

# Mathématiques pour informaticien

Séance 1 — Mise en matière

# Pourquoi des maths pour l'info ?

Écrire des conditions justes (si/alors) =  
logique

Données sans doublons, clés, relations =  
ensembles et fonctions

Raisonner sur des algorithmes =  
invariants, récurrence (light)

Évaluer des cas/exceptions =  
combinatoire & probabilités

Comprendre les limites (temps/mémoire)  
= complexité (aperçu)

# Objectifs d'apprentissage (semestre)

**Lire et interpréter des propositions logiques -**  
priorité des connecteurs, tables de vérité

**Utiliser des équivalences clés -**  
implication $\leftrightarrow$ disjonction, De Morgan,  
contraposée (simplifier/justifier)

**Modéliser simplement un problème -**  
ensembles, relations, fonctions ; passer du  
langage naturel à une forme claire

**Appliquer l'arithmétique des entiers -** division  
euclidienne, PGCD, congruences (parité,  
périodicité, hachage simple)

**Dénombrer des cas simples -** produit,  
permutations/arrangements/combinatoires ;  
estimer la taille d'un espace

**Calculer des probabilités élémentaires -**  
espaces finis, indépendance ;  
Bernoulli/Binomiale ; interpréter l'espérance

**Expliquer une démarche courte de preuve -**  
exemples/contre-exemples, contraposée,  
récurrence "douce", invariants

**Relier aux objets d'informatique -** vocabulaire  
de base des graphes ; lire  $O/\Omega/\Theta$  pour  
comparer des approches

# Compétences pratiques visées

**Raisonnement logique** — connecteurs, tables de vérité, implication/équivalence, contraposée

**Modélisation discrète** — ensembles, relations, fonctions pour décrire des données/processus

**Arithmétique des entiers** — divisibilité, PGCD, congruences modulo (applications : parité, hachage simple)

**Combinatoire élémentaire** — principe multiplicatif, permutations/arrangements/combinations ; estimation de cas

**Probabilités discrètes** — événements, indépendance, Bernoulli/Binomiale, espérance (lecture de risques)

**Induction & invariants (initiation)** — justifier brièvement une boucle/suite simple

**Graphes — notions** — vocabulaire, exemples (chemins, arbres) en lien avec l'algorithmique

**Complexité — intuition** — lire  $O/\Omega/\Theta$  pour comparer des croissances simples

# Panorama du semestre (12 semaines)

**S1–S2** — Logique & équivalences (connecteurs, tables, De Morgan, contraposée)

**S3** — Ensembles (opérations, diagrammes, dédoublonnage)

**S4** — Relations & Fonctions (numériques/usuelles, application à la modélisation)

**S5** — Entiers & modulo (division euclidienne, PGCD, congruences ; parité, hachage simple)

**S6–S7** — Combinatoire I–II (produit, arrangements/combinaisons, principe des tiroirs)

**S8** — Probabilités discrètes I (événements, indépendance)

**S9** — Statistiques descriptives (moyenne, médiane, variance, diagrammes)

**S10** — Probabilités discrètes II (Bernoulli/Binomiale, **espérance**)

**S11** — Graphes (notions) & Optimisation — initiation (sac à dos simple, allocation)

**S12** — Optimisation (suite) & Complexité — aperçu ( $O/\Omega/\Theta$ ) + révision intégrée

Ce que CE  
cours n'est  
pas

Pas de démonstrations très  
techniques

Pas besoin d'être fort·e en  
calcul formel

La programmation servira  
surtout en démonstration  
(enseignant)

# Liens avec les autres UE (S1–S2)

## Intro algo

- conditions correctes, dénominvements simples

## Python

- booléens, `if/while`, `set` pour dédoublonner

## Analyse de données

- comptages, fréquences, probas simples, statistiques descriptives

## Web

- validation de formulaire (logique)

## Outils & supports

Notebook S1 (logique, tables, équivalences)

Formulaire d'introduction (PDF/HTML) pour mieux vous connaître

Bibliographie (DOCX/PDF/HTML)

Jupyter/Colab

# Évaluations (simples et régulières)

Participation dans le cours: (10%)

Évaluation de mi-session: (20%)

Épreuve finale: 50%

Projet “Python discret” en binôme  
 $(2 \times 10\%) = 20\%$

Descriptifs précis fournis avant  
chaque évaluation

## Rituels de cours

Courtes activités en binôme

Quiz rapides (feedback immédiat)

Ticket de sortie en fin de séance

Règles : respect, entraide, droit à l'erreur, questions bienvenues

## Conseils pour réussir

Relire les équivalences clés (2–3) entre chaque séance

Refaire les mini-exercices avec vos propres exemples

Comparer vos réponses en binôme et poser des questions

Participer régulièrement dans le cours

# Bibliographie essentielle (accessible)

## Rosen

- chapitres ciblés : logique, ensembles, combinatoire

## [Mathematics for Computer Science](#) (MIT, gratuit)

- preuves/inductions/graphs/probability  
- extraits

## Jaume et al. (Ellipses)

- logique pour l'info - très adapté S1–S2



**ON APPREND EN FAISANT — PAR  
PETITES ÉTAPES ET SANS PEUR DE SE  
TROMPER.**



**BIENVENUE !**