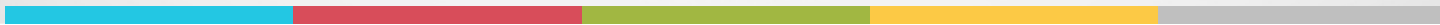
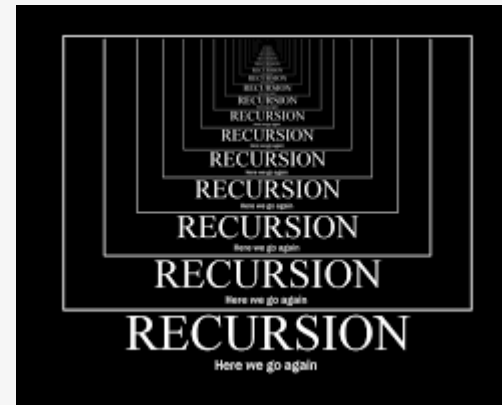


Methoden - rekursiv



Definition von Rekursion

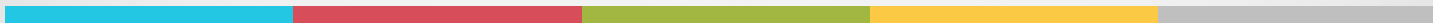
Ein Objekt heißt rekursiv, wenn es sich selbst als Teil enthält:



Beispiel: Die Fee

Auf dem Weg durch den Wald begegnet uns eine Fee. Sie spricht zu uns:
»Du hast drei Wünsche frei«....

```
static void feeBegegnetUns() {  
    wunsch();  
    wunsch();  
    feeBegegnetUns();  
}
```



Beispiel: Fakultät I

Mathematische Definition:

$$n! = \begin{cases} 1 & n=0 \\ n! = n \cdot (n-1)! & n \geq 1 \end{cases}$$

Beispiel:

$$7! = 1 \cdot 2 \cdot 3 \cdot 4 \cdot 5 \cdot 6 \cdot 7$$

Rekursive Definition:

$$n = 0 \Rightarrow 0! = 1$$

$$n! = n \cdot (n-1)!$$

Beispiel: Fakultät II

Kodierung(nicht-rekursiv):

```
double faku(int n)
{
    double res=1 ;
    for(int i=1; i<=n; i++)
        res = res*i ;
    return res;
}
```

Beispiel: Fakultät III

Kodierung(rekursiv):

```
double faku(int n)
{
    if(n==0)
        return 1;
    return faku(n-1)*n;
}
```

