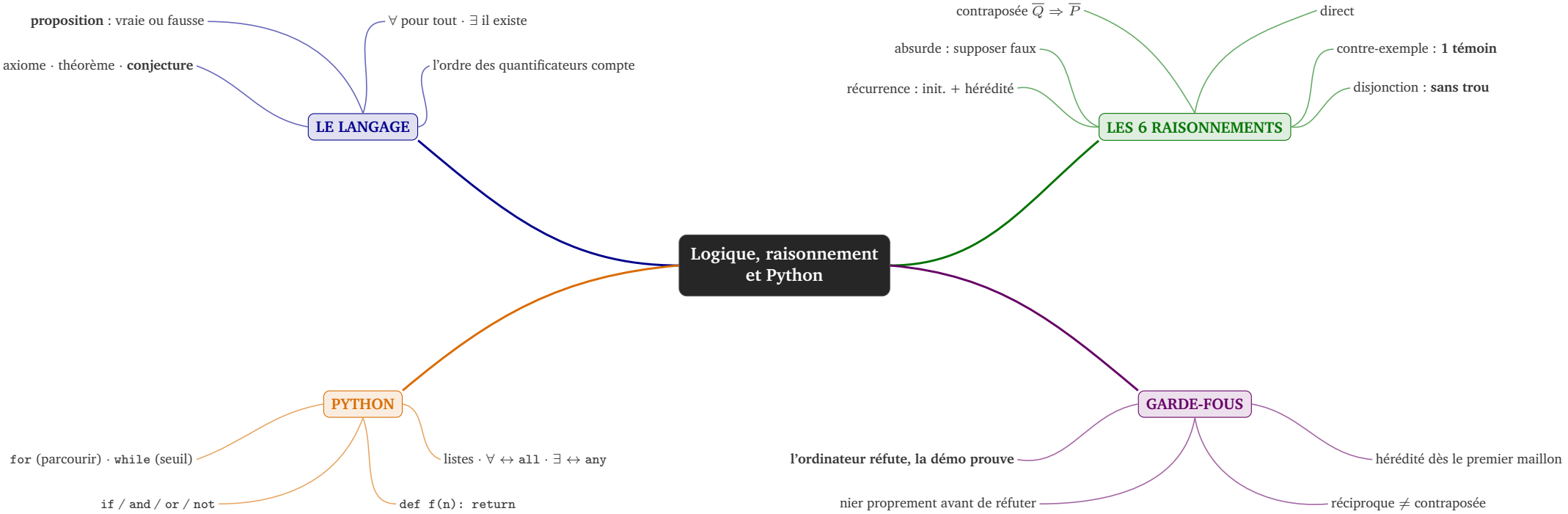
Logique, raisonnement et Python
Chapitre A0



Nier un énoncé ($\forall$ et $\exists$ s'échangent, la propriété se nie) :

$$\overline{\forall x,\ P(x)} = \exists x,\ \overline{P(x)} \qquad \overline{\exists x,\ P(x)} = \forall x,\ \overline{P(x)}$$

$$\overline{P \text{ et } Q} = \overline{P} \text{ ou } \overline{Q} \qquad \overline{P \text{ ou } Q} = \overline{P} \text{ et } \overline{Q}$$

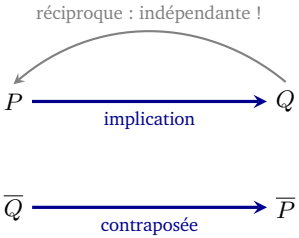
Le dictionnaire logique-ensembles :

P et Q	P ou Q	non P	$P \Rightarrow Q$
$A \cap B$	$A \cup B$	$\overline{A}$	$A \subset B$

Quel raisonnement, quand, comment ?

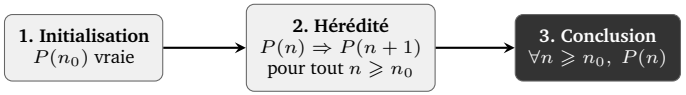
Raisonnement	Quand ?	Comment ?
Direct	le cas général	chaîne d'implications
Contre-exemple	réfuter un $\forall$	**un seul** témoin
Disjonction	la propriété change de visage	cas **sans trou**
Contraposée	hypothèse pauvre	prouver $\overline{Q} \Rightarrow \overline{P}$
Absurde	« impossible », « n'existe pas »	supposer faux, contredire
Récurrence	énoncé dépendant de n	voir ci-contre

L'implication et ses deux sœurs



$$(P \Rightarrow Q) \Leftrightarrow (\overline{Q} \Rightarrow \overline{P})$$

La récurrence en trois temps



Rédaction : « Soit $n \geqslant n_0$ tel que $P(n)$ soit vraie ; montrons $P(n+1)$. »

Python minimal

Besoin	Python
Tester	`if/elif/else`, `and`, `or`, `not`
Parcourir	`for k in range(a, b)`
Attendre un seuil	`while condition:`
Encapsuler	`def f(n): return ...`
Rassembler	`[n**2 for n in range(100)]`
Quantifier	$\forall \leftrightarrow$ `all`, $\exists \leftrightarrow$ `any`

Les trois pièges qui coûtent des points. Confondre une implication et sa **réciproque** · oublier que l'hérédité doit valoir **dès le premier maillon** (les chevaux unicolores) · croire qu'un million de vérifications démontre un $\forall$: **l'ordinateur réfute, la démonstration prouve**.

Le détail, les démonstrations et les histoires : [COURS-A0] et le compagnon numérique du chapitre.