

Python ⇄ Algorithmique

Fiche de correspondance rapide — convention algorithmique du Baccalauréat

Aa Mot-clé

Aa Fonction /
built-in

Aa Chaîne

Aa Nombre

Aa Booléen

Aa Affectation

(1) VARIABLES & AFFECTATION

OPÉRATION	PYTHON	ALGORITHMIQUE
Affecter une valeur	<code>x = 10</code>	<code>x ← 10</code>
Changer la valeur	<code>x = "Sara"</code>	<code>x ← "Sara"</code>
Variable globale	<code>global x</code>	(déclarer x hors des paramètres)

(2) ENTRÉE / SORTIE

OPÉRATION	PYTHON	ALGORITHMIQUE
Lire une saisie	<code>x = input()</code>	<code>Lire(x)</code>
Lire avec message	<code>x = input("Msg")</code>	<code>Lire(x, "Msg")</code>
Afficher	<code>print(x)</code>	<code>Écrire(x)</code>
Afficher plusieurs valeurs	<code>print(a, b)</code>	<code>Écrire(a, b)</code>
Afficher avec saut de ligne	<code>print(x, "\n")</code>	<code>Écrire_nl(x)</code>

(3) TYPES DE DONNÉES & CONVERSION

OPÉRATION	PYTHON	ALGORITHMIQUE
Nombre entier	<code>int</code>	<code>Entier</code>
Nombre réel	<code>float</code>	<code>Réel</code>
Vrai / Faux	<code>bool</code>	<code>Booléen</code>
Texte	<code>str</code>	<code>Caractère / Chaîne</code>
Texte → Entier	<code>int(ch)</code>	<code>Valeur(ch)</code>
Texte → Réel	<code>float(ch)</code>	<code>Valeur(ch)</code>
Nombre → Texte	<code>str(x)</code>	<code>Convch(x)</code>
Tester si numérique	<code>ch.isdecimal()</code>	<code>Estnum(ch)</code>

(4) CONDITIONS

OPÉRATION	PYTHON	ALGORITHMIQUE
Si	<code>if cond:</code>	<code>Si Condition Alors ... FinSi</code>
Si / Sinon	<code>if cond:</code> <code>...</code> <code>else:</code> <code>...</code>	<code>Si...Alors...Sinon...FinSi</code>
Sinon si	<code>elif cond2:</code>	<code>Sinon Si Condition2 Alors</code>
Sélection multiple	<code>match x:</code> <code>case v: ...</code>	<code>Selon <Sélecteur> ... FinSelon</code>

(5) BOUCLES

OPÉRATION	PYTHON	ALGORITHMIQUE
Boucle bornée (for)	<code>for i in range(deb, fin+1, pas):</code>	Pour i de deb à fin [Pas=pas] Faire ... FinPour
Boucle conditionnelle (while)	<code>while cond:</code>	Tant que Condition Faire ... FinTantque
Répéter...Jusqu'à	<code>while True:</code> ... if cond: break	Répéter ... Jusqu'à Condition

(6) TABLEAUX (numpy array)

OPÉRATION	PYTHON	ALGORITHMIQUE
Importer	<code>from numpy import array</code>	(déclaration dans TDNT)
Déclarer tableau 1D	<code>T = array([0] * N)</code>	T = Tableau de N Entier
Déclarer tableau 2D	<code>T = array([[0]*m]*n)</code>	T = Tableau de n lignes * m colonnes
Accéder à l'élément i	<code>T[i]</code>	T[i]
Modifier l'élément i	<code>T[i] = val</code>	T[i] ← val
Taille du tableau	<code>len(T)</code>	(taille N donnée dans TDNT)
Afficher (jamais print(T))	<code>for i in range(len(T)):</code> print(T[i])	Pour i de 0 à N-1 Faire Écrire (T[i]) FinPour

(7) ENREGISTREMENTS

OPÉRATION	PYTHON	ALGORITHMIQUE
Déclarer le type	<code>P = dict(nom="", age=0)</code>	P = Enregistrement nom: Chaîne age: Entier Fin
Accéder à un champ	<code>P['nom']</code>	P.nom
Modifier un champ	<code>P['age'] = 18</code>	P.age ← 18

(8) FONCTIONS & PROCÉDURES

OPÉRATION	PYTHON	ALGORITHMIQUE
Définir une fonction	<code>def f(p1, p2):</code> ... return r	Fonction f(p1, p2): Type DEBUT ... Retourner r FIN
Définir une procédure	<code>def f(p1, p2):</code> ...	Procédure f(p1, p2) DEBUT ... FIN
Appeler une fonction	<code>r = f(a, b)</code>	r ← f(a, b)
Appeler une procédure	<code>f(a, b)</code>	f(a, b)
Passage par référence	<code>def f(T, n):</code> (tableau/dict = auto)	Procédure f(@T, n)

(9) OPÉRATEURS

OPÉRATION	PYTHON	ALGORITHMIQUE
Addition / Soustraction	+ -	+ -
Multiplication	*	*
Division réelle	/	/
Division entière	//	Div
Reste (modulo)	%	Mod
Puissance	**	^
Égal / Différent	== !=	= ≠
Comparaisons	> >= < <=	> ≥ < ≤
ET / OU / NON logiques	and or not	Et Ou Non
Appartenance	in	∈

(10) FONCTIONS PRÉDÉFINIES

OPÉRATION	PYTHON	ALGORITHMIQUE
Arrondir	round(x)	Arrondi(x)
Racine carrée	sqrt(x) # import math	RacineCarré(x)
Nombre aléatoire	randint(a,b) # import random	Aléa(a,b)
Valeur absolue	abs(x)	Abs(x)
Partie entière	int(x)	Ent(x)
Code ASCII ↔ caractère	ord(c) chr(d)	Ord(c) Chr(d)
Longueur d'une chaîne	len(ch)	Long(ch)
Position d'une sous-chaîne	ch.find(sc)	Pos(sc, ch)
Extraire une sous-chaîne	ch[d:f]	Sous_chaine(ch, d, f)
Supprimer de d à f	ch[:d] + ch[f:]	Effacer(ch, d, f)
Mettre en majuscules	ch.upper()	Majus(ch)

(11) FICHIERS TEXTE

OPÉRATION	PYTHON	ALGORITHMIQUE
Ouvrir	f = open(chemin, mode)	Ouvrir(chemin, f, mode)
Lire tout le contenu	ch = f.read()	Lire(f, ch)
Lire une ligne	ch = f.readline()	Lire_ligne(f, ch)
Écrire	f.write(ch)	Écrire(f, ch)
Écrire avec saut de ligne	f.write(ch + "\n")	Écrire_nl(f, ch)
Tester fin de fichier	ch == ""	Fin_fichier(f)
Fermer	f.close()	Fermer(f)

(12) FICHIERS DE DONNÉES (binaire)

OPÉRATION	PYTHON	ALGORITHMIQUE
Importer	<code>from pickle import load, dump</code>	–
Ouvrir (lecture/écriture/ajout)	<code>"rb" / "wb" / "ab"</code>	<code>"rb" / "wb" / "ab"</code>
Lire un enregistrement	<code>x = load(f)</code>	<code>Lire(f, x)</code>
Écrire un enregistrement	<code>dump(x, f)</code>	<code>Ecrire(f, x)</code>
Tester fin de fichier	<code>try: load(f) except: fin = True</code>	<code>Fin_fichier(f)</code>
Fermer	<code>f.close()</code>	<code>Fermer(f)</code>