Funktionen und Rekursion

D. Komm, J. Závodný
29.01.2020

Funktionen

Fakultät: $n! = n * (n - 1) * (n - 2) * \ldots * 3 * 2 * 1$

Fakultät

Fakultät: $n! = n * (n - 1) * (n - 2) * \ldots * 3 * 2 * 1$

```
def fakultaet(n):
  result = 1
  for k in range(1, n+1):
    result = result * k
  return result
```

Fakultät

Fakultät: $n! = n * (n - 1) * (n - 2) * \ldots * 3 * 2 * 1$

```
def fakultaet(n):
  result = 1
  for k in range(1, n+1):
    result = result * k
  return result

result = 4
print fakultaet(5)
print result
```

Fakultät

Fakultät: $n! = n * (n - 1) * (n - 2) * \ldots * 3 * 2 * 1$

```
def fakultaet(n):
  result = 1
  for k in range(1, n+1):
    result = result * k
  return result

result = 4
print fakultaet(5)
print result
```

120

4

Fakultät

Fakultät: $n! = n * (n - 1) * (n - 2) * \ldots * 3 * 2 * 1$

```
def fakultaet(n):
  result = 1
  for k in range(1, n+1):
    result = result * k
  return result

result = 4
print fakultaet(5)
print result
```

```
result: 4
```

Fakultät

Fakultät: $n! = n * (n - 1) * (n - 2) * \ldots * 3 * 2 * 1$

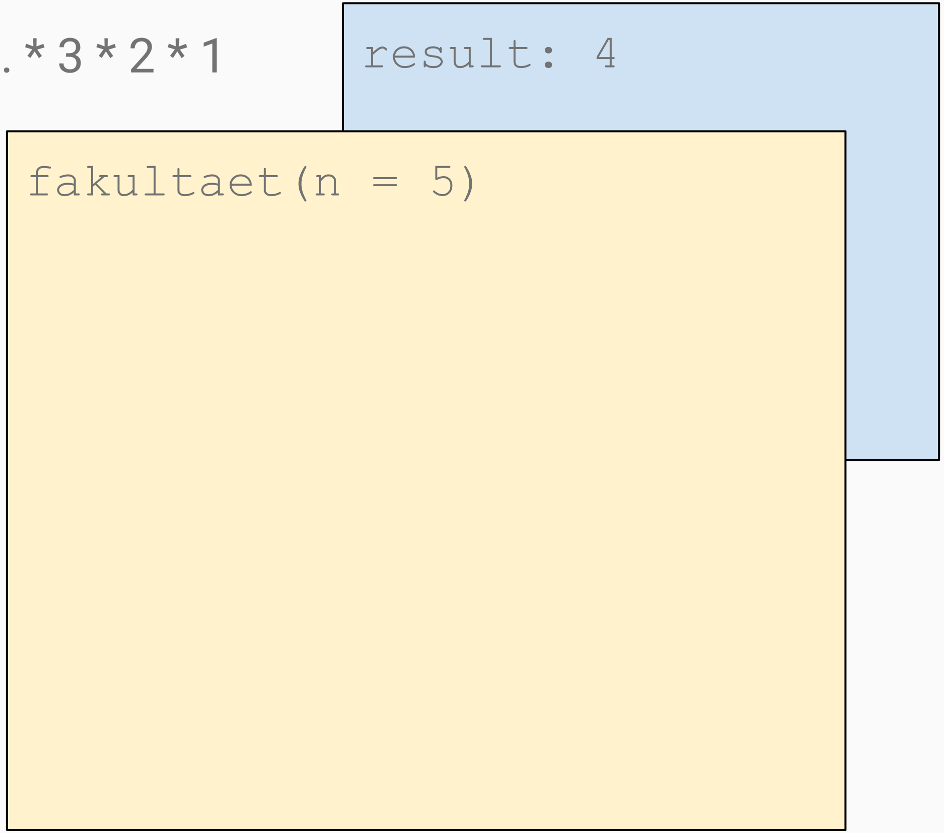
```
def fakultaet(n):
  result = 1
  for k in range(1, n+1):
    result = result * k
  return result

result = 4
print fakultaet(5)
print result
```

```
result: 4
```

Fakultät

Fakultät: $n! = n * (n - 1) * (n - 2) * \ldots * 3 * 2 * 1$

```
def fakultaet(n):
  result = 1
  for k in range(1, n+1):
    result = result * k
  return result

result = 4
print fakultaet(5)
print result
```



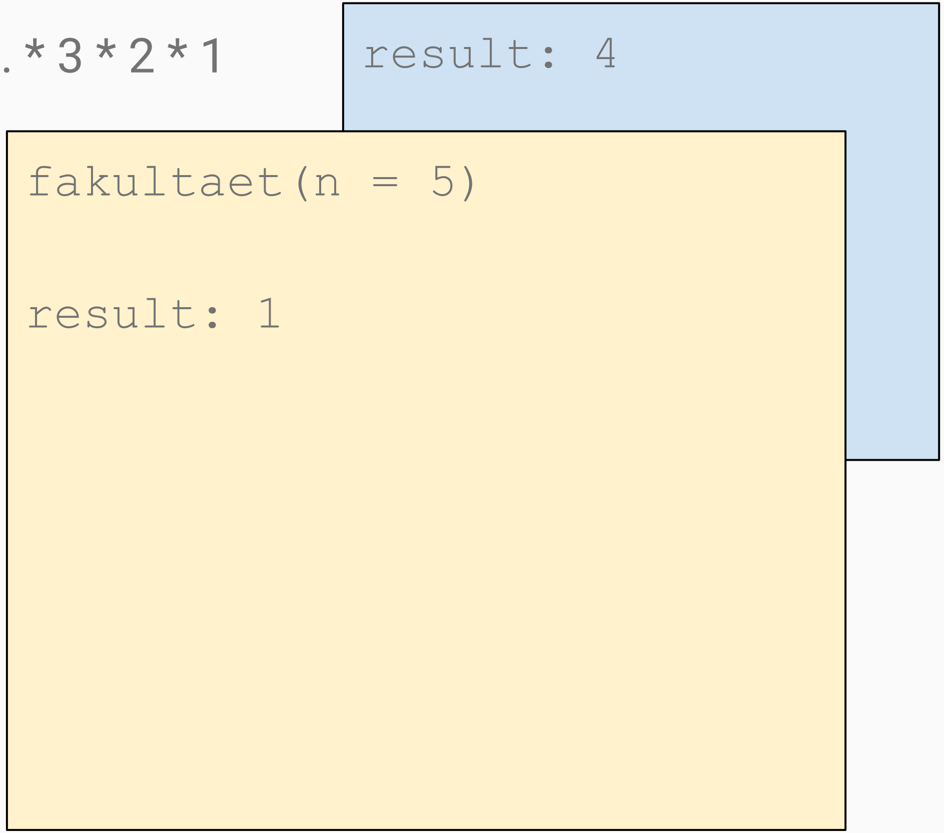
```
result: 4
```

```
fakultaet(n = 5)
```

Fakultät

Fakultät: n! = n * (n - 1) * (n - 2) * … * 3 * 2 * 1

```
def fakultaet(n):
  result = 1
  for k in range(1, n+1):
    result = result * k
  return result

result = 4
print fakultaet(5)
print result
```



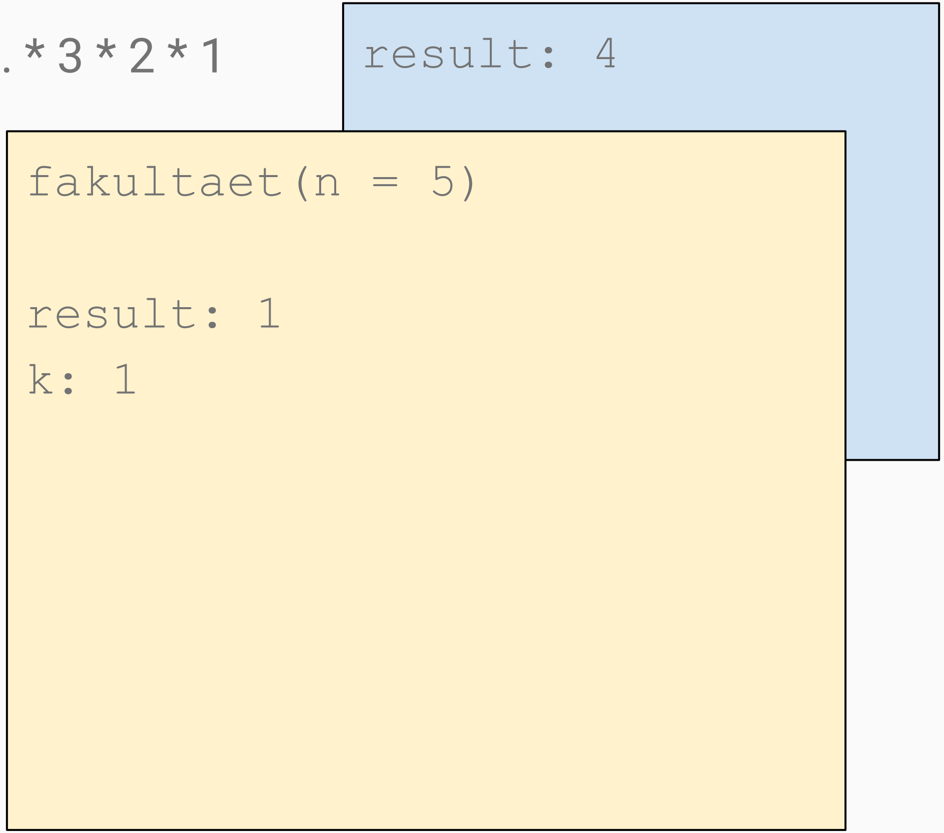
```
result: 4
```

```
fakultaet(n = 5)

result: 1
```

Fakultät

Fakultät: $n! = n * (n - 1) * (n - 2) * \ldots * 3 * 2 * 1$

```
def fakultaet(n):
  result = 1
  for k in range(1, n+1):
    result = result * k
  return result

result = 4
print fakultaet(5)
print result
```



result: 4

fakultaet(n = 5)

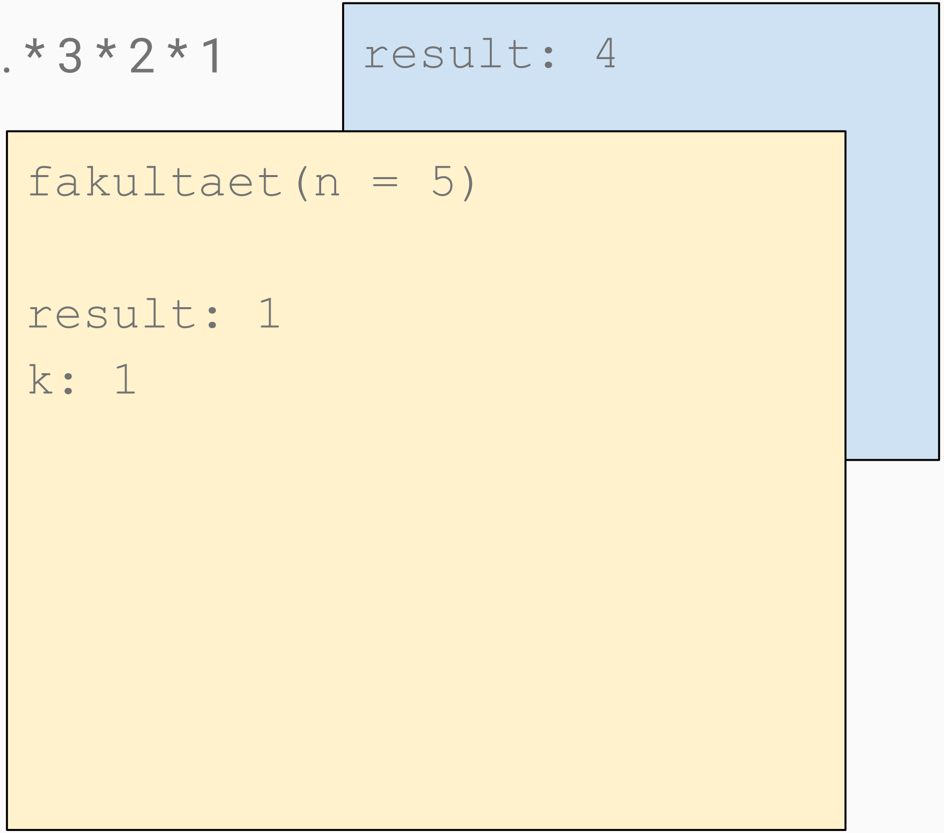
result: 1

k: 1

Fakultät

Fakultät: $n! = n * (n - 1) * (n - 2) * \ldots * 3 * 2 * 1$

```
def fakultaet(n):
  result = 1
  for k in range(1, n+1):
    result = result * k
  return result

result = 4
print fakultaet(5)
print result
```



