

Architecture interne des ordinateurs



Pr. Dominique Ginhac
dginhac@u-bourgogne.fr

D. Ginhac



\$ ~ whoami

Je suis **D. GINHAC**

Professeur à Polytech Dijon, j'enseigne l'informatique (Architecture des ordinateurs et Programmation Orientée Objet C++).

Chercheur au laboratoire ICB, je développe des travaux de recherche en Vision par ordinateur et Intelligence artificielle embarquée en temps réel.

Expert référent en Informatique au Ministère de l'Enseignement Supérieur et de la Recherche.



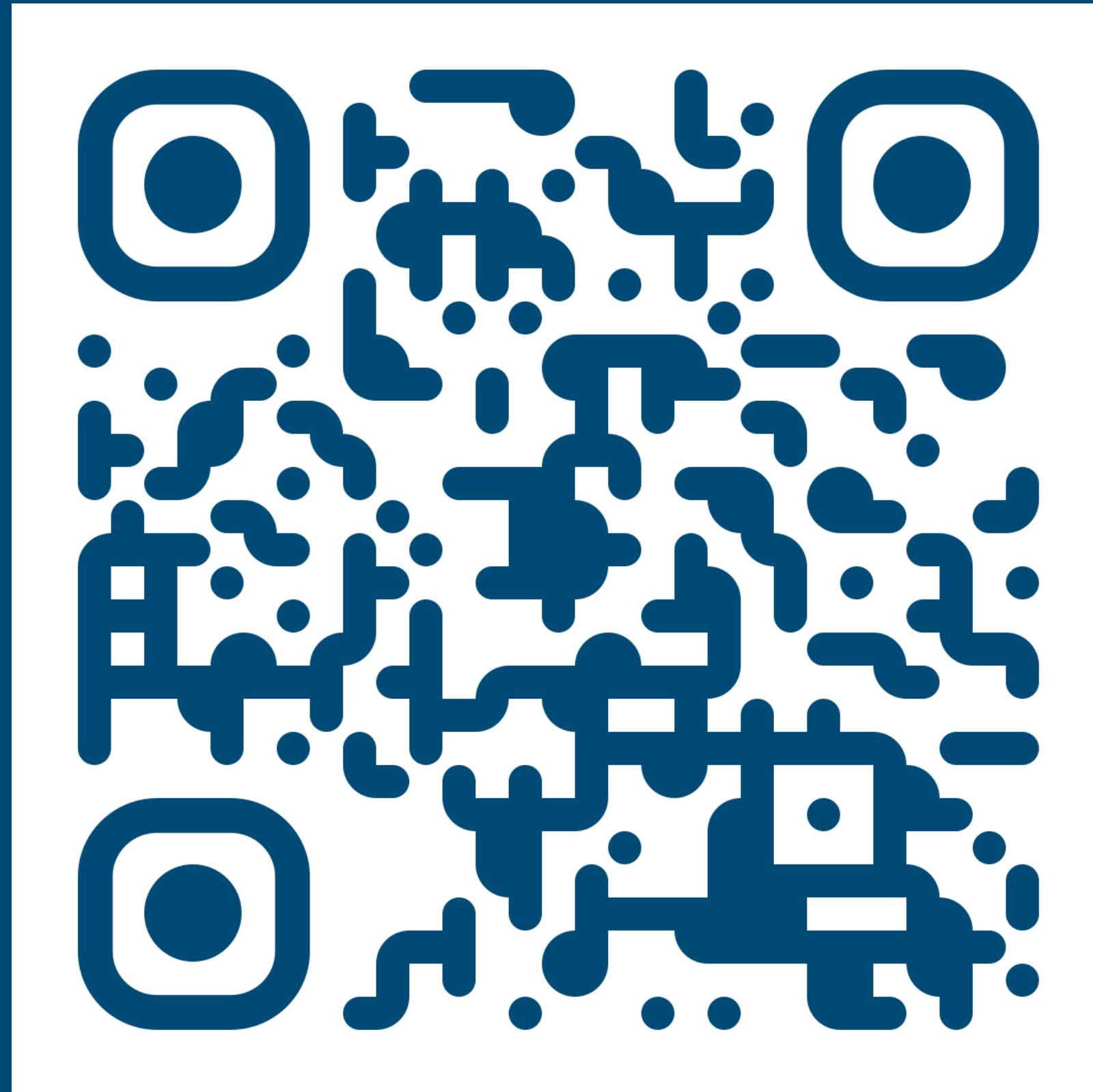
dginhac@u-bourgogne.fr



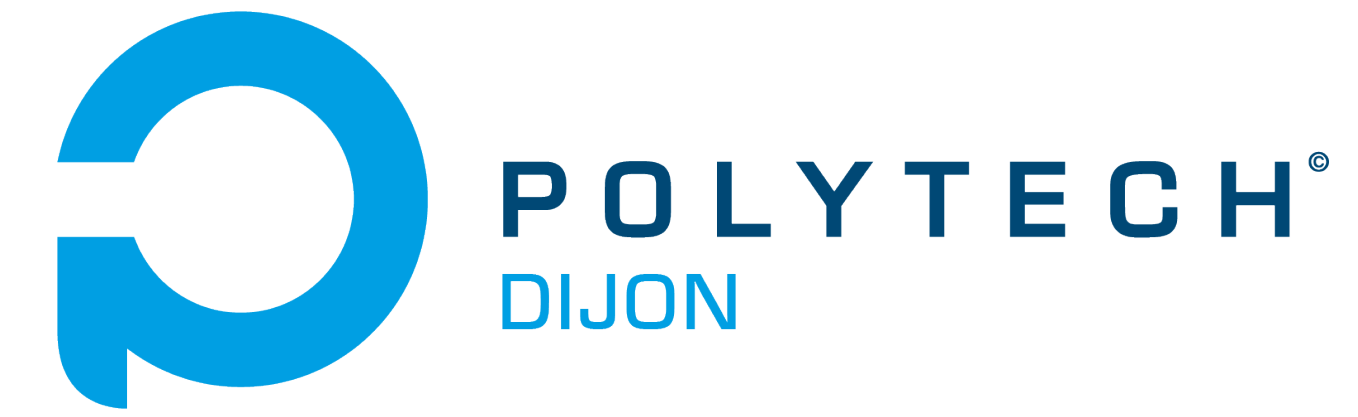
[dginhac](https://github.com/dginhac)



