

## 1.5.A Anwendung/Umgebungsvariablen – Arbeitsblatt

- Bearbeitungsdauer: 45 min
- Ergebnissicherung: Auf diesem Blatt
- Hilfsmittel: Lerntext

### Aufgabe 1: Inhalt spezieller Variablen

Manche Variablen sind bereits durch den Kommandointerpreter vorgelegt und haben eine spezielle Bedeutung. Bei der Shell (bzw. Bash) ist z. B. die Variable `PS1` zuständig für die Eingabeaufforderung (*prompt*). Ihren Inhalt erhält man so:

```
schueler@debian964:~$ echo "$PS1"
\u@\h:\w\$
schueler@debian964:~$
```

Bestimmen Sie ebenso den Inhalt der folgenden Variablen und finden Sie heraus, was die Bedeutung dieser Variablen sein könnte:

- a) Name: `BASH_VERSION`  
Inhalt:

Bedeutung:

- b) Name: `USER`  
Inhalt:

Bedeutung:

- c) Name: `HOME`  
Inhalt:

Bedeutung:

- d) Name: `LANG`  
Inhalt:

Bedeutung:

- e) Name: `PWD`  
Inhalt:

Bedeutung:

f) Name: OLDPWD  
Inhalt:

Bedeutung:

g) Name: RANDOM  
Inhalt:

Bedeutung:

h) Name: SECONDS  
Inhalt:

Bedeutung:

## Aufgabe 2: Ändern spezieller Variablen

Manche dieser speziellen Variablen können von Hand geändert werden, und das Verhalten des Kommandointerpreters ändert sich (leicht). Bei der Shell (bzw. Bash) kann man z. B. die Variable `PS1` ändern, und man erhält eine andere Eingabeaufforderung:

```
Terminal
schueler@debian964:~$ PS1="Hallo, gib was ein: "
Hallo, gib was ein:
Hallo, gib was ein:
```

Nun sollen Sie bestimmte Variablen setzen und sehen, was passiert:

a) Name: `PROMPT_COMMAND`  
Neuer Inhalt: `"echo -----"`  
Befehlszeile:

Ergebnis:

b) Name: `HOME`  
Neuer Inhalt: `"/tmp"`  
Befehlszeile:

Ergebnis bei Aufruf von `cd`:

c) Name: `HISTTIMEFORMAT`  
Neuer Inhalt: `"%d.%m.%Y, %H:%M:%S > "`  
Befehlszeile:

Ergebnis bei Aufruf von `history`:

### Aufgabe 3: alias statt Variable

Mit dem Shell-Befehl `alias` kann man Ersetzungen von Befehlen vornehmen:

```
Terminal
schueler@debian964:~$ k="cal 4 2015" # alias setzen
schueler@debian964:~$ k           # alias verwenden
    April 2015
So Mo Di Mi Do Fr Sa
      1  2  3  4
 5  6  7  8  9 10 11
12 13 14 15 16 17 18
19 20 21 22 23 24 25
26 27 28 29 30
```

Damit erlaubt man sich, in Zukunft `k` statt `cal 4 2015` eingeben zu können.

Nun sollen Sie genauso erlauben, dass man statt eines Linux-Befehls den entsprechenden Windows-Befehl eingeben kann.

- Erlauben Sie, `del` (Windows) statt `rm` (Linux) eingeben zu können!  
Befehlszeile:
- Erlauben Sie, `dir` statt `ls -l` eingeben zu können!  
Befehlszeile:
- Erlauben Sie, `copy` (Windows) statt `cp` (Linux) eingeben zu können!  
Befehlszeile:
- Erlauben Sie, `rename` statt `mv` eingeben zu können!  
Befehlszeile:
- Erlauben Sie, `cd ..` statt `cd ..` eingeben zu können!  
Befehlszeile:

### Aufgabe 4: Dauerhaftes Anbringen von Änderungen

Die Datei `.bashrc` in Ihrem persönlichen Verzeichnis (absoluter Pfadname: `~/.bashrc`) wird bei jedem Start einer Bash-Shell einmal eingelesen. Sie enthält möglicherweise bereits eine Reihe von Einstellungen, mit denen Ihre Arbeit personalisiert und vereinfacht werden kann.

- Öffnen Sie die Datei mit einem Editor, z. B. `gedit`.  
Befehlszeile:
- Fügen Sie am Schluss der Datei die folgende Zeile an:  
`"PATH=$PATH:."`  
Damit sorgen Sie dafür, dass die Shell auch in Ihrem momentanen Verzeichnis (abgekürzt durch den Punkt) nach Programmen sucht. Das hilft Ihnen später, wenn Sie selbst Programme schreiben und Sie diese Programme aufrufen wollen.  
Die letzte Zeile der Datei `.bashrc` im persönlichen Verzeichnis lautet jetzt also?

- c) Speichern Sie die Datei und schließen Sie den Editor. Lesen Sie die Datei ein mit dem Befehl:  
`source ~/.bashrc`  
Wie ist jetzt der Inhalt der Variablen `PATH`?

Hinweis: Genauso kann man auch die Ersetzungen aus der vorigen Aufgabe in derselben Datei speichern (in jede Zeile eine Ersetzung). Dann sind sie dauerhaft wirksam.