

Programming

Lesson14 - Recursive function

Saeed Isa

Recursive function

- ▶ Recursive → function call itself

```
def func(x):
```

```
    ...
```

```
    func(prev(x))
```

```
    ...
```

- ▶ func(x)

- ▶ Assume function solves func(prev(x))
- ▶ Do something with x

Example

► `def func(x):`
 `return x + func(x-1)`

Try: `func(3)`

► `3 + func(2)` →

► `3 + 2 + func(1)` →

► `3 + 2 + 1 + func(0)` →

► ...?

► `3 + 2 + 1 + 0 + -1 + -2 + -1000000000`

Stop condition

```
▶ def func(x):  
    if x == 1:  
        return 1  
    return x + func(x-1)
```

Try: func(3)

- ▶ $3 + \text{func}(2) \rightarrow$
- ▶ $3 + 2 + \text{func}(1) \rightarrow$
- ▶ $3 + 2 + 1 \rightarrow 6$

Let's try...😊

- ▶ Open PyCharm:
 - ▶ Create new file: recursive_functions.py
 - ▶ Start coding 😊

Questions - Factorial

- ▶ Factorial function $\rightarrow n!$
 $\text{factorial}(n) = n * (n-1) * (n-2) * \dots * 1$
 $5! = 5 * 4 * 3 * 2 * 1 = 120$
 $\rightarrow \text{factorial}(n) = n * \text{factorial}(n-1)$

Questions - Fibonacci sequence

- ▶ 0, 1, 1, 2, 3, 5, 8, 13, 21, 34...
- ▶ Definition
 - ▶ $n_0: 0$
 - ▶ $n_1: 1$
 - ▶ $n_2: n_0 + n_1$
 - ▶ $n_3: n_2 + n_1$
- ▶ `fibonacci(idx)` → return Fibonacci num in index idx
 - `fibonacci(2)` → 1
 - `fibonacci(7)` → 13

“

Thank you 😊!

”

Stay tuned for more!