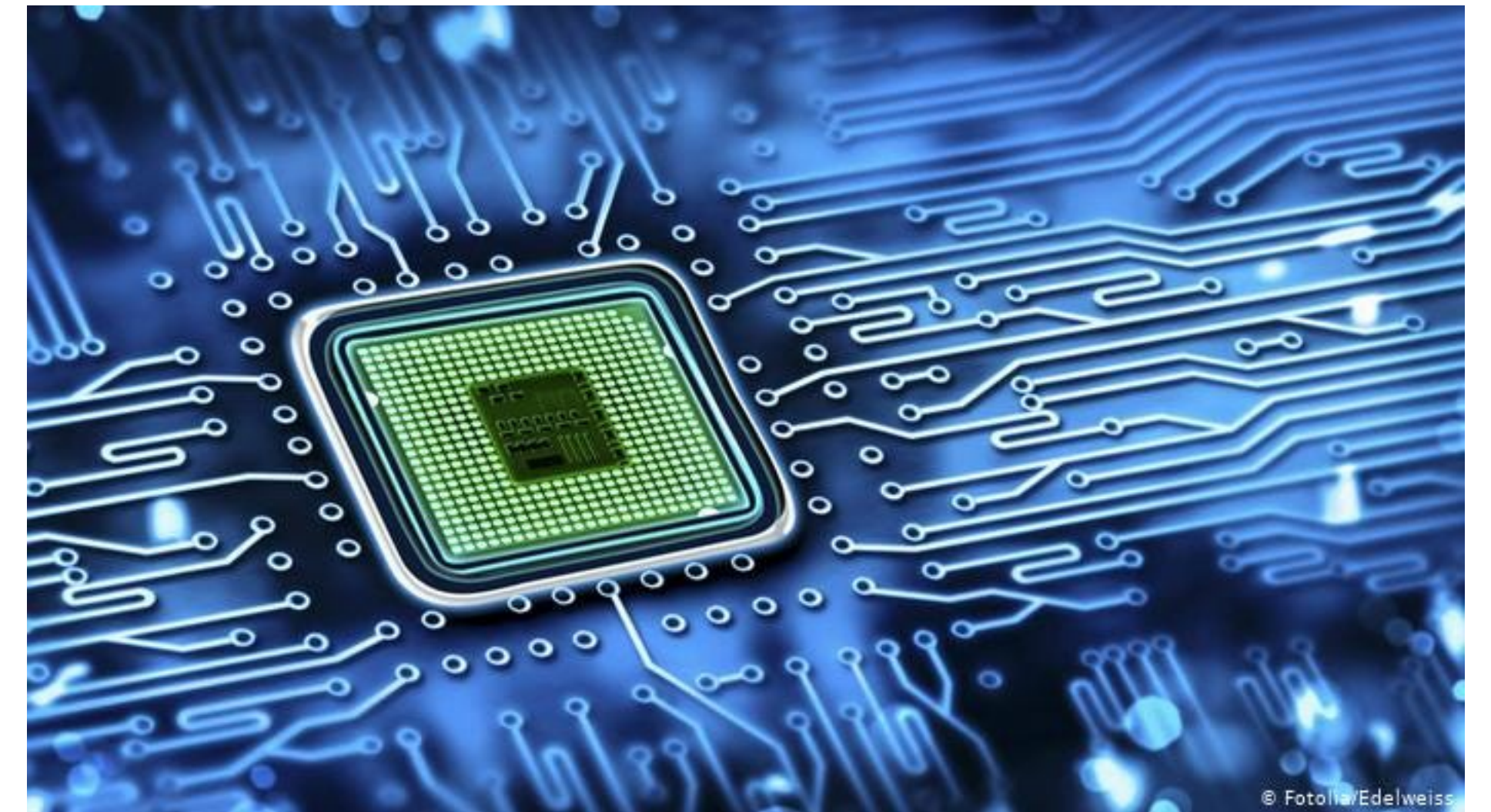
[dginhac](https://www.linkedin.com/in/dginhac)



<https://ginhac.com/archi/00-intro.pdf>



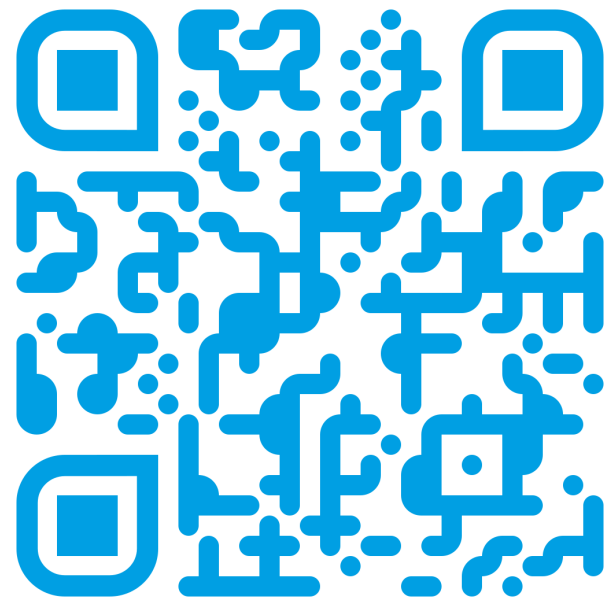
Architecture interne des ordinateurs



Du binaire à la programmation bas niveau.

<https://github.com/dginhac/esirem-archi>

Tout est disponible sur GitHub.



GitHub repository page for **esirem-archi** (Public).

Navigation: <> Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, Settings.

Repository details: **esirem-archi** (Public), 1 Branch, 0 Tags. Actions: Pin, Unwatch (1), Fork (0), Star (0).

