ServerOS Essentials

*Week 10 - Oefeningen*

* Pas je prompt aan, zodanig dat deze enkel uit het “>”-teken bestaat

Zorg ervoor dat de prompt opnieuw op de originele waarde komt, zonder uit te loggen, te herstarten of een sub-shell te starten. Hint: de prompt wordt gezet in ~/.bashrc

***student@studentvm:~$ PS1=”>”***

***student@studentvm:~$ ./.bashrc***

* Maak een script dat de naam van het script teruggeeft, alsook de eerste parameter: Let op: De naam van het script en niet het volledig pad.

Voorbeeld:

student@desktop:~$ /home/student/bin/oef23\_03.sh 12345

De naam van het script is: oef23\_03.sh

De eerste parameter is: 12345

***#! /bin/bash***

***echo "De naam van het script is:" $(echo $0 | rev | cut -d/ -f1 | rev)***

***echo "De*** [***eerste***](https://www.messenger.com/t/Limoentaart#) ***parameter is:" $1***

* Maak een script, genaamd grootstegetal.sh, dat het grootste getal teruggeeft van de getallen die als parameter worden meegegeven.

***#! /bin/bash  
biggest=$1  
shift  
while [[ $1 != "" ]]  
do  
  
test $1 -gt $biggest && biggest=$1  
  
shift  
done  
echo "Het grootste getal was" $biggest***

* Pas het voorgaand script aan.

Voeg een functie, genaamd syntax, toe.

Deze functie toont een foutmelding en de juiste syntax waarmee het script moet worden aangeroepen

U hebt een onjuist commando opgegeven Syntax: script.sh getal1 getal2 getal3 … getaln

De functie eindigt ook de uitvoer van het script met een foutcode.

Indien er geen parameter wordt meegegeven, moet de foutmelding en juiste syntax worden getoond.

***#! /bin/bash***

***function syntax  
{  
 echo "U hebt een onjuist commando opgegeven Syntax: script.sh getal1 getal2 getal3 ... getaln"  
 exit -1  
}  
  
if [[ $1 == "" ]]  
then  
syntax  
fi  
  
biggest=$1  
shift  
while [[ $1 != "" ]]  
do  
  
test $1 -gt $biggest && biggest=$1  
  
shift  
done   
echo "Het grootste getal was" $biggest***

* Pas het voorgaand script aan.

Controleer of alle parameters ook getallen zijn. Indien het niet allemaal getallen zijn, dan dient de juiste syntax getoond te worden.

***#! /bin/bash  
  
function syntax  
{  
 echo "U hebt een onjuist commando opgegeven Syntax: script.sh getal1 getal2 getal3 ... getaln"  
 exit -1  
}  
  
if [[ $1 == "" ]]  
then  
syntax  
fi  
  
if ! [[ $1 =~ ^[0-9]+$ ]]  
then  
syntax  
fi  
  
biggest=$1  
shift  
while [[ $1 != "" ]]  
do  
if ! [[ $1 =~ ^[0-9]+$ ]]  
then  
syntax  
fi  
test $1 -gt $biggest && biggest=$1  
shift  
done  
echo "Het grootste getal was" $biggest***

* Maak een scriptje dat twee getallen als parameters ontvangt. In het script worden deze parameters doorgegeven aan een funtie, genaamd vermenigvuldig. Deze functie vermenigvuldigd de twee parameter-waarden met behulp van het commando “let” en stopt de uitkomst in een nieuwe variabele uitkomst. Print buiten de functie de waarde van de variabele uitkomst af naar het scherm.

***#! /bin/bash  
function vermenigvuldig  
{  
 let uitkomst=$1\*$2  
}  
vermenigvuldig $1 $2  
echo $uitkomst***

* Probeer vorig scriptje, maar nu met volgende regel toe te voegen als nieuwe eerste regel van de functie: local uitkomst=0.

Werkt dit nog? Waarom/Waarom niet?

***#! /bin/bash  
function vermenigvuldig  
{***

***local uitkomst=0  
 let uitkomst=$1\*$2  
}  
vermenigvuldig $1 $2  
echo $uitkomst***

***Nee, dit werkt niet meer omdat ‘uitkomst’ nu een locale variabele in de functie is en niet buiten dat bereik gekend is.***

* Los vorig probleem op door in de functie de waarde te tonen via het echo-commando. Vang deze echo op in een variabele buiten de functie (door gebruik te maken van een subshell) en print vervolgens deze variabele af buiten de functie.

***#! /bin/bash  
function vermenigvuldig  
{  
 local uitkomst=0  
 let uitkomst=$1\*$2  
 global=$(echo $uitkomst)  
}  
vermenigvuldig $1 $2  
echo $global***

* Maak een scriptje, genaamd list.sh, dat :

– de opties a,d en l opvangt

– de parameter a laat alle files zien (ook de hidden files)

– de parameter l laat de files in een lijst zien ipv in kolommen

– de optie d ontvangt ook een parameter (de directory om te listen)

– een nederlandstalige foutboodschap toont indien een foute optie wordt opgegeven

– dezelfde nederlandstalige foutboodschap toont indien de optie h wordt meegegeven

– een gewone ls doet van de huidige directory indien geen opties werden meegegeven

– een ls uitvoert met de nodige meegegeven opties van de eventueel meegegeven directory

***#! /bin/bash  
  
function foutboodschap  
{  
 echo "Er is een foutieve optie" $1 "meegegeven"  
 exit -1  
}  
  
commando="ls"  
  
while getopts ":ald:" option  
do  
 case $option in  
 a) commando="$commando -a" ;;  
 l) commando="$commando -l" ;;  
 d) commando="$commando $OPTARG" ;;  
 \*) foutboodschap ;;  
 esac  
done  
  
$commando***

* + Maak een quiz die een random getal van 0 tot 9 laat raden. Geef bij het juist raden weer in hoeveel pogingen het is geraden. Indien het getal geraden wordt in 4 pogingen of minder, dan dien je de quizzer ook te feliciteren.

Voor een random getal te berekenen gebruik je volgende regels:

randomgetal=$(( RANDOM % 10 )) ​#rest na /10 (modulo 10) = getal van 0-9

***#! /bin/bash  
  
randomgetal=$(( RANDOM % 10))  
  
echo "Geef een getal in van 0-9:"  
read gok  
aantalGokken=1  
  
while [[ $gok != $randomgetal ]]  
do  
echo "Dat was niet correct! Probeer opnieuw een getal in te geven van 0-9:"  
read gok  
let aantalGokken=$aantalGokken+1  
done  
  
test $aantalGokken -le 4 && echo "Proficiat, je hebt het in 4 gokken of minder $***