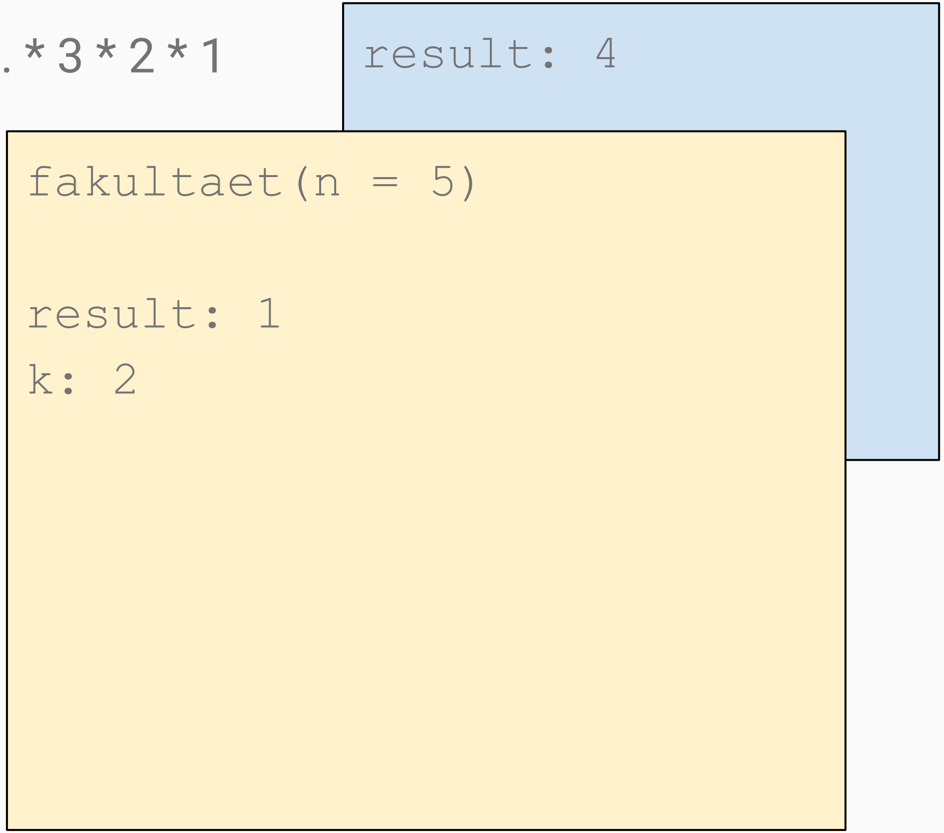
```
result: 4
```

```
fakultaet(n = 5)

result: 1
k: 1
```

Fakultät

Fakultät: $n! = n * (n - 1) * (n - 2) * \ldots * 3 * 2 * 1$

```
def fakultaet(n):
  result = 1
  for k in range(1, n+1):
    result = result * k
  return result

result = 4
print fakultaet(5)
print result
```



result: 4

fakultaet(n = 5)

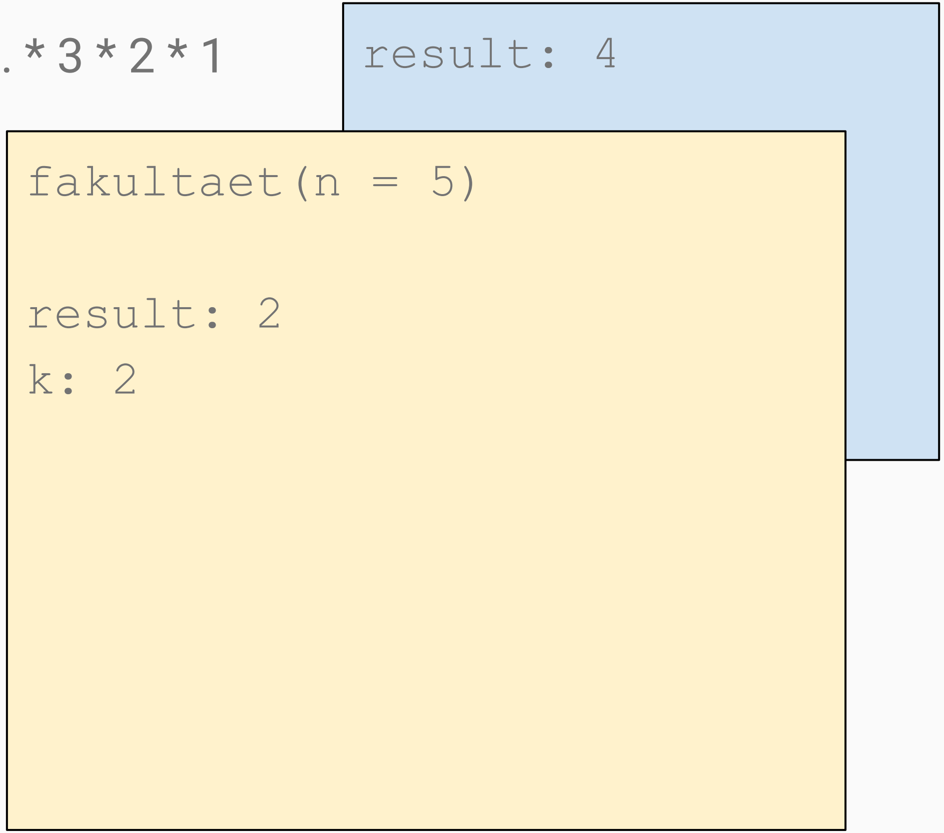
result: 1

k: 2

Fakultät

Fakultät: $n! = n * (n - 1) * (n - 2) * \ldots * 3 * 2 * 1$

```
def fakultaet(n):
  result = 1
  for k in range(1, n+1):
    result = result * k
  return result

result = 4
print fakultaet(5)
print result
```



result: 4

fakultaet(n = 5)

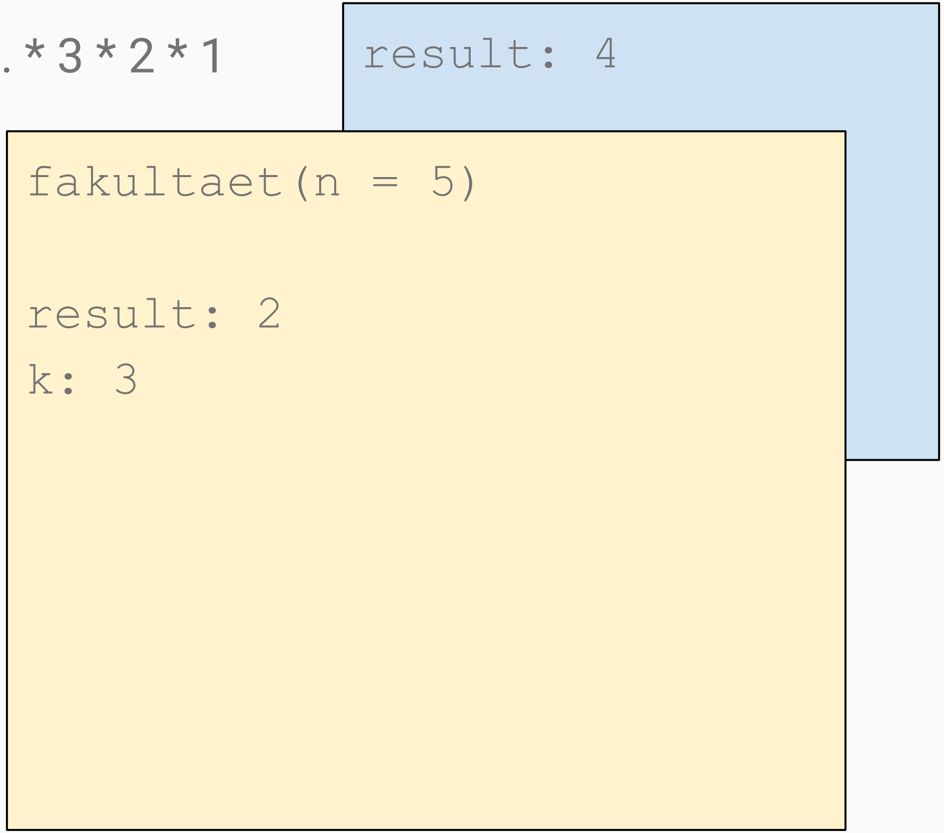
result: 2

k: 2

Fakultät

Fakultät: $n! = n * (n - 1) * (n - 2) * \ldots * 3 * 2 * 1$

```
def fakultaet(n):
  result = 1
  for k in range(1, n+1):
    result = result * k
  return result

result = 4
print fakultaet(5)
print result
```



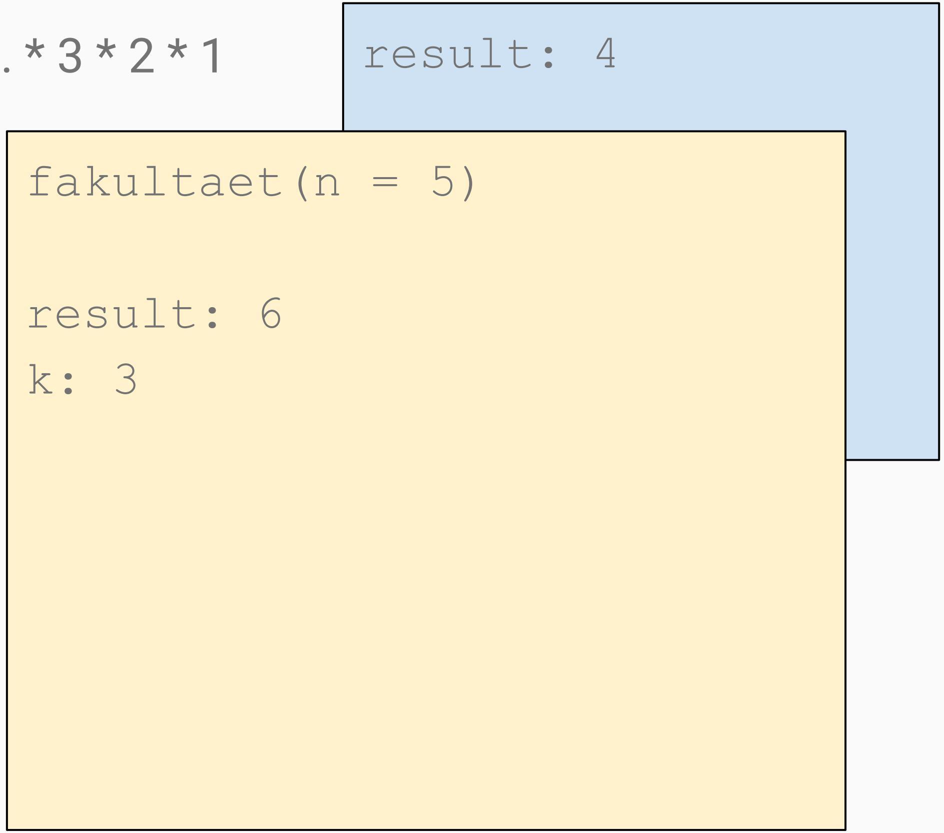
```
result: 4
```

```
fakultaet(n = 5)

result: 2
k: 3
```

