

TIW 8 - Technologies Web synchrones et multi-dispositifs

Introduction au cours

<https://aurelient.github.io/tiw8/>

Plan

- **Introduction au cours**
- L'informatique Ubiquitaire
- Rappels Stack JavaScript

Technologies Web synchrones et multi-dispositifs

Suite de MIF13

Objectifs :

- Aller au delà de l'état de l'art (applis Web desktop ou mobiles).
- Concevoir et réaliser des applications multi-dispositifs
- Qui fonctionnent sur plusieurs dispositifs à la fois (tablette, télé, ordi, mobile, etc.)
- Qui sont collaboratives temps réel (visio, éditeur de documents type google doc)
- Qui sont embarquées : IoT, utilisation de capteurs.
- Réfléchir aux enjeux de l'informatique ambiante : temps réel, synchronisation, consommation, performance, vie privée, etc

Contenu de l'UE

- Outils de programmation en JavaScript / TypeScript
- Programmation réactive (React, Easy Peasy V5)
- Multi-dispositif
- Nouvelles modalités de présentation et d'interaction
- Temps réel et WebRTC
- Web of Things
- Performance

Organisation de l'UE

- 5 CM de 1h30

- 9 séances de TP de 1h30 ou 3h

(Nouveau) Cours adaptable à votre niveau ! (Nouveau)

- **3 Niveaux:** Débutant, Adepté, Confirmé

Débutants, suivez ce cours.

Adeptes, tendez l'oreille tout de même mais allez vous renseigner sur:

TypeScript

Webpack

Easy Peasy V5

Organisation de l'UE

Confirmés, vous pouvez toujours faire attention au cours, mais mettez en place conteneurisé sous docker:

Un front React (en TS, avec Webpack, EasyPeasyV5, TailwindCSS)

(Optionel) Un back (en TS avec le framework NestJS et avec Inversion de dépendance et Injection)



Évaluation

3 Grands TPs (66%)

Un examen en février (33%)

Poser des questions sur le cours (1%)

Modalités d'évaluation

Rendu des TPs

- Mettez votre code sur la forge
- Ajoutez les encadrants de TP en tant que *Reporter* sur les projets
- **README.md OBLIGATOIRE** avec **4 sections**: Etat actuel de l'app, Dépendances (version de node, packages, ...), Setup (commandes), Utilisation (commandes à lancer). Bonus si Makefile.
- Remplissez les champs Tomuss pour le TP

L'entraide c'est bien

La copie de code non, elle sera détectée

TP: Problèmes de rendu à éviter

"Pourtant ça fonctionne chez moi":

- Argument invalide
- Abstraire votre projet de son environnement (à l'aide d'un .env par exemple)
- Attention aux URLs absolues locales
- Une application dockerisée = Une app qui fonctionne partout

Le manque de documentation

Des fichiers manquants

Pas d'accès donnés au repo

Peu importe la taille d'un projet de TP, c'est important qu'il soit **qualitatif**, pour nous, examinateur, comme vous, développeur.

Pour ceux que ça interesse

[LyonTechHub](#)

[LyonJS](#)