Cheat sheet “Functional Programming”

Functional programming is the way of programming with function calls instead of using variables.

A - Immutability (*clean code*)

This is the basis of the FP paradigm. Using a variable is a risk that another program will mutate it. Don’t use variables or constants.

B - Side effect free (*clean code*)

Side effects are unexpected behavior due to the use of global variables which are mutated by some function execution.

I - High Order Function (*HOF*)

HOF are functions that take a function in argument and/or return another function. This process allows you to create specific functions from more generic ones.

II - Function composition

This process allows you to create a specific workflow which is built from a set of more generic functions. This allows you to chain multiple function calls.

III - Currying

If you need to use a binary (*or more*) function in a “compose”, currying rewrite this function in order to provide you an unary function.

IV - Unary functions

Unary functions are functions with only one argument. You have to create/use unary functions to make them composable (*use in a “compose”*).

V - Point free

It is the fact of declaring a function without having to mention the arguments explicitly.

Dig deeper in functional programming (ingin.fr)

Author : Gaël Cadoret