Fakultät

Fakultät: $n! = n * (n - 1) * (n - 2) * \ldots * 3 * 2 * 1$

```
def fakultaet(n):
  result = 1
  for k in range(1, n+1):
    result = result * k
  return result

result = 4
print fakultaet(5)
print result
```



result: 4

fakultaet(n = 5)

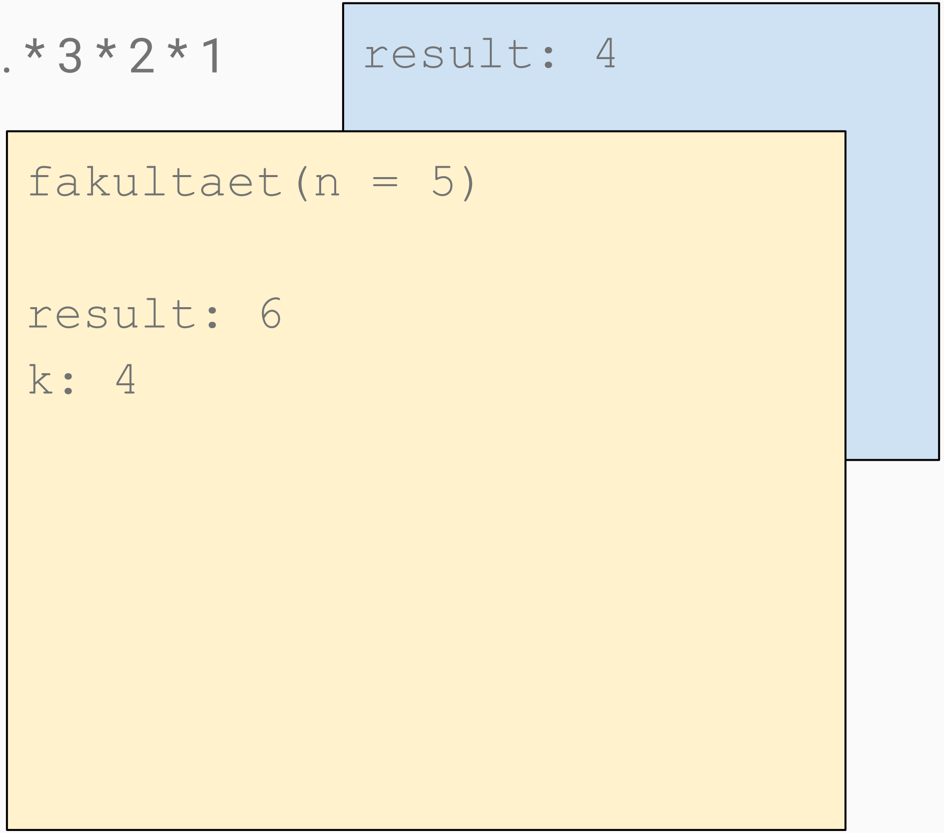
result: 6

k: 3

Fakultät

Fakultät: $n! = n * (n - 1) * (n - 2) * \ldots * 3 * 2 * 1$

```
def fakultaet(n):
  result = 1
  for k in range(1, n+1):
    result = result * k
  return result

result = 4
print fakultaet(5)
print result
```



result: 4

fakultaet(n = 5)

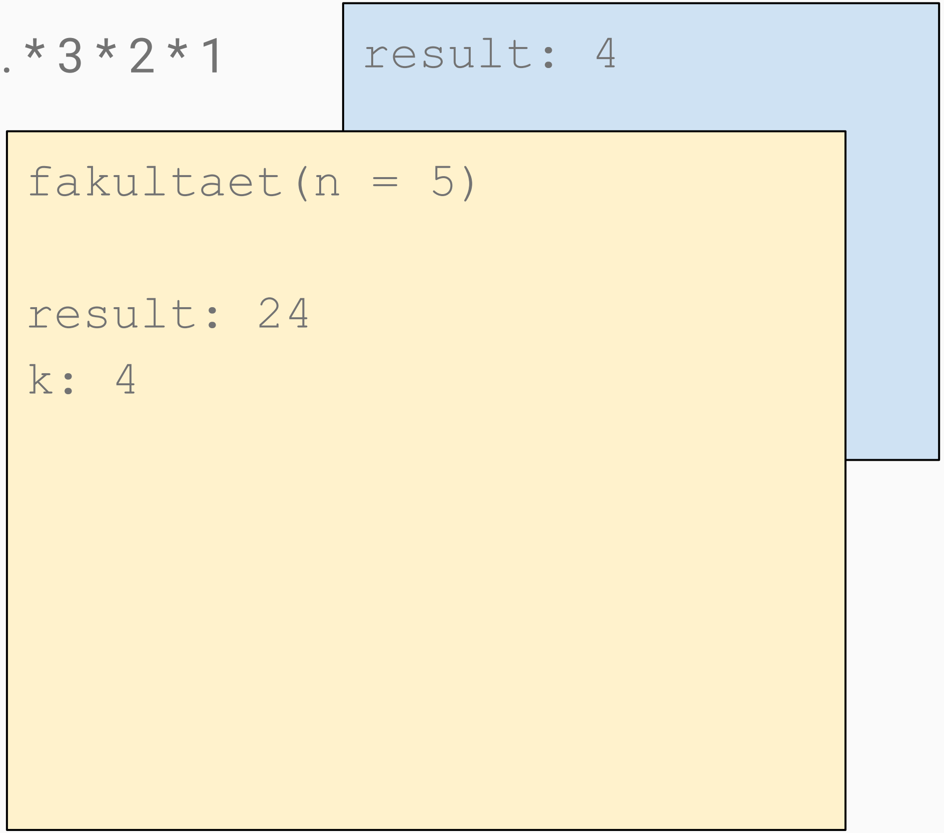
result: 6

k: 4

Fakultät

Fakultät: n! = n * (n - 1) * (n - 2) * … * 3 * 2 * 1

```
def fakultaet(n):
  result = 1
  for k in range(1, n+1):
    result = result * k
  return result

result = 4
print fakultaet(5)
print result
```



result: 4

fakultaet(n = 5)

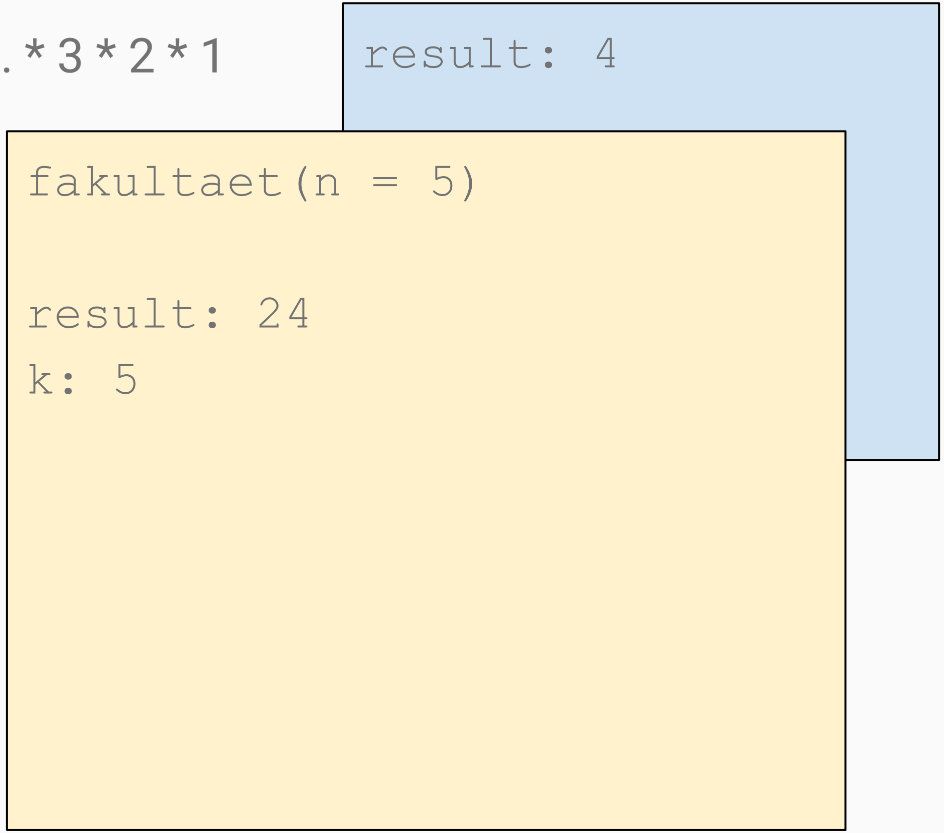
result: 24

k: 4

Fakultät

Fakultät: n! = n * (n - 1) * (n - 2) * … * 3 * 2 * 1

```
def fakultaet(n):
  result = 1
  for k in range(1, n+1):
    result = result * k
  return result

result = 4
print fakultaet(5)
print result
```



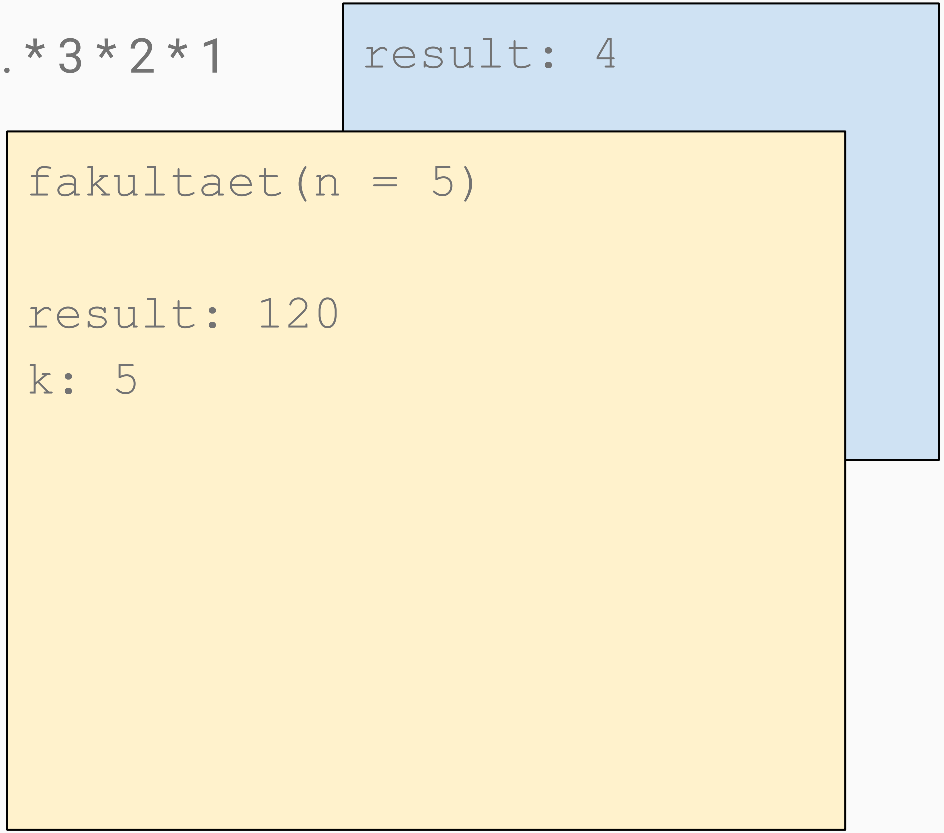
```
result: 4
```

```
fakultaet(n = 5)

result: 24
k: 5
```

