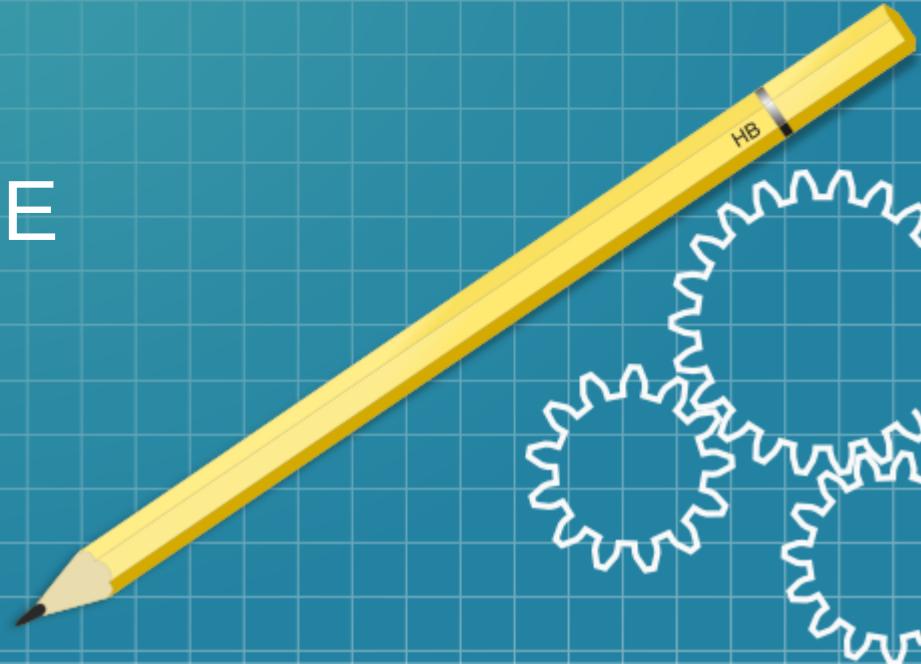


Algorithmique & Programmation Objet APO

L3 MIAGE



Présentation



- Équipe pédagogique
- Quelques Mots sur les LLM
- Algorithmique et Programmation Objet
- Contrat pédagogique (pédagogie inversée)
- Support pédagogique : Caseine
- Contenu et organisation des cours
- Évaluation initiale et création des groupes

