Fondamentaux

Concepts clés de la Programmation Fonctionnelle

Auteur / Enseignant :
Alexandre Leroux (alex@shrp.dev) - 2024

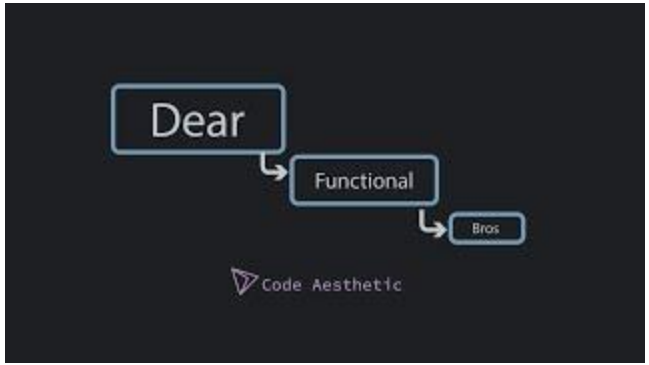
Objectifs de la Programmation Fonctionnelle

La Programmation Fonctionnelle regroupe un ensemble de concepts et de pratiques dont l'ambition est de **produire du code :**

- **fiable,**
- **expressif,**
- **prédictible,**
- **adaptable,**
- **réutilisable,**
- **testable.**

Concepts et objectifs fondamentaux de la Programmation Fonctionnelle

- Absence d'effet de bord,
- Immutabilité,
- Transparence Référentielle,
- Expressivité,
- Pureté,
- Récursivité,
- Memoization,
- Composition,
- Lazy Evaluation,
- Curryfication…



Concepts clés de la Programmation Fonctionnelle

Functional Programming

Immutability

Referential Transparency

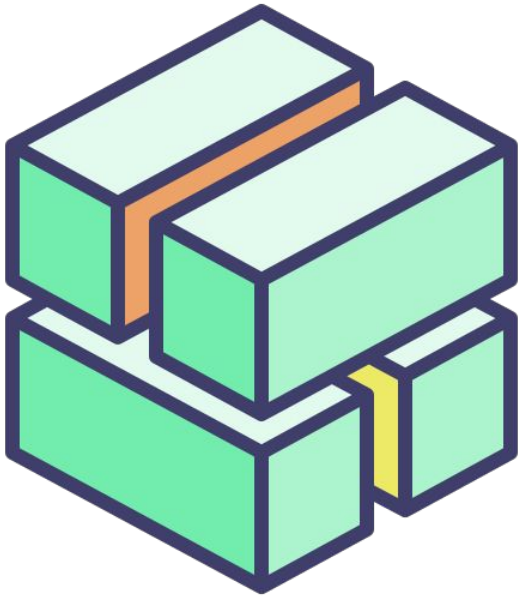
Higher Order Functions

Lazy Evaluation

Absence d'effet de bord

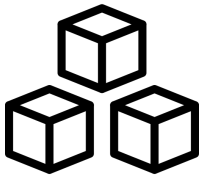
La Programmation Fonctionnelle considère l'absence d'effets de bord comme l'une des conditions *sine qua non* pour disposer d'un **code fiable.**

- **Immutabilité** des données,
- Emploi d'**expressions** plutôt que d'instructions,
- **Transparence référentielle** rendue possible par l'emploi de **fonctions pures et totales.**



Paradigme déclaratif

La Programmation Fonctionnelle est issue du **Paradigme Déclaratif**.
Ce paradigme repose sur la **séparation entre :**



