

# PROJET BLOC 0

DJEHONNON Jésugnon Geoffroy

Filière : Génie logiciel

Date : 18 / 02 / 2026

Institut de Formation et de Recherche en Informatique (IFRI)

# Vision globale de l'informatique

## □ Définition de l'informatique

- **Étymologie** : Contraction des mots "**Information**" et "**Automatique**".
- **Concept clé** : C'est la science du traitement rationnel et automatique de l'information à l'aide de machines (ordinateurs, serveurs, smartphones).
- **Le rôle** : Elle sert à stocker, organiser, transformer et transmettre des données pour résoudre des problèmes humains.

# Différences fondamentales

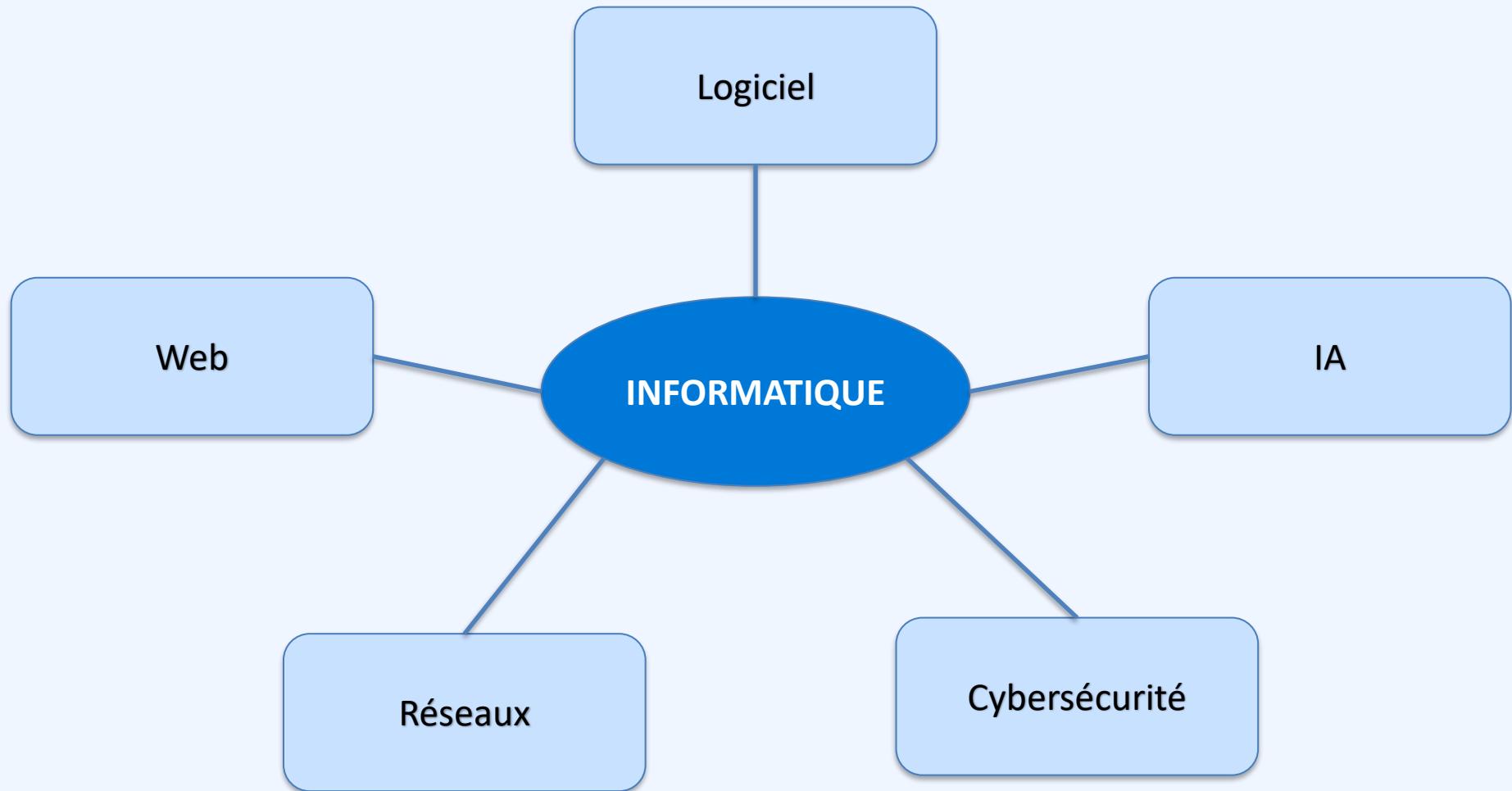
## Différence entre informatique, bureautique et simple usage

