

**NE PAS
OUBLIER!**

Les opérateurs

* Les opérateurs arithmétiques sur les nombres

En algorithme	+	-	*	/	div	mod
En Python	+	-	*	/	//	%

* Les opérateurs relationnels (de comparaison) sur les nombres

En algorithme	=	≠	<	≤	>	≥	∈
En Python	==	!=	<	<=	>	>=	in

Remarque : L'opérateur d'appartenance **∈** (**in** en Python) n'est utilisé qu'avec les ensembles et les intervalles des entiers et des caractères et non pas avec les chaînes et les tableaux (même si c'est possible en Python).

* Les opérateurs logiques

En algorithme	non	ou	et
En Python	not	or	and

Les fonctions arithmétiques standards sur les nombres

Nom algorithmique	En Python	Rôle	Exemples
Arrondi (x)	round (x)	Retourne l'entier le plus proche de x.	R ← arrondi(6.1) R vaut 6. R ← arrondi(6.6) R vaut 7. R ← arrondi(7.5) R vaut 8.
Racine carrée(x)	sqrt (x)	Retourne la racine carrée de x.	R ← racine carrée(9) R vaut 3. R ← racine carrée(16) R vaut 4.
Ent (x)	int (x)	Retourne la partie entière de x.	R ← Ent (2.2) R vaut 2. R ← Ent (2.8) R vaut 2. R ← Ent (-2.7) R vaut -3.
Abs (x)	abs (x)	Retourne la valeur absolue de x	R ← Abs (-2.5) R vaut 2.5 R ← Abs (6) R vaut 6.
Aléa (A, B)	randint (A,B)	Génère un nombre entier aléatoire compris entre A et B.	A ← aléa (0,3) A peut être 4. ou 2 ou 3 ou 0.

NB : la fonction **sqrt** nécessite l'importation de la bibliothèque **math** et la fonction **randint** nécessite l'importation de la bibliothèque **random**.

Les fonctions prédéfinies sur les caractères

Nom algorithmique	En Python	Rôle	Type du résultat	Exemples
ord (c)	ord (c)	Retourne le code ASCII du caractère c.	Entier	x ← ord ("A") x vaut 65 y ← ord ("a") y vaut 97.
chr (x)	chr (x)	Retourne le caractère correspondant au code ascii x.	Caractère	a ← chr (66) a vaut "B" b ← chr (100) b vaut "d"

Les fonctions prédéfinies sur les chaînes de caractères

Notation algorithmique	En Python	Rôle	Exemple
Long (ch)	len (ch)	Retourne le nombre de caractères d'une chaîne ch.	Long ("Algo") vaut 4
Pos (ch2, ch1)	ch1.find (ch2)	Retourne la première position d'une chaîne ch2 dans une chaîne ch1.	ch ← "secondes" Pos ("e", ch) vaut 1 Pos ("f", ch) vaut -1
Valeur (ch)	int (ch)	Convertit la chaîne ch en un entier si c'est possible.	valeur ("2.4") vaut 2.4 valeur ("20") vaut 20
	float (ch)	Convertit la chaîne ch en un réel si c'est possible.	valeur ("4Si") donne <i>Erreur</i>
Convch (x)	str (x)	Convertit le nombre x en une chaîne de caractères.	convch (100) vaut "100"
Estnum (ch)	ch.isdecimal()	Retourne Vrai si la chaîne ch est convertible en une valeur numérique. Sinon, elle retourne Faux	Estnum ("180") vaut Vrai Estnum ("2.4") vaut Vrai Estnum ("algo") vaut Faux
Majus (ch)	ch.upper ()	Convertit la chaîne ch en majuscules.	Majus ("bac") vaut "BAC" Majus ("2024") vaut 2024
Sous_chaine (ch, d, f)	ch [d : f]	Retourne une partie de la chaîne ch à partir de la position d jusqu'à la position f. (f exclue avec d ≤ f)	ch ← "strictement" ch1 ← Sous chaîne (ch,1,4) ch1 vaut "tri" ch2 ← Sous chaîne (ch,0,7) ch2 vaut "stricte"
Effacer (ch, d, f)	ch = ch [: d] + ch [f :]	Efface des caractères de la chaîne ch à partir de la position d jusqu'à la position f. (f exclue avec d ≤ f)	ch ← "métier" ch ← Effacer (ch,1,4) ch devient "mer"

Opérateur de concaténation

Notation algorithmique	En Python	Rôle	Exemple
+	+	Concaténation de deux ou plusieurs chaînes.	"4SI" + "2" vaut "4SI2".....