bloc 2	
Boucles	
for n in range(0, 100) :	pour n allant de de 0 inclus à 100 exclus
print(n)	affiche les entiers 0; 1 ;2 ; ... 99
for n in range(7, 12, 3) :	pour n allant de de 7 inclus à 12 exclus, de 3 en 3
print(n)	affiche 7 et 10
n=2	n prend la valeur 2
while n < 10000:	tant que n est plus petit que 10000
n=n**3	n prend pour valeur son cube

Aléas	
from random import randint randint(0, 10)	 donne un entier tiré au hasard entre 0 et 10 inclus
Listes	
L = [0, 1, 1, 2, 3, 5]	déclare une liste t contenant 6 entiers
L = [0] * 100	déclare une liste L de taille 100, dont toutes les valeurs sont égales à 0
L=[i**2 for i in range (2,11,3)]	déclare une liste L=[4,25,64]
89 in L	est-ce que 89 est dans la liste L(booléen)
L[3]	la quatrième valeur de la liste t
L[2] = 42	modifie la troisième valeur de la liste
len(t)	le nombre d'éléments de la lite t
L2 = list(L1)	copie le contenu de la liste L1 dans une liste nommée L2
for x in t: print(x)	x parcourt la liste t affiche les éléments de t un à un dans l'ordre
Dictionnaires	
d = {}	déclare un dictionnaire d, vide
d['toto'] = 42	associe l'entier 42 à la clé 'toto'
'toto' in d	est-ce que 'toto' est une clé ? (booléen)
d['titi']	la valeur associée à la clé 'titi'
len(d)	le nombre de clé dans le dictionnaire d
for x, v in d.items(): print(x, '->', v)	parcourt les clés/valeurs de d affiche ici toutes les clés/valeurs
d2=dict(d1)	copie le contenu du dictionnaire d1 dans un dictionnaire nommé d2 (copie superficielle)
import copy	pour pourvoir réaliser une copie profonde avec le module copy
d2=copy.deepcopy(d1)	copie le contenu du dictionnaire d1 dans un dictionnaire nommé d2 (copie profonde)
Fonctions	
def nom_fonction(): bloc	La fonction exécute le bloc à l'appel de son nom suivi de ().
def nom_fonction(param1, ...) :	La fonction donne un nom au bloc et peut éventuellement faire appel à des paramètres.
bloc	les variables déclarées dans le bloc sont locales.
return expression	termine et renvoie la valeur de expression.