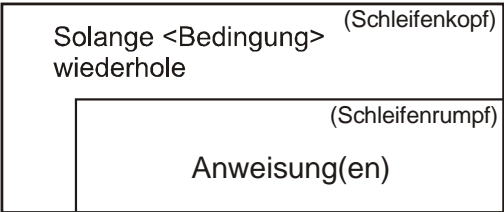
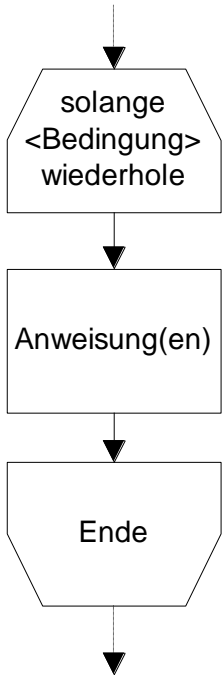


### KOPFGESTEUERTE SCHLEIFE

Eine kopfgesteuerte Schleife wird solange ausgeführt, wie die im Schleifenkopf angegebene Bedingung erfüllt (*true*) ist. Ist die Bedingung schon beim ersten Eintritt in die Schleife nicht erfüllt (*false*), dann werden die Anweisungen innerhalb der Schleife überhaupt nicht ausgeführt.

Eine kopfgesteuerte Schleife wird in Python über die **while** - Anweisung realisiert.

#### Darstellung der kopfgesteuerten Schleife im Struktogramm bzw. PAP

Struktogramm nach DIN 66261	Bemerkungen	PAP DIN 66001
	<p>Die kopfgesteuerte Schleife wird solange ausgeführt, wie die Bedingung im Schleifenkopf erfüllt (<i>true</i>) ist.</p> <p>Eine kopfgesteuerte Schleife kann ggf. kein einziges Mal durchlaufen werden.</p>	

#### Syntax der offenen kopfgesteuerten Schleife in PYTHON

<p><b>Syntax</b></p> <pre> while Bedingung:     Anweisung     Anweisung     ... # optional, falls die # Bedingung nie gilt else:     Anweisung     ... # Einrückung zurück     ...         </pre>	<p>Solange die Bedingung im Schleifenkopf wahr (<i>true</i>) ist, werden die Anweisungen im Schleifenrumpf ausgeführt.</p> <p>Falls die Bedingung nie (also schon beim ersten Mal nicht zutrifft), wird ein optionaler „else-Pfad“ abgearbeitet</p>
---	---

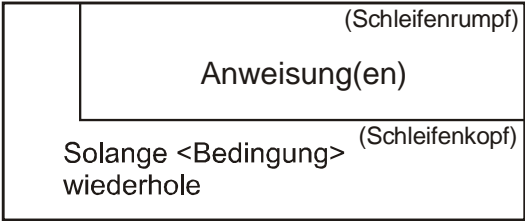
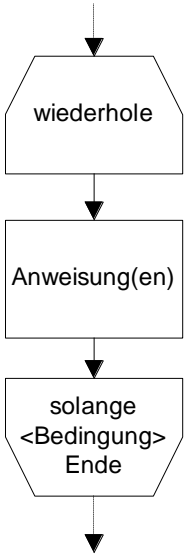
Die Codierung der while-Schleife erfolgt mit dem reservierten Wort **while** der anschließende Bedingungsausdruck wird bzgl. des Wahrheitswertes zu Beginn jedes Schleifendurchlaufs überprüft.

## FUSSGESTEUERTE SCHLEIFE

Eine fußgesteuerte Schleife wird ausgeführt, solange die im Schleifenkopf\* angegebene Bedingung erfüllt (*true*) ist. Die Bedingung einer fußgesteuerten Schleife wird erst am Ende der Schleife überprüft, aus diesem Grund werden die Anweisungen innerhalb der Schleife immer mindestens einmal ausgeführt.

\* auch wenn die Bedingung am Ende der Schleife (also am Fuß) steht, wird vom Schleifenkopf gesprochen!