Équipe Pédagogique

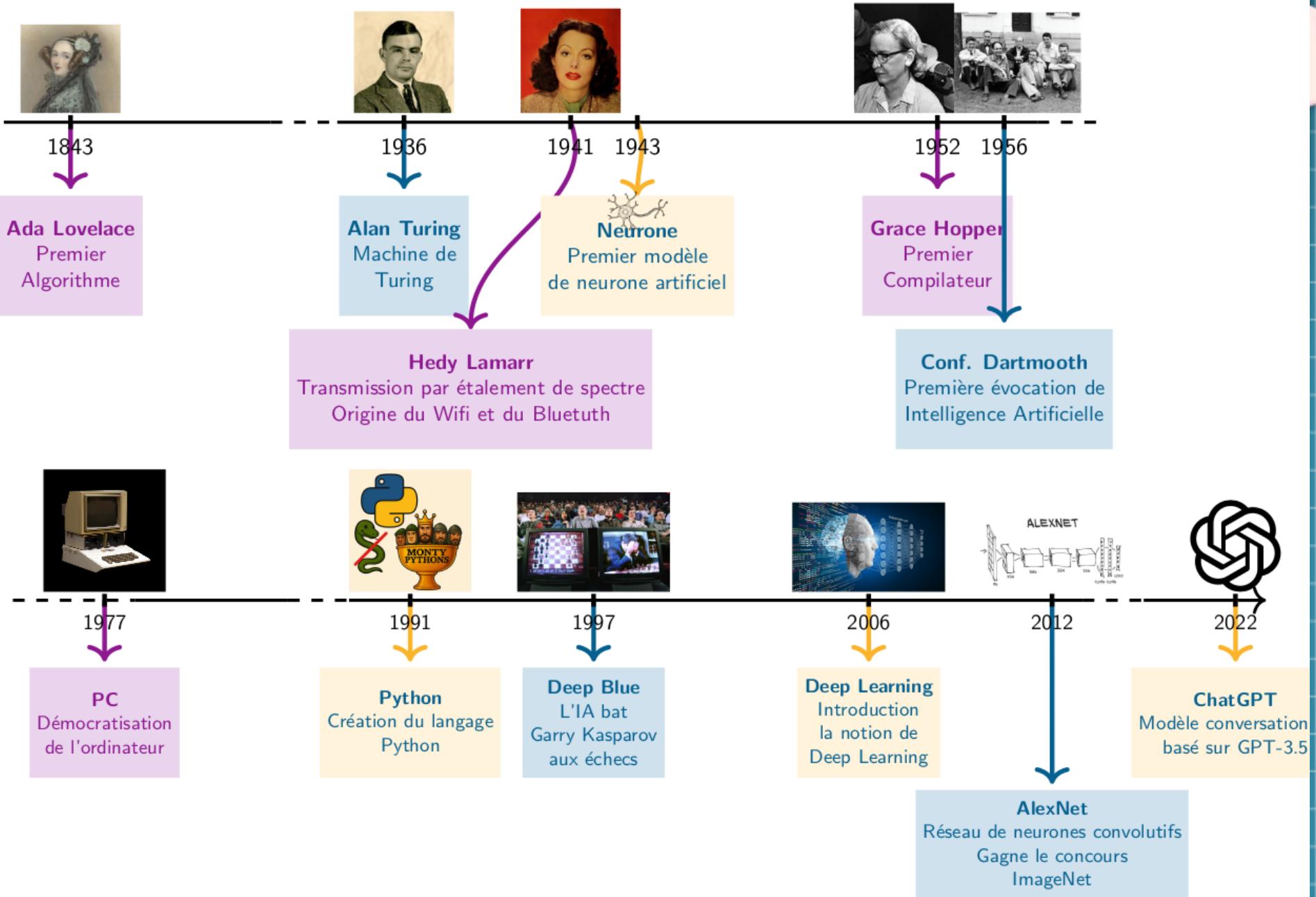
- Céline Fouard (1 groupe CTD)
 - Resp. Algorithmique et Programmation Objet (APO)
 - Recherche : GMCAO (thésaurisation/sécurisation de code)
- Alexandre Demeure (1 groupe CTD + 1 TD)
 - Resp. d'année / Interface Homme Machine (IHM)
 - Recherche : IHM
- Gilles Sérasset (1 groupe CTD + 1 TD + 1 TP)
 - Recherche : Traitement automatique des langages
- Philippe Genoud
 - Recherche : Graphes de connaissances
- Kadir Uraz Alcali
 - Ingénieur logiciel chez Moody's

Présentation



- Équipe pédagogique
- Quelques remarques sur les LLM
- Algorithmique et Programmation Objet
- Contrat pédagogique (pédagogie inversée)
- Support pédagogique : Caseine
- Contenu et organisation des cours
- Évaluation initiale et création des groupes
-

L'IA, ça date pas d'hier...



