FUNCTIONAL SWIFT IN FIVE MINUTES

BUT FIRST.

THE WHY

LET'S UNDERSTAND THE PROBLEMS

GOOD CODE CAN BE REASONED ABOUT.

FUNCTIONAL CODE CAN BE REASONED ABOUT.

∴ FUNCTIONAL CODE IS GOOD!

CODE THAT CAN BE REASONED ABOUT IS GOOD.
FUNCTIONAL CODE CAN BE REASONED ABOUT.

.:. FUNCTIONAL CODE IS GOOD!

(YAY)

SIDE EFFECTS

MUTABILITY

LACK OF TESTABILITY

TIPS

BUT NO TRICKS

STRUCTS
OVER CLASSES

ENUMS ALL THE WAY DOWN

PURE FUNCTIONS

BAD

```
func uppercaseHelloName(name: String) -> String {
    self.name = name
    self.uppercaseName()
    return "Hello, " + self.name
}
```

GOOD

```
func uppercaseHelloName(name: String) -> String {
    return "Hello, " + name.uppercaseString
}
```

AVOID INSTANCE METHODS

BAN THE `for` LOOP

NOT PURE OR IMMUTABLE

```
var sum = 0
for val in [1, 2, 3, 4] {
  sum += val
}
sum // 10
```

PURE

```
func sum(start: Int, array: [Int]) -> Int {
    if let f = array.first {
        let rest = array.skip(1)
        return sum(start + f, array: rest)
    }
    else {
        return start
    }
}

sum(0, array: [1, 2, 3, 4]) // 10
```

FILTER AND MAP LIKE THERE IS NO TOMORROW

THANK YOU

@THATERIKPERSON