Shell Scripting

Inleiding tot Scripting

DE HOGESCHOOL
MET HET NETWERK

Hogeschool PXL – Dep. PXL-IT – Elfde-Liniestraat 26 – B-3500 Hasselt
www.pxl.be - www.pxl.be/facebook

Een script aanmaken

- Aanmaken van een nieuw script
  - doe je best in een nieuwe map “bin” van je homedir
    - cd
      mkdir bin
      cd bin
      vi <scriptnaam> OF nano <scriptnaam>
  - een script kan je de extensie “.sh” geven

Een script aanmaken

- Interpreter
  - geeft aan wat er moet worden gebruikt om de commando's te verstaan (interpreteren)
  - wordt gespecificeerd op de eerste regel van het script
  - she-bang (eerste twee tekens)
    - `#!/bin/bash --`
      - De laatste twee min-tekens kunnen worden geplaatst uit veiligheid

Een script aanmaken

- Commentaar
  - regel die start met een #-teken
    - # Auteur: Gert Van Waeyenberg
    - # Datum: 20 oktober 2019
    - # Versie: 1.0
    - # Gebruik: ./script.sh <parameter:getal>
  - of ergens in de regel
    - vanaf het #-teken begint de commentaar
      - vDatum=2019   #plaats in deze variabele het huidig jaartal

Een script aanmaken

- Commando’s
  - de bedoeling van een script is om meerdere commando’s samen te brengen om zo een welbepaald klein programma te vormen
  - naar automatisatie toe is het makkelijker om bijvoorbeeld dagelijks een script uit te voeren, dan alle commando’s afzonderlijk

Een script aanmaken

- Uitvoerbaar maken
  - een script kan uitvoerbaar gemaakt worden
  - als je een zelfgemaakt script enkel uitvoerbaar wilt maken voor jezelf, kan je dit met volgend commando:
    - chmod u+x <scriptnaam>

Een script aanmaken - voorbeeld

```
vwg@laptop: ~/bin
vwg@laptop:~/bin$ vi voorbeeldscript.sh
```

```
vwg@laptop: ~/bin
#!/bin/bash --

# Auteur: Gert Van Waeyenberg
# Datum:  20 oktober 2024
# Versie: 1.0

echo "Mijn eerste script"

:wq
```

```
vwg@laptop: ~/bin
vwg@laptop:~/bin$ chmod u+x voorbeeldscript.sh
```

```
vwg@laptop: ~/bin
vwg@laptop:~/bin$ ./voorbeeldscript.sh
Mijn eerste script
vwg@laptop:~/bin$
```

Input vragen

- Input vragen
  - tijdens de uitvoer van een script kan er naar input gevraagd worden
    - We stellen eerst de vraag
      - `echo -n "Geef een getal:"`
        - de optie -n zorgt er voor dat de cursor achter de vraag blijft staan
    - Dan vragen we een waarde en kennen deze toe aan een variabele
      - `read vGetal`
        - Deze variabele kunnen we verder gebruiken in het script

Input vragen - voorbeeld

vwg@laptop: ~/bin

```
#!/bin/bash --

# Auteur: Gert Van Waeyenberg
# Datum:  20 oktober 2024
# Versie: 1.1

echo -n "Geef een getal:"
read vGetal
echo "Het getal dat u gaf was: $vGetal"

~
:wq
```

vwg@laptop: ~/bin

```
vwg@laptop:~/bin$ ./voorbeeldscript.sh
Geef een getal:19
Het getal dat u gaf was: 19
vwg@laptop:~/bin$
```

Het test-commando

- test-commando
  - wordt gebruikt om waarden te vergelijken
    - test $vGetal -gt 100 && echo “Groter” || echo “Kleiner”
      - Indien de inhoud van de variabele groter is dan 100 wordt de tekst “Groter” getoond, anders wordt de tekst “Kleiner” getoond
    - [$vGetal -gt 100] && echo “Groter” || echo “Kleiner”
      - Dit is de verkorte schrijfwijze van het test-commando
      - Let wel op de spatie die verplicht is aan de binnenkant van de vierkante haken

