

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
#Created on Tue Oct 24 13:53:52 2017
#@author: christianehehzel
```

1 Rekursiver Funktionsaufruf

Rekursive Funktionen rufen sich wieder selbst auf. Wichtig hierbei ist, irgendwo eine Abbruchbedingung einzubauen, sonst wird die Funktion nie beendet.

1.1 Beispiel Fakultät

$n! := 1 * 2 * 3 * \dots * n$

iterative Implementierung der Fakultät. Hier wird eine Schleife durchlaufen.

```
def iterative_factorial(n):
    result = 1
    for i in range(2, n+1):
        result *= i
    return result

print(iterative_factorial(10))
```

| 3628800

Und hier die rekursive Version. Sie kommt ohne Schleife aus

```
def rekursive_factorial(n):
    if n == 1:
        return 1
    else:
        return n * rekursive_factorial(n-1)

print(rekursive_factorial(10))
```

| 3628800

In dieser Version kann man auch sehen, wie die Funktion arbeitet

```
def rekursive_factorial(n):
    print("factorial is called with n=", n)
    if n == 1:
        return 1
    else:
        res = n * rekursive_factorial(n-1)
        print("intermediate result for ", n, " * factorial(", n-1, "): ", res)
        return res

print(rekursive_factorial(5))
```

```
factorial is called with n= 5
factorial is called with n= 4
factorial is called with n= 3
factorial is called with n= 2
factorial is called with n= 1
intermediate result for 2 * factorial( 1 ): 2
intermediate result for 3 * factorial( 2 ): 6
intermediate result for 4 * factorial( 3 ): 24
intermediate result for 5 * factorial( 4 ): 120
120
```

Die obigen Beispiele und weitere Informationen finden Sie unter
https://www.python-kurs.eu/python3_rekursive_funktionen.php