



Laboration – Grundläggande skriptning

OBS det kan förekomma små skillnader vad gäller sökvägar samt befintliga kommandon mellan olika distributioner. T.ex. mellan CentOS och Ubuntu,

Material: För att genomföra laborationen behöver man ha tillgång till en dator (vm) med Linux installerat. Windows Subsystem for Linux går att använda också. För vissa övningar kan lokal administratörsbehörighet behövas. I labben utgår det från att man är inloggad med en användare som heter **sysadmin**

Mål: I denna laboration kommer du att utföra följande uppgifter:

- Använd vi-redigeraren för att skapa och redigera textfiler.
- Skapa enkla shell-skript.
- Skapa shell-skript med villkorsstyrd körning.
- Använd loopar i skriptet för upprepning.

Grundläggande texteditering

De flesta Linux-distributioner har fler än en texteditor. Dessa kan inkludera enkla textbaserade editorer, som **nano**, eller grafiska editorer, som **gedit**.

I denna uppgift kommer du att utforska några av de grundläggande funktionerna för textredigering i **vi**-editorn. Alla distributioner har någon version av **vi**. **vi**-editorn är en kraftfull texteditor med en viss inlärningskurva, men den klarar av att hantera en mängd olika textredigeringsuppgifter.

vi-editorn har två lägen: insert (infoga) och command (kommando). I insert-läget lägger du till text i ett dokument. I command-läget utförs operationer som navigering, sökning, sparande och avslutning av editorn.

För att skapa en ny fil, kör följande kommando:

```
vi myfile
```

```
sysadmin@localhost:~$ vi myfile
```

Tryck på **i** för att gå in i **vi**:s "insert"-läge (mer om detta senare). Skriv sedan in följande text:

```
Welcome to the vi editor.  
It is a very powerful text editor.  
Especially for those who master it.
```

Tryck sedan på **Esc** för att lämna insert-läget. Skriv **:wq** för att spara filen och avsluta.

Observera

Varje kommando ovan kommer att gås igenom mer i detalj senare i denna labb. Syftet med stegen ovan var att skapa en fil att arbeta med under labben.

Starta **vi**-editorn för att redigera filen du skapade. När **vi** startas hamnar du som standard i *kommandoläge* :

```
vi myfile
```



```
sysadmin@localhost:~$ vi myfile
```

Ditt resultat ska likna följande:

```
Welcome to the vi editor.  
It is a very powerful text editor.  
Especially for those who master it.  
~  
~  
~  
~  
~  
~  
~  
"myfile" 3 lines, 97 characters
```

Observera att längst ner till vänster visas filnamnet, antal rader och antal tecken i filen.

Tryck på var och en av följande tangenter två gånger och se hur markören rör sig. Kom ihåg att du är i kommandoläget:

Tangent Funktion

j	Flyttar markören en rad ner (samma som nedåtpilen)
k	Flyttar markörens uppåtlinje (samma som upp-pilen)
l	Flyttar markören till höger ett tecken (samma som höger-pilen)
h	Flyttar markören till vänster ett tecken (som vänster pil)
w	Flyttar markören till början av nästa ord
e	Flyttar markören till slutet av ordet
b	Flyttar markören till början av föregående ord

Warning: Om du trycker på några andra tangenter än de som står ovan kan du hamna i insert-läge. Ingen fara! Tryck på **Esc** och skriv sedan **:q!** följt av **Enter**. Då avslutas **vi** utan att spara ändringar. Starta sedan **vi myfile** igen så är du tillbaka i **vi**-editorn!

Fler vi-kursornavigeringar: tryck på följande tangenter och se hur markören flyttar sig:

Tangent Funktion

\$	Flyttar markören till slutet av nuvarande rad (samma som END -tangenter)
0 (zero)	Flyttar markörens början på nuvarande rad (samma som HOME -tangenter)
3G	Hoppar till tredje linjen (nG hoppar till n:te linjen)
1G	Hopp till första linjen
Shift+G	Hoppar till sista raden

Flytta markören till början av ordet "very" genom att trycka på följande tangenter:

```
3G (håll ned siffran "3" och tryck på bokstaven "g")
```

```
k
```

```
8l (det är siffran åtta följt av bokstaven "l")
```



