

# Rekursion mit Python

Verallgemeinerung  
der Vorgehensweise  
bei der Rekursion

# Rekursion mit Python

- Muster\_Rekursion\_allgemein.py

```
def rekursiveFkt(Parameter)
    if abbruchTest(Parameter):
        return wert_im_Abbruchfall(Parameter)
    else:
        return bearbeite(\
            Parameter,
            rekursiveFkt(update (Parameter)))
```

ein oder mehrere Parameter

# Rekursion mit Python

- Muster\_Rekursion\_allgemein.py

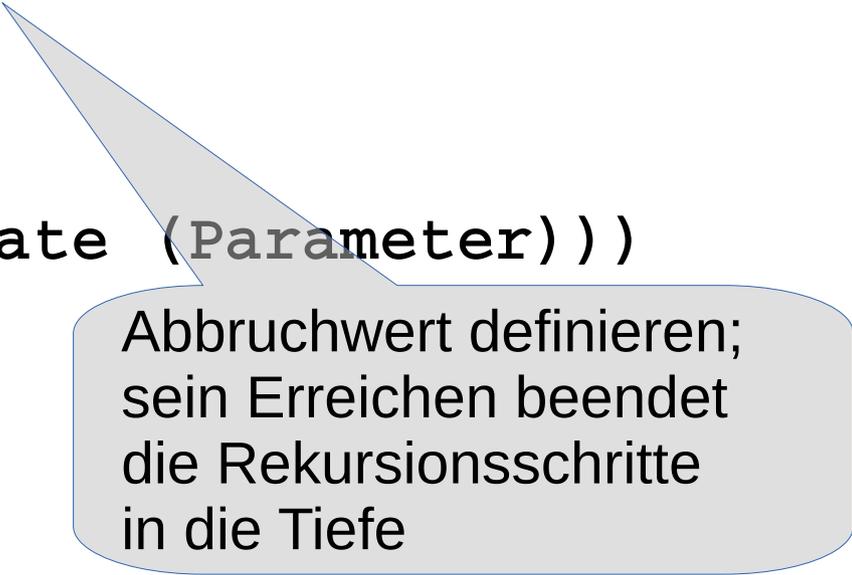
```
def rekursiveFkt (Parameter)
  if abbruchTest (Parameter) :
    return wert_im_Abbruchfall (Parameter)
  else:
    return bearbeite (\
      Parameter,
      rekursiveFkt (update (Parameter)))
```

Abbruchfall testen

# Rekursion mit Python

- Muster\_Rekursion\_allgemein.py

```
def rekursiveFkt (Parameter)
  if abbruchTest (Parameter) :
    return wert_im_Abbruchfall (Parameter)
  else:
    return bearbeite (\
      Parameter,
      rekursiveFkt (update (Parameter)) )
```



Abbruchwert definieren;  
sein Erreichen beendet  
die Rekursionsschritte  
in die Tiefe

*Anmerkung:*

*Der Abbruchwert kann ebenfalls (wie hier) durch  
eine Funktion gegeben sein.*

# Rekursion mit Python

- Muster\_Rekursion\_allgemein.py

```
def rekursiveFkt (Parameter)
    if abbruchTest (Parameter) :
        return wert_im_Abbruchfall (Parameter)
    else:
        return bearbeite (\
            Parameter,
            rekursiveFkt (update (Parameter)) )
```

In der Regel erfolgt  
noch eine Bearbeitung ...

# Rekursion mit Python

- Muster\_Rekursion\_allgemein.py

```
def rekursiveFkt (Parameter)
    if abbruchTest (Parameter) :
        return wert_im_Abbruchfall (Parameter)
    else:
        return bearbeite (\
            Parameter,
            rekursiveFkt (update (Parameter)))
```

auf der Basis der aktuellen  
Parameterwerte ...

# Rekursion mit Python

- Muster\_Rekursion\_allgemein.py

```
def rekursiveFkt (Parameter)
    if abbruchTest (Parameter) :
        return wert_im_Abbruchfall (Parameter)
    else:
        return bearbeite (\
            Parameter,
            rekursiveFkt (update (Parameter)))
```

...  
des beim Schritt in die  
Tiefe erzielten Wertes

# Rekursion mit Python

- Muster\_Rekursion\_allgemein.py

```
def rekursiveFkt (Parameter)
    if abbruchTest (Parameter) :
        return wert_im_Abbruchfall (Parameter)
    else:
        return bearbeite (\
            Parameter,
            rekursiveFkt (update (Parameter)))
```

Die konkrete Problemlösung hängt von der Bearbeitungsfunktion und der konkreten update-Fkt ab.

