Functional Programming

Part 1

map
reduce
filter
zip

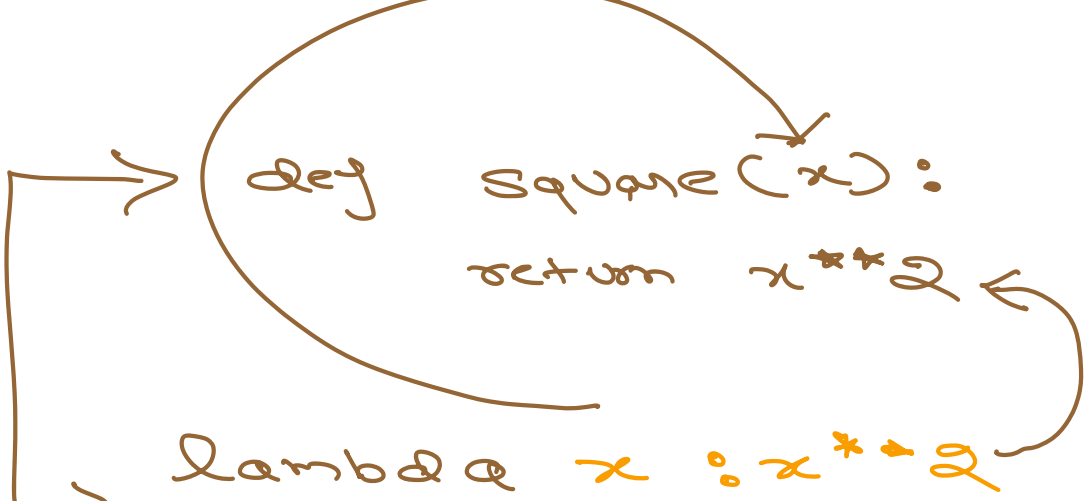
- Lambda Functions
- Higher Order Functions
- Decorators

What is FP?

FP is a programming paradigm where focus is to view code as Mathematical functions

1) Focus on immutability
   - avoids changing data inplace (State change)
2) Declarative style of programming

$$y = f(x)$$

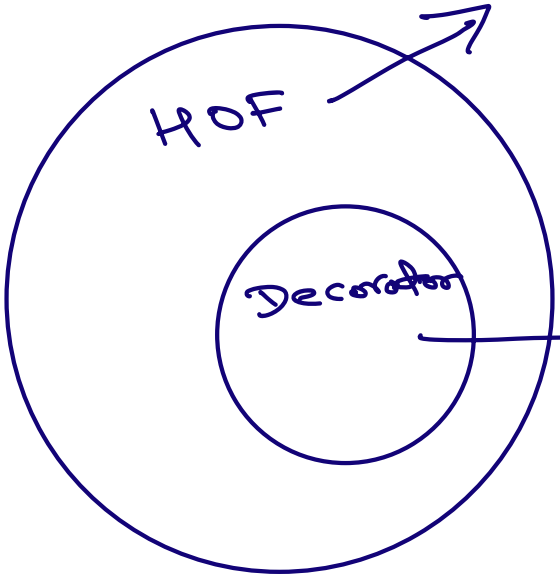


* Anonymous function
we can store and call lambda functions later

$$y = f(g(x))$$

Higher Order function

→ A function which returns another function



→ accepts a function and returns another

Use Case: When we want to create multiple variations of any function

```
def gen_exp(n):
    def exp(x):
        return x**n
    return exp
```

n=5

```
f_5 = gen_exp(5)
```

```
def exp(x):
    return x**5
```

```
f_5(10) → 10**5
```

x

gen_exp(2)(10)

x x

```
f_2 = gen_exp(2)
```

```
def exp(x):
    return x**2
```

```
f_2(10) → 2**10
f_2(3) → 2**3
```

Decorators

- Type of HOF that accepts a function as well