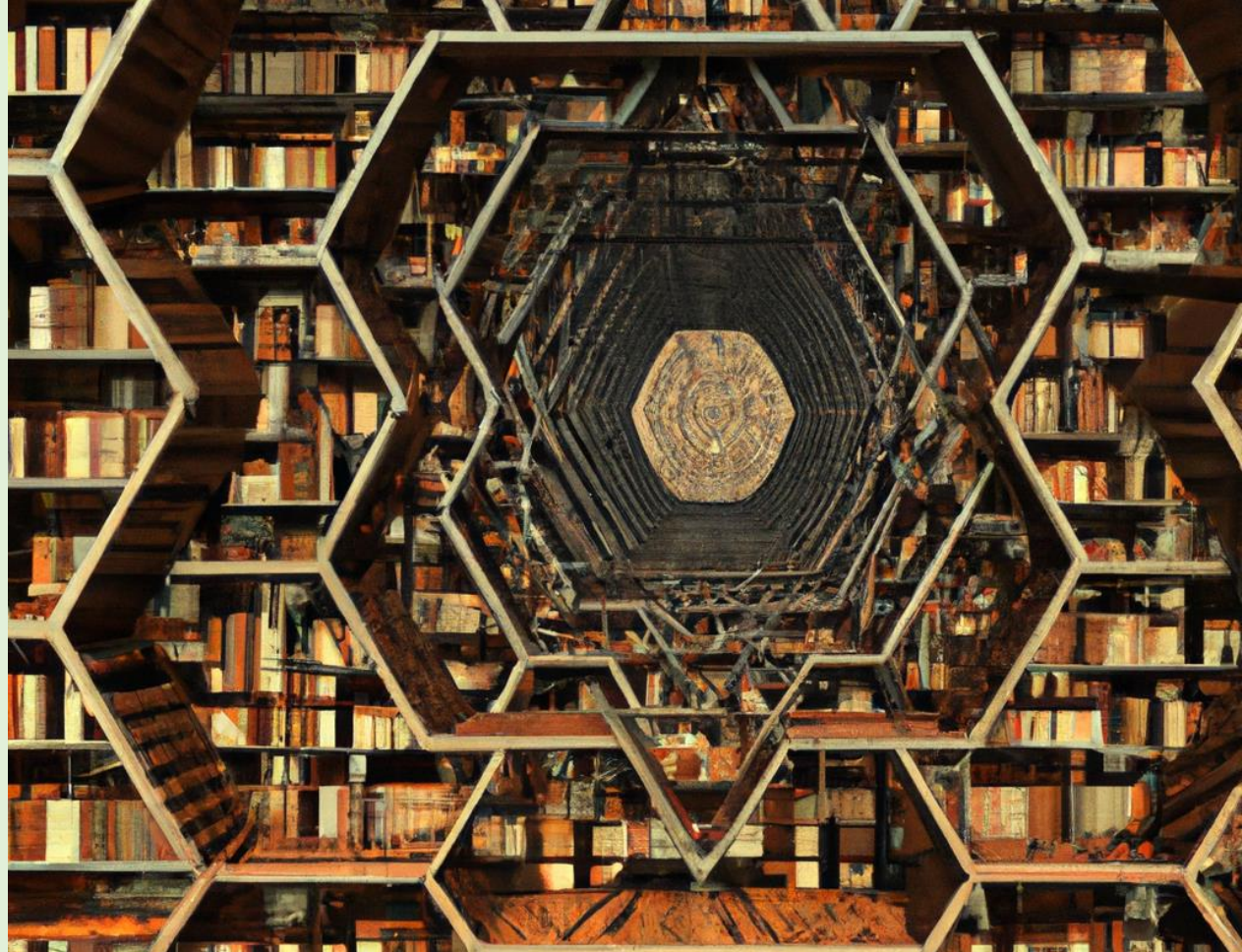




EINFÜHRUNG IN PROGRAMMIERUNG UND DATENBANKEN

JOERN PLOENNIGS

REKURSION IN FUNKTIONEN



DALL-E 2: Library of Babel

REKURSION - ALLGEMEIN

Wikipedia: „ Als Rekursion (lateinisch recurrere ‚zurücklaufen‘) wird ein prinzipiell unendlicher Vorgang, der sich selbst als Teil enthält oder mithilfe von sich selbst definierbar ist, bezeichnet. “

Praktisch gesprochen: Rekursion geschieht immer dann, wenn eine Funktion *sich selbst* aufruft.

