nguyen minh quan

[Draw your reader in with an engaging abstract. It is typically a short summary of the document. When you’re ready to add your content, just click here and start typing.]

RamdaJS

[Document subtitle]

Table of Contents

**Type chapter title (level 1)1**

Type chapter title (level 2)2

Type chapter title (level 3)3

**Type chapter title (level 1)4**

Type chapter title (level 2)5

Type chapter title (level 3)6

# What is RamdaJS and why we need it?

It borrows fundamental FP ideas, including

* Emphasis on pure functions
* Everything’s curried
* Pipe/Compose are included and encouraged
* Functions take their data last

# Các khái niệm trước khi vào ramda

## Pure function

Function được consider là pure nếu:

* Same inputs *always* return same outputs
* No side-effects

Lý do Ramda prefer pure function vì Ramda được viết để tối ưu FP, giúp code với FP dễ hơn

## Higher-Order Function

HOFs có thể coi là 1 abtraction của 1 action, và return 1 function.

Why this great

import { median, pluck, pipe, reject } from 'ramda';

import employees from './employees.json';

const getMedianSixFigurePaycheck = pipe(

  getSalaries,

  reject(isBelowSixFigures),

  median,

  calculateMonthlyPaycheck,

  toUSD

);

const result = getMedianSixFigurePaycheck(employees);

console.log({ result });

Output

{ result: '$50,725.21' }

Với .pipe() dễ thấy :

* Get all salaries from a list
* Exclude salaries below six figures (< $100,000)
* Find the median salary
* Calculate how much they make each month
* Format in dollars (USD)

Việc sử dụng pipe còn giúp cho ta tránh khỏi phải tạo nhiều tên biến

## Composition.

Compose means [to create by combining things](https://www.educative.io/courses/functional-programming-patterns-with-ramdajs/N87WVwGGj3m).

Compose function giống như:

f(x) = x^2 +1

g(x) = x + 2

Compose Function: f(g(x)) = (x+2)^2 + 1

Vì tính chất thay thế nên compose function duyệt từ phải qua trái

const upperAndReverseFirstNames = (users) => {

  return users.map(upperAndReverseFirstName);

};

Thành

const upperAndReverseFirstNames = map(upperAndReverseFirstName);

Có thể hiểu compose là đầu tiên giải quyết g(x) sau đó giải quyết f(x)

Còn pipe() thì ngược lại, đầu tiên tính f(x) sau đó tính g(x)