Recursion

Definition

Applying the same logic repeatedly to solve a problem

- usually a smaller and smaller problem

Problem-solving process stops when base-case is reached

factorial

informally:

5! = 5*4*3*2*1

formally

1! = 1

n! = n * (n - 1)!

iterative approach

```
n = 5

result = 1

for i in range(1,n+1,1):

        result *= i
```

iterative approach

```
def factorial_iter(n: int) -> int:
    result = 1
    for i in range(1,n+1,1):
        result *= i
    return result
```

recursive approach

```
def factorial(n: int) -> int:
    if n == 1:
        return 1
    else:
        return n * factorial(n - 1)
```

base case (recursion stops)

recursive call with progression of the input

What happens…

- if base case is missing?
  - infinite loop
- if there is no progression of the input?
  - infinite loop
- if progression is the wrong direction?
  - infinite loop