Fakultät

Fakultät: $n! = n * (n - 1) * (n - 2) * \dots * 3 * 2 * 1$

```
def fakultaet(n):
  result = 1
  for k in range(1, n+1):
    result = result * k
  return result

result = 4
print fakultaet(5)
print result
```



result: 4

fakultaet(n = 5)

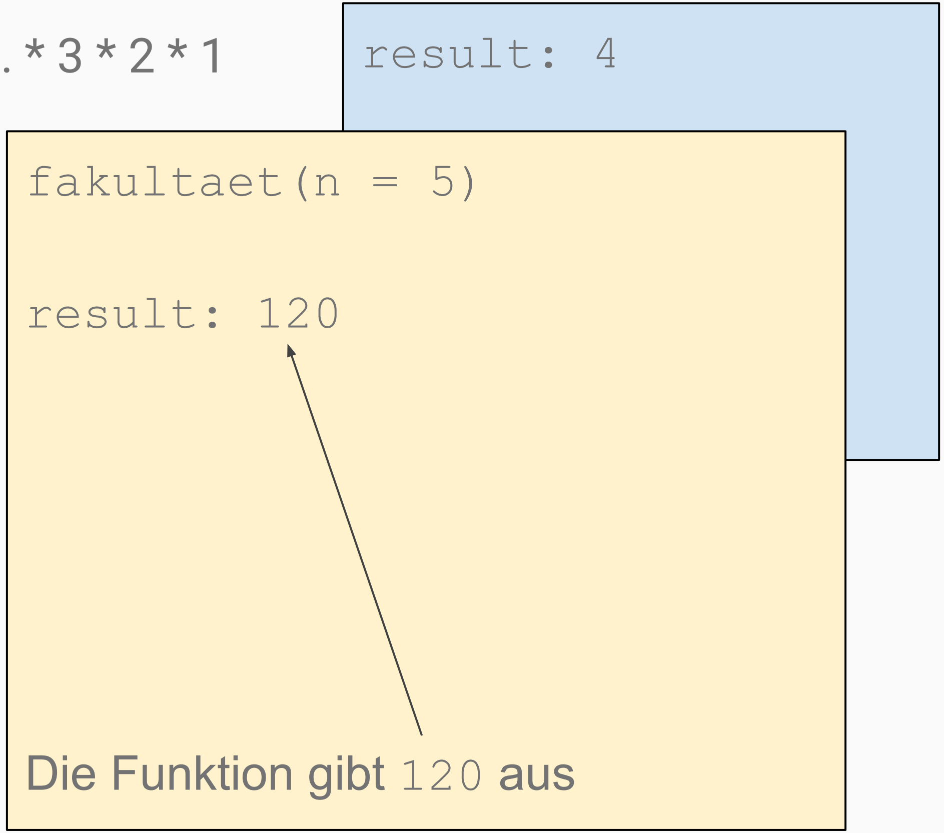
result: 120

k: 5

Fakultät

Fakultät: $n! = n * (n - 1) * (n - 2) * \ldots * 3 * 2 * 1$

```
def fakultaet(n):
  result = 1
  for k in range(1, n+1):
    result = result * k
  return result

result = 4
print fakultaet(5)
print result
```



result: 4

fakultaet(n = 5)

result: 120

Die Funktion gibt 120 aus

Fakultät

Fakultät: $n! = n * (n - 1) * (n - 2) * \ldots * 3 * 2 * 1$

```
result: 4
```

```
def fakultaet(n):
  result = 1
  for k in range(1, n+1):
    result = result * k
  return result

result = 4
print fakultaet(5)
print result
```

druckt 120 aus

Fakultät

Fakultät: $n! = n * (n - 1) * (n - 2) * \ldots * 3 * 2 * 1$

```
def fakultaet(n):
  result = 1
  for k in range(1, n+1):
    result = result * k
  return result

result = 4
print fakultaet(5)
print result
```

```
result: 4
```

druckt 120 aus

druckt 4 aus

Fakultät: Rekursion

Fakultät: $n! = n * (n - 1) * (n - 2) * \ldots * 3 * 2 * 1$

```
def fakultaet(n):
  result = 1
  for k in range(1, n+1):
    result = result * k
  return result
```

Fakultät: Rekursion

Fakultät: $$n! = n * \underbrace{(n - 1) * (n - 2) * \ldots * 3 * 2 * 1}_{(n - 1)!}$$

```
def fakultaet(n):
  result = 1
  for k in range(1, n+1):
    result = result * k
  return result
```

Fakultät: Rekursion

Fakultät: $n! = n * \underbrace{(n - 1) * (n - 2) * \dots * 3 * 2 * 1}_{(n - 1)!}$

```
def fakultaet(n):
  return n * fakultaet(n - 1)
```

Fakultät: Rekursion

Fakultät: $n! = n * \underbrace{(n - 1) * (n - 2) * \ldots * 3 * 2 * 1}_{(n - 1)!}$

```
def fakultaet(n):
  if n > 1:
    return n * fakultaet(n - 1)
  else:
    return 1
```

Fakultät: Rekursion

```
> fakultaet(5)
```

```
fakultaet(n = 5)
```

```
def fakultaet(n):
  if n > 1:
    return n * fakultaet(n - 1)
  else:
    return 1
```

Fakultät: Rekursion

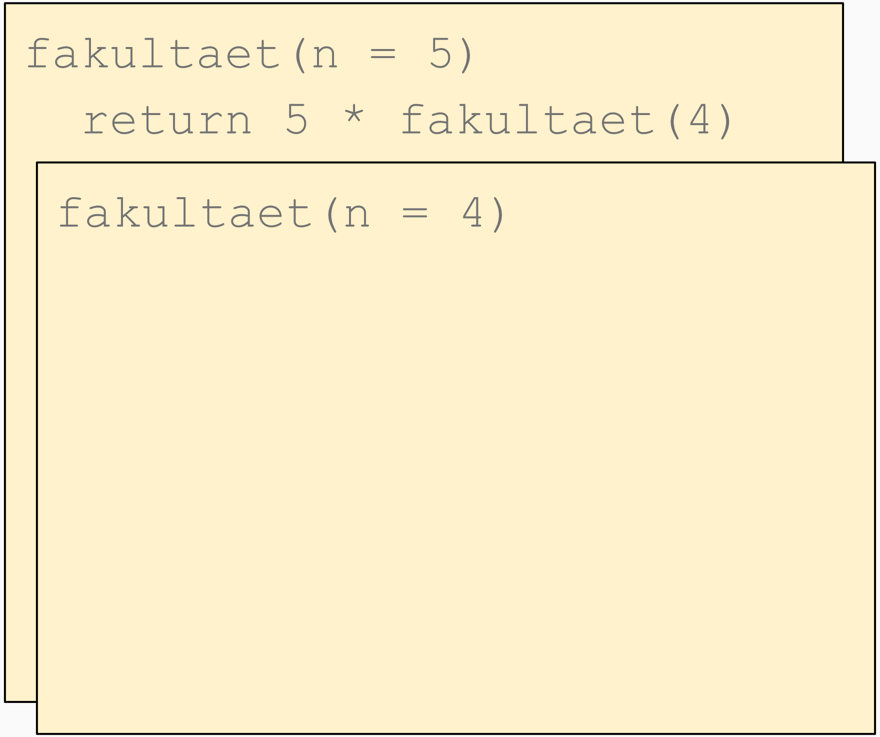
```
> fakultaet(5)
```

```
fakultaet(n = 5)
  return 5 * fakultaet(4)
```

```
def fakultaet(n):
  if n > 1:
    return n * fakultaet(n - 1)
  else:
    return 1
```

Fakultät: Rekursion

```
> fakultaet(5)
```



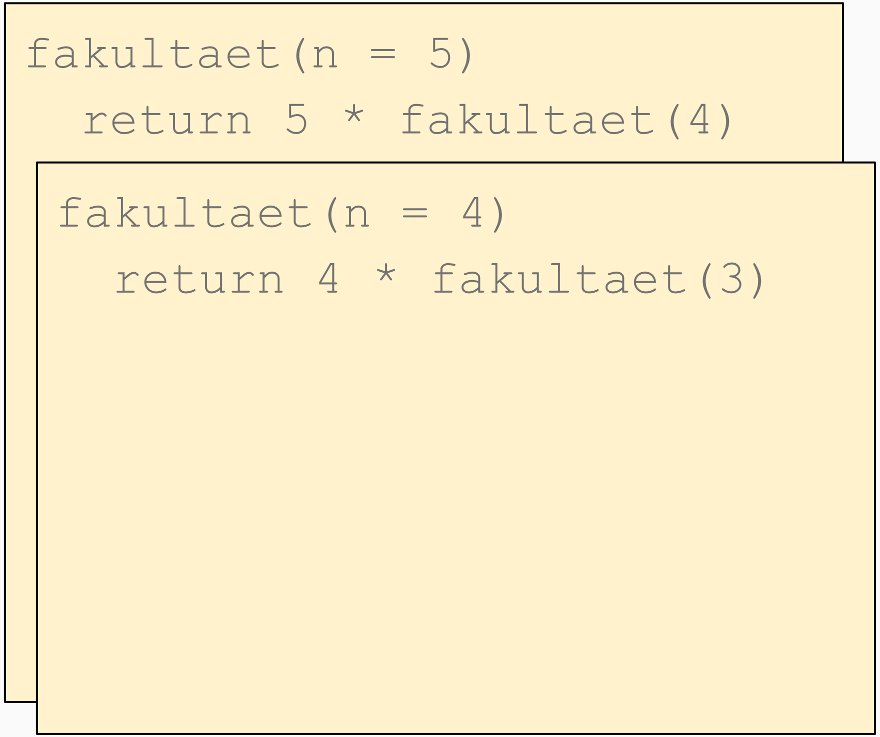
```
fakultaet(n = 5)
  return 5 * fakultaet(4)
```

```
fakultaet(n = 4)
```

```
def fakultaet(n):
  if n > 1:
    return n * fakultaet(n - 1)
  else:
    return 1
```