# Rekursion mit Python

Warum update?

Bearbeitung mit unveränderten Parameterwerten bedeutet bei funktionaler Programmierung

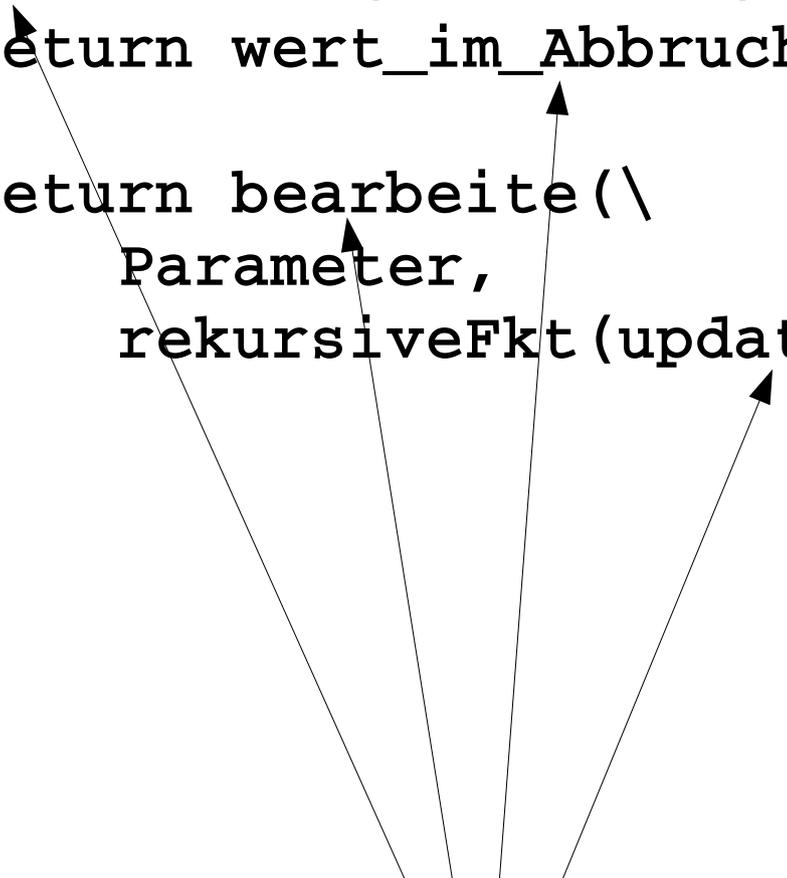
- identische Bearbeitung,
- identische Werte und
- führt zu einer Endlosschleife

Die update-Funktion muss prinzipiell auf die Abbruchbedingung hin führen

# Rekursion mit Python

- Beispiel: Rekursion über Zahlen

```
def rekursiveFkt (Parameter)
  if abbruchTest (Parameter) :
    return wert_im_Abbruchfall (Parameter)
  else:
    return bearbeite (\
      Parameter,
      rekursiveFkt (update (Parameter)))
```

The diagram consists of four arrows originating from a common point at the bottom center. One arrow points to the 'rekursiveFkt' argument in the recursive call 'rekursiveFkt (update (Parameter))'. Another arrow points to the 'Parameter' argument in the same recursive call. A third arrow points to the 'wert\_im\_Abbruchfall (Parameter)' return statement in the base case. The fourth arrow points to the 'Parameter' argument in the base case return statement.

*mit den konkreten Funktionen ...*

# Rekursion mit Python

- Beispiel: Rekursion über Zahlen

```
def abbruchTest (Parameter) :  
    return Parameter<=1
```

```
def wert_im_Abbruchfall (Parameter) :  
    return 1
```

```
def bearbeite (Parameter, rekursionswert) :  
    return Parameter * rekursionswert
```

```
def update (Parameter) :  
    return Parameter - 1
```

*... erhält man hier die Fakultäts-Funktion.*

# Rekursion mit Python

Nicht für jeden Parameter ist eine update-Funktion notwendig, die ihn verändert.

Umgekehrt gilt:

- Bei mehreren Parametern ist für mindestens einen eine update-Funktion notwendig, die seinen Wert verändert.