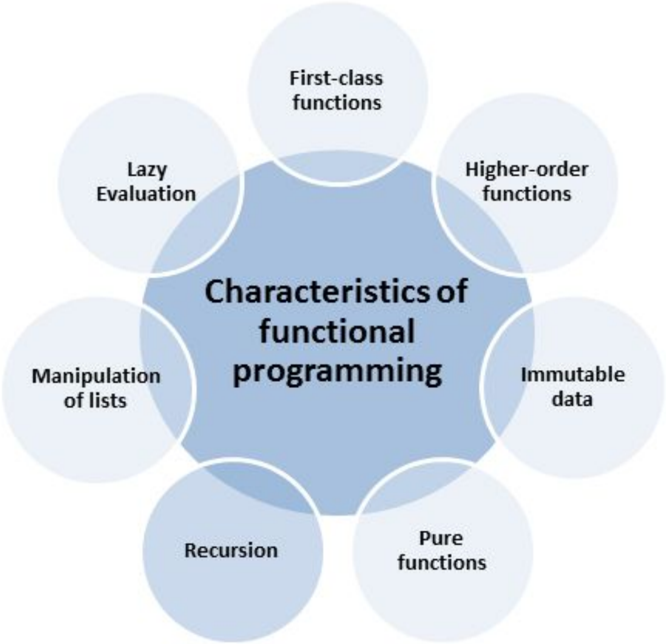
Ashot Nalbandyan
https://github.com/analbandyan

FP Principles

Agenda

- FP Characteristics
- Principles
  - Pure Functions
  - Immutability
  - Referential Transparency
  - Functions as first-class entities
  - Higher order functions
  - Disciplined state
  - Type systems
- Side Effects
  - IO
  - Exceptions
- Data structures
- Characteristics and techniques
  - Loops
  - Recursion
  - Currying
  - Memoization
  - Lazy evaluation
- OOP vs FP

Functional Programming Characteristics



Characteristics of functional programming

First-class functions

Higher-order functions

Immutable data

Pure functions

Recursion

Manipulation of lists

Lazy Evaluation

Pure functions

$$y=f(x)$$

Immutability

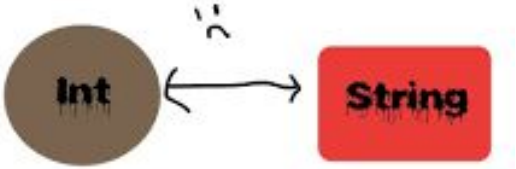
Referential transparency

Functions as first-class entities

Higher order functions

Disciplined state

Type systems



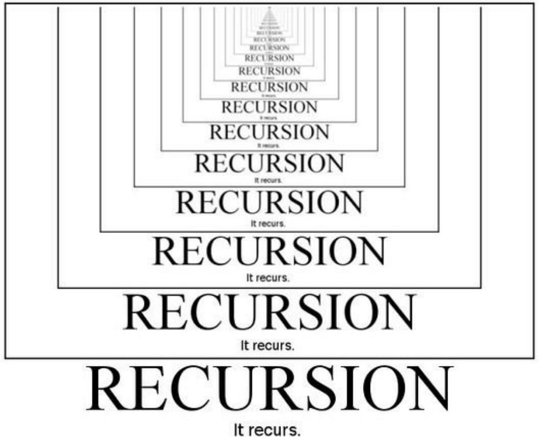
IO

Exceptions

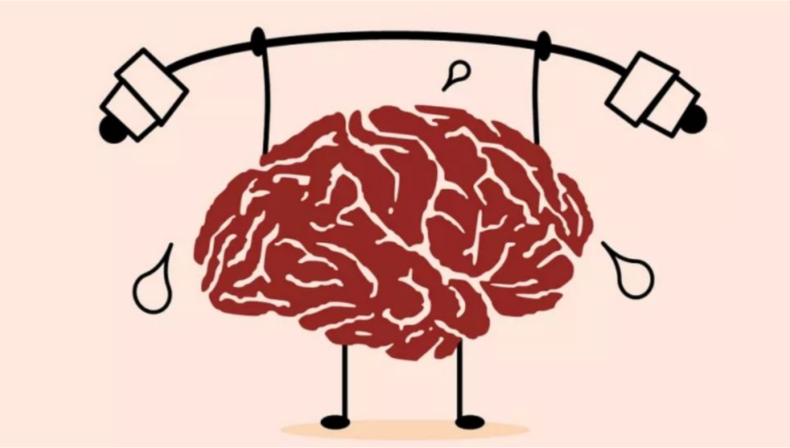
Data structures

Loops

Recursion



Memoization



Lazy evaluation

OOP vs FP

OOP vs FP

OO pattern/principles

- Single Responsibility Principle
- Open/Closed Principle
- Dependency Inversion Principle
- Interface Segregation Principle
- Factory Pattern
- Strategy Pattern
- Decorator Pattern
- Visitor Pattern

FP pattern/principles

- Functions
- Functions
- No way! Functions
- Again, Functions
- Still, Functions
- What the… what? Functions!
- Not funny!... Functions!
- Objects! Just kidding, Functions!

~~OOP vs FP~~

↓

OOP & FP

or simply

OOFP

Q&A

Thank you!