Fakultät: Rekursion

```
> fakultaet(5)
```



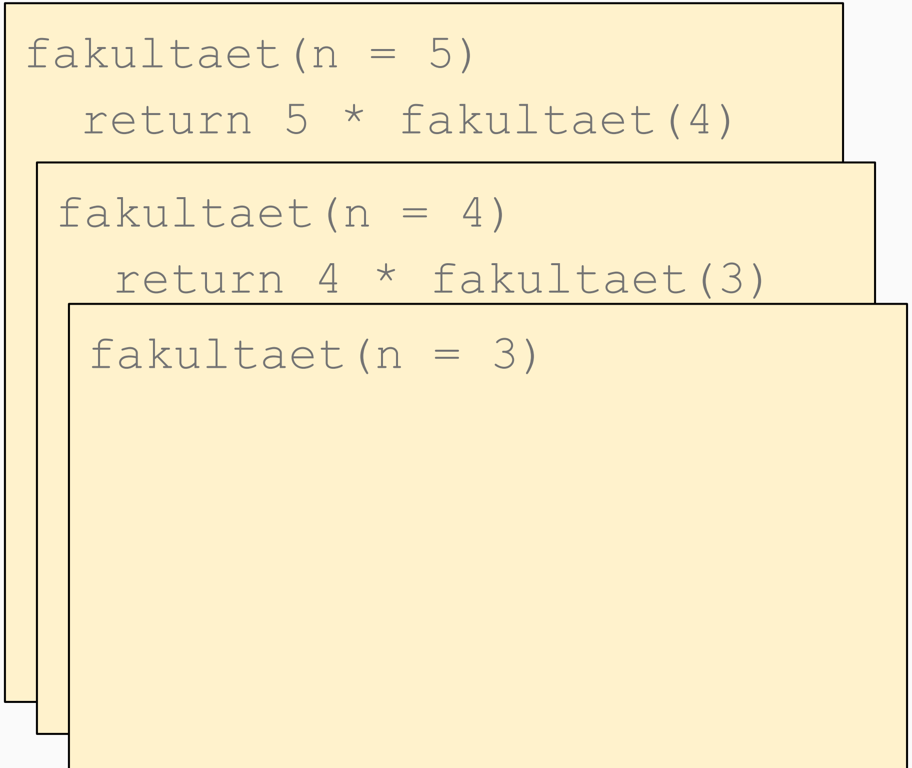
```
fakultaet(n = 5)
  return 5 * fakultaet(4)
```

```
fakultaet(n = 4)
  return 4 * fakultaet(3)
```

```
def fakultaet(n):
  if n > 1:
    return n * fakultaet(n - 1)
  else:
    return 1
```

Fakultät: Rekursion

```
> fakultaet(5)
```



```
fakultaet(n = 5)
  return 5 * fakultaet(4)
```

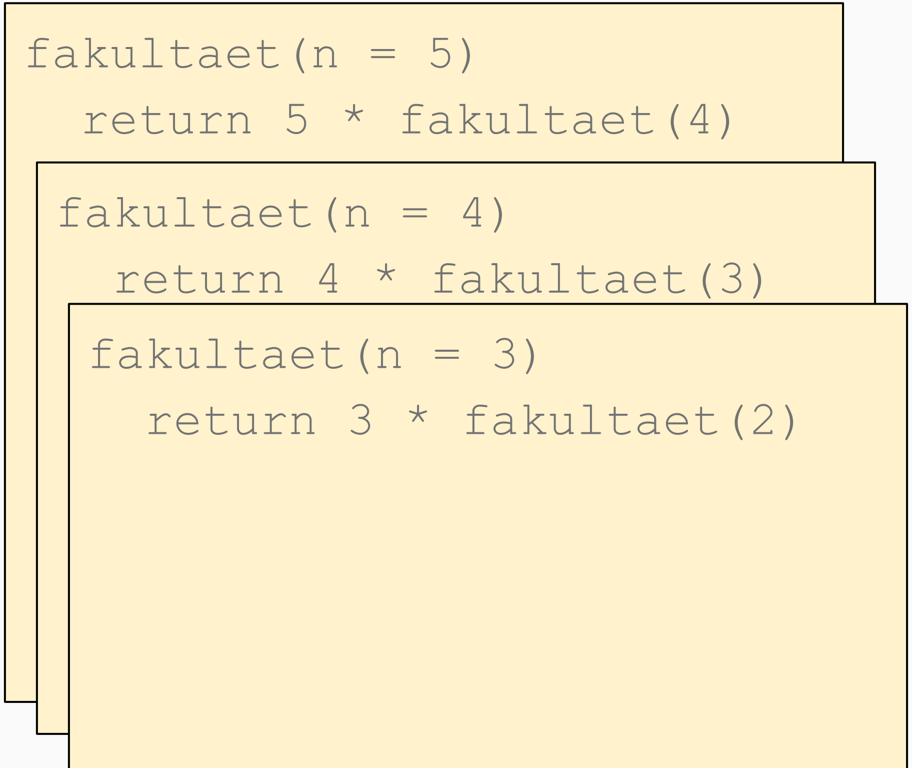
```
fakultaet(n = 4)
  return 4 * fakultaet(3)
```

```
fakultaet(n = 3)
```

```
def fakultaet(n):
  if n > 1:
    return n * fakultaet(n - 1)
  else:
    return 1
```

Fakultät: Rekursion

```
> fakultaet(5)
```



```
fakultaet(n = 5)
  return 5 * fakultaet(4)
```

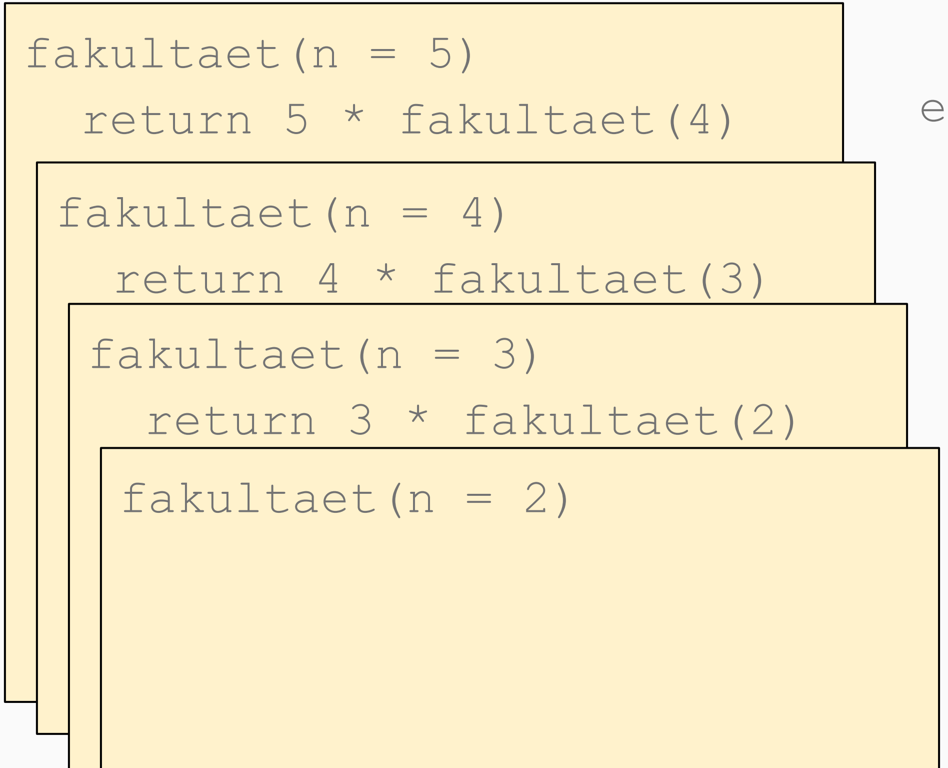
```
fakultaet(n = 4)
  return 4 * fakultaet(3)
```

```
fakultaet(n = 3)
  return 3 * fakultaet(2)
```

```
def fakultaet(n):
  if n > 1:
    return n * fakultaet(n - 1)
  else:
    return 1
```

Fakultät: Rekursion

```
> fakultaet(5)
```



```
fakultaet(n = 5)
  return 5 * fakultaet(4)
```

```
fakultaet(n = 4)
  return 4 * fakultaet(3)
```

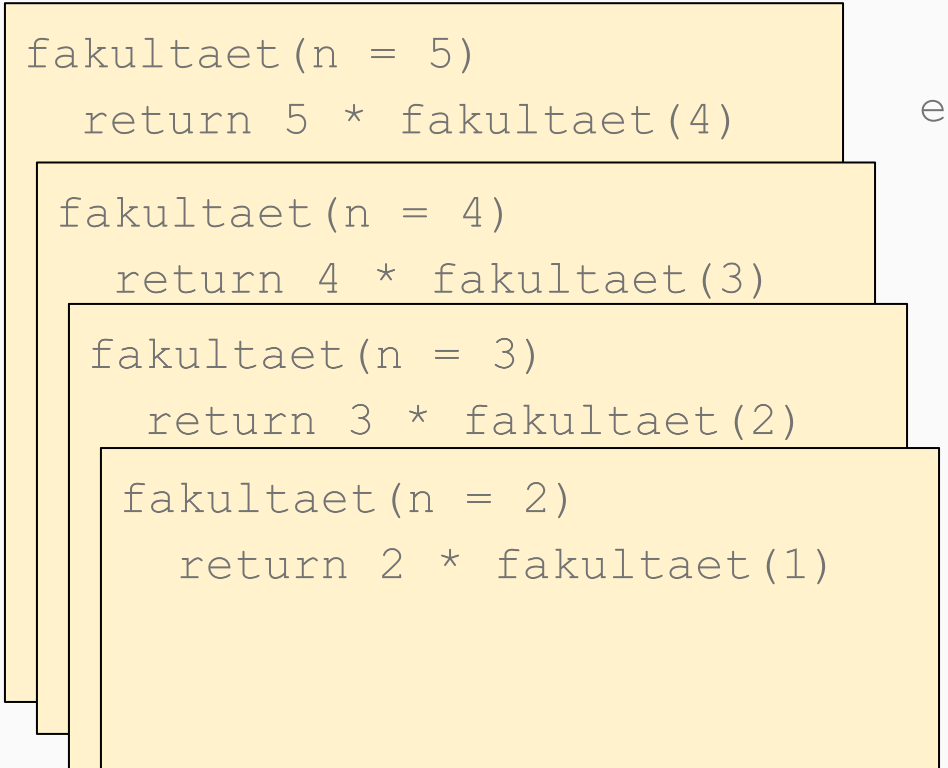
```
fakultaet(n = 3)
  return 3 * fakultaet(2)
```

```
fakultaet(n = 2)
```

```
def fakultaet(n):
  if n > 1:
    return n * fakultaet(n - 1)
  else:
    return 1
```

Fakultät: Rekursion

```
> fakultaet(5)
```



```
fakultaet(n = 5)
  return 5 * fakultaet(4)
```

```
fakultaet(n = 4)
  return 4 * fakultaet(3)
```

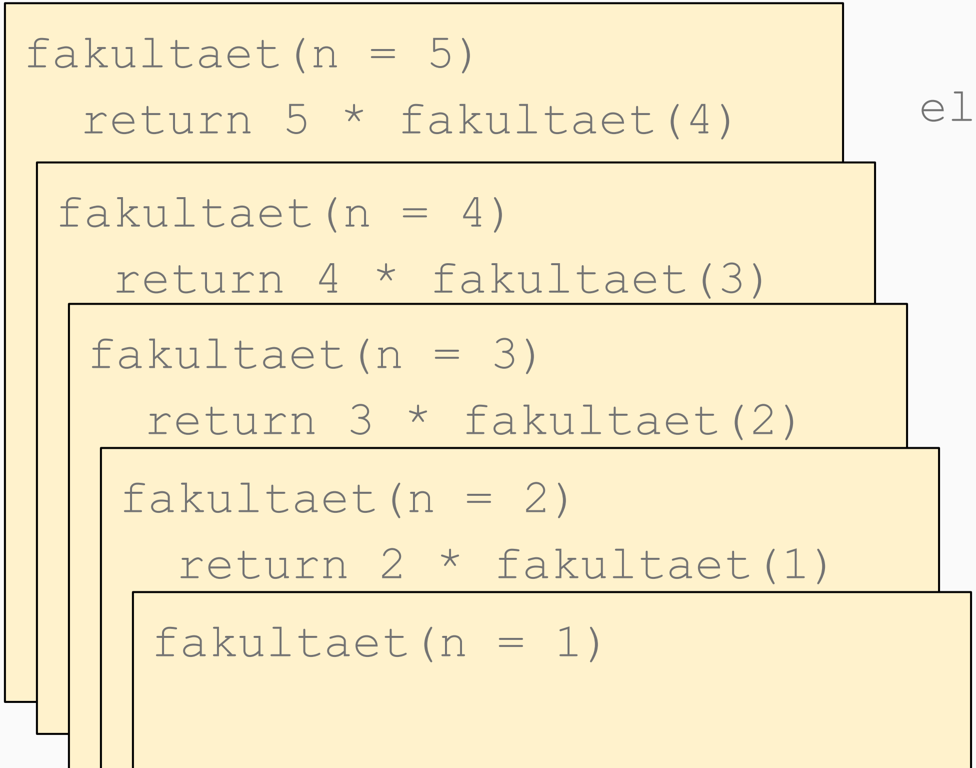
```
fakultaet(n = 3)
  return 3 * fakultaet(2)
```

```
fakultaet(n = 2)
  return 2 * fakultaet(1)
```

```
def fakultaet(n):
  if n > 1:
    return n * fakultaet(n - 1)
  else:
    return 1
```