Het test-commando

- test-commando
  - operatoren
    - **-lt:less than**
    - **-gt:greater than**
    - **-ge:greather or equal to**
    - **-le:less or equal to**
    - **-eq:equal to**
    - **-ne:not equal to**

    - **= :equals a string**
    - **!= :Not equals a string**

    - **-d:does dir exist**
    - **-f:does file exist**

Het test-commando - voorbeeld

vwg@laptop: ~/bin

```
#!/bin/bash --

# Auteur: Gert Van Waeyenberg
# Datum:  20 oktober 2024
# Versie: 1.1

echo -n "Geef een getal van 1 tot 20:"
read vGetal
echo "Het getal dat u gaf was: $vGetal"

test $vGetal -lt 10 && echo "Het getal is kleiner dan 10" || echo "Het getal is groter dan 10"

[ $vGetal -lt 20 ] && echo "Het getal is kleiner dan 20" || echo "Het getal is te groot!"

~
~
"voorbeeldscript.sh" 14L, 366C                                1,1           All
```

vwg@laptop: ~/bin

```
vwg@laptop:~/bin$ ./voorbeeldscript.sh
Geef een getal van 1 tot 20:16
Het getal dat u gaf was: 16
Het getal is groter dan 10
Het getal is kleiner dan 20
vwg@laptop:~/bin$
```

Het test-commando

- test-commando
  - Testen op meerdere expressies tegelijk
    - -a voor de AND-operator
    - -o voor de OR-operator

Het test-commando - voorbeeld

vwg@laptop: ~/bin

```
#!/bin/bash --

# Auteur: Gert Van Waeyenberg
# Datum:  20 oktober 2024
# Versie: 1.3

echo -n "Geef een getal van 1 tot 20:"
read vGetal

[ $vGetal -lt 0 -o $vGetal -gt 20 ] && echo "U gaf geen getal van 1 tot 20" || echo "Het getal da
t u gaf is: $vGetal"

                                                                    1,1           All
```

vwg@laptop: ~/bin

```
vwg@laptop:~/bin$ ./voorbeeldscript.sh
Geef een getal van 1 tot 20:-3
U gaf geen getal van 1 tot 20
vwg@laptop:~/bin$ ./voorbeeldscript.sh
Geef een getal van 1 tot 20:15
Het getal dat u gaf is: 15
vwg@laptop:~/bin$ ./voorbeeldscript.sh
Geef een getal van 1 tot 20:27
U gaf geen getal van 1 tot 20
vwg@laptop:~/bin$
```

if-then-else

- if-then-else
  - in plaats van met && en || te werken is het veel duidelijker om te werken met if-then-else
  - de if-then-else-structuur wordt samen gebruikt met het test-commando of zijn verkorte schrijfwijze
  - Opgelet: deze structuur wordt afgesloten met “fi”
    - Dus:

```
if
then
else
fi
```

if-then-else - voorbeeld

vwg@laptop: ~/bin

```
#!/bin/bash --

# Auteur: Gert Van Waeyenberg
# Datum:  20 oktober 2024
# Versie: 1.4

echo -n "Geef een getal van 1 tot 20:"
read vGetal

if [ $vGetal -lt 0 -o $vGetal -gt 20 ]
then
   echo "U gaf geen getal van 1 tot 20"
else
   echo "Het getal dat u gaf is: $vGetal"
fi
                                                              1,1           All
```

if-then-elif

- if-then-elif
  - Je kan if-then-else -structuren in elkaar nesten met if-then-elif
  - Syntax:

```
if
then
elif
then
elif
then
...
else
then
fi
```

Dit is een opbouwende structuur:

Indien expr1
dan ...
anders indien expr2 (en dus niet expr1)
dan ...
anders indien expr3 (en dus niet expr1 en ook niet expr2)
dan ...
anders (wilt dus zeggen in alle andere gevallen)
dan ...
ende indien

if-then-elif - voorbeeld

vwg@laptop: ~/bin

```
#!/bin/bash --

# Auteur: Gert Van Waeyenberg
# Datum:  20 oktober 2024
# Versie: 1.5

echo -n "Geef een getal van 1 tot en met 20:"
read vGetal

if [ $vGetal -le 0 ]
then
   echo "U gaf een te klein getal"
elif  [ $vGetal -lt 10 ]
then
   echo "U gaf een geldig getal in de range van 1 tot en met 9"
elif  [ $vGetal -le 20 ]
then
   echo "U gaf een geldig getal in de range van 10 tot en met 20"
else
   echo "U gaf een te groot getal"
fi
                                                              1,1           All
```

for-loop

```
for <expressie>
do
   commando's
done
```

- for-loop
  - Om de commando's die tussen de do en done van de for-lus staan meerdere keren uit te voeren
  - de expressie bestaat uit een variabele die gebruikt wordt als teller en een range die de telling aangeeft
    - De telling kan bestaan uit
      - losse items (for teller in 1 2 3 4 5)
      - een range (for teller in {1..5}) (for teller in `seq 1 5`)
      - bestanden verkregen door file-globbing (for file in `ls *`)

for-loop - voorbeeld

vwg@laptop: ~/bin

```
#!/bin/bash --

# Auteur: Gert Van Waeyenberg
# Datum:  20 oktober 2024
# Versie: 1.0

echo -n "Tot welk getal wil je dat ik tel:"
read vGetal

echo "OK, hier beginnen we..."

for vTeller in `seq 1 $vGetal`
do
   echo -n "$vTeller "
done

echo " "                                        # prompt op volgende lijn
~
                                                              1,1           All
```

vwg@laptop: ~/bin

```
vwg@laptop:~/bin$ ./voorbeeldscript2.sh
Tot welk getal wil je dat ik tel:5
OK, hier beginnen we...
1 2 3 4 5
vwg@laptop:~/bin$
```

for-loop - voorbeeld 2

vwg@laptop: ~/bin

```
#!/bin/bash --

# Auteur: Gert Van Waeyenberg
# Datum:  20 oktober 2024
# Versie: 1.0

echo -n "Van welke directory wil je de bestanden zien?:"
read vDir

echo "Bestanden van $vDir"

for vBestand in `ls $vDir`
do
   if [ -f $vDir/$vBestand ]
   then
      echo -n "$vBestand - "
   else
      echo -en "\e[94m$vBestand\e[39m - "              # directories in blauw
   fi
done

echo " "                                             # prompt op volgende lijn
~
                                                                1,1           All
```

vwg@laptop: ~/bin

```
vwg@laptop:~/bin$ ./voorbeeldscript3.sh
Van welke directory wil je de bestanden zien?:/home/vwg
Bestanden van /home/vwg
bin - Calibre - Library - Desktop - Documents - Downloads - examples.desktop - f
orever.sh - fotowall-background1.jpg - launchy.db - launchy.ini - lijst - lijst2
 - mijnrapport - mijnscript.sh - mijntekst - Music - PDF - persoonlijk - Picture
s - PlayOnLinux's - virtual - drives - Public - selectscript2.sh - selectscript.
sh - Templates - test - test! - testdir - Videos - vmware - weby-icon-cache -
vwg@laptop:~/bin$
```

http://misc.flogisoft.com/bash/tip_colors_and_formatting

while-loop

```
while <voorwaarde>
do
   commando's
done
```

- while-loop
  - Om de commando's die tussen de "do" en "done" van de while staan te blijven herhalen **zolang als** aan de voorwaarde voldaan is
  - als voorwaarde gebruikt men meestal het test-commando
    - Kan bvb. gebruikt worden om de vraag naar invoer te herhalen totdat een juist commando wordt ingegeven

