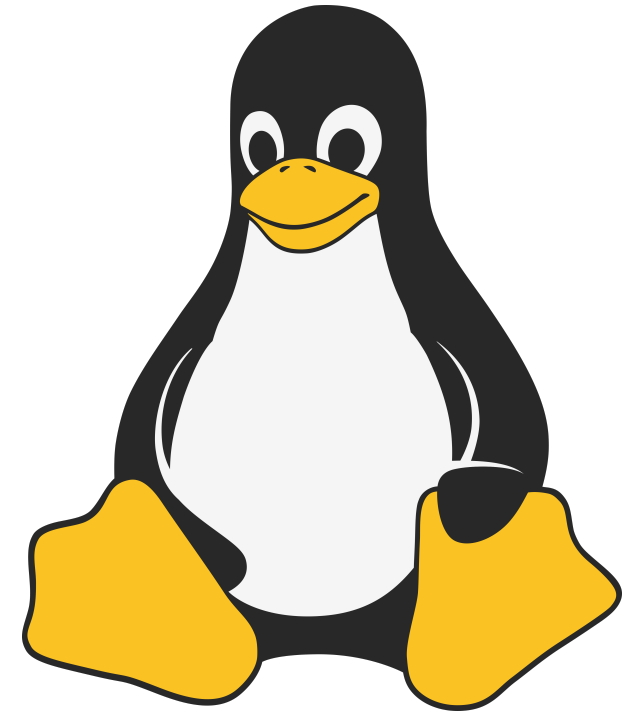


# Shell-Praxis

## Kommandos gruppieren



# Kommandos gruppieren mit `()`

- Mehrere Kommandos können in runden Klammern `()` zusammengefasst werden.
- Die Klammern `()` bilden eine Gruppe von Kommandos, die von der Shell als ein einziges Kommando behandelt werden.
- Diese Kommandogruppierung wird typischerweise verwendet, um die Ausgabe mehrerer Kommandos gemeinsam in eine Datei oder einen Pipe umzulenken.

Syntax: `( command1 ; command2 ; ... ; commandN )`

- Nach der öffnenden runden Klammer `(` muss **kein** Leerzeichen oder ein Zeilenumbruch folgen.
- Vor der schließenden runden Klammer `)` muss **kein** Kommando-Trenner stehen.

```
hermann@debian:~$ # single line command group
hermann@debian:~$ (echo "USERNAME:PASSWORD:UID:GID:COMMENT:HOME:SHELL"; head -3 /etc/passwd | sort)
USERNAME:PASSWORD:UID:GID:COMMENT:HOME:SHELL
bin:x:2:2:bin:/bin:/usr/sbin/nologin
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
root:x:0:0:root:/root:/bin/bash
```

```
hermann@debian:~$ # single line command group
hermann@debian:~$ (echo "USERNAME:PASSWORD:UID:GID:COMMENT:HOME:SHELL"; head -3 /etc/passwd | sort) | nl
  1  USERNAME:PASSWORD:UID:GID:COMMENT:HOME:SHELL
  2  bin:x:2:2:bin:/bin:/usr/sbin/nologin
  3  daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
  4  root:x:0:0:root:/root:/bin/bash
```

```
hermann@debian:~$ ( # multi line command group
> echo "USERNAME:PASSWORD:UID:GID:COMMENT:HOME:SHELL"
> head -3 /etc/passwd | sort
> ) | nl
  1  USERNAME:PASSWORD:UID:GID:COMMENT:HOME:SHELL
  2  bin:x:2:2:bin:/bin:/usr/sbin/nologin
  3  daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
  4  root:x:0:0:root:/root:/bin/bash
```

# Kommandos gruppieren mit `{ }`

- Mehrere Kommandos können in geschweiften Klammern `{ }` zusammengefasst werden.
- Die Klammern `{ }` bilden eine Gruppe von Kommandos, die von der Shell als ein einziges Kommando behandelt werden.
- Diese Kommandogruppierung wird typischerweise verwendet, um eine Funktion zu definieren (wird hier nicht behandelt).
- Man verwendet es auch, um die Ausgabe mehrerer Kommandos gemeinsam in eine Datei oder einen Pipe umzulenken. Hierzu wird jedoch meist das Konstrukt in runden Klammern `( )` verwendet.

- Wenn möglich führt die Shell die geklammerte Kommandogruppe in der aktuellen Shell aus. In manchen Fällen wird jedoch eine Subshell erzeugt (z.B. bei der Umlenkung der Ausgabe in eine Pipe oder Datei).

Syntax: `{ command1; command2; command3; ...; commandN; }`

- Nach der öffnenden geschweiften Klammer `{` muss ein Leerzeichen oder ein Zeilenumbruch folgen.
- Vor der schließenden geschweiften Klammer `}` muss ein Kommando-Trenner (Semikolon `;` oder ein Newline-Zeichen) stehen.

```
hermann@debian:~$ { echo "USERNAME:PASSWORD:UID:GID:COMMENT:HOME:SHELL"; head -3 /etc/passwd | sort; }
USERNAME:PASSWORD:UID:GID:COMMENT:HOME:SHELL
bin:x:2:2:bin:/bin:/usr/sbin/nologin
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
root:x:0:0:root:/root:/bin/bash
```

```
hermann@debian:~$ { echo "USERNAME:PASSWORD:UID:GID:COMMENT:HOME:SHELL"; head -3 /etc/passwd | sort; } | nl
 1  USERNAME:PASSWORD:UID:GID:COMMENT:HOME:SHELL
 2  bin:x:2:2:bin:/bin:/usr/sbin/nologin
 3  daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
 4  root:x:0:0:root:/root:/bin/bash
```



```
hermann@debian:~$ {  
> echo "USERNAME:PASSWORD:UID:GID:COMMENT:HOME:SHELL"  
> head -3 /etc/passwd | sort  
> }  
USERNAME:PASSWORD:UID:GID:COMMENT:HOME:SHELL  
bin:x:2:2:bin:/bin:/usr/sbin/nologin  
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin  
root:x:0:0:root:/root:/bin/bash
```

# Subtiler Unterschied zwischen `()` und `{}`

- In den obigen Beispielen ist der Unterschied zwischen den beiden Konstrukten mit `()` und `{}` nicht sichtbar.
- Der Unterschied liegt in der Ausführung der Kommandogruppe in der aktuellen Shell bei `{}` und in einer Subshell bei `()`.