Fakultät: Rekursion

```
> fakultaet(5)
```

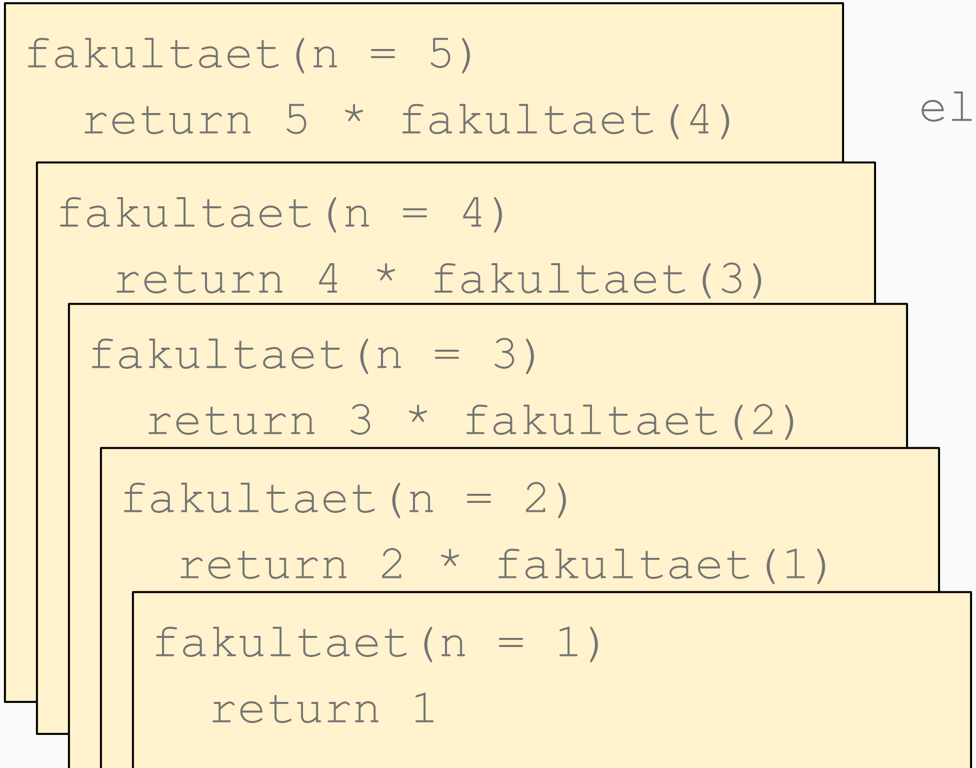


```
fakultaet(n = 5)
  return 5 * fakultaet(4)
fakultaet(n = 4)
  return 4 * fakultaet(3)
fakultaet(n = 3)
  return 3 * fakultaet(2)
fakultaet(n = 2)
  return 2 * fakultaet(1)
fakultaet(n = 1)
```

```
def fakultaet(n):
  if n > 1:
    return n * fakultaet(n - 1)
  else:
    return 1
```

Fakultät: Rekursion

```
> fakultaet(5)
```



```
fakultaet(n = 5)
  return 5 * fakultaet(4)
```

```
fakultaet(n = 4)
  return 4 * fakultaet(3)
```

```
fakultaet(n = 3)
  return 3 * fakultaet(2)
```

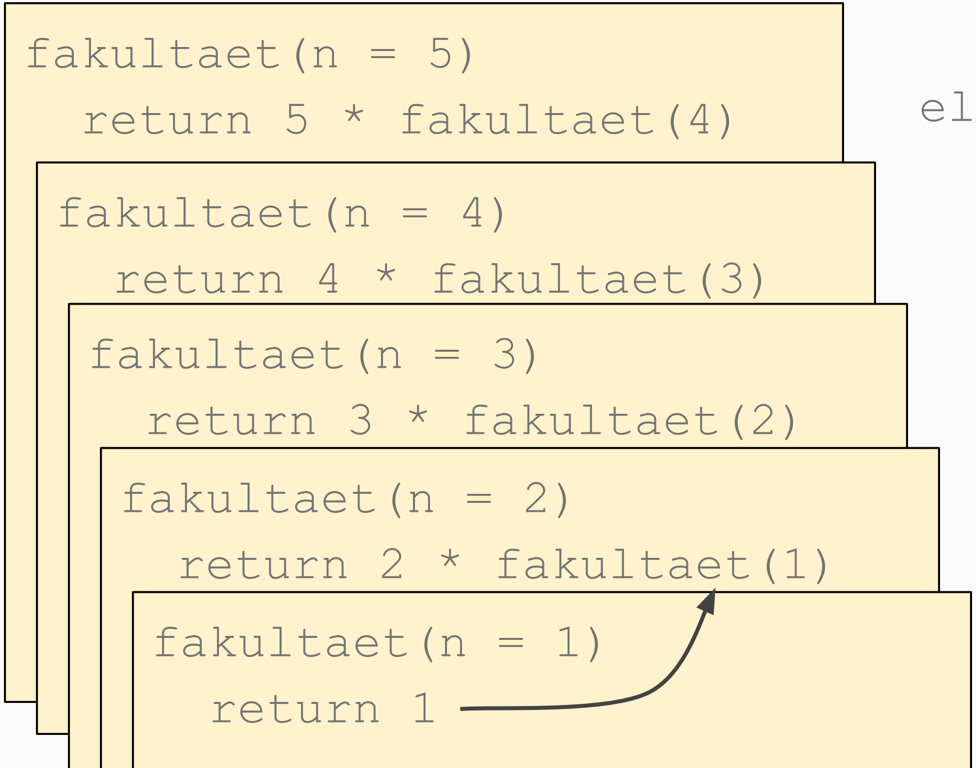
```
fakultaet(n = 2)
  return 2 * fakultaet(1)
```

```
fakultaet(n = 1)
  return 1
```

```
def fakultaet(n):
  if n > 1:
    return n * fakultaet(n - 1)
  else:
    return 1
```

Fakultät: Rekursion

```
> fakultaet(5)
```

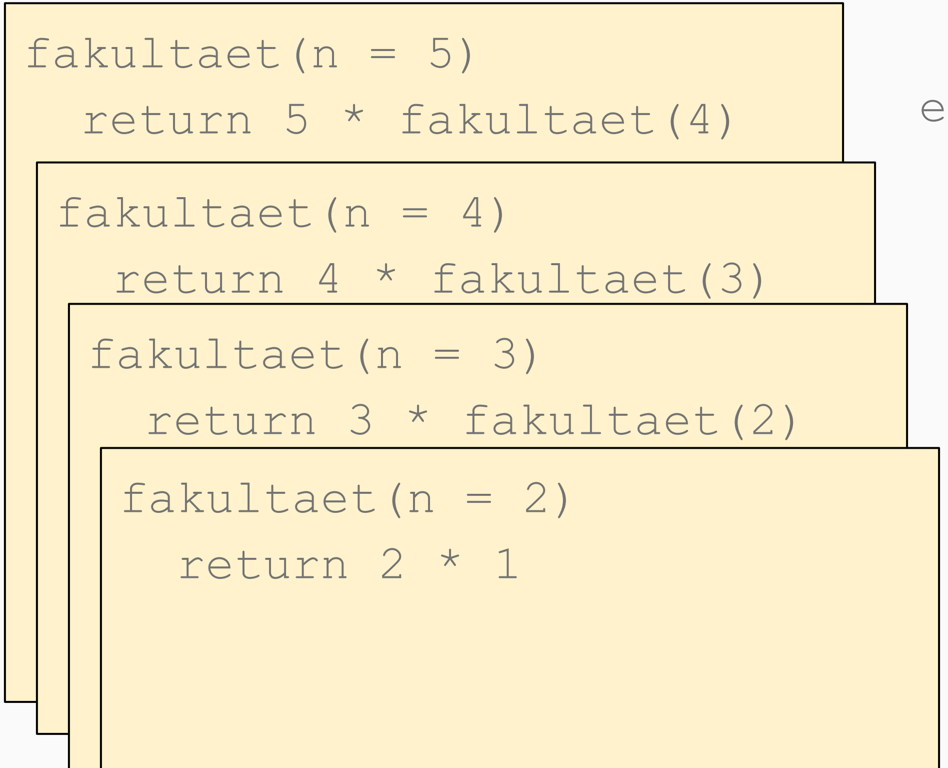


```
fakultaet(n = 5)
  return 5 * fakultaet(4)
fakultaet(n = 4)
  return 4 * fakultaet(3)
fakultaet(n = 3)
  return 3 * fakultaet(2)
fakultaet(n = 2)
  return 2 * fakultaet(1)
fakultaet(n = 1)
  return 1
```

```
def fakultaet(n):
  if n > 1:
    return n * fakultaet(n - 1)
  else:
    return 1
```

Fakultät: Rekursion

```
> fakultaet(5)
```



```
fakultaet(n = 5)
  return 5 * fakultaet(4)
```

```
fakultaet(n = 4)
  return 4 * fakultaet(3)
```

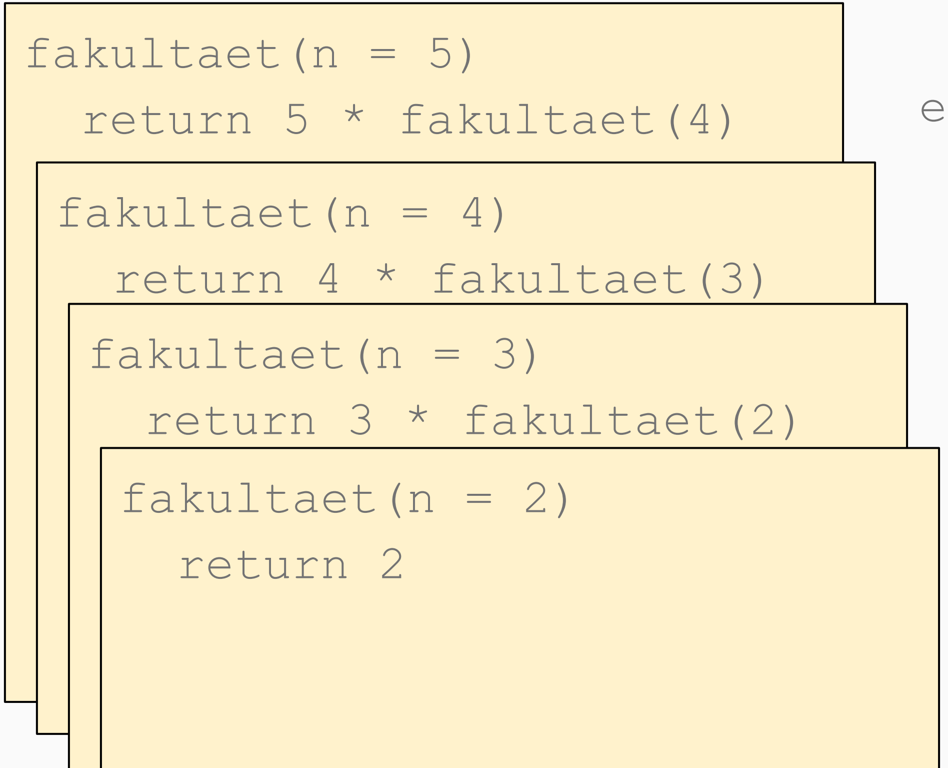
```
fakultaet(n = 3)
  return 3 * fakultaet(2)
```

```
fakultaet(n = 2)
  return 2 * 1
```

```
def fakultaet(n):
  if n > 1:
    return n * fakultaet(n - 1)
  else:
    return 1
```