Markören ska nu stå på bokstaven **V** i ordet "very" som visas nedan:

```
Welcome to the vi editor.  
It is a Very powerful text editor.  
Especially for those who master it.  
~  
~
```

Ta bort ordet "very" genom att använda kommandot **dw** (**d** elete **w** ord):

```
dw
```

Din skärm ska nu se ut ungefär så här:

```
Welcome to the vi editor.  
It is a powerful text editor.  
Especially for those who master it.  
~
```

Ångra den senaste åtgärden:

```
u
```

Din skärm ska nu se ut ungefär så här:

```
Welcome to the vi editor.  
It is a Very powerful text editor.  
Especially for those who master it.  
~  
~
```

Ta bort två ord:

```
2dw
```

Din skärm ska nu se ut ungefär så här:

```
Welcome to the vi editor.  
It is a text editor.  
Especially for those who master it.  
~  
~
```

Ångra den senaste åtgärden:

```
u
```

Din skärm ska nu se ut ungefär så här:

```
Welcome to the vi editor.  
It is a Very powerful text editor.  
Especially for those who master it.  
~  
~
```

Ta bort fyra tecken, ett i taget:

```
xxxx
```



Din skärm ska nu se ut ungefär så här:

```
Welcome to the vi editor.  
It is a powerful text editor.  
Especially for those who master it.  
~  
~
```

Ångra de senaste fyra åtgärderna och återställ de borttagna tecknen:

```
4u
```

Din skärm ska nu se ut ungefär så här

```
Welcome to the vi editor.  
It is a very powerful text editor.  
Especially for those who master it.  
~  
~
```

Ta bort 14 tecken:

```
14x
```

Din skärm ska nu se ut ungefär så här:

```
Welcome to the vi editor.  
It is a text editor.  
Especially for those who master it.  
~  
~
```

Ångra den senaste åtgärden:

```
u
```

Din skärm ska nu se ut ungefär så här:

```
Welcome to the vi editor.  
It is a very powerful text editor.  
Especially for those who master it.  
~  
~
```

Ta bort de fem tecknen till vänster om markören (skriv **5** och tryck sedan **Shift+x**):

```
5X
```

Din skärm ska nu se ut ungefär så här:

```
Welcome to the vi editor.  
It very powerful text editor.  
Especially for those who master it.  
~  
~
```

Ångra den senaste åtgärden:

```
u
```



Din skärm ska nu se ut ungefär så här:

```
Welcome to the vi editor.  
It is a very powerful text editor.  
Especially for those who master it.  
~  
~
```

Ta bort den aktuella raden:

```
dd
```

Din skärm ska nu se ut ungefär så här:

```
Welcome to the vi editor.  
Especially for those who master it.  
~  
~
```

Det som senast raderades eller kopierades kan "klistras in". Klistra in de borttagna raderna under den aktuella raden:

```
p
```

Din skärm ska nu se ut ungefär så här:

```
Welcome to the vi editor.  
Especially for those who master it.  
It is a very powerful text editor.  
~  
~
```

Ångra de två senaste åtgärderna:

```
2u
```

Din skärm ska nu se ut ungefär så här:

```
Welcome to the vi editor.  
It is a very powerful text editor.  
Especially for those who master it.  
~  
~
```

Ta bort två rader, den aktuella och nästa:

```
2dd
```

Din skärm ska nu se ut ungefär så här:

```
Welcome to the vi editor.  
~  
~
```

Ångra den senaste åtgärden:

```
u
```



Din skärm ska nu se ut ungefär så här:

```
Welcome to the vi editor.  
It is a very powerful text editor.  
Especially for those who master it.  
~  
~
```

Gå till det fjärde ordet och radera sedan från den aktuella positionen till radens slut **Shift+D** :

```
4w  
D
```

Din skärm ska nu se ut ungefär så här:

```
Welcome to the vi editor.  
It is a very  
Especially for those who master it.  
~  
~
```

Kommando **d\$** raderar också till radens slut. Tecknet **\$** tar dig, som tidigare nämnt, till slutet av raden. Alltså raderar **d\$** till radens slut.

Ångra den senaste åtgärden:

```
u
```

Din skärm ska nu se ut ungefär så här:

```
Welcome to the vi