- **Simple usage** : C'est être un "consommateur". Utiliser une application (Instagram, Netflix) sans comprendre comment elle fonctionne.
- **Bureautique** : C'est une spécialisation de l'informatique liée au travail de bureau. Utiliser des outils pour produire des documents (Word), des calculs (Excel) ou des présentations (PowerPoint).
- **Informatique (Sens large)** : C'est la "cuisine". C'est comprendre la création des outils, la logique des programmes, la gestion des serveurs et la structure des données.

## Grands domaines

- **Web** : Création de sites et d'applications accessibles via un navigateur (Front-end : ce qu'on voit ; Back-end : la logique cachée).
- **Logiciel** : Développement d'applications natives pour PC ou mobiles (ex: Photoshop, WhatsApp).
- **Intelligence Artificielle (IA)** : Création de systèmes capables de simuler l'intelligence humaine (apprentissage, reconnaissance visuelle).
- **Réseaux** : L'infrastructure qui permet aux machines de communiquer entre elles (Internet, Wi-Fi).
- **Cybersécurité** : La protection des systèmes et des données contre les attaques malveillantes.

# Carte mentale des domaines



# Pensée algorithmique

## ❑ Notion d'algorithme

Un **algorithme** est une *suite d'instructions précises et ordonnées* qui permet de résoudre un problème ou d'accomplir une tâche.

- **Analogie** : Une recette de cuisine est un algorithme. Les ingrédients sont les données d'entrée, la cuisson est l'algorithme, et le plat est le résultat.

# Pensée algorithmique

- **Problème 1 : Vérifier si un utilisateur est majeur**
- **Raisonnement :** On demande l'âge. Si le chiffre est supérieur ou égal à 18, c'est bon. Sinon, c'est refusé.
- **Pseudo-code :**

DÉBUT

    AFFICHER ("Quel est votre âge ? ");

    LIRE (age\_utilisateur);

    SI (age\_utilisateur >= 18) ALORS

        AFFICHER ("Accès autorisé");

    SINON

        AFFICHER ("Accès refusé");

    FIN SI

FIN

# Pensée algorithmique

- **Problème 2 : Calculer la moyenne de deux notes**
- **Raisonnement :** Additionner les deux nombres, puis diviser le total par 2.
- **Pseudo-code :**

DÉBUT

```
AFFICHER("Entrez deux nombres");
LIRE (note1, note2);
somme = note1 + note2;
resultat = somme / 2;
AFFICHER (resultat);
```

FIN

# Environnement informatique

## ❖ Système d'exploitation (OS) utilisé: *Windows*

•**Rôle :** C'est le chef d'orchestre. Il gère le matériel (clavier, écran) et les logiciels.

## ❖ Organisation des dossiers du projet

Une bonne structure est cruciale pour ne pas se perdre:

- ***PROJET\_INFO/***
  - *Cahier Des Charges – Projet Bloc 0.pdf/*
  - *PROJET\_BLOC\_0\_Geoffroy\_Djebonnon.pptx/*

## ❖ Commandes de base du terminal

Le terminal permet de parler directement à l'ordinateur sans interface graphique.

- ls (ou dir sur Windows)*** : Liste les fichiers présents dans le dossier actuel.
- cd (Change Directory)*** : Permet d'entrer dans un dossier (ex: *cd documents*).
- mkdir (Make Directory)*** : Crée un nouveau dossier.
- Touch( type nul >)*** : Crée un fichier vide (ex: *touch index.html*).

# Environnement informatique

 MINGW64;/c/Users/CCG

```
CCG@Thanks-you-God MINGW64 ~ (master)
$ git init
Reinitialized existing Git repository in C:/Users/CCG/.git/
CCG@Thanks-you-God MINGW64 ~ (master)
$ C:\Users\CCG\.git\PROJET_INFO\Bloc 0\PROJET_BLOC_0_Geoffroy_Djehonnon.pptx
bash: C:UsersCCG.gitPROJET_INFOBloc: command not found

CCG@Thanks-you-God MINGW64 ~ (master)
$ git add C:\Users\CCG\.git\PROJET_INFO\Bloc 0\PROJET_BLOC_0_Geoffroy_Djehonnon.pptx
fatal: Unable to create 'C:/Users/CCG/.git/index.lock': File exists.
```