Fakultät: Rekursion

```
> fakultaet(5)
```



```
fakultaet(n = 5)
  return 5 * fakultaet(4)
```

```
fakultaet(n = 4)
  return 4 * fakultaet(3)
```

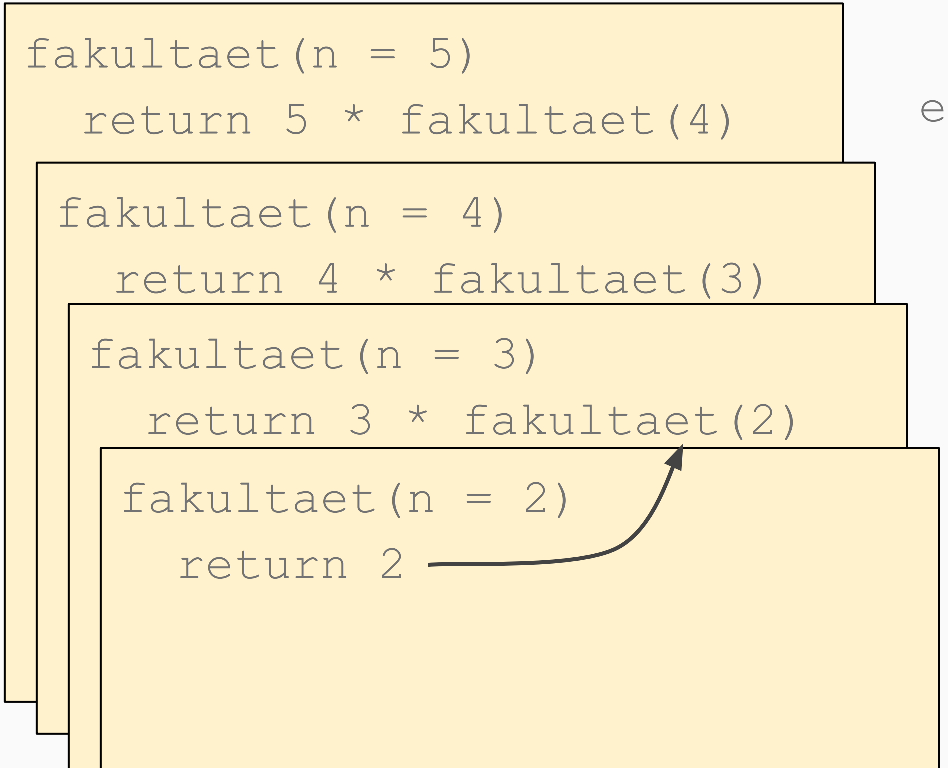
```
fakultaet(n = 3)
  return 3 * fakultaet(2)
```

```
fakultaet(n = 2)
  return 2
```

```
def fakultaet(n):
  if n > 1:
    return n * fakultaet(n - 1)
  else:
    return 1
```

Fakultät: Rekursion

```
> fakultaet(5)
```



```
fakultaet(n = 5)
  return 5 * fakultaet(4)
```

```
fakultaet(n = 4)
  return 4 * fakultaet(3)
```

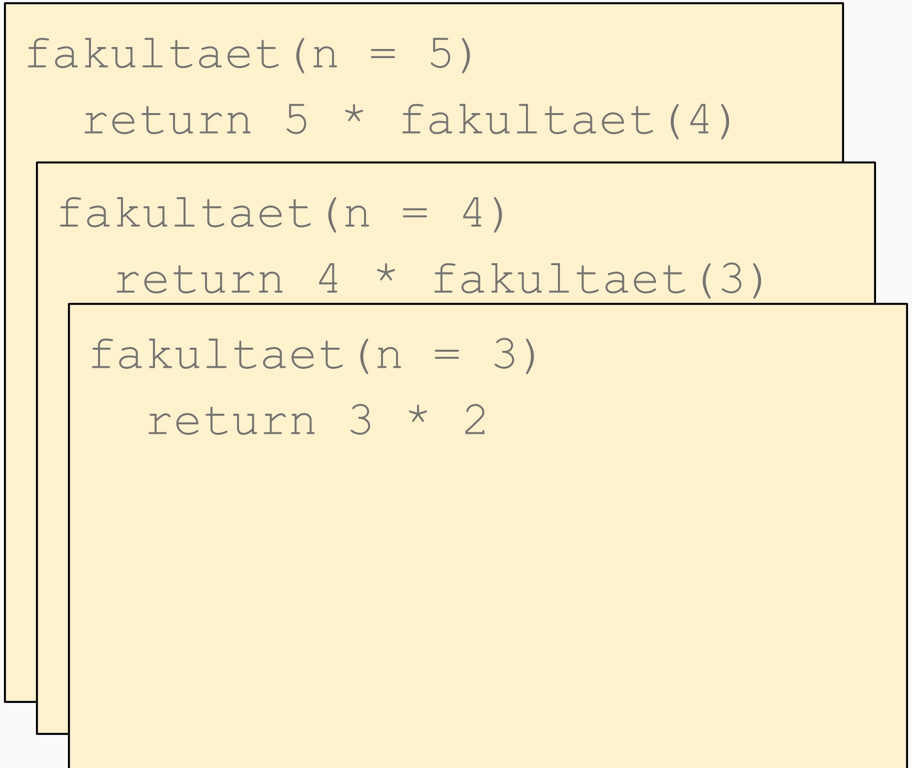
```
fakultaet(n = 3)
  return 3 * fakultaet(2)
```

```
fakultaet(n = 2)
  return 2
```

```
def fakultaet(n):
  if n > 1:
    return n * fakultaet(n - 1)
  else:
    return 1
```

Fakultät: Rekursion

```
> fakultaet(5)
```



```
fakultaet(n = 5)
  return 5 * fakultaet(4)
```

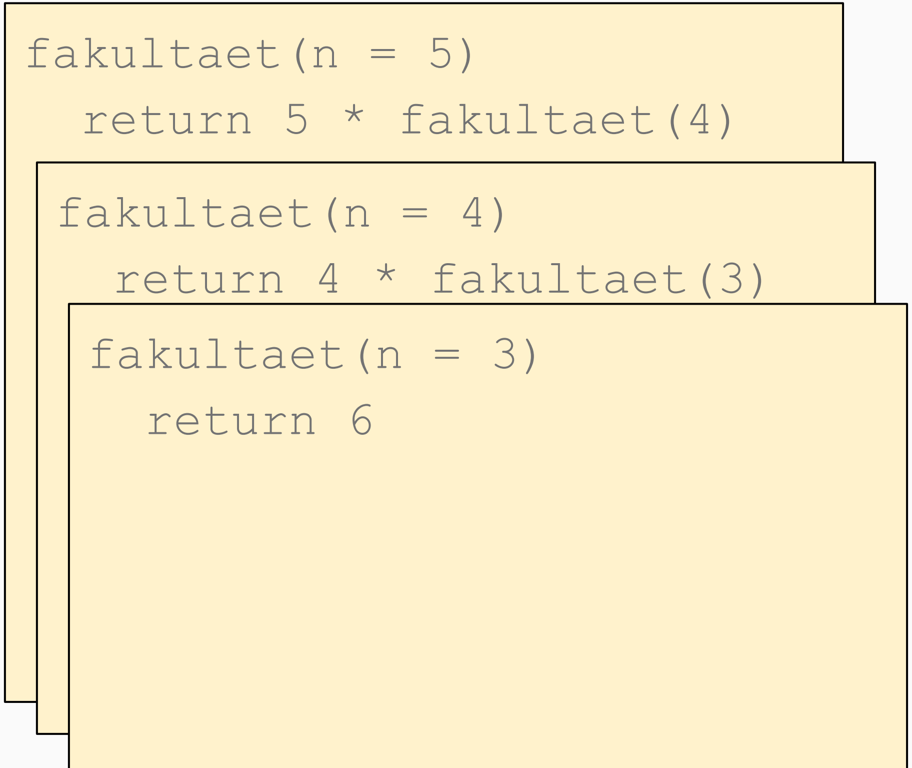
```
fakultaet(n = 4)
  return 4 * fakultaet(3)
```

```
fakultaet(n = 3)
  return 3 * 2
```

```
def fakultaet(n):
  if n > 1:
    return n * fakultaet(n - 1)
  else:
    return 1
```

Fakultät: Rekursion

```
> fakultaet(5)
```



```
fakultaet(n = 5)
  return 5 * fakultaet(4)
```

```
fakultaet(n = 4)
  return 4 * fakultaet(3)
```

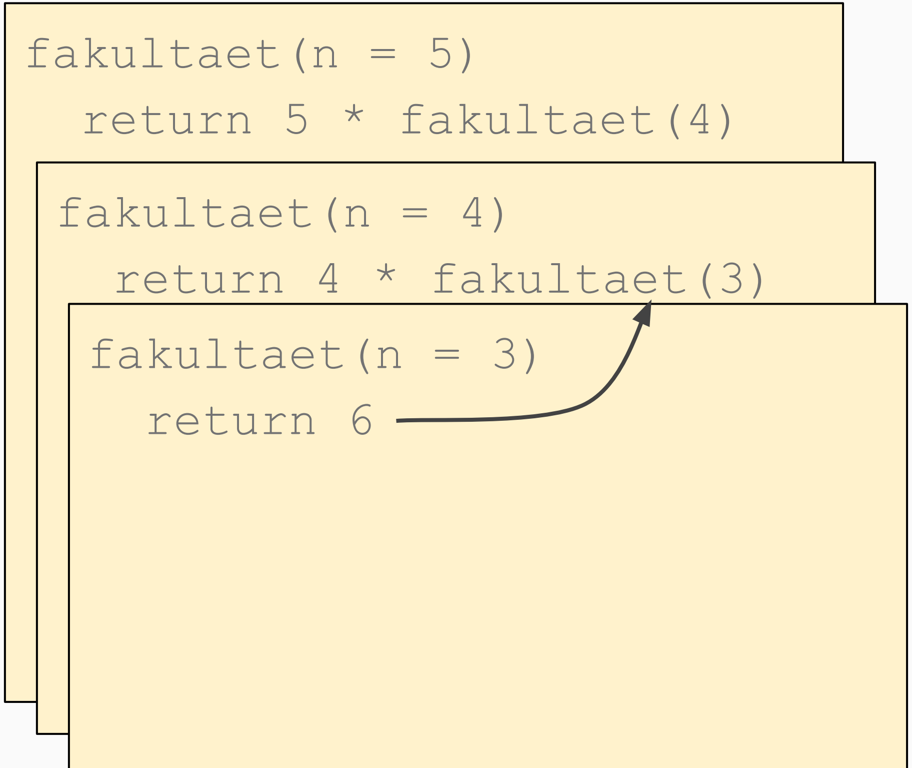
```
fakultaet(n = 3)
  return 6
```

```
def fakultaet(n):
  if n > 1:
    return n * fakultaet(n - 1)
  else:
    return 1
```

