Functional Programming λ

WHY DO YOU LIKE FUNCTIONAL PROGRAMMING SO MUCH? WHAT DOES IT ACTUALLY *GET* YOU?
TAIL RECURSION IS ITS OWN REWARD.

How to invert a given binary tree ?

```
def invertTree(root):
    if root is None:
        return None
    root.left, root.right = invertTree(root.right), invertTree(root.left)
    return root
```

```
def invert(node):
    if node is None:
        return None
    else
        return Tree(node.value, invert(node.right), invert(node.left))
```

We describe the mapping between data instead of how to transform(mutate) the data.

No side effects

Referential transparency

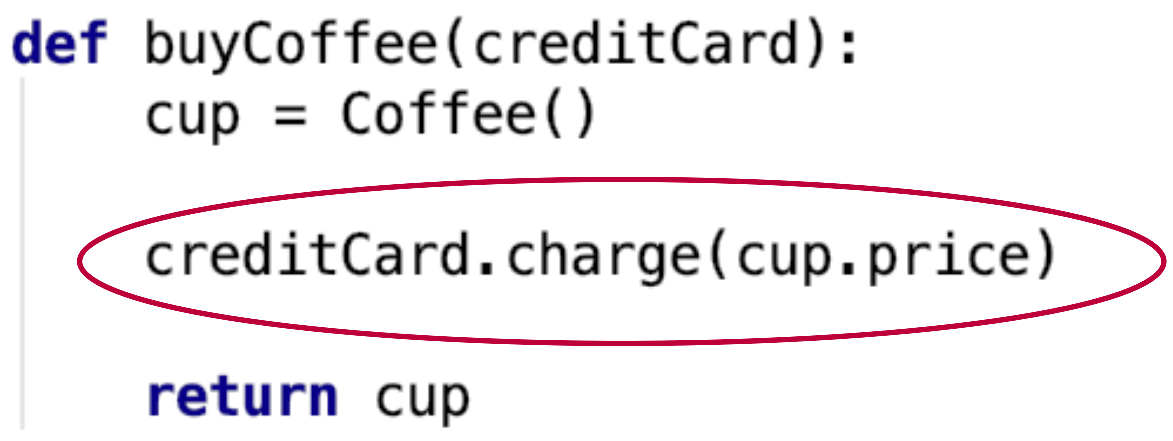
Side effect means if it does something other than simply return a result.

Modifying a data structure in place

Modifying a variable

Setting a field on an object

…………



```
def buyCoffee(creditCard):
    cup = Coffee()

    creditCard.charge(cup.price)

    return cup
```

We do not want our tests to actually contact the credit card company and charge the card!

Contacting the credit card company via some web service

Authorising the transaction

Charging the credit card

…………………

```python
def buyCoffee(creditCard):
    cup = Coffee()
    return cup, Charge(creditCard, cup.price)
```

“Charge” is a data type containing the credit card and the amount to charge.

Functional programming often speaks of implementing programs with a pure core and a thin layer on the outside that handles effects.

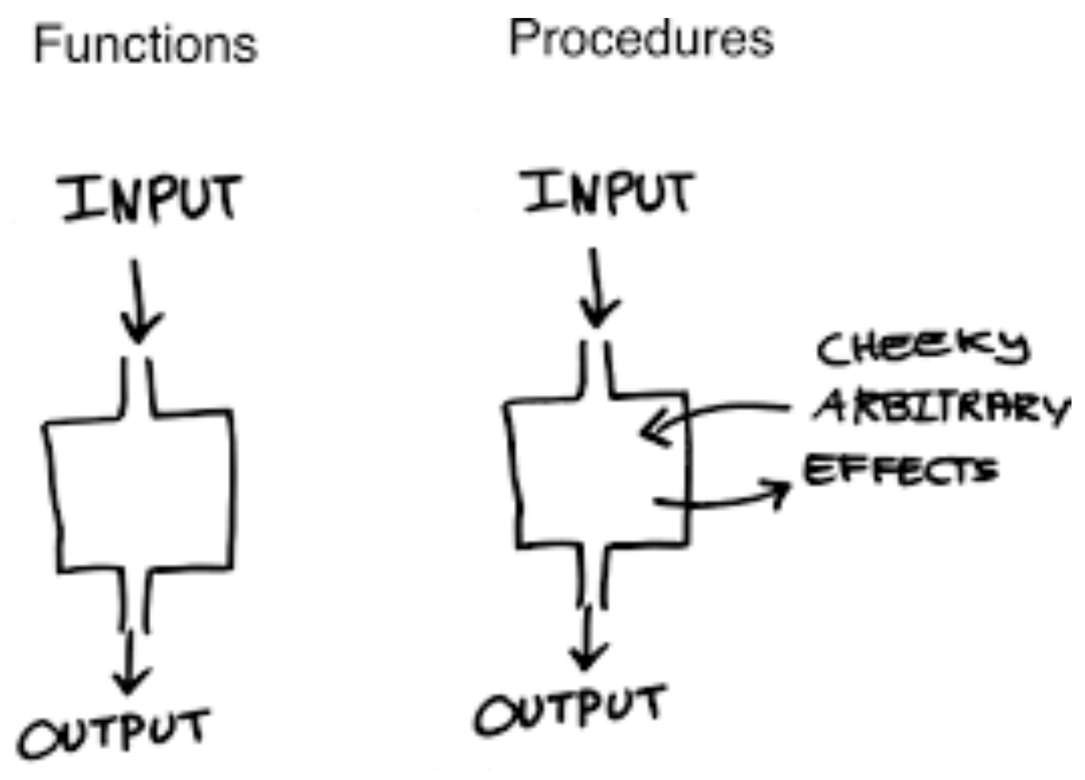
Referential transparency

“Everything a function does is represented by the value it returns”

$\searrow$ We call it pure function

Enables equational reasoning…

$$\begin{tikzcd}
f(x) + g(y) \arrow[d] & \\
8 + 7 \arrow[r] & 15
\end{tikzcd}$$



Mapping

No side effects

Referential transparency