Intelligence Artificial

Programmation Traditionnelle

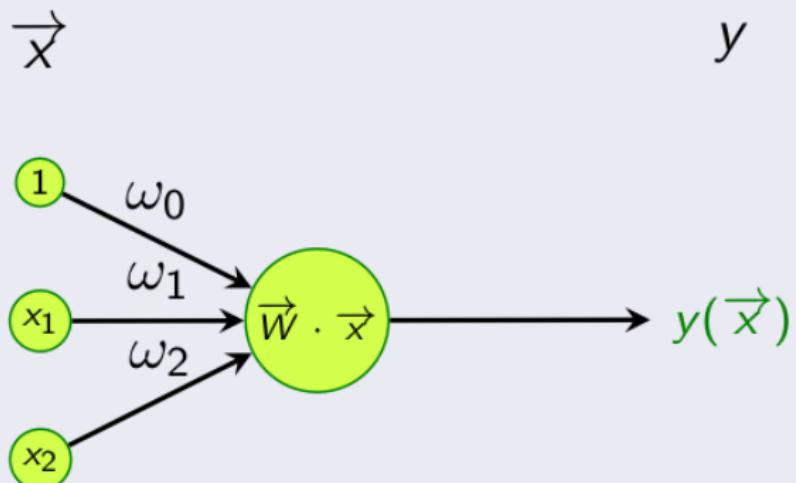
Apprentissage Machine

Réseaux de Neurones
Deep Learning

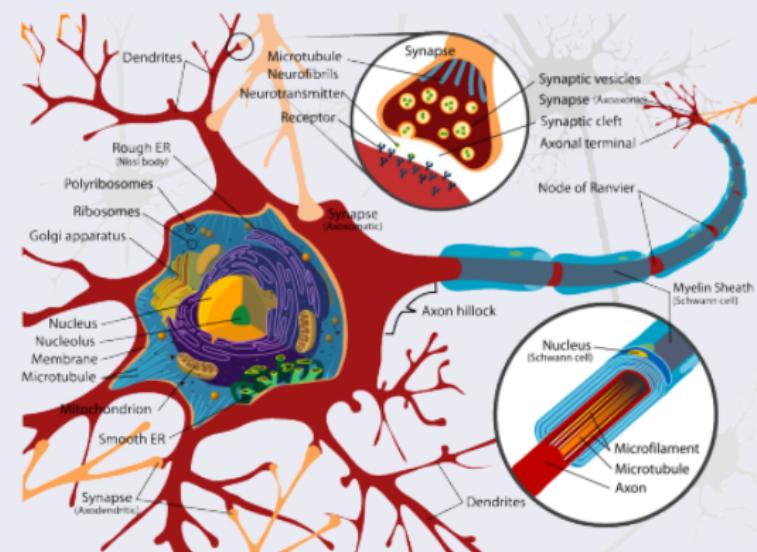
IA Générative

LLM

Perceptron (*Neurone machine*)



Neurone Humain



Clker-Free-Vector-Images from Pixabay

Présentation

- Équipe pédagogique
- Quelques remarques sur les LLM
- **Algorithmique et Programmation Objet**
- Contrat pédagogique (pédagogie inversée)
- Support pédagogique : Caseine
- Contenu et organisation des cours
- Évaluation initiale et création des groupes

Paradigmes de programmation

- Paradigme impératif

(Fortran, COBOL, BASIC, Pascal, C, Ada...)

```
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char *argv[]) {
    int a, b, tmp;

    if (argc != 3) {
        fprintf(stderr, "Arguments invalides\n");
        exit(1);
    }
    a = atoi(argv[1]);
    b = atoi(argv[2]);

L5:
    if (a >= b)
        goto L4;
    tmp = a;
    a = b;
    b = tmp;
L4:
    a -= b;
L3:
    if (b != 0)
        goto L5;

    printf("%d\n", a);

    return 0;
}
```

Paradigmes de programmation

- Paradigme impératif

(Fortran, COBOL, BASIC, Pascal, C, Ada...)

```
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char *argv[]) {
    int a, b, tmp;

    if (argc != 3) {
        fprintf(stderr, "Arguments invalides\n");
        exit(1);
    }
    a = atoi(argv[1]);
    b = atoi(argv[2]);
```

- 1968 – Dijkstra dénonce les méfaits de l'instruction GOTO
- Théorème de Böhm-Jacobi :
Toute fonction calculable peut être exprimée en terme de
 - Séquences
 - Sélections
 - Itérations

Paradigmes de programmation

- Paradigme impératif

(Fortran, COBOL, BASIC, Pascal, C, Ada...)