Fakultät: Rekursion

```
> fakultaet(5)
```

```
def fakultaet(n):
  if n > 1:
    return n * fakultaet(n - 1)
  else:
    return 1
```



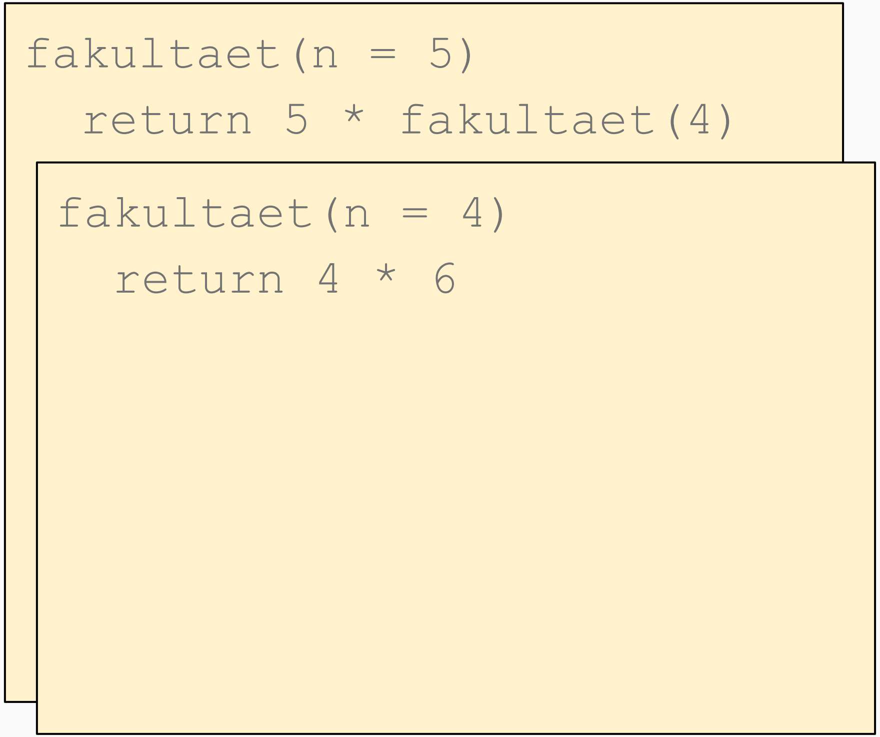
```
fakultaet(n = 5)
  return 5 * fakultaet(4)
```

```
fakultaet(n = 4)
  return 4 * fakultaet(3)
```

```
fakultaet(n = 3)
  return 6
```

Fakultät: Rekursion

```
> fakultaet(5)
```



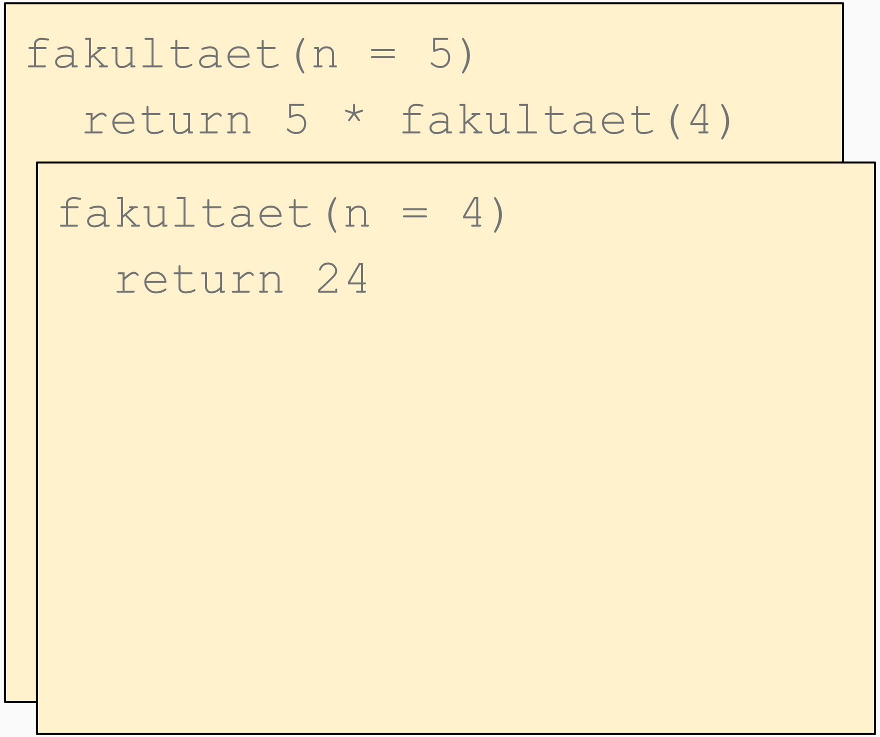
```
fakultaet(n = 5)
  return 5 * fakultaet(4)
```

```
fakultaet(n = 4)
  return 4 * 6
```

```
def fakultaet(n):
  if n > 1:
    return n * fakultaet(n - 1)
  else:
    return 1
```

Fakultät: Rekursion

```
> fakultaet(5)
```



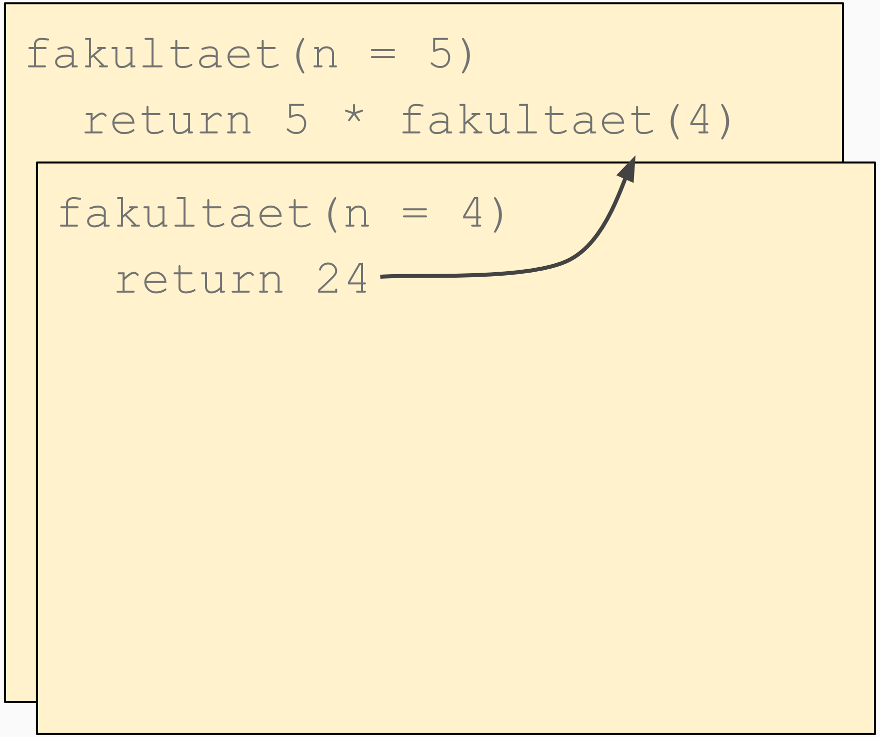
```
fakultaet(n = 5)
  return 5 * fakultaet(4)
```

```
fakultaet(n = 4)
  return 24
```

```
def fakultaet(n):
  if n > 1:
    return n * fakultaet(n - 1)
  else:
    return 1
```

Fakultät: Rekursion

```
> fakultaet(5)
```



```
fakultaet(n = 5)
  return 5 * fakultaet(4)
```

```
fakultaet(n = 4)
  return 24
```

```
def fakultaet(n):
  if n > 1:
    return n * fakultaet(n - 1)
  else:
    return 1
```

Fakultät: Rekursion

```
> fakultaet(5)
```

```
fakultaet(n = 5)
  return 5 * 24
```

```
def fakultaet(n):
  if n > 1:
    return n * fakultaet(n - 1)
  else:
    return 1
```

Fakultät: Rekursion

```
> fakultaet(5)
```

```
fakultaet(n = 5)
  return 120
```

```
def fakultaet(n):
  if n > 1:
    return n * fakultaet(n - 1)
  else:
    return 1
```

Fakultät: Rekursion

```
> fakultaet(5)
120
```

```
def fakultaet(n):
  if n > 1:
    return n * fakultaet(n - 1)
  else:
    return 1
```

Fakultät: Rekursion

Fakultät: $n! = n * (n - 1) * (n - 2) * \ldots * 3 * 2 * 1$

```
def fakultaet(n):
  if n > 1:
    return n * fakultaet(n - 1)
  else:
    return 1
```

Rekursion: Doppel-Fakultät

Doppel-Fakultät: n!! = n * (n - 2) * (n - 4) * (n - 6) * …

```
def doppel_fakultaet(n):
  ???
```

Rekursion: Doppel-Fakultät

Doppel-Fakultät: $n!! = n * (n - 2) * (n - 4) * (n - 6) * \ldots$

```
def doppel_fakultaet(n):
  if n > 1:
    return n * doppel_fakultaet(n - 2)
  else:
    return 1
```

Rekursion: Durchschnitt-Mieten

In Haus #0 ist die Miete == 1, in Haus #4 ist die Miete == 9. In den Häusern dazwischen ist die Miete der Durchschnitt der Mieten der Nachbarn. Was ist die Miete im Haus #1?

1	?	?	?	9
0	1	2	3	4

Rekursion: Durchschnitt-Mieten

In Haus #0 ist die Miete == 1, in Haus #4 ist die Miete == 9. In den Häusern dazwischen ist die Miete der Durchschnitt der Mieten der Nachbarn. Was ist die Miete im Haus #1?

1	?	?	?	9
0	1	2	3	4

```
def miete(n):
  if n == 0:
    return 1
  elif n == 4:
    return 9
  else:
    return (miete(n - 1) + miete(n + 1)) / 2.0
```

Rekursion: Durchschnitt-Mieten

In Haus #0 ist die Miete == 1, in Haus #4 ist die Miete == 9. In den Häusern dazwischen ist die Miete der Durchschnitt der Mieten der Nachbarn. Was ist die Miete im Haus #1?

1	?	?	?	9
0	1	2	3	4

```
def miete(n):
  if n == 0:
    return 1
  elif n == 4:
    return 9
  else:
    return (miete(n - 1) + miete(n + 1)) / 2.0
```

Rekursion: Listen

Gib alle Elemente einer Liste aus.

Schleife:

```
def print_all(liste):
  for e in liste:
    print e
```

Rekursion:

```
def print_all(liste):
  if len(liste) > 0:
    print liste[0]
    print_all(liste[1:])
```

Rekursion: Listen

Berechne die Summe aller Elementen einer Liste.

Schleife:

```
def summe(liste):
  result = 0
  for e in liste:
    result = result + e
  return result
```

Rekursion:

```
def summe(liste):
  if len(liste) == 0:
    return 0
  else:
    return liste[0] + summe(liste[1:])
```

Aufgaben