

# Shell-Praxis (Teil 10)



## Initialisierung der Bash-Shell

# Inhaltsverzeichnis

- [Aliase dauerhaft verfügbar machen](#)
- [Kommando](#) `source`
- [Bash-Initialisierung in](#) `~/ .bashrc`

# Aliase dauerhaft verfügbar machen

Wir haben im vorherigen Tutorial Aliase definiert, die nur für die aktuelle Shell-Sitzung gelten.

Um Aliase dauerhaft verfügbar zu machen, müssen wir sie in die Datei `~/.bash_aliases` eintragen.

Textdateien können mit jedem Texteditor bearbeitet werden. Wir verwenden `nano`. Dieser Editor ist einfach und intuitiv zu bedienen. Er ist anfängertauglich und wird auf fast allen Linux-Systemen mitgeliefert.

# Alias des Dozenten

```
alias a=alias  
alias cl=clear  
alias h=history  
alias l='ls -l'  
alias la='ls -al'  
alias cx='chmod u+x'  
alias path='echo $PATH'  
alias ssdn='sudo shutdown now'  
alias srbn='sudo reboot now'
```

## Empfehlung: Aliase des Dozenten (zum Start) verwenden

- Verwenden zunächst die Aliase des Dozenten, die in `~/linux-trainings/Linux-Handouts/alias/bash_aliases` zu finden sind.
- Hängen Sie diese Aliase an die Datei `~/.bash_aliases` an:  
`cat ../alias/bash_aliases >> ~/.bash_aliases`.
- Bearbeiten Sie die Datei `~/.bash_aliases` danach nach Ihren Wünschen mit `nano`: Löschen Sie Aliase, die Sie nicht benötigen, und/oder fügen Sie eigene hinzu.

Kontrollieren Sie mit `cat`, ob die Aliase korrekt in der Datei `~/.bash_aliases` eingetragen sind.

Allein durch den Eintrag in der Datei `~/.bash_aliases` sind die Aliase noch nicht aktiv. Sie müssen dafür sorgen, dass `~/.bash_aliases` neu geladen wird.

Sie schließen die aktuelle Terminal-Sitzung und öffnen eine neue. Beim Start der neuen Sitzung wird `~/.bash_aliases` automatisch geladen. Die Aliase sind dann verfügbar.

# Kommando `source`

Mit dem Kommando `source` kann eine Skriptdatei wie z. B. `~/ .bash_aliases` in die aktuelle Shell geladen werden.

Dies wird meist verwendet, um Änderungen an der aktuellen Shell-Umgebung vorzunehmen (z.B. Definition von Umgebungsvariablen oder Aliasen)

Damit kann man sich das Ab- und Anmelden zur Initialisierung der Shell ersparen.

Das Kommando `.` ist ein Synonym für `source`.

## Veränderte Alias-Definitionen in `~/ .bash_aliases` laden

```
hermann@debian:~/my-tests$ source ~/.bash_aliases
```

```
hermann@debian:~/my-tests$ a
alias a='alias'
alias cl='clear'
alias h='history'
alias l='ls -l'
alias la='ls -al'
alias cx='chmod u+x'
alias path='echo $PATH'
alias ssdn='sudo shutdown now'
alias srbn='sudo reboot now'
```



```
hermann@debian:~/my-tests$ a path
alias path='echo $PATH'
hermann@debian:~/my-tests$ a l
alias l='ls -l'
```

- `a` ist jetzt ein Alias für `alias`.
- `a` allein zeigt alle Aliase an.
- `a path` zeigt die Definition des Aliases `path`.
- `a l` zeigt die Definition des Aliases `ls -l`.

```
hermann@debian:~/my-tests$ l
insgesamt 8
drwxrwxr-x 3 hermann hermann 4096 Nov 10 13:05 my-etc
-rw-rw-r-- 1 hermann hermann  223 Nov  9 17:09 users.txt
```

```
hermann@debian:~/my-tests$ echo $PATH
/usr/local/bin:/usr/bin:/bin:/usr/local/games:/usr/games
hermann@debian:~/my-tests$ path # is the same as echo $PATH
/usr/local/bin:/usr/bin:/bin:/usr/local/games:/usr/games
```

## Alias löschen: `unalias`

Mit dem Kommando `unalias <alias_name>` kann ein Alias gelöscht werden.

```
hermann@debian:~/my-tests$ alias h
alias h='history'
hermann@debian:~/my-tests$ unalias h # undefine alias h
hermann@debian:~/my-tests$ alias h
bash: alias: h: not found
hermann@debian:~/my-tests$ alias h=history # redefine alias h
hermann@debian:~/my-tests$ alias h
alias h='history'
```

# Bash-Initialisierung in `~/ .bashrc`

- Die Datei `~/ .bashrc` wird automatisch beim Start einer interaktiven Bash-Shell geladen.
- Die dient der Initialisierung der `bash`. Sie enthält Befehle, die bei jedem Start einer interaktiven Bash-Shell ausgeführt werden sollen. Typischerweise sind das:
  - Definition von Umgebungsvariablen,
  - Definition von Aliase,
  - Definition von Funktionen (werden hier nicht behandelt),
  - Setzen von Shell-Optionen (werden hier nicht behandelt).

## Inhalt der Datei `~/.bashrc`

- Sehen Sie sich die Datei `~/.bashrc` mit den Kommandos `nl -ba` und `less` an: `nl -ba ~/.bashrc | less`.
- Diese Kommandos sind "ungefährlich". Sie ändern die Datei nicht, sondern zeigen nur ihren Inhalt an.
- Im nächsten Tutorial werden wir die Datei `~/.bashrc` auch bearbeiten.

## Alias-Definitionen in `~/.bashrc`

- Einige Aliase sind in `~/.bashrc` bereits vordefiniert.
- Die eigenen Aliase werden typischerweise in `~/.bash_aliases` definiert.
- `~/.bashrc` lädt `~/.bash_aliases` mit dem Kommando `source ~/.bash_aliases`.

## Auszug aus `~/.bashrc`:

```
# Alias definitions.  
# You may want to put all your additions into a separate file like  
# ~/.bash_aliases, instead of adding them here directly.  
# See /usr/share/doc/bash-doc/examples in the bash-doc package.  
  
if [ -f ~/.bash_aliases ]; then  
    . ~/.bash_aliases  
fi
```

Wenn die Datei `~/.bash_aliases` existiert, wird sie mit dem Kommando `.` in die aktuelle Shell geladen. `.` ist ein Synonym für `source`.

## `~/ .bashrc` erweitern

- Damit Sie sie bei einem Fehler zur Not wiederherstellen können, erstellen Sie eine Sicherungskopie:

```
hermann@debian:~$ cp ~/.bashrc ~/.bashrc.org
```

- Öffnen Sie nun die Datei `~/ .bashrc` mit dem Editor `nano`:

```
hermann@debian:~$ nano ~/.bashrc
```



- Fügen Sie die folgenden Zeilen am Ende der Datei ein und beenden Sie danach den Editor:

```
cowsay -f tux "cowsay $(echo Heute ist $(date +%A).)"
```

- Beenden Sie die aktuelle Shell und öffnen Sie eine neue Shell-Sitzung. Was geschieht?
- Starten Sie eine Subshell mit dem Kommando `bash`. Was geschieht?
- Beenden Sie die Subshell mit `exit` oder `Ctrl+D`.