Rekursionen werden insbesondere bei Divide-and-Conquer-Algorithmen oder bei kombinatorischen Problemen verwendet.

REKURSION - IN PYTHON

Wir benötigen nichts weiter als eine Funktionsdefinition die sich wieder selbst aufruft

```
def rekursive_funktion():  
    Statement1  
    rekursive_funktion()
```

REKURSION - BEISPIEL: FAKULTÄT

Die Fakultät einer Zahl ist definiert als Produkt der natürlichen Zahlen von 1 bis n

```
def factorial_rekursiv(x):  
    if x > 1:  
        return x * factorial_rekursiv(x-1)  
    else:  
        return 1
```

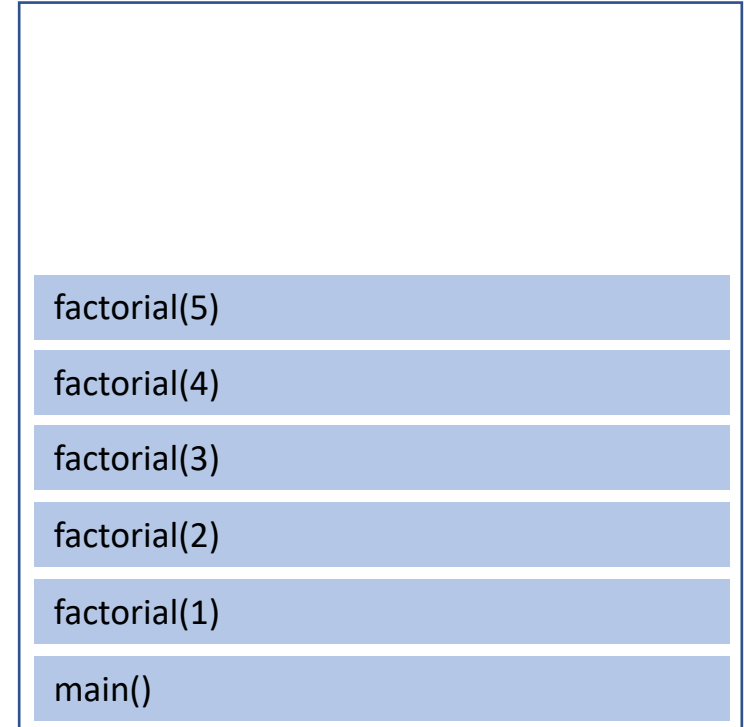

INFORMATIK-EXKURS: ENDLOSREKURSION, STACK

Bei rekursiven Funktionen gibt es das selbe Problem, wie bei while-Schleifen: **Sie können sich (theoretisch) endlos verzweigen!**

Deshalb sollte mindestens ein Pfad in der Funktion nicht rekursiv sein.

Praktisch kommt es allerdings bei endloser Rekursion zu **Stack-Overflow-Fehlern**. Das liegt daran, dass Python nachvollziehen muss, welche Funktion welche andere aufgerufen hat, um beim beenden in die richtige Funktion und Position zurückzukehren. Das geschieht über eine Liste die sich **Stack** nennt.