- Impératif Structuré

```
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char *argv[]) {
    int a, b, tmp;

    if (argc != 3) {
        fprintf(stderr, "Arguments incorrects\n");
        exit(1);
    }
    a = atoi(argv[1]);
    b = atoi(argv[2]);

    while (b != 0) {
        if (a < b) {
            tmp = a;
            a = b;
            b = tmp;
        }
        a -= b;
    }
    printf("%d\n", a);

    return 0;
}
```

Paradigmes de programmation

- Paradigme impératif

(Fortran, COBOL, BASIC, Pascal, C, Ada...)

- Impératif Structuré
- Impératif Procédural

```
#include <stdio.h>
#include <stdlib.h>

void echange(int **x, int *y) {
    int tmp = *x;
    *x = *y;
    *y = tmp;
}

int pgcd(int a, int b) {
    if (b == 0)
        return a;
    else {
        if (a < b)
            echange(&a, &b);
        return pgcd(a-b, b);
    }
}

int main(int argc, char *argv[]) {
    int a, b;

    if (argc != 3) {
        fprintf(stderr, "Arguments invalides\n");
        exit(1);
    }
    a = atoi(argv[1]);
    b = atoi(argv[2]);

    a = pgcd(a, b);
    printf("%d\n", a);

    return 0;
}
```

Paradigmes de programmation



- Paradigmes déclaratifs

Flot d'exécution

- Programmation Fonctionnelle

- (Lisp, Scheme, Erlang, Haskell, etc.)

```
(de pgcd (N P) (cond
((< N P) (pgcd P N))
((eq N P) N)
((eq P 0) N)
(t (pgcd (- N P) P)))))
```

- Programmation logique

- (Prolog, etc.)

```
pgcd(X,X,X).
pgcd(X,Y,D) :- X<Y, Nouvel_y is Y-X, pgcd(X,Nouvel_y,D).
pgcd(X,Y,D) :- X>Y, pgcd(Y,X,D).
```

Paradigmes de programmation

- Paradigme Objet

Basé sur les données

```
class Couple {
    int a, b; // Attributs

    Couple(int a, int b) {
        this.a = a;
        this.b = b;
    }
    int pgcd() {
        int tmp;
        while (b != 0) {
            if (a < b) {
                tmp = a;
                a = b;
                b = tmp;
            }
            a -= b;
        }
        return a;
    }
}
class Pgcd {
    public static void
        main(String args[]) {
            int a =
                Integer.parseInt(args[0]);
            int b =
                Integer.parseInt(args[1]);
            Couple couple =
                new Couple(a, b);
            System.out.println(
                couple.pgcd()
            );
        }
}
```

Paradigmes de programmation

- Autre Paradigmes
 - Programmation évènementielle
 - Programmation concurrente
 - Programmation par contrat

Algorithmique



- Algorithmique vs programmation
 - Algorithmique :
 - principes d'un paradigme
 - peut être adapté dans différents langages →
 - Les langages implémentent différents paradigmes
 - Syntaxe différente

Algorithmique et Programmation Objet

- Dans CE cours
 - Maîtrise de l'algorithmique
 - Impératif structuré et procédural
 - Objet
 - Utilisation de Java
 - Uniquement sur les paradigmes impératifs et objet
 - Spécificités de l'API Java
 - Facilité de portage vers d'autres langages

De la bidouille à l'algorithmique

« *Mais pourtant ça marche !!??!!* »

valeur	35	42	56	38	25
indice	0	1	2	3	4

- Ça compile ?
- Ça s'exécute sans crash ?
- Ça donne le résultat attendu ?
- C'est juste ?

```
boolean estDansTableau(int[] tab, int val) {  
    for(int i = 0 ; i < tab.length ; i++) {  
        if (tab[i] == val) {  
            return true ;  
        } else {  
            return false ;  
        }  
    }  
    return false ;  
}
```



Objectifs pédagogiques de l'APO

- Comprendre et maîtriser les concepts clés
 - Du paradigme impératif
 - Du paradigme orienté objet
 - Un peu de fonctionnel...
- Maîtriser la démarche informatique
 - Décomposer un problème
 - Déterminer les structures algorithmiques optimales pour la résolution du problème (Éviter de tomber dans les pièges flagrants qui conduisent à des erreurs difficiles à débugger)
 - Comprendre et maîtriser le flux d'exécution d'un programme (diagramme APO)
 - Comprendre et corriger du code existant

