LU06a - Einführung in Selektionen in Bash

Selektionen (auch Kontrollstrukturen genannt) ermöglichen es, in Bash-Skripten Entscheidungen zu treffen und den Programmfluss basierend auf Bedingungen zu steuern. Die beiden häufigsten Mechanismen dafür sind **`if`-Bedingungen** und **`case`-Anweisungen**.

`if`-Bedingungen

Die `if`-Anweisung wird verwendet, um Code auszuführen, wenn eine Bedingung wahr ist.

Grundstruktur

```
if [ Bedingung ]; then
    # Befehle, falls Bedingung wahr ist
elif [ andere Bedingung ]; then
    # Befehle, falls andere Bedingung wahr ist
else
    # Befehle, falls keine Bedingung wahr ist
fi
```

Beispiel:

```
#!/bin/bash
zahl=10

if [ $zahl -gt 5 ]; then
    echo "Die Zahl ist größer als 5"
elif [ $zahl -eq 5 ]; then
    echo "Die Zahl ist genau 5"
else
    echo "Die Zahl ist kleiner als 5"
fi
```

Wichtige Operatoren:

• Zahlen:

```
`-eq`: gleich
`-ne`: ungleich
`-lt`: kleiner als
`-le`: kleiner oder gleich
`-gt`: größer als
`-ge`: größer oder gleich

Zeichenketten:

`=`: gleich
`!=`: ungleich
`-z`: leer
```

o `-n`: nicht leer

`case`-Anweisungen

Die `case`-Anweisung ist nützlich, wenn Sie mehrere Bedingungen basierend auf einem einzigen Wert prüfen möchten.

Grundstruktur

```
case Variable in
   Muster1)
      # Befehle für Muster1
      ;;;
Muster2)
      # Befehle für Muster2
      ;;;
*)
      # Standardfall (wenn nichts passt)
      ;;
esac
```

Beispiel:

```
#!/bin/bash
farbe="blau"

case $farbe in
    "rot")
        echo "Die Farbe ist rot"
        ;;
    "blau")
        echo "Die Farbe ist blau"
        ;;
    *)
        echo "Unbekannte Farbe"
        ;;
esac
```

Vergleich von `if` und `case`

	`if`-Bedingungen	`case`-Anweisungen
Nützlichkeit	Für numerische oder logische Vergleiche	Für Mustervergleiche
Lesbarkeit	Schwer lesbar bei vielen Bedingungen	Einfach bei vielen Optionen

https://wiki.bzz.ch/ Printed on 2025/11/20 01:50

Logische Operatoren in Selektionen

Bedingungen können kombiniert werden mit:

• `&&`: UND
• `||`: ODER

Beispiel:

```
#!/bin/bash
zahl=10

if [ $zahl -gt 5 ] && [ $zahl -lt 20 ]; then
        echo "Die Zahl liegt zwischen 5 und 20"
else
        echo "Die Zahl liegt außerhalb des Bereichs"
fi
```

Zusammenfassung

- Verwenden Sie `if`, wenn Sie logische oder numerische Bedingungen prüfen möchten.
- Nutzen Sie `case`, um mehrere Werte in einer übersichtlichen Weise zu prüfen.
- Kombinieren Sie Bedingungen mit `&&` und `||`, um komplexe Entscheidungen zu treffen.

M122-LU06



From:

https://wiki.bzz.ch/ - BZZ - Modulwiki

Permanent link:

https://wiki.bzz.ch/modul/m122/learningunits/lu06/selektion?rev=1731912322

Last update: 2024/11/18 07:45