Files:

File	Commit	Time
archives/2023-2024	lancement année 2024-2025	17 minutes ago
images	ajout TP sur calculatrice Notation Polonaise Inverse	10 months ago
LICENSE-CC-BY-NC	initial commit with README and Licenses	last year
LICENSE-GPLV3	initial commit with README and Licenses	last year
README.md	typo sur le nom du fichier intro.pdf	15 minutes ago

About: No description, website, or topics provided. Includes Readme, Licenses (Unknown, GPL-3.0), Activity, 0 stars, 1 watching, 0 forks.

Releases: No releases published. [Create a new release](#)

Packages: No packages published. [Publish your first package](#)

Contributors (2): **dginhac** (d0m Ginhac), **dev1bryant**

Languages: Assembly 100.0%

Architecture interne des ordinateurs

License: CC BY-NC 4.0, GPLV3

Introduction

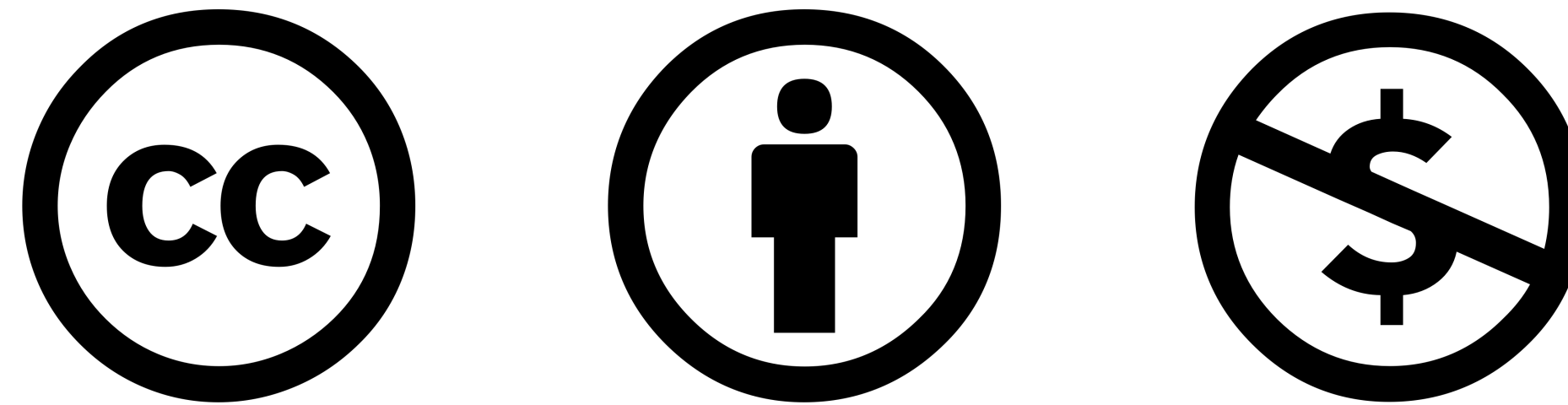
Les documents disponibles sur ce dépôt GitHub sont à destination des étudiants de la deuxième année de classe préparatoire de l'école d'ingénieurs [Polytech Dijon](#) - Dijon - France. Ils sont susceptibles d'évoluer au fil du temps.

Année 2024 - 2025

Les news

This document is **licensed** under a
Creative Commons Attribution-NonCommercial 4.0 International License.