### Darstellung der fußgesteuerten Schleife im Struktogramm bzw. PAP

Struktogramm nach DIN 66261	Bemerkungen	PAP DIN 66001
	<p>Die fußgesteuerte Schleife wird solange ausgeführt, wie die Bedingung im Schleifenkopf erfüllt (<i>true</i>) ist.</p> <p>Eine fußgesteuerte Schleife wird immer mindestens einmal durchlaufen.</p>	

### Syntax der offenen fußgesteuerten Schleife in Python

Eine fußgesteuerte Schleife wird in Python nicht direkt unterstützt, sondern muss über die „**while** – Anweisung“ realisiert werden.

<b>Syntax</b>	<pre> <b>first = True</b> <b>while</b> Bedingung <b>or</b> first:     Anweisung     Anweisung     ...     <b>first = False</b>  # Einrückung zurück ... </pre>	<p>Der <b>erstmalige Eintritt in die Schleife</b> kann sichergestellt werden, indem eine <b>wahre Aussage</b> mit der Schleifenbedingung mittels <b>or</b> verknüpft wird.</p> <p>Anschließend werden die Anweisungen im Schleifenrumpf ausgeführt, solange die Bedingung im Schleifenkopf wahr (<i>True</i>) ist.</p>
---------------	--	--

Wiederholungen von Programmabschnitten (Iterationen) können in den meisten Programmiersprachen mit vorprüfenden Schleifen, nachprüfenden Schleifen oder mit Zählschleifen realisiert werden.

## Zählschleifen

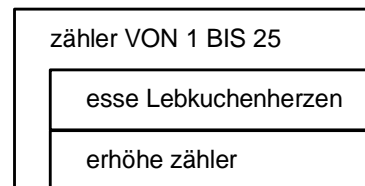
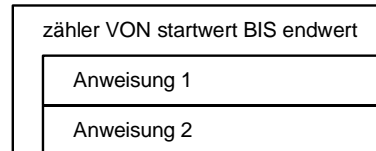
Die Anzahl der Wiederholungen wird vorher festgelegt.

### Pseudocode

```
FÜR Zähler VON <Startwert> BIS <Endwert>
    <Anweisung 1>
    <Anweisung 2>
```

```
FÜR Zähler VON 1 BIS 25
    esse Lebkuchenherzen
```

### (manchmal verwendetes) Struktogramm



In Python wird die Zählschleife mit der **for-Anweisung** realisiert. Allerdings handelt es sich nicht um eine reine Zählschleife. Vielmehr findet hier eine Iteration über eine Liste (von Elementen) statt. Bei jedem Durchlauf wird das nächste Element aus der Liste einer Schleifenvariablen zugewiesen.

Die Liste kann im einfachsten Fall als Folge ganzer Zahlen mittels der **range**-Anweisung erstellt werden.

### Syntax:

```
for i in range(0, n): # [0,...,n-1]
    Anweisung
    Anweisung
```

Im Initialisierungsteil wird die Zählvariable der Schleife initialisiert, also ihr Anfangswert festgelegt; die Schleife wird dann solange wiederholt, bis die Wiederholungsbedingung false ergibt. Im Inkrementierungsteil ("Erhöhungsteil") wird die Zählvariable verändert.

Programmausschnitt	BildschirmAusgabe
<pre>for i in range(2, 6):     print(i, end=", ")</pre>	2,_3,_4,_5,_
<pre>for i in range(7, 4, -1):     print(i, end=", ")</pre>	7,_6,_5,_

Erläutern Sie die Wirkung des folgenden Programmausschnitts.

```
print("Anzahl der Zahlen: ")
anzahl = int(input())
summe = 0
produkt = 1
if (anzahl == 0)
    produkt = 0
for i in range(1, anzahl+1, 1):
    zahl = int(input(("Zahl eingeben: ")))
    summe += zahl
    produkt *= zahl
print("Summe:" , summe, "-- Produkt:", produkt)
```

1. Zählvariable initialisieren

2. Abbruchswert

3. „inkrementieren“

anzahl = 2	produkt	zahl	summe
	1		0
i = 1		1	
i = 2		3	
Letztes Element der Liste abgearbeitet			
Schleife beendet			
i wird "vergessen"			