

Anweisungen in Python

Prof. Dr. Rüdiger Weis

Beuth Hochschule für Technik Berlin

Sommersemester 2011

1 if

2 elif

3 while

4 for

5 Schleifenabbruch

Blockbildung

- Blockbildung durch Einrücken
- Keine Klammerung
- Empfehlungen
 - Leerzeichen verwenden
 - 4 Leerzeichen einrücken
 - Unterstützenden Editor (z.B. emacs) verwenden

if Anweisung

if..else

```
if <bedingung> : <anweisung>  
[else : <anweisung>]
```

- : statt THEN
- else ist optional.

Beispiel

```
antwort = raw_input("Was ist der Sinn des Lebens?")
if antwort == '42' :
    print("Falscher Film!")
else:
    print(antwort + ' ist eine interessante Antwort.')
```

if..elif

if..elif

```
if <bedingung> : <anweisung>  
[elif <bedingung> : <anweisung>]
```

- Kurzform von else : if <Bedingung>
- Syntactic sugar
- Spart Einrücken
- elif und else sind kombinierbar.

Beispiel mit elif

```
antwort = raw_input("Was ist der Sinn des Lebens?")
if antwort == '42' :
    print("Falscher Film!")
elif antwort == 'Geld' :
    print('Wenn Sie meinen.')
```

Beispiel mit else if

```
antwort = raw_input("Was ist der Sinn des Lebens ? ")
if antwort == '42' :
    print("Falscher Film!")
else:
    if antwort == 'Geld' :
        print('Wenn Sie meinen.')
```

while Anweisung

while

```
while <bedingung> : <anweisung>  
[else : <anweisung>]
```

- else ist optional.
- else wird nur ausgeführt,
wenn die Schleife ohne break verlassen wird.

for Anweisung

for Anweisung

```
for <element> in <folge> : <anweisung>  
[else : <anweisung>]
```

- for iteriert über eine Sequenz
 - Liste
 - Tupel
 - String
- Die 'Schleifenvariable' <Elemente> wird nacheinander mit den Elementen der Sequenz belegt.
- else wird nur ausgeführt, wenn die Schleife ohne break verlassen wird.

for bei Tupeln

```
>>> bsp = (1, ('a', 'b'), 'Spam')
>>> for element in bsp : print(element * 3)
...
3
('a', 'b', 'a', 'b', 'a', 'b')
SpamSpamSpam
```

for bei Listen

```
>>> bsp = [1,('a','b'),'Spam']
>>> for element in bsp : print(element * 3)
...
3
('a', 'b', 'a', 'b', 'a', 'b')
SpamSpamSpam
```

Schleifenabbruch

break versus continue

- break Anweisung springt **hinter** die umgebenden for oder while Schleife.
- continue Anweisung springt zur **Schleifen-Bedingung-Zeile** der umgebenden for oder while Schleife.

Ein else block der for oder while Anweisung wird übersprungen.

Beispiel: Autouebersetzer

```
woerterbuch = {}
satz = raw_input("Bitte Satz eingeben : ")
while satz != "":
    woerter = satz.split()
    for wort in woerter :
        if wort not in woerterbuch :
            print("Bitte Uebersetzung von " + wort + " eingeben : "),
            uebersetzung = raw_input()
            woerterbuch[word] = uebersetzung
    for wort in woerter :
        print woerterbuch[word],
    print("*** Auto Uebersetzung ***")
    satz = raw_input("Bitte Satz eingeben : ")
```

The logo for ©opyleft, featuring a white copyright symbol inside a circle followed by the text "opyleft" in a white sans-serif font, all on a dark blue background.

- Erstellt mit Freier Software
- © Rüdiger Weis, Berlin 2005–2011
- unter der GNU Free Documentation License.