# Der Call-Stack und die Rekursion

Ein populäres Beispiel für rekursive Algorithmen ist die Fakultätsfunktion:

```
5! = 5*4*3*2*1
fakultaet(5) = 120
fakultaet(3) = 3*2*1 = 6
```

---

### (A1) Iterativ

Implementiere in BlueJ eine iterative Version der Fakultätsfunktion, die als Argument die Zahl entgegennimmt, deren Fakultät berechnet werden soll.

---

### (A2) Rekursiv

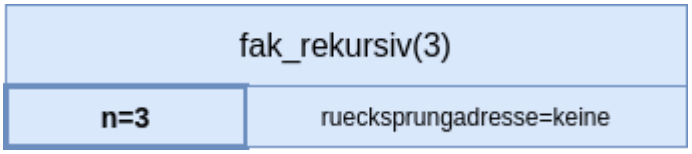
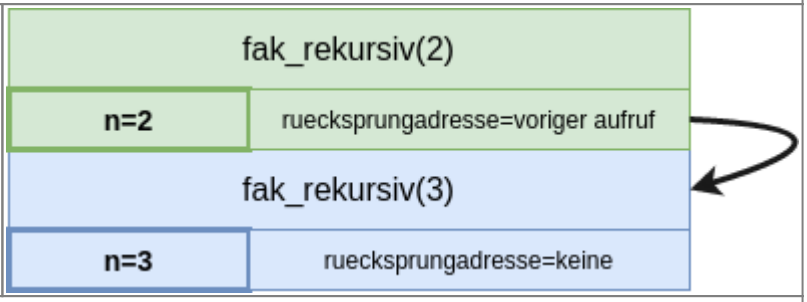
Implementiere anhand des folgenden Pseudocodes eine rekursive Version `fak_rekursiv`.

```
fak_rekursiv(int n):
  wenn n=1:
    return 1
  sonst:
    return n*fak_rekursiv(n-1)
```

- Was ist der Rekursionsfall, was der Basisfall?
- Teste deine rekursive Methode

## Dataillierte Betrachtung des Call-Stacks bei der Rekursion

| Was passiert | Wie sieht der Stack aus? |
|---|---|
| | |

| **Was passiert** | **Wie sieht der Stack aus?** |
|---|---|
| `fak_rekursiv(3)` wird aufgerufen.<br>Auf dem Stack wird Speicher für diesen Aufruf reserviert. Es gibt keine Rücksprungadresse.<br>Innerhalb dieses Aufrufs wird `fak_rekursiv(2)` (nächster Schtritt) aufgerufen,<br>da die Fallunterscheidung nicht zum Basisfall<br>führt sondern zum Rekursionsfall. |  |
| | |
| `fak_rekursiv(2)` wird aus dem vorhergehenden Aufruf heraus aufgerufen.<br>Auf dem Stack wird Speicher für diesen Aufruf reserviert. Es gibt keine Rücksprungadresse |  |
| | |

From:
https://www.info-bw.de/ -

Permanent link:
**https://www.info-bw.de/faecher:informatik:oberstufe:algorithmen:rekursion:callstack_rekursion:start?rev=1642073485**

Last update: **13.01.2022 11:31**