

Chapitre 22 – Corrections

Raisonnement, ensembles et Python : outils transversaux

20 exercices progressifs – Solutions détaillées

Exercice 1 – Contre-exemple

Non : $x = -2$ vérifie $x^2 = 4$ mais pas $x = 2$.

Exercice 2 – Négation

Les négations sont respectivement $x \geq 5$ et $x < -2$.

Exercice 3 – Et et ou

La première condition donne $[1; 4[$; la seconde $] - \infty; 1[\cup [4; +\infty[$, complémentaire de la première dans $\mathbb{R}$.

Exercice 4 – Réciproque

La réciproque affirme que tout entier pair est multiple de 4. Elle est fausse : 2 est un contre-exemple.

Exercice 5 – Contraposée

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

Exercice 6 – Cardinal

$\text{Card}(A) = 4$ et $\text{Card}(B) = 2$.

Exercice 7 – Complémentaire

$E \setminus A = \{1; 3; 5\}$.

Exercice 8 – Produit cartésien

Les quatre couples sont $(0; 2), (0; 3), (1; 2), (1; 3)$.

Exercice 9 – Affectations

Après la deuxième instruction, $x = 7$. La troisième donne $y = 14$.

Exercice 10 – Égalité

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

Exercice 11 – Fonction simple

On peut écrire `def f(x):` puis, à la ligne indentée, `return 3*x-2`. Par exemple `f(4)` renvoie 10.

Exercice 12 – Test logique

La première condition est `True` car les deux comparaisons sont vraies. La seconde est `False` car aucune ne l'est.

Exercice 13 – Boucle bornée

Les valeurs sont 0, 1, 2, 3. La borne 4 est exclue.

Exercice 14 – Somme répétée

La trace donne 3, 6, 9, 12 ; la valeur finale est 12.

Exercice 15 – Condition de terminaison

Les valeurs successives sont 1, 2, 4, 8, 16, 32. La condition devient fausse à 32 ; cinq tours sont exécutés.

Exercice 16 – Approfondissement : somme des entiers

Initialiser `s=0`, puis utiliser `for k in range(1,n+1):` avec le bloc indenté `s=s+k`. La somme finale est `s` ; pour $n = 0$, la boucle vide laisse $s = 0$.

Exercice 17 – Approfondissement : valeur absolue

Définir `def valeur(x):`, puis tester `if x>=0: return x`, sinon `return -x`, avec les indentations correspondantes. Les deux branches réalisent la définition par morceaux.

Exercice 18 – Approfondissement : filtre

La condition est `(-2 <= x and x <= 3) or x == 10`. Elle accepte les deux bornes et la valeur isolée 10.

Exercice 19 – Approfondissement : recherche de seuil

Les valeurs sont 110, 121, 133, 146, 161, 051. Le premier dépassement arrive après 5 tours : $n = 5$.

Exercice 20 – Approfondissement : erreur de boucle

Les valeurs deviennent négatives et restent inférieures à 5. Remplacer le bloc par `x=x+1` permet d'atteindre 5 après cinq tours.