<https://creativecommons.org/licenses/by-nc/4.0/>



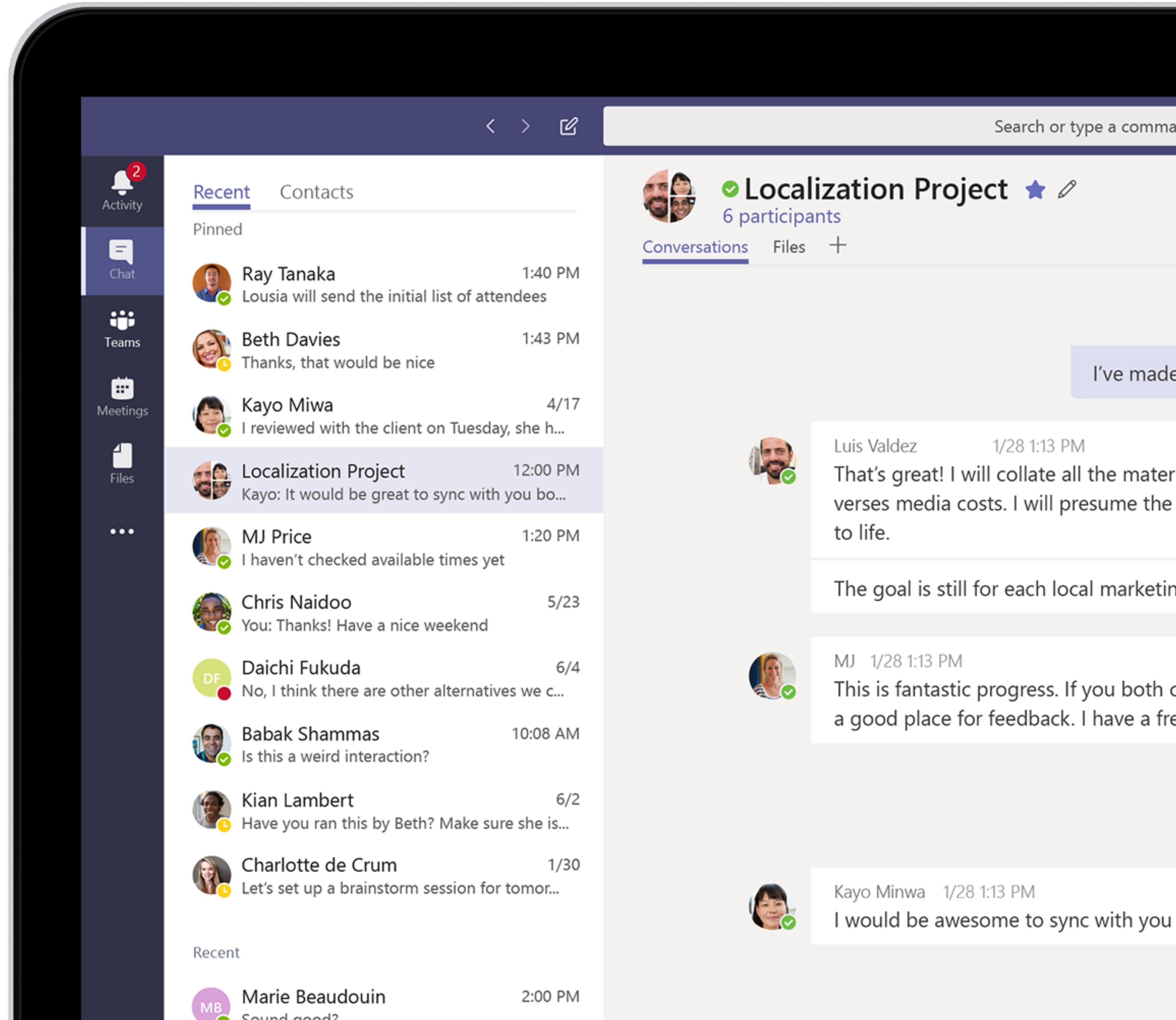
The code samples are **free software**: you can redistribute it and/or modify it under the terms of
the GNU General Public License- version 3 as published by the Free Software Foundation.

<https://www.gnu.org/licenses/gpl-3.0.html>



Microsoft TEAMS

pour toute
information
supplémentaire.



