

LU05.A11: Factorial



Schreibe ein Programm, das die Fakultät einer Zahl berechnet.

Die Fakultät von n , bezeichnet als $n!$, wird berechnet als $1 * 2 * 3 * \dots * n$. Zum Beispiel ist die Fakultät von 4 24 oder $4! = 1 * 2 * 3 * 4 = 24$. Außerdem wurde festgelegt, dass die Fakultät von 0 gleich 1 ist, also $0! = 1$.



Fakultäten werden vor allem in der Wahrscheinlichkeitsrechnung verwendet, wenn verschiedene mögliche Ordnungen einer Menge untersucht werden. Zum Beispiel kann eine Gruppe von fünf Personen $5!$ verschiedene Reihen bilden, und ein Kartenspiel mit 52 Karten kann $52!$ verschiedene Reihenfolgen haben. Die Faktorrechnung kann auch zur Berechnung von [Kombinationen](#) verwendet werden. Es ist zum Beispiel möglich, $52! / (5! * (52-5)!)$ verschiedene Blätter aus einem 52-Kartenspiel austeilen, und man kann $40! / (7! * (40 - 7)!)$ verschiedene 7er-Lotterielinien aus 40 Zahlen bilden.

Auftrag

Erstelle die Programmlogik für das Programm:

1. Der Benutzer gibt eine natürliche Zahl ein.
2. Das Programm berechnet die Fakultät dieser Zahl und gibt das Resultat aus.

Schreibe die Programmlogik einmal als `while`-Schleife in `main_while` und einmal als `for`-Schleife in `main_for`.

Beispiel

```
Give a number: **3**  
Factorial: 6
```

Die interne Berechnung war hier $1 * 2 * 3 = 6$

```
Give a number: **10**  
Factorial: 3628800
```

Und nun war die interne Berechnung $1 * 2 * 3 * \dots * 8 * 9 * 10 = 3628800$

Vorgehen

1. Akzeptiere das GitHub Classroom Assignment im Moodlekurs.
2. Kclone das Repository in PyCharm.
3. Codiere die Programmlogik in `main.py`.
4. Teste dein Programm mit den Testfällen in `main_test.py`.
5. Führe einen Commit und einen Push durch.

Abgabe

Die Abgabe erfolgt durch den Push ins GitHub Repository. In Moodle ist keine Abgabe vorgesehen oder möglich.

Vorlage

```
def main_while():  
    # Write your program here and remove the line "pass"  
    pass  
  
def main_for():  
    # Write your program here and remove the line "pass"  
    pass  
  
if __name__ == '__main__':  
    main_while()  
    main_for()
```

Anmerkung: Kümmere dich im Moment nicht zu sehr um `if __name__ == '__main__':`. Dieser Programmblock legt fest, welche Funktion beim Ausführen gestartet wird. Wir werden dies im Verlauf des Moduls noch genauer betrachten.

M319-LU05



Kevin Maurizi, Marcel Suter

Diese Aufgabe ist eine übersetzte und angepasste Aufgabe von [Scott Morgan](#), verwendet unter CC BY NC SA.

From:

<https://wiki.bzz.ch/> - **BZZ - Modulwiki**

Permanent link:

<https://wiki.bzz.ch/de/modul/m319/learningunits/lu05/aufgaben/factorial>

Last update: **2025/06/23 07:45**