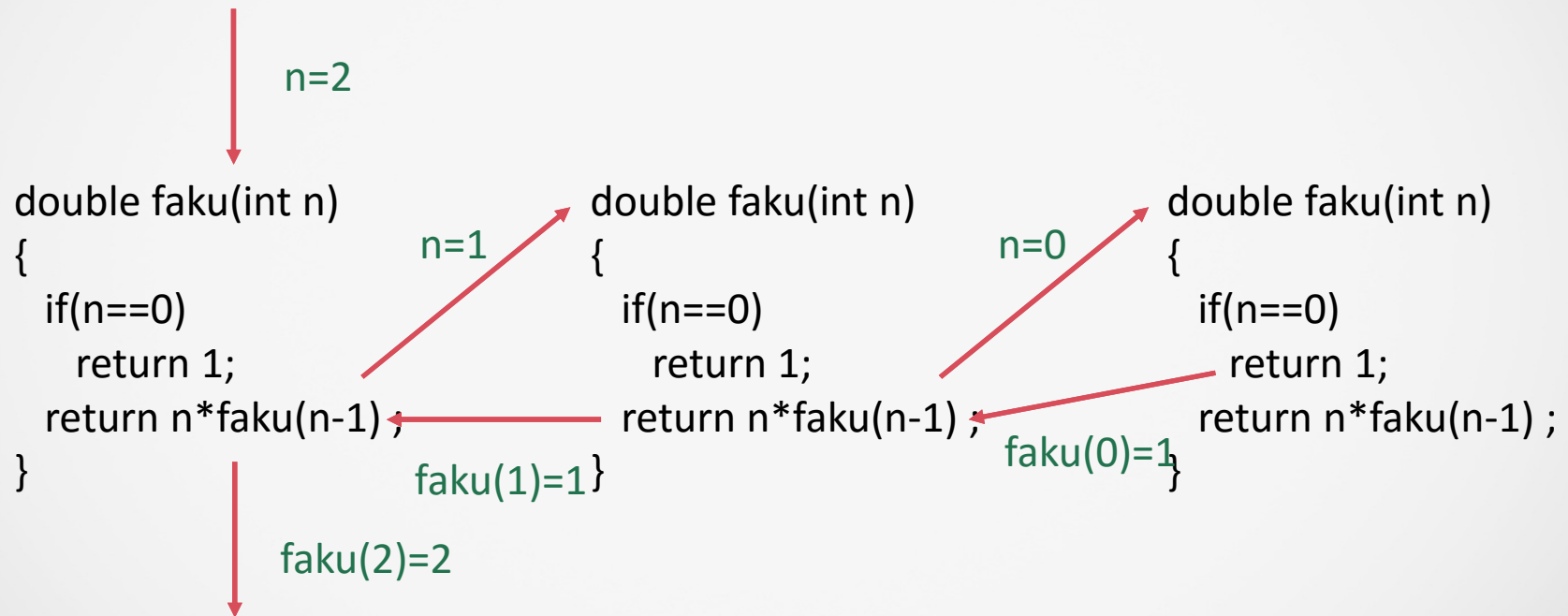
Basis Fall

$$n! = \begin{cases} 1 & n=0 \\ n! = n*(n-1)! & n \geq 1 \end{cases}$$

Rekursiver Fall

Beispiel: Fakultät IV

Visualisierung I:



Beispiel: Fakultät V

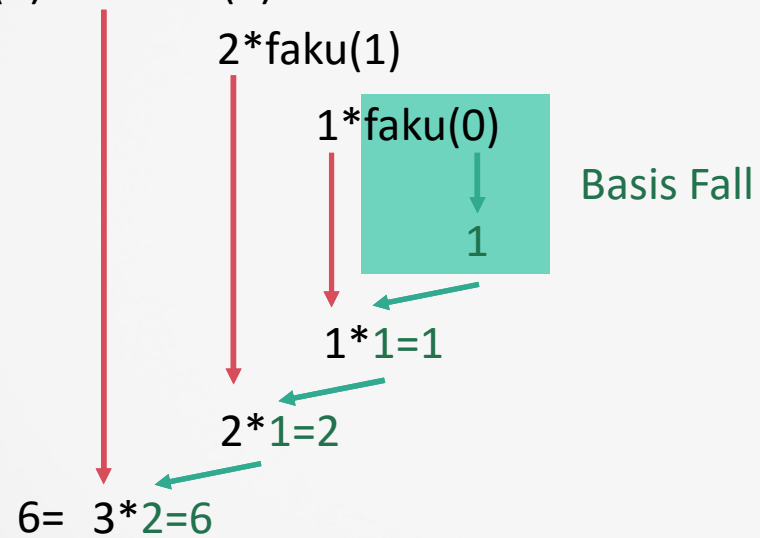
Visualisierung II:

$$\begin{aligned} \text{fak}(5) &= 5 * \text{fak}(4) \\ &= 5 * (4 * \text{fak}(3)) \\ &= 5 * ((4 * (3 * \text{fak}(2)))) \\ &= 5 * ((4 * (3 * (2 * \text{fak}(1))))) \\ &= 5 * ((4 * (3 * (2 * (1 * \text{fak}(0))))) \\ &\text{Hier kann wieder eingesetzt werden:} \\ &= 5 * ((4 * (3 * (2 * (1 * 1)))) \\ &= 120 \end{aligned}$$

Beispiel: Fakultät VI

Visualisierung III:

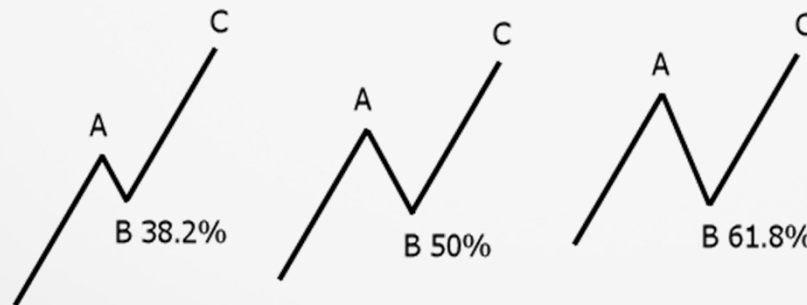
$$\text{faku}(3) = 3 * \text{faku}(2)$$



Beispiel: Fibonacci-Zahlen I

0, 1, 1, 2, 3, 5, 8, 13, ...
(optional) 0+1, 1+1, 1+2, 2+3, 3+5, 5+8

- **Leonardo Fibonacci(1202):** Wachstum einer Kaninchenpopulation
- **Griechen, Inder:** Bereits im Altertum bekannt
- **Goldener Schnitt:** Quotient zweier Folgenglieder nähert sich dem Goldenen Schnitt(1,618033...)
- **Chartanalyse:** Retracement



Beispiel: Fibonacci-Zahlen II

Kodierung:

```
public static double fibo(int n)
{
  if (n<=1)
    return n;
  return fibo(n-1) + fibo(n-2);
}
```

Basis Fall



$$a_0=0; a_1=1;$$
$$a_{n+2}=a_{n+1}+a_n \quad n \geq 2$$

gleichwertig:

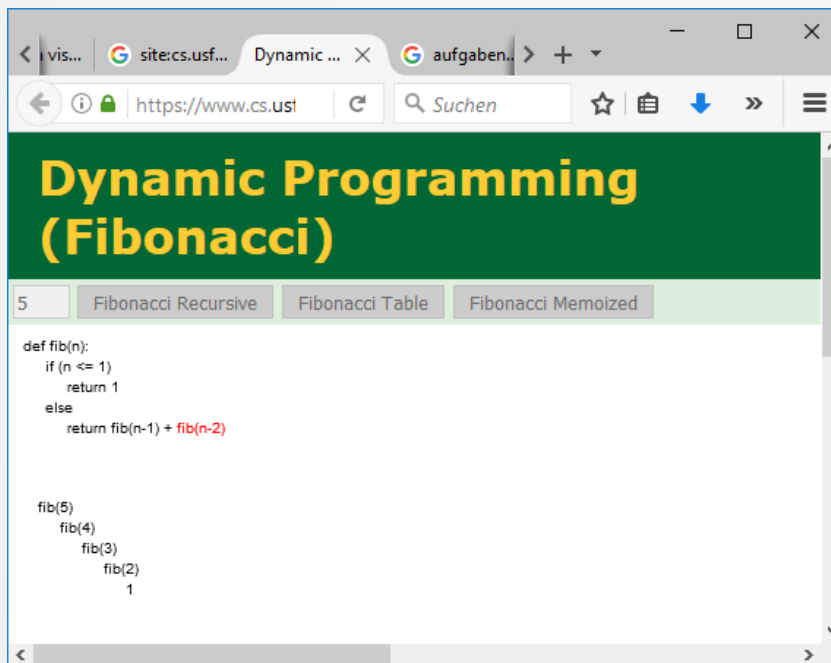
$$a_n=a_{n-1}+a_{n-2} \quad n \geq 2$$

Rekursiver Fall



Beispiel: Fibonacci-Zahlen III

Simulation:



The screenshot shows a web browser window with the URL <https://www.cs.usfca.edu/~galles/visualization/DPFib.html>. The page title is "Dynamic Programming (Fibonacci)". There are three tabs: "5", "Fibonacci Recursive", and "Fibonacci Memoized". The "5" tab is selected. The code displayed is:

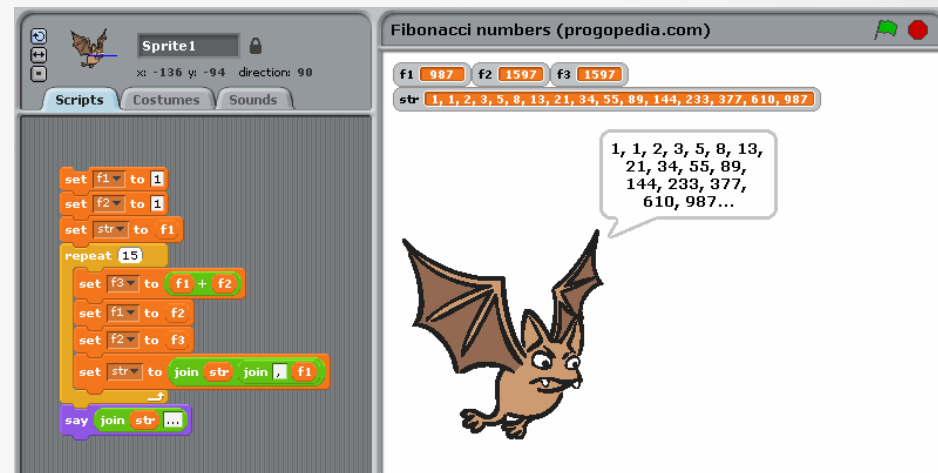
```
def fib(n):  
    if n <= 1:  
        return 1  
    else:  
        return fib(n-1) + fib(n-2)  
  
fib(5)  
fib(4)  
fib(3)  
fib(2)  
1
```

<https://www.cs.usfca.edu/~galles/visualization/DPFib.html>

Beispiel: Fibonacci-Zahlen in Scratch

Kodierung:

```
set f1 to 1
set f2 to 1
set str to f1
repeat 15
  set f3 to (f1 + f2)
  set f1 to f2
  set f2 to f3
  set str to join (str (join (,) f1))
say join (str (...))
```



<http://progopedia.com/version/scratch-1.4/>

Beispiel: Summe von den ersten n Zahlen

Kodierung:

```
public static double summeN(int n)
{
    if (n==0)
        return 0 ;
    return n+ summeN(n-1);
}
```

Basis Fall



$S_0=0;$	$n=0$
$S_n=n+S_{n-1}$	$n\geq 1$

Rekursiver Fall



Beispiel: Potenz einer Zahl mit fester Basis

Kodierung:

```
public static double PotenzN (double a,int n)
{
  if (n==0)
    return 1 ;
  return a* PotenzN (a,n-1);
}
```

Basis Fall



$P_0=1;$	$n=0$
$P_n=a*P_{n-1}$	$n\geq 1$

Rekursiver Fall