C'est votre Capitaine qui vous parle



**Merci de ranger vos téléphones
Nous allons décoller.**



<https://ginhac.com/archi/00-intro.pdf>

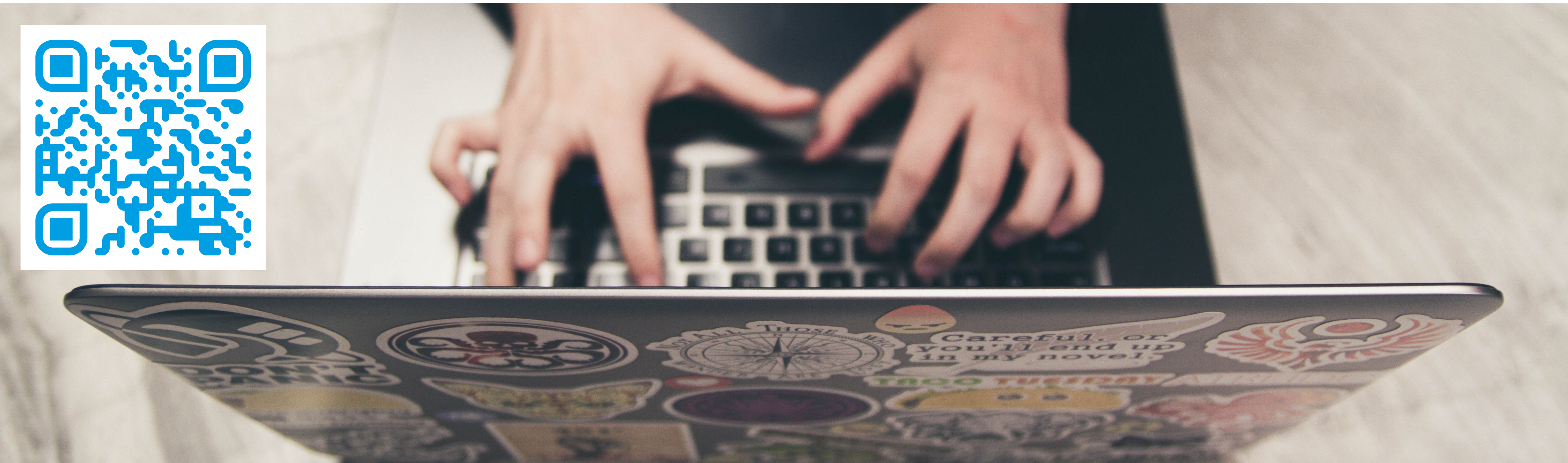
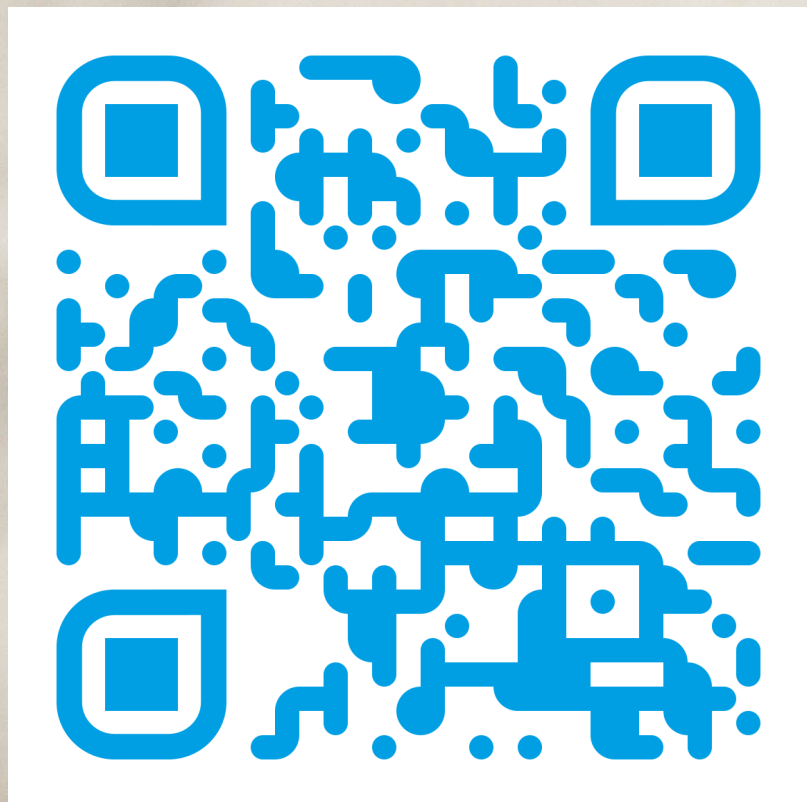


Photo par [NeONBRAND](#) sur [Unsplash](#)

Donner un aperçu du **cours** en
présentant les principales **notions fondamentales**.