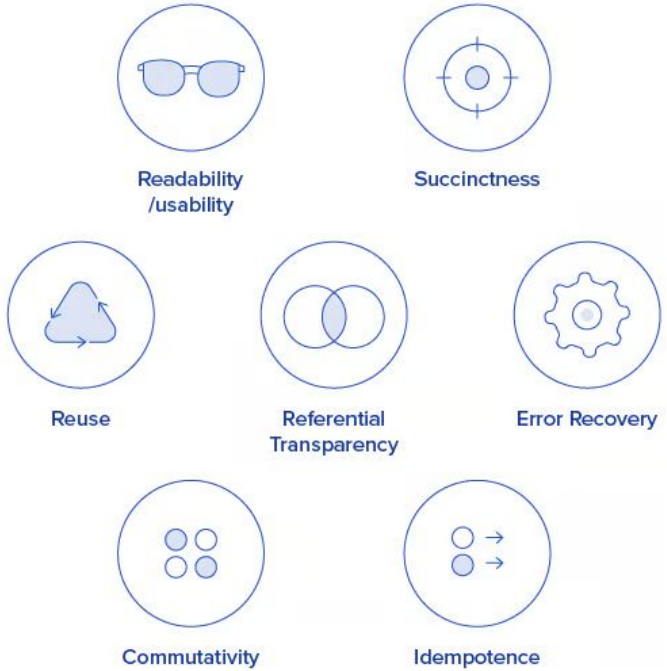
La gestion de l'état
soit, la valeur des données.

La gestion du comportement
qui produit des effets sur l'état

Expression d'intention

Dans le paradigme Déclaratif, le développeur exprime des "intentions" de résultat, et laisse la machine effectuer les calculs nécessaires pour y parvenir.

Ainsi, le développeur se concentre davantage sur le *"quoi"* que sur le *"comment"*.
Ce qui est **en parfaite opposition à l'approche impérative** (Programmation procédurale, Programmation Orientée Objet).

