COMP105 Lecture 2

What is functional programming?

What is functional programming?

In functional programming **everything is a pure function**

- We build our program entirely out of pure functions
- We combine simple functions to build more complex functions

Leads to a **very** different style of programming

- But we can still do everything that imperative programs can!

Building functions

Every line of a functional program will be of the form:

```
f(x) = <some other function>
```

Functions are built up from other functions

eg.

```
f(x) = square(x) + x
```

```
g(x) = h(i(x), j(x))
```

Building functions

Imagine an imperative language where every subroutine

- has only one line
- immedieatly returns a value

```
def square(x):
    return x * x

def square_plus_one(x):
    return square(x) + 1
```

What we won't see in functional programming

Functional programming has no concept of a **variable**

- Variables rely on side effects to operate
- So a function cannot use variables

Functional programming does not allow **loops**

- Loops need variables to operate
- Recursion is used instead

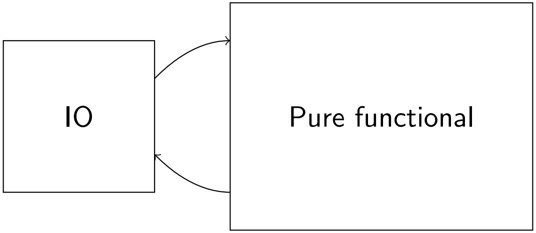
Infact there is no notion of **control flow** at all!

- Everything is just function application

But don’t we need side effects?

We want our programs to do interesting things

- So don’t we **need** side effects?



Yes, but

- Only a small amount of our code needs to communicate
- The rest can be pure functional
- We will study the pure functional bit first