

Effects



Takeaways

Pure functional program = a big expression computing a value

- referential transparency = can replace an expression with its value without changing behavior

Expressions performing side effects are not replaceable

- i.e. break referential transparency

Effect = data type which

- embodies a computational concept (e.g. side effects, absence of value)
- is referentially transparent

Effect properties

- it describes what kind of computation it will perform
- the type signature describes the *value* it will calculate
- it separates effect *description* from effect *execution* (when externally visible side effects are produced)

Scala rocks

