

unix / linux



```
#!/bin/sh -e
### BEGIN INIT INFO
# Provides:          ifupdown
# Required-Start:    ifupdown-clean
# Required-Stop:     $local_fs
# Default-Start:     S
# Default-Stop:      0 6
# Short-Description: Prepare the system for taking up
### END INIT INFO
```

```
[ -x /sbin/ifup ] || exit 0
[ -x /sbin/ifdown ] || exit 0
```

Shellscripte

```
MYNAME="${0##*/}"
report() { echo "${MYNAME}: $*" ; }
report_err() { log_failure_msg "$*" ; }
```

Ziele



- Einsatzgebiete und grundlegende Syntax von Shellskripten kennen
- Shellskripte aufrufen können
- `#!`-Zeilen verstehen und formulieren können

Agenda



- **Einleitung**
- Aufruf
- Aufbau
- BASH
- Shellskripte schreiben
- Fehlererkennung

Einleitung



- Alles was direkt mit der Shell möglich ist, ist auch in einem Shellskript möglich
- Einsetzbar für immer wiederkehrende Aufgaben (Automatisierung)
- Vereinfacht komplexe Aufgaben
- Ersetzt keine "echte" Programmiersprache

Agenda



- Einleitung
- **Aufruf**
- Aufbau
- BASH
- Shellskripte schreiben
- Fehlererkennung



Aufruf von Shellskripten

- Übergabe des Skripts an eine Bash
`$bash script.sh`
- Ausführbare Skripte sind direkt aufrufbar
`$chmod u+rx script.sh ; ./script.sh`
`$chmod 755 script.sh ; ./script.sh`
- Direkt in der aktuellen Shell
`$source script.sh`
`$. script.sh`

Agenda



- Einleitung
- Aufruf
- **Aufbau**
- BASH
- Shellskripte schreiben
- Fehlererkennung

Aufbau von Shellskripten



- Folge von Shellkommandos (stdin von Textdatei)
- Zeilenumbrüche trennen Kommandos
<kommando1>
<kommando2>
- Kommentarzeichen #
- Escape Zeichen \

Agenda



- Einleitung
- Aufruf
- Aufbau
- **BASH**
- Shellskripte schreiben
- Fehlererkennung



! /bin/bash

- Standardmässig übergibt der Kernel sämtliche ausführbaren Textdateien an `/bin/sh`
- Eigentlich startet der Kernel `/bin/sh <datei`
- `#!<bin>` als erste Zeile ändert das Verhalten
- Maximale Länge 127 Zeichen
- Parameterübergabe ist möglich

Agenda



- Einleitung
- Aufruf
- Aufbau
- BASH
- **Shellskripte schreiben**
- Fehlererkennung

Shellskripte schreiben



- Ist ein Shellskript die richtige Lösung?
 - Keine komplexen Datenstrukturen
 - Kritische Masse \sim 100-1000 Zeilen
- Iteratives Vorgehen
- Erste Version = Kommando-Eingabe
- Kommentare!

Agenda



- Einleitung
- Aufruf
- Aufbau
- BASH
- Shellskripte schreiben
- **Fehlererkennung**



Fehlererkennung

- Shell kennt nur die Syntax von internen Kommandos
- Ablaufverfolgung mit Ausführung
`set -x`
- Ablaufverfolgung ohne Ausführung
`set -n`
- Ausgabe des Kommandos vor Ausführung
`set -v`

Fragen?

