

Aufgabe 1:

```
#!/bin/bash

#Wie überprüfen sie, ob ein Shellskript mit Parametern aufgerufen wurde, oder ohne?

# "$#" gibt die Anzahl an Param. ab
#echo "$#"

# "$@" gibt alle Param in ein String ab
#echo "$@"

if [[ "$#" -eq 0 ]]; then
echo 'No Arguments!'
exit 1
else
# gebe alle Parameter in jeweils eine andere Zeile ab
for a in "$@"; do
echo "$a"
done
fi

read -n1 -r -p "Press any key to continue..." key
```

Aufgabe 2:

```
#!/bin/bash

# Wie schreiben sie in einem bash-Skript eine Schleife, die beliebig viele
Kommandozeilenparameter abarbeitet?

# "$@" gib alle Param in ein String
#echo "$@"

for a in "$@"; do
echo "$a"
#switch "$a"
done

read -n1 -r -p "Press any key to continue..." key
```

Aufgabe 3:

```
#!/bin/bash

DSTRING= ``
```

```

#grep
#r -> recursive
#n -> get the line Number if found
#I -> no Binary file
#o -> only matching

# Gibt den Dateinamen aus
DSTRING=`grep -rI Date: ${PWD}`

# oder geht auch mit
#DSTRING=`grep -rI Date: *`

read -n1 -r -p "Press any key to continue..." key

```

Aufgabe 4:

```

#!/bin/bash

DSTRING= ``

#grep
#r -> recursive
#n -> get the line Number if found
#I -> no Binary file
#o -> only matching

# Gibt den Dateinamen aus
#DSTRING=`grep -rI Date: * | cut -d ":" -f 1`
#email.eml storemail.sh storemail.sh storemail.sh storemail.sh~

# Gibt den Müll aus
#DSTRING=`grep -rI Date: * | cut -d ":" -f 2`
#Date DSTRING=`grep -rI Date DSTRING=`grep -rI Date #DSTRING=`grep -rI Date
DSTRING=`grep -rI Date #DSTRING=`grep -rI Date #DSTRING=`grep -rI Date

# Gibt den Datum aus (DAY, DD MM YYYY HH)
#DSTRING=`grep -rI Date: eml/ | cut -d ":" -f 3`
DSTRING=`grep -rI Date: Beispielmails/ | cut -d ":" -f 3`

# Gibt die Minuten anzahl aus (MM)
#DSTRING=`grep -rI Date: eml/ | cut -d ":" -f 4`
DSTRING=`grep -rI Date: Beispielmails/ | cut -d ":" -f 4`

# Gibt die Sekunden aus (SS)
#DSTRING=`grep -rI Date: eml/ | cut -d ":" -f 5`
DSTRING=`grep -rI Date: Beispielmails/ | cut -d ":" -f 5`

# Kontrolle ob DSTRING die Liste der Datum enstpricht
#echo ${DSTRING}

for a in "${DSTRING}"; do

```

```
echo "$a"
done
read -n1 -r -p "Press any key to continue..." key
```

ODER:

```
#!/bin/bash

_yea=""
_mnt=""
_day=""
_hrs=""
_min=""
_sec=""

DSTRING= ``

# Gibt den Datum aus
#DSTRING=`grep -rI Date: eml/ | cut -d ":" -f 3`
DSTRING=`grep -rI Date: Beispielmails/ | cut -d ":" -f 3`

# Mon, 19 Mar 2018 15
_yea=$(echo $DSTRING | cut -d " " -f 4)
_mnt=$(echo $DSTRING | cut -d " " -f 3)
_day=$(echo $DSTRING | cut -d " " -f 2)
_hrs=$(echo $DSTRING | cut -d " " -f 5)

_min=`grep -rI Date: Beispielmails/ | cut -d ":" -f 4`
_sec=`grep -rI Date: Beispielmails/ | cut -d ":" -f 5`

echo "-----"
echo "Year: $_yea"
echo "Mnth: $_mnt"
echo "Day : $_day"
echo "Hour: $_hrs"

echo "Min : $_min"
echo "Sec : $_sec"
echo "-----"

# Kontrolle ob DSTRING die Liste der Datum enstpricht
#echo ${DSTRING}

for a in "${DSTRING}"; do
echo "$a"
done

read -n1 -r -p "Press any key to continue... " key
```

Aufgabe 5:

```
#!/bin/bash
```

```
echo "Year: $_yea"
```

```
echo "Mnth: $_mnt"
```

```
echo "Day : $_day"
```

```
echo "Hour: $_hrs"
```

```
echo "Min : $_min"
```

```
echo "Sec : $_sec"
```

```
echo "-----"
```

```
case "$_mnt" in
```

```
"Jan")
```

```
    _mnt=1
```

```
;;
```

```
"Feb")
```

```
    _mnt=2
```

```
;;
```

```
"Mar")
```

```
    _mnt=3
```

```
;;
```

```
"Apr")
```

```
    _mnt=4
```

```
;;
```

```
"Mai")
```

```
    _mnt=5
```

```
;;
```

```
"Jun")
```

```
    _mnt=6
```

```
;;
"Jul")
    _mnt=7
;;
"Aug")
    _mnt=8
;;
"Sep")
    _mnt=9
;;
"Okt")
    _mnt=10
;;
"Nov")
    _mnt=11
;;
"Dez")
    _mnt=12
;;
*)
    echo "Fuck! Month not found!!!"
;;
esac

echo $_mnt
```

```
read -n1 -r -p "Press any key to continue... " key
```

Aufgabe 6:

```
#!/bin/bash
```

```
mkdir -p Shit/a/b/g/a/u
```

#Wenn das Verzeichnis schon vorhanden ist, werden die nächsten Vewrzeichnisse einfach in den existenten Ordner gecrieben.

Aufgabe 7:

```
#modification time ändern
```

```
touch -a -m -t 201512180130.09 fileName.ext
```

```
# -a = accessed
```

```
# -m = modified
```

```
# -t = timestamp - use [[CC]YY]MMDDhhmm[.ss] time format
```

Aufgabe 8:

Wie können sie erreichen, dass sie das Shellskript storemails alleine durch Aufrufendes Namens starten können?

#Die Umgebungsvar. PATH ändern und den Verzeichnis von Storemails.sh hinzufügen

#Oder Stotrmails.sh in /bin kopieren