Enjoy! 😊

Organisation des **cours**



CM

A partir de la semaine 47

12 x 1.75 h
D. Ginhac



TD

A partir de la semaine 50

8 x 1.75 h
D. Ginhac



TP

A partir de la semaine 04

8 x 2 h
Autres intervenants



Examens

Dates à définir

Théorique + Pratique

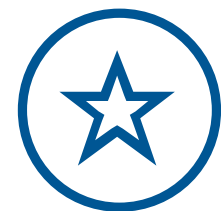
Votre mission (si vous l'acceptez ...)



Ecoutez attentivement.

Le contenu des cours, TD, TP, tutoriaux sont ou seront [disponibles](#) sur [GitHub](#).

Vous êtes donc encouragés à [venir en cours](#) et [participer activement](#) lors des CM, TD et TP.



Soyez proactif.

Mes slides sont peu denses (à part celle-ci) et je parle beaucoup. Donc n'oubliez pas de [prendre des notes](#) pour compléter votre cours.

Vous pouvez [utiliser votre ordinateur](#) pour prendre des notes sur mon cours et pour tester du code.

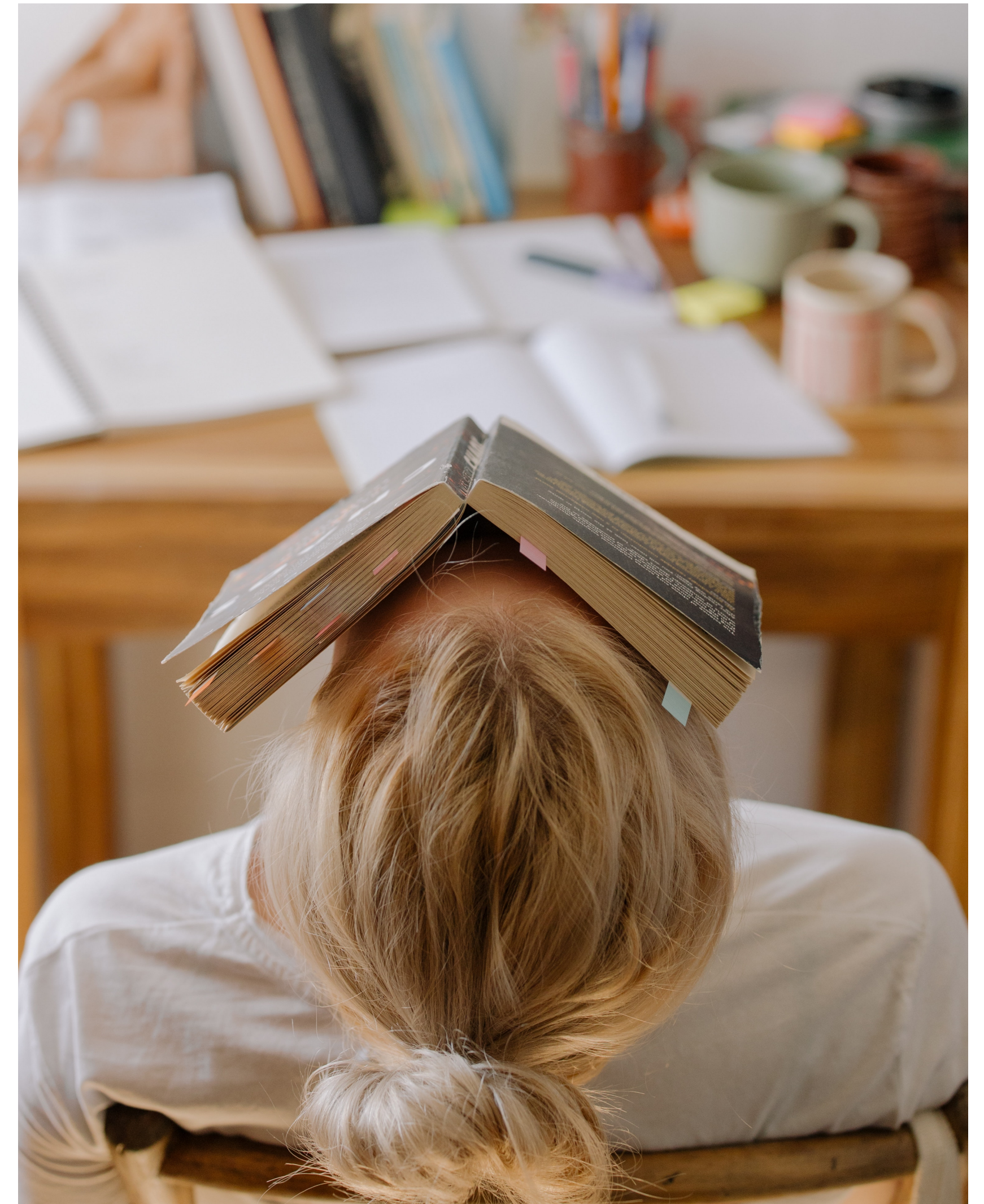
Vous pouvez même utiliser votre smartphone pour [prendre des screenshots](#) du tableau si je donne des explications complémentaires.



