

# Réussir son S1 de 3MIC Starter Pack™

Les bases à avoir avant d'arriver en cours:

•

## Les liens ressources

- La page **Cyber-Aide** : [lien](#)
- [Drive GAMMA](#)
- Le **Club Info** organise des **tutorats** notamment en Git (utile pour le S2 et le projet du S1)
- Le site annales contient encore quelques ressources intéressantes : [lien](#).
- Un google drive assez complet (merci Théo) : [lien](#).
- D'autres ressources étudiantes sont disponibles :
  - Le site de Louis : [lien](#).

## Conseils plus pratiques

Général:

**Tronc commun :**

**UF - Analyse Numérique, Optimisation, Chaînes de Markov :**

Analyse Numérique et Optimisation :

Pendant les TPs, utilisation de Python en Jupyter Notebook. Il est donc conseillé de revoir ses cours pour ne pas perdre de temps (notamment les affichages la matplotlib). Aussi, les TPs sont indispensables pour bien connaître la seconde partie du cours, donc ne les prenez pas à la légère, voir les refaire.

Chaînes de Markov :

## **UF - Système d'exploitation, Réseaux, Base de données :**

Système d'exploitation :

Revoir l'architecture matérielle aidera beaucoup : une partie du cours est en commun.

*Disclaimer : Si ça n'a pas été corrigé, les contrôles ne se reposent pas sur le cours, mais sont des copier-coller des contrôles des années précédentes (donc des cours en partie plus vus). Donc faites les annales, sinon vous n'aurez pas une partie des points :)*

Réseaux :

Site du MIT avec des vidéos très bien expliquées et des schémas/questions de cours et d'application :

[https://gaia.cs.umass.edu/kurose\\_ross/lectures.php](https://gaia.cs.umass.edu/kurose_ross/lectures.php)

Base de données 2 :

Bien connaître sa BDD 1 n'est pas obligatoire, mais conseillé.

La note ne se fait que par projet, mais attention ! Contrairement à la BDD1, corriger un bug ou un manque peut prendre extrêmement longtemps. Il est conseillé de passer un maximum de temps sur la **partie théorique** de conception des bases avant d'implémenter.

Pour les fans d'IA, utiliser ChatGPT peut être pratique, notamment pour implémenter parfois des dizaines de lignes avec des valeurs aléatoires pour des tests. Néanmoins, n'en abusez pas, la BDD2 sera utile pour les prochains semestres, donc plus vous pratiquez, mieux vous maîtriserez. :)

Enfin, vous pouvez utiliser **git** (sur **GitEud**) ou **Visual Studio Code** avec l'option **Live Share** pour pouvoir mettre vos tables en commun.

## **Option Mathématiques Appliquées :**

**UF - EDO et leur résolution numérique :**

**UF - Calcul intégral et probabilités :**

## Option Informatique et Réseaux :

### UF - Base de données et programmation web :

Base de données 1 :

Programmation Web :

Vous y verrez l'HTML, le CSS et le Javascript. Pour ce dernier, revoir vos cours de C peut aider.

Nous vous recommandons d'utiliser WebStorm de la suite [JetBrains](#)

Pour la mise en commun : Utiliser **GitEtud** peut être très pratique pour le projet, donc n'hésitez pas à apprendre les bases (Créer un projet, commit, push, merge). Sinon, **Visual Studio Code** avec l'option **Live Share** peut fonctionner pour les moins vaillants.

Pour voir votre projet : Tous les étudiants de l'INSA ont accès à un **hébergement gratuit** sur le **serveur etud**. Si vous avez besoin d'aide, vous pouvez poser toutes vos questions sur le **serveur du Club Info**. Utile pour le rendu du rapport. (nécessitant un lien pour accéder à vos pages). Il y a aussi un preview sur git.

Sinon, pour pouvoir avoir le rendu en direct, vous pouvez **ouvrir le fichier main.html dans votre navigateur** ou utiliser l'extension **Live Server** sur **Visual Studio Code**.

### UF - Outils théoriques pour l'informatique :

Algorithmique Avancée :

Vous étudierez la complexité des algorithmes, et certains théorèmes pour mieux comprendre la complexité des algorithmes. Mais ceci n'est pas un cours de programmation, vous ne toucherez donc pas aux ordinateurs pendant ce cours. Il n'y a pas de pré-requis particuliers.

Expressions Régulières :

L'examen de cette matière est facultatif, et vous pouvez valider cette matière si vous venez au TP (qui n'est pas noté). Cependant, essayez quand même de travailler cette matière, car elle vous sera forcément très utile par la suite dans vos scripts ou pour chercher quelque chose dans de longs fichiers de code. (un peu l'équivalent d'un contrôle F, mais en beaucoup plus poussé). Il n'a pas non plus de prérequis à avoir.

## Programmation linéaire :

Il n'y a pas de prérequis à avoir. C'est une matière pas très difficile qui permet d'avoir une bonne note si on travaille bien. Pour cela, je recommande de suivre attentivement les TD, car cette matière peut permettre d'avoir une bonne note pour rattraper Théorie de l'information et codages, qui est une matière très incertaine où on peut avoir de très mauvaises notes, même en ayant compris.

## Théorie de l'information et codages :

---

Révision #16

Créé 5 juillet 2021 12:43:18 par Raphael

Mis à jour 29 juin 2023 10:35:00 par Raphael