Denis Shevchenko

- ☑ Haskell Developer at IOHK
- ☑ Co-founder of ruHaskell
- ☑ Code since 2005

@dshevchenko_biz

pyerevan

August 2019

A practical introduction to **functional** programming in Python

FP

→ essense

→ benefit

→ myth

FP → **essense**

I Purity

II Immutability

III Higher-order functions

Purity

```
func greet(person: String) -> String {
  let greeting = "Hi, " + person + "!"
  return greeting
}
```

Purity

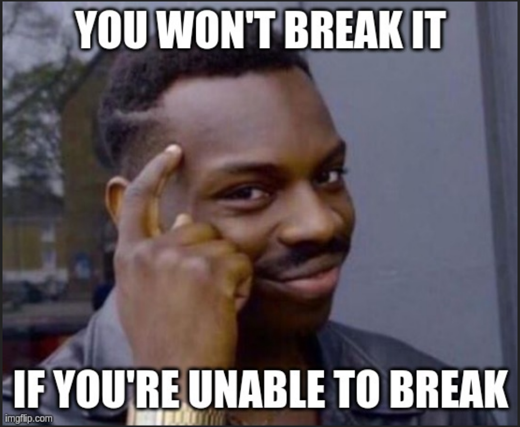
```swift
func greet(person: String) -> String {
  launchMissiles() // Ops!
  let greeting = "Hi, " + person + "!"
  return greeting
}
```

Purity

```haskell
greet :: String -> String
greet person = greeting
  where
    greeting = "Hi, " <> person <> "!"
```

Cannot launch missiles!

Immutability



Immutability

```
class Car {
  var make: String
  var model: String
}
```

Immutability

```
let car = Car(make: "VW", model: "Polo")
...
car.make = "Toyota" // Toyota Polo??
```

Immutability

```swift
struct Car {
  let make: String
  let model: String
}
```

Immutability

```
let car = Car(make: "VW", model: "Polo")
// VW Polo forever!
```

Higher-order functions

```
let values = [11, 13, 14, 17, 21, 33, 22]
var even : [Int] = []
for v in values {
  if v % 2 == 0 {
    even.append(v)
  }
}
```

Higher-order functions

```
let values = [11, 13, 14, 17, 21, 33, 22]
let even = values.filter { $0 % 2 == 0 }
```

FP → **benefit**

Code with **less** surprises

Easier to **reason** about

Easier to **maintain**

Easier to **test**

FP

- ☑ Love purity and immutability
- ☑ Code with less surprises

What’s **next**?

«A practical introduction
to functional programming»

bit.do/**fpPython**

Thank you!

Questions?