Posez des questions.

[Apprendre / Enseigner](#) l'architecture des ordinateurs n'est [pas simple](#) et peut devenir rapidement incompréhensible.


Soyez donc actifs et n'hésitez pas à m'interrompre dès que quelque chose n'est pas claire ou que vous voulez des informations complémentaires. Je prendrai toujours le temps de réexpliquer.



Une définition

L'architecture des ordinateurs désigne l'organisation et le fonctionnement des principaux composants d'un ordinateur (CPU, mémoire, bus).

Elle décrit comment ces composants interagissent pour **exécuter des instructions, gérer les données** et assurer le fonctionnement optimal de la machine.



Des objectifs d'apprentissage

1. Découvrir l'**architecture interne** des processeurs
2. Mieux comprendre les **principes de fonctionnement** des processeurs
3. Acquérir des **connaissances bas niveau** de la programmation

Quel intérêt d'avoir des connaissances en **programmation bas niveau** ?

①

Comprendre l'Architecture bas niveau des processeurs pour écrire du code de haut niveau "plus propre"

②

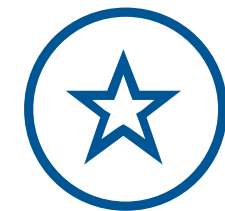
Optimiser les performances avec un accès plus direct aux ressources matérielles

③

Maitriser le développement de systèmes très contraints (systèmes embarqués, IoT)

Ces compétences sont requises pour de nombreux domaines de l'informatique tels que les systèmes embarqués, la cybersécurité, l'intelligence artificielle ou les jeux vidéos.

Quelles sont les **notions abordées** ?



Logique Booléenne et Portes Logiques

La logique booléenne, les opérateurs comme AND, OR, NOT ainsi que les tables de vérité, sont souvent utilisés en assembleur pour manipuler les bits. Ces concepts sont fondamentaux pour réaliser des comparaisons ou des branchements conditionnels.



Encodage des données en Binaire/Hexadécimal

La programmation en assembleur nécessite de comprendre le binaire (par exemple, pour gérer les drapeaux de statut, les masques de bits, et les opérations sur les bits) et l'hexadécimal, qui est fréquemment utilisé pour représenter des adresses mémoire et des valeurs numériques compactes.



Architecture des Processeurs

Comprendre l'architecture d'un processeur, y compris les concepts de registres, de mémoire, du bus de données, et des modes d'adressage. Cette compréhension est cruciale car l'assembleur interagit directement avec le matériel, et chaque instruction est pensée pour manipuler ces composants matériels de manière efficace.



Programmation Assembleur ARM

Avoir des connaissances en programmation assembleur permet de comprendre comment le matériel exécute réellement le code, ce qui est crucial pour écrire du code adapté à la plateforme cible, analyser les erreurs complexes (notamment en manipulant directement la mémoire ou les registres) et optimiser les performances.

Questions





Contacts

Pr. Dominique Ginhac

@ dginhac@u-bourgogne.fr

Retrouvez toutes les infos sur :

 <https://github.com/dginhac/esirem-archi>

