

# LU06b - Einführung in Iterationen in Bash

Iterationen (Schleifen) in Bash ermöglichen es, wiederholt eine Gruppe von Befehlen auszuführen, bis eine bestimmte Bedingung erfüllt ist. Bash bietet dafür mehrere Arten von Schleifen, die für unterschiedliche Zwecke verwendet werden können.

—

## ### 1. `for`-Schleife

Die **`for`-Schleife** wird verwendet, um durch eine Liste von Werten oder Dateien zu iterieren.

#### **Grundstruktur:** ``` bash for variable in Liste; do`

```
# Befehle
```

```
done ``
```

#### **Beispiel:** ``` bash #!/bin/bash for name in Alice Bob Charlie; do`

```
echo "Hallo, $name!"
```

```
done ``
```

**Ausgabe:** ``` Hallo, Alice! Hallo, Bob! Hallo, Charlie! ```

#### **`for`-Schleife für Dateien:** ``` bash #!/bin/bash for file in *.txt; do`

```
echo "Datei: $file"
```

```
done ``
```

—

## ### 2. `while`-Schleife

Die **`while`-Schleife** führt Befehle aus, solange eine Bedingung wahr ist.

#### **Grundstruktur:** ``` bash while [ Bedingung ]; do`

```
# Befehle
```

```
done ``
```

#### **Beispiel:** ``` bash #!/bin/bash zahl=1 while [ $zahl -le 5 ]; do`

```
echo "Zahl: $zahl"
zahl=$((zahl + 1))
```

```
done ``
```

**Ausgabe:** `` Zahl: 1 Zahl: 2 Zahl: 3 Zahl: 4 Zahl: 5 ``

—

### ### 3. `until`-Schleife

Die `until`-Schleife ist das Gegenteil der `while`-Schleife: Sie läuft, bis eine Bedingung wahr wird.

#### Grundstruktur: `` bash until [ Bedingung ]; do

```
# Befehle
```

```
done ``
```

#### Beispiel: `` bash #!/bin/bash zahl=1 until [ \$zahl -gt 5 ]; do

```
echo "Zahl: $zahl"
zahl=$((zahl + 1))
```

```
done ``
```

**Ausgabe:** `` Zahl: 1 Zahl: 2 Zahl: 3 Zahl: 4 Zahl: 5 ``

—

### ### 4. Schleifensteuerung

Bash bietet Befehle, um Schleifen zu kontrollieren: - **`break`**: Beendet die Schleife vorzeitig. - **`continue`**: Überspringt den aktuellen Schleifendurchlauf.

#### Beispiele: `` bash # Mit break for i in {1..10}; do

```
if [ $i -eq 5 ]; then
    break
fi
echo "Zahl: $i"
```

```
done ``
```

`` bash # Mit continue for i in {1..5}; do

```
if [ $i -eq 3 ]; then
    continue
fi
echo "Zahl: $i"
```

```
done ``
```

—

### ### Zusammenfassung

Schleife	Beschreibung	Verwendung
-----	-----	-----
`for`	Iteriert über eine Liste von Werten oder Dateien	Wiederholung bei bekannter Anzahl von Werten
`while`	Läuft, solange eine Bedingung wahr ist	Dynamische Bedingung
`until`	Läuft, bis eine Bedingung wahr wird	Dynamische Bedingung, Gegenteil von `while`
`break`	Beendet die Schleife	Vorzeitiges Ende
`continue`	Überspringt den aktuellen Schleifendurchlauf	Fortsetzen der Schleife

Mit diesen Schleifen können Sie wiederkehrende Aufgaben effizient automatisieren und komplexe Abläufe strukturieren.

From:

<https://wiki.bzz.ch/> - **BZZ - Modulwiki**

Permanent link:

<https://wiki.bzz.ch/modul/m122/learningunits/lu06/iteration?rev=1731912473>

Last update: **2024/11/18 07:47**

