

# TP : AUTH FRONTEND

## COMPOSANT REACT

Seul 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 COMPOSANT AVEC DES CLASS, N'UTILISEZ PAS LES LIFECYCLES**

# TP : AUTH FRONTEND

## STATE MANAGEMENT REACT

A la différence d'une fonction graphique (ocaml-vdom / elm), un composant peut disposer d'un état local.

Nous utiliserons Redux pour gérer l'état application avec le middleware Redux-loop qui reprend le pattern Elm comme vu avec ocaml-vdom

Nous appliquerons la stratégie vu au cours 4:

- La gestion de la logique applicative doit être stockée dans l'état global (Redux)
- La logique de composant réutilisable peut être stockée dans l'état local (datepicker, info sur un champs, drag&drop, ...)

*A priori vous n'avez pas besoin d'état local pour ce TP, sauf si vous vouliez vous amuser à mettre indicateur de complexité sur un mot de passe*