Another git process seems to be running in this repository, e.g.  
an editor opened by 'git commit'. Please make sure all processes  
are terminated then try again. If it still fails, a git process  
may have crashed in this repository earlier:  
remove the file manually to continue.

```
CCG@Thanks-you-God MINGW64 ~ (master)
$ git commit -m "Mon projet bloc 0"
```

# Git et GitHub

## ❑ Explication du versionnement

- **Le problème :** Quand on travaille à plusieurs, on finit souvent avec des fichiers nommés projet\_final V2 final vrai.zip.
- **La solution (Git) :** C'est une "machine à remonter le temps". Il enregistre chaque modification. Si vous faites une erreur, vous pouvez revenir à la version d'hier en un clic.

## Création d'un dépôt GitHub

- **GitHub :** C'est un site web (un "cloud") qui héberge vos projets Git pour que tout le groupe puisse y accéder.
- **Dépôt (Repository) :** C'est le dossier du projet en ligne.

## Étapes principales (Le flux de travail)

1. **git init** : "Réveille" Git dans votre dossier (crée le système de suivi).
2. **git add .** : Sélectionne tous les changements que vous voulez sauvegarder.
3. **git commit -m "Message"** : Valide la sauvegarde avec un petit commentaire explicatif.
4. **git push** : Envoie vos changements sur GitHub pour vos camarades.

## Lien vers le dépôt

- URL de mon dépôt: <https://github.com/Geoffroy-Dhn/Mon-Projet>

# Conclusion

## □ Ce que le groupe a appris

Le projet m'a permis de passer du statut de **consommateur** à celui d'**acteur** du numérique.

- **La rigueur de la machine** : j'ai compris qu'un ordinateur ne "devine" rien ; il n'exécute que des instructions strictement logiques. Une simple erreur de syntaxe ou d'indentation bloque tout le processus.
- **L'importance de l'abstraction** : Apprendre à modéliser un problème (via le pseudo-code) avant de sauter sur le clavier est une étape cruciale pour éviter les erreurs de logique.
- **La puissance de l'écosystème** : L'informatique n'est pas qu'un écran noir avec du code, c'est un ensemble vaste (IA, Réseaux, Web) qui interconnecte le monde entier.

# Conclusion

## ❑ Compétences acquises

Ce Bloc 0 a posé les "briques" fondamentales de mon profil technique:

- **Compétence Algorithmique** : Capacité à décomposer un problème complexe en une suite d'étapes simples et logiques.
- **Maîtrise de l'environnement technique** : Utilisation du **Terminal** (ligne de commande) pour naviguer et manipuler des fichiers, une compétence indispensable pour tout développeur professionnel.
- **Collaboration via le Versionnement** : Initiation à **Git et GitHub**, nous permettant de travailler en équipe sur un même projet sans écraser le travail des autres et en gardant un historique complet.
- **Organisation structurée** : Gestion rigoureuse de l'arborescence des fichiers et des dossiers.

J'ai été confrontés à **la gestion des droits d'accès sur GitHub**. Cet incident nous a permis de comprendre concrètement le rôle d'administrateur de dépôt et l'importance de la validation des invitations pour sécuriser un projet collaboratif.

# Conclusion

## □ Lien avec le développement web à venir (Bloc 1)

Le Bloc 0 est la fondation directe sur laquelle va s'appuyer le Bloc 1 (HTML/CSS/JavaScript) :

- **Logique de programmation** : Le pseudo-code que nous avons écrit est l'ancêtre direct du **JavaScript**. Les conditions (*SI... ALORS*) et les variables seront les moteurs de nos futurs sites web interactifs.
- **Infrastructure Git/GitHub** : Dans le Bloc 1, nous utiliserons GitHub non plus pour un PowerPoint, mais pour héberger nos sites web et collaborer sur du code réel en temps réel.
- **Structure et Rigueur** : L'organisation des dossiers apprise ici est exactement la même que celle d'un projet web (où l'on sépare les images, le style CSS et les fichiers HTML).
- **Navigation Terminal** : De nombreux outils de développement web (comme les serveurs locaux ou les outils de build) se lancent exclusivement via les commandes apprises dans la Partie « **Environnement informatique** ».

**Merci pour votre attention**

**Questions ?**