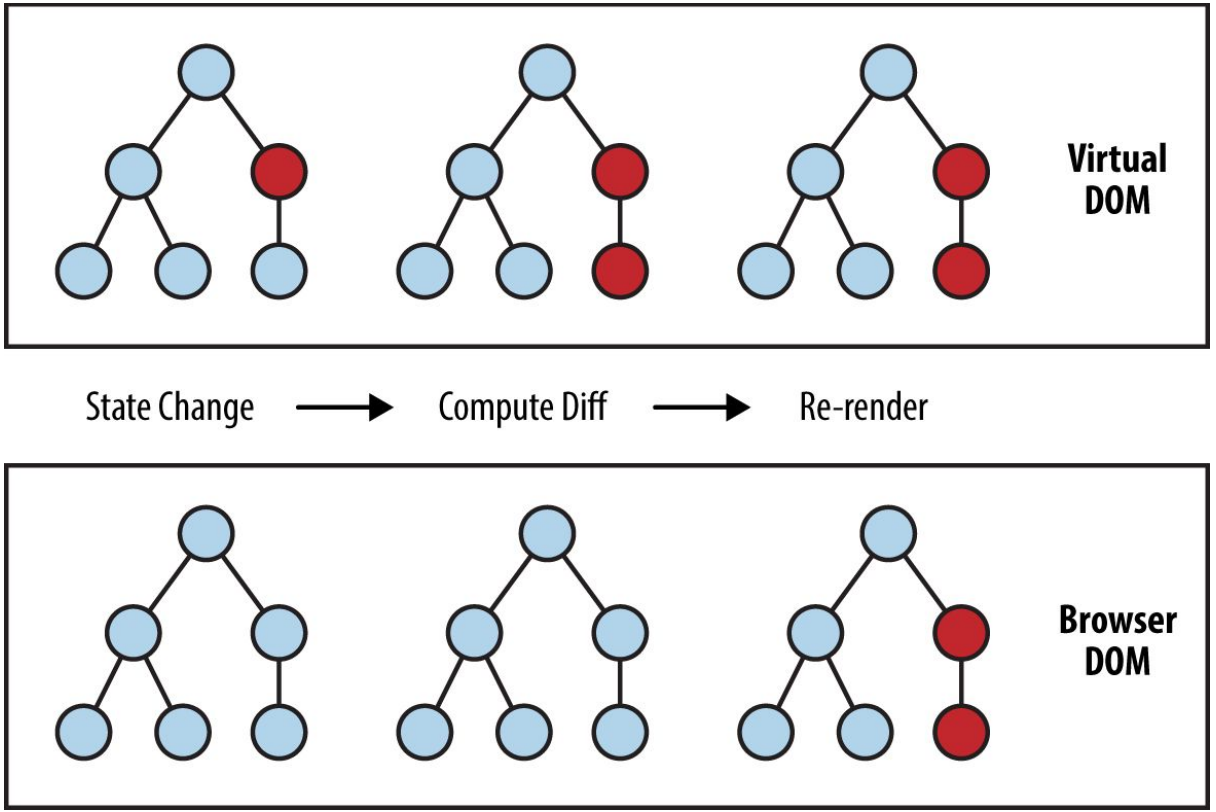


Concepts du Paradigme Déclaratif

https://www.toptal.com/software/declarative-programming

L'évolution de l'état doit être encadrée pour éviter toute incohérence.

Dans les applications Front End (cf. React, Flutter…), la construction de l'UI résulte de la mutation de l'état.



Changement d'état et mise à jour d'interface dans React (Virtual DOM / Browser DOM)

Fonction sans état

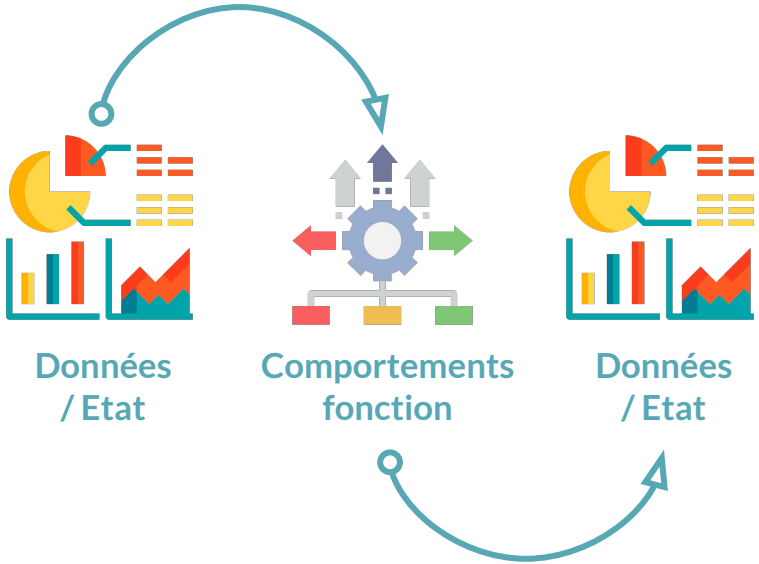
En Programmation Fonctionnelle, **les fonctions sont dénuées d'état (stateless)**.

Seules les données entrantes (input) sont considérées pour générer des données sortantes (output).

"Same input,
Same output."

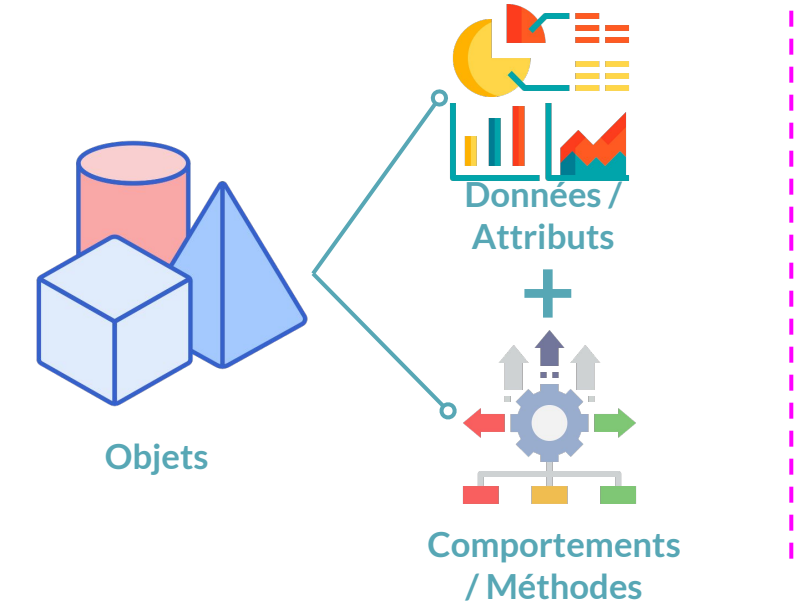
A partir des mêmes données entrantes, la fonction produira toujours les mêmes données sortantes.