while-loop - voorbeeld

vwg@laptop: ~/bin

```
#!/bin/bash --

# Auteur: Gert Van Waeyenberg
# Datum:  20 oktober 2024
# Versie: 1.0

vGetal=0

while [ $vGetal -lt 1  -o  $vGetal -gt 10 ]
do
   echo -n "Geef een getal van 1 tot en met 10:"
   read vGetal
done

echo "U heeft het getal $vGetal opgegeven!"
                                                    1,1           Top
```

vwg@laptop: ~/bin

```
vwg@laptop:~/bin$ ./voorbeeldscript5.sh
Geef een getal van 1 tot en met 10:-3
Geef een getal van 1 tot en met 10:0
Geef een getal van 1 tot en met 10:11
Geef een getal van 1 tot en met 10:10
U heeft het getal 10 opgegeven!
vwg@laptop:~/bin$
```

while-loop - voorbeeld 2

vwg@laptop: ~/bin

```
#!/bin/bash --

# Auteur: Gert Van Waeyenberg
# Datum:  20 oktober 2024
# Versie: 1.0

vGetal=$(( ( RANDOM % 10 ) + 1 ))
vGok=0

while [ $vGok -ne $vGetal ]
do
   echo -n "Raad een getal van 1 tot en met 10:"
   read vGok
done

echo "U heeft het getal $vGetal geraden !"
                                                    1,1           Top
```

vwg@laptop: ~/bin

```
vwg@laptop:~/bin$ ./voorbeeldscript4.sh
Raad een getal van 1 tot en met 10:1
Raad een getal van 1 tot en met 10:2
U heeft het getal 2 geraden !
vwg@laptop:~/bin$
```

until-loop

```
until <voorwaarde>
do
   commando's
done
```

- until-loop
  - Om de commando's die tussen de "do" en "done" van de until staan te blijven herhalen **totdat** aan de voorwaarde voldaan is
  - als voorwaarde gebruikt men meestal het test-commando
    - Kan bvb. gebruikt worden om de vraag naar invoer te herhalen totdat een juist commando wordt ingegeven

until-loop - voorbeeld

vwg@laptop: ~/bin

```
#!/bin/bash --

# Auteur: Gert Van Waeyenberg
# Datum:  20 oktober 2024
# Versie: 1.0

vGetal=0

until [ $vGetal -ge 1  -a  $vGetal -le 10 ]
do
   echo -n "Geef een getal van 1 tot en met 10:"
   read vGetal
done

echo "U heeft het getal $vGetal opgegeven!"
                                                    1,1           Top
```

vwg@laptop: ~/bin

```
vwg@laptop:~/bin$ ./voorbeeldscript6.sh
Geef een getal van 1 tot en met 10:-1
Geef een getal van 1 tot en met 10:0
Geef een getal van 1 tot en met 10:11
Geef een getal van 1 tot en met 10:10
U heeft het getal 10 opgegeven!
vwg@laptop:~/bin$
```

until-loop - voorbeeld 2

vwg@laptop: ~/bin

```
#!/bin/bash --

# Auteur: Gert Van Waeyenberg
# Datum:  20 oktober 2024
# Versie: 1.0

vGetal=$(( ( RANDOM % 10 ) + 1 ))
vGok=0

until [ $vGok -eq $vGetal ]
do
   echo -n "Raad een getal van 1 tot en met 10:"
   read vGok
done

echo "U heeft het getal $vGetal geraden !"
                                                    1,1           All
```

vwg@laptop: ~/bin

```
vwg@laptop:~/bin$ ./voorbeeldscript7.sh
Raad een getal van 1 tot en met 10:4
Raad een getal van 1 tot en met 10:5
Raad een getal van 1 tot en met 10:6
U heeft het getal 6 geraden !
vwg@laptop:~/bin$
```