Chapitre 22 – Cours

Raisonnement, ensembles et Python : outils transversaux

Dans ce chapitre, tu vas apprendre à :

- Raisonner par implication, contraposée et contre-exemple.
- Utiliser ensembles, couples et connecteurs logiques.
- Écrire des fonctions Python avec conditions et boucles puis tester leurs résultats.

Les mots pour comprendre

Définition – Une condition

Une condition est une expression qui peut être vraie ou fausse. Elle permet à un programme de choisir les instructions à exécuter.

Définition – Une boucle

Une boucle répète des instructions. La boucle `for` parcourt une collection ; la boucle `while` continue tant qu'un test est vrai.

1. Des propositions au raisonnement

Une proposition mathématique est une phrase dont on peut déterminer la vérité dans un cadre précisé. « Le carré de tout réel est positif ou nul » est vraie ; « tout entier impair est premier » est fausse, comme le montre 9.

Pour réfuter une affirmation générale, un contre-exemple suffit. Pour la prouver, plusieurs exemples ne suffisent pas : on établit une propriété pour un nombre quelconque du domaine.

L'implication « si A, alors B » est différente de sa réciproque « si B, alors A ». Sa contraposée est « si B est fausse, alors A est fausse » ; elle est logiquement équivalente à l'implication initiale.

Une équivalence demande les deux sens. Les expressions « pour tout » et « il existe » sont formulées en mots à ce niveau, sans imposer les symboles de quantification.

2. Connecteurs et ensembles

Le connecteur « et » exige deux conditions simultanées ; le « ou » mathématique est inclusif : au moins une des deux conditions est vraie, éventuellement les deux.

La négation de « $x > 3$ » est « $x \leq 3$ », pas « $x < 3$ ». La négation d'une condition « A et B » est « non A ou non B ».

Dans un ensemble E, le complémentaire de A est $E \setminus A$. L'intersection traduit « et », la réunion traduit « ou ».

Le cardinal compte les éléments d'un ensemble fini. Un couple $(a; b)$ est ordonné : $(a; b)$ et $(b; a)$ peuvent être différents.

Le produit cartésien $A \times B$ est l'ensemble des couples dont la première coordonnée appartient à A et la seconde à B.

3. Variables, types et fonctions Python

Une affectation avec `=` donne une nouvelle valeur à une variable ; un test d'égalité utilise `==`. Python distingue notamment entiers, flottants, booléens et chaînes de caractères.

Les calculs flottants sont approchés : un résultat affiché ne constitue pas nécessairement une valeur exacte. Une fonction définie par `def` reçoit des arguments et renvoie une valeur avec `return`.

Le domaine mathématique doit être respecté : une fonction qui divise par un argument ne peut pas accepter zéro sans traitement particulier. Le bloc est délimité par l'indentation ; elle fait partie du sens du programme.

4. Conditions et répétitions

Une condition utilise `if`, éventuellement `elif` et `else`.

Les connecteurs `and`, `or`, `not` traduisent les opérations logiques. Une boucle `for` parcourt un nombre connu d'étapes ; `range(n)` fournit les entiers de 0 à $n - 1$.

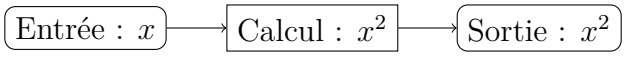
Une boucle `while` répète un bloc tant qu'une condition reste vraie : il faut expliquer pourquoi elle termine.

5. Contrôler un programme

Avant d'exécuter, on prévoit un résultat sur un exemple simple. On teste ensuite les bornes du domaine, les cas nuls autorisés et plusieurs situations.

Une trace des valeurs successives permet de localiser une erreur. L'algorithmique et la logique se réinvestissent dans les nombres, les fonctions, la géométrie et les probabilités ; cette fiche est une référence transversale, non un cours détaché des problèmes mathématiques.

Voir la notion



Le schéma distingue ce que l'on reçoit, ce que l'on calcule et ce que l'on renvoie.

Deux exemples, pas à pas

Exemple résolu – 1 – Contraposée

Question. Écrire la contraposée de la même implication.

Résolution. « Si un entier n'est pas pair, alors il n'est pas multiple de 4. » Elle est vraie comme l'implication initiale.

Exemple résolu – 2 – Égalité

Question. Expliquer la différence entre `x=5` et `x==5`.

Résolution. La première instruction affecte 5 à x. La seconde teste si la valeur actuelle de x est égale à 5 et renvoie un booléen.