

# DEV FRONT END

## JS EVERYWHERE

Javascript est un langage complexe

Demande l'utilisation de beaucoup lib tierces combler ses manques (Ramda, Immutable.js, Flow, eslint, ...)

Evolue très vite sans jamais faire table rase du passé (don't break the web)

=> Ça explique que JS soit de plus en plus utilisé comme un « bytecode » pour des langages qui compilent vers JS, avec Typescript en première ligne !  
mais aussi Elm, Rescript, ClojureScript, Purescript, KotlinJs, Ocaml ...

# DEV FRONT END

CHOISIR UNE OPTION



Dans le cadre du cours, nous allons nous appuyer sur:

- Langage: Typescript (verbeux et mauvaise inférence, mais nécessaire à maîtriser en 2023)
- Framework : React
  - Seule lib front "mainstream" à avoir une approche expressive !
  - Un composant React = Une fonction qui prend en paramètre un objet props et retourne un objet de type React.Element
  - Un composant monté dans le DOM correspond donc à un objet instancié par un pattern Factory
  - Vous pouvez utiliser le DSL JSX pour décrire les Elements

```
import React from "react";  
  
const Welcome = ({ name } /* destructuring de props */) => <h1>Hello, {name} </h1>;
```

**NE CREEZ PAS DE COMPOSANTS AVEC DES CLASS, N'UTILISEZ PAS LES LIFECYCLES**