Présentation

- Équipe pédagogique
- Quelques remarques sur les LLM
- Algorithmique et Programmation Objet
- Contrat pédagogique (pédagogie inversée)
- Support pédagogique : Caseine
- Contenu et organisation des cours
- Évaluation initiale et création des groupes

Contrat pédagogique APO (et LW)



- Cours en pédagogie inversé
 - Polycopié de cours à lire/apprendre à la maison
 - CTD : **Quick**, puis exercices et questions (pas de rappel du cours!)
 - TD : exercices d'approfondissement **papier / crayon** en équipe
 - TP en Java, sur Caseine **sans IDE / LLM** travail individuel
- Note de l'UE :
 - 2/3 de la note : examen terminal
 - 1/3 de la note : contrôle continu
 - Moyenne des Quicks
 - Partiels papier et machine

Les labs Caseine des Tps sont uniquement à but formatif

Leur note n'est PAS prise en compte dans la moyenne

=> Tricher ou faire faire ses Tps par un LLM ne sert à rien !

On vous donne les Clés... ... C'est à vous d'acquérir les connaissances...

Investissement hors Cours

- Lire/étudier le polycopié
- Faire les exercices corrigés (lire une correction ne sert à rien)
- Faire les exercices d'entraînement Caseine
- Conseil : travailler en binôme ou en groupe

=> Prévoir du temps de travail conséquent en dehors des cours !!!

Investissement en CTD / TD

- Venir équipé
 - Réflexion sur papier
 - Amener le poly de TD (et de cours)
- Venir préparé
 - Avoir lu/étudié le nouveau cours
 - Inutile de venir non préparé (vous ne comprendrez pas un mot du TD...)
- Etre actif en CTD / TD
 - Pas de téléphone portable/d'ordinateur
 - Discussions hors sujet pendant les pauses
 - Poser des questions sur ce qui n'est pas compris

Présentation

- Équipe pédagogique
- Algorithmique et Programmation Objet
- Contrat pédagogique (pédagogie inversée)
- Support pédagogique : Caseine
- Contenu et organisation des cours
- Evaluation initiale et création des groupes

Supports : Caseine



- <https://caseine.org/>
 - Cours : « Algorithmique et Programmation Orientée Objet (UGA, L3 MIAGE) »
 - Accès par login obligatoire à partir de mercredi
 - Mot de passe d'inscription :
 - apo-2025

Caseine

https://moodle.caseine.org

Non connecté ([Connexion](#))

Avancer d'une page (⌘→)
Cliquer tout en déplaçant la souris vers le bas pour afficher l'historique

Caseine

Caseine



CaselnE est une plateforme d'apprentissage en génie industriel, informatique et mathématiques. Son objectif est de favoriser l'apprentissage et l'autonomie des étudiants tout en améliorant la qualité du temps que l'enseignant leur consacre. Basée sur Moodle elle permet

1. d'évaluer automatiquement le code informatique et les modèles mathématiques des étudiants,
2. d'assurer un suivi pédagogique des étudiants,
3. de partager des contenus entre les enseignants à travers une communauté d'utilisateurs.

[Contacts](#) - [Infos](#) - [Les enseignants](#) - [Pour en savoir plus](#) - [Film de présentation](#)

Rechercher des cours [Valider](#) [?](#)

Cours

▼ Open courses [Tout replier](#)

- [Découvrir caseine](#)
- [Base de données - SQL](#)
- [Introduction à l'informatique et à la programmation - Python](#)
- [Informatique - Java](#)

Caseine: Se connecter sur le site +

https://moodle.caseine.org/login/index.php 120 % ☆

Caseine

etudiantbeta
.....

Se souvenir du nom d'utilisateur

Connexion

You have forgotten your name, user name and/or password ?

Your browser must support cookies ?

Courses can be accessible to anonymous visitors

Connexion anonyme

Connect using account :

Connexion Grenoble (circled in red)

Autres comptes universitaires

Première visite sur ce site ?