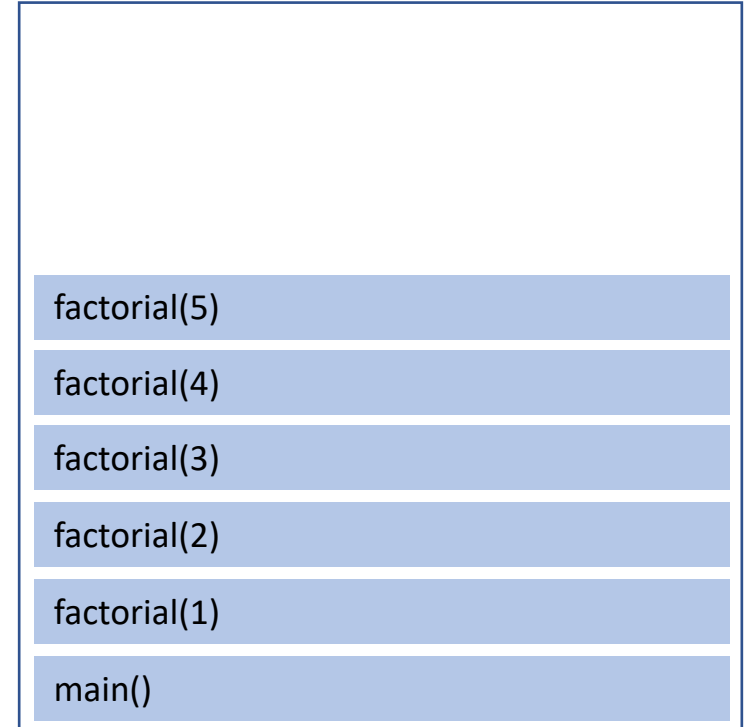
Stack-Größe



INFORMATIK-EXKURS: ENDLOSREKURSION, STACK

Der Stack ist eine LIFO-Liste (Last In First Out) bei der das letzte Element das hinzugefügt wird (die neu aufgerufene Funktion) auch das erste sein muss, was entfernt wird (wenn die neu aufgerufene Funktion wird beendet). Quasi wie ein Bücherstapel.

Stack-Größe

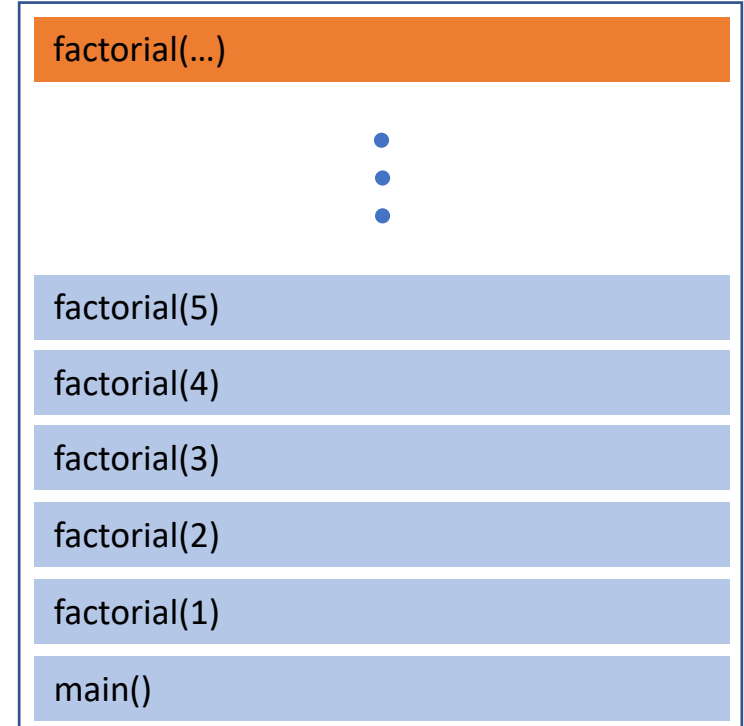


INFORMATIK-EXKURS: ENDLOSREKURSION, STACK

Die Stack wird bei endloser Rekursion so groß, dass er nicht mehr in den vorgesehenen Speicher passt, so dass es ein **Stack-Overflow-Fehler** ist.

Das ist ein fataler Fehler der nicht abgefangen werden kann und zum Absturz des Programmes führt.

Stack-Größe



Übrigens werden die Variablen nicht im Stack abgespeichert, sondern liegen im **Heap**. Dieser kann im Vergleich zum Stack so groß werden wie der Arbeitsspeicher und gleicht mehr einer FIFO-Liste (First In First Out), da der Speicher von alten Variablen, die nicht mehr Gültigkeit haben frei gegeben wird.

Heap
(Variablen)

Stack
(Funktionsaufrufe)

HÖRSAMFRAGE

FRAGEN?



DALL-E 2: A psychedelic DJ with a question mark for a head