Evolution de l'état
de façon horizontale
(traçabilité facilitée)

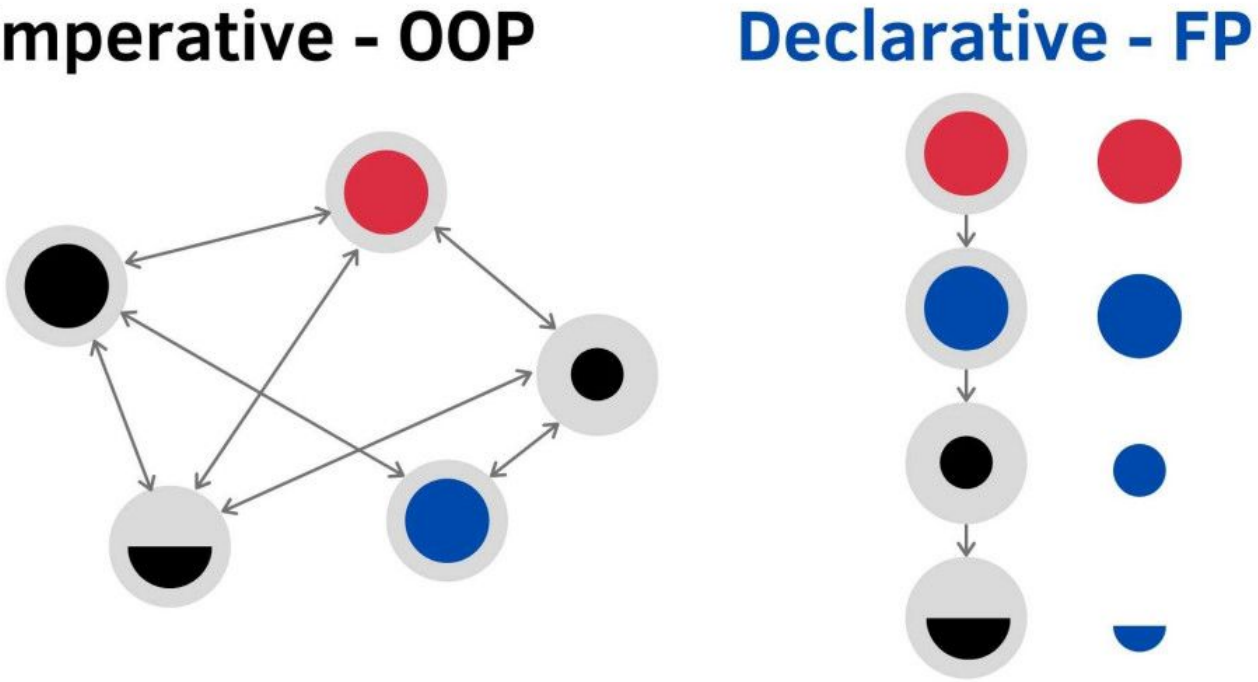


Programmation Fonctionnelle

Evolution de l'état
potentiellement horizontale et verticale
(traçabilité complexe)



Programmation Orientée Objet



Programmation Orientée Objet vs Programmation Fonctionnelle

POO vs PF ou POO + PF ?

La ***Programmation Fonctionnelle*** **(PF)** et la ***Programmation Orienté Objet*** (POO) sont 2 approches qui visent des **objectifs communs** en proposant chacune des **solutions différentes**.

Sur certains sujets, les mécanismes de résolution sont également similaires (composition, séparation des responsabilités, injection de dépendance, polymorphisme…). Les 2 paradigmes majeurs de programmation peuvent donc être complémentaires.