Attention : pour les **étudiants et enseignants de Grenoble**, vous devez utiliser l'option **Connexion Grenoble** et vos identifiants académiques pour vous connecter.

Pour les **étudiants et enseignants d'autres structures académiques** françaises, utilisez l'option **Autres comptes universitaires**.

Si ce n'est pas votre cas, vous pouvez faire une demande de compte ci-dessous.

Créer un compte

Service d'Authentification de la

https://cas-simsu.grenet.fr/login?service=https%3A%2F%2Fmoodle.caseine.org%2F

Service d'authentification

Accès aux services numériques

Bienvenue sur le service d'authentification de la communauté des établissements d'enseignement supérieur du site "Grenoble Savoie" (UGA, Grenoble'INP, U.Savoie, IEP, COMUE "Université Grenoble Alpes") mis en oeuvre par le SIMSU de Grenoble.

Vous souhaitez accéder à un service qui nécessite une authentification.

Entrez votre nom d'utilisateur et votre mot de passe puis cliquez sur le bouton **Connexion** ci-dessous pour continuer.

Utilisateur Mot de passe

SE CONNECTER **EFFACER**

Pour des raisons de sécurité, veuillez vous déconnecter et fermer votre navigateur lorsque vous avez fini d'accéder aux services authentifiés.

Si vous avez oublié votre mot de passe, contactez la DSI de votre établissement

[Aide sur les services numériques](#)

**sim
sue** SERVICE INFORMATIQUE MUTUALISÉ DU SITE UNIVERSITAIRE

Caseine

https://moodle.caseine.org

120 %

Etudiant Bêta

Accueil

- Tableau de bord
- Calendrier
- Fichiers personnels
- Mes cours
- Discovery
- APO L3Miage
- TIM
- Pédagogie hybride à l'IM2AG
- Espace de partage

Caseine

Caseine



CaselnE est une plateforme d'apprentissage en génie industriel, informatique et mathématiques. Son objectif est de favoriser l'apprentissage et l'autonomie des étudiants tout en améliorant la qualité du temps que l'enseignant leur consacre. Basée sur Moodle elle permet

1. d'évaluer automatiquement le code informatique et les modèles mathématiques des étudiants,
2. d'assurer un suivi pédagogique des étudiants,
3. de partager des contenus entre les enseignants à travers une communauté d'utilisateurs.

[Contacts](#) - [Infos](#) - [Les enseignants](#) - [Pour en savoir plus](#) - [Film de présentation](#)

Mes cours

Découvrir caseine



Algorithmique et Programmation Orientée Objet (UGA, L3 MIAGE)

Traitement d'Images -Python (UGA, M1 Informatique)

Pédagogie hybride (UGA - UFR IM2AG)



APO L3Miage

Participants

Badges

Compétences

Notes

Session 01 - Installation et Évaluation initiale

Révisions et Approfondissement en Algorithmique en Autonomie

Session 02 - Classe, Instance et Diagramme APO

Session 03: Tableaux d'instances

Session 04: Tableaux à 2 dimensions

Session 05: Piles et Files

Sessin 06 - On récapitule

Semaine 07a - La classe Java, packages

Algorithmique et Programmation Orientée Objet (UGA, L3 MIAGE)

Accueil / Mes cours / APO L3Miage

Forums et Documents génériques pour APO

Annonces générales pour APO

Java Code Conventions

Polycopier cours Algorithmique et Programmation Objet (pour impression)

Polycopier de TD Algorithmique et Programmation Objet (pour impression)

Correction de code G1

Correction de code G2

Correction de code G3

Session 01 - Installation et Évaluation initiale

Étiquettes: 7 Fichier: 1 Virtual programming lab: 1

Progression : 0 / 1

Révisions et Approfondissement en Algorithmique en Autonomie

Présentation

- Équipe pédagogique
- Algorithmique et Programmation Objet
- Contrat pédagogique (pédagogie inversée)
- Support pédagogique : Caseine
- Contenu et organisation des cours
- Evaluation initiale et création des groupes