Intro to Functional Programming

Drew Ripberger

January 15, 2020

Our Language of Choice

Haskell

"it is a polymorphically statically typed, lazy, purely functional language, quite different from most other programming languages"
haskell.org

Why learn functional programming?

- Concise syntax
- Functional purity simplifies large codebases
- Recursion fits many problems extremely well
- Higher order functions are seen in numerous imperative languages
- More logically consistent with math
- Many functional concepts show up across languages

The Changes with Functional

To state it simply, everything is a function.

- No variables
- No loops

Instead functions and recursion can be leaned on for all of these common tasks like iteration.

An Example

Counting down from 50

Haskell

```
main = mapM_ print [50,49..1]
```

Python

```
for i in range(50,0,-1):
    print(i)
```

Haskell Breakdown

Haskell

```
main = mapM_ print [50,49..1]
```

Here *mapM_* is a function that maps the *print* function to each of the elements of the list *[50,49..1*
This results in a countdown from 50 to 1 as such:
50
49
48
...
1