Les bases du langage Python 3

Opérations	opérations de base : `+ - * /`, puissance : `**` division euclidienne : reste `%`, quotient `//`
Variables et affectations	*types* : entier **`int`**, flottant **`float`**, booléen **`bool`**, texte **`str`**, liste **`list`** affectation simple : `ma_variable = -5` instructions sur la même ligne : `a = 1 ; b = 2` affectations parallèles : `a, b = -4, 1.2`

Entrées	`texte = input("message")` → la valeur renvoyée est toujours de type **`str`**
	`nombre = eval(input("message"))`
	`eval` évalue et convertit le texte saisi dans le type approprié
	→ permet de saisir nombres, expressions, listes, variables multiples
	`a, b = eval(input("saisir a, b = "))`

Sorties	`print("la longueur est", x, "cm")`
Commentaires	en ligne : `# ...` ; bloc : `"""..."""` ou `'''...'''`
Tests	tests simples : `==, !=, <, <=, >, >=` → renvoie un booléen (`True`, `False`) tests combinés : **`and`**, **`or`**, **`not`**, `a < b < c`

Structure alternative	`if condition:`
	`  instructions`
	`else:`
	`  instructions`
	– la *condition* doit être suivie par un double-points
	– c'est l'*indentation* qui indique la structure
	– **`else:`** est optionnel, il n'y a pas de end
	– on peut insérer des **`elif`** `condition:` (else if)

Boucle while	`while condition:`
	`  instructions`
	→ **tant que** condition est vraie **faire** instructions
	– double-points + indentation, pas de end

Boucle for	`for i in range(5):`
	`  instructions`
	→ **pour** `i` de 0 à 4 **faire** instructions (*range*=plage)
	– **`for i in range(1, 5)`** : pour `i` de 1 à 4
	– **`range(m, n, p)`** : `m` inclus → `n` exclu, `p` = pas

Définition de fonction	`def f(x, y): # nb quelconque de paramètres, y compris aucun` `  return x**2 + y**2 # valeur renvoyée par la fonction f`
Fonctions mathématiques	fonctions chargées automatiquement : **`max`**, **`min`**, **`abs`**, **`round`** autres fonctions (et constantes) mathématiques → dans le *module* `math` : `sqrt, sin, cos, tan, exp, log, floor, gcd, pi, e, ...` importation : **`from`** `math` **`import`** `sin, cos`
Nombres aléatoires	**`from`** `random` **`import`** `uniform, randint` `random()` → nombre (flottant) aléatoire dans [0 ; 1[, `uniform(a, b)` → [a, b] `randint(a, b)` → entier aléatoire compris entre `a` et `b` (inclus)
Listes	`L = [-1, 3.2, 1/3, 1e-3]` (les éléments peuvent être de type quelconque) élément (modifiable) : `L[1] = 5` (doit exister, l'indice commence à 0) fonctions de liste : **`len`** (longueur), **`max`**, **`min`**, **`sum`** supprimer l'élément d'indice `i` : **`del`** `L[i]` ajouter un élément, une liste : `L.append(-2.5)`, `L.extend([1, 0])` **`if`** `x` **`in`** `L:` → teste si `x` est dans `L` (fonctionne aussi pour les chaînes) **`for`** `x` **`in`** `L:` → `x` parcourt la liste `L` (fonctionne aussi pour les chaînes)
Chaînes de caractères	`texte = "bonjour"` ou `'bonjour'` ; **`len`**`(texte)` → longueur *concaténation* : `"mathé" + "mathiques"` donne `"mathématiques"` `texte[i]` → `i`+1-ème caractère (l'indice commence à 0, non-modifiable)
Graphiques	**`import`** `matplotlib.pyplot as plt` table des x : `Xlist = plt.np.linspace(a, b, n)` ou `Xlist = plt.np.arange(a, b, pas)` (ici `b` est exclu) liste des y : `Ylist = [f(x)` **`for`** `x` **`in`** `Xlist]` courbe : `plt.plot(Xlist, Ylist, options)` nuage de points : `plt.scatter(Xlist, Ylist, options)` diagramme à barre : `plt.bar(Xlist, Ylist, options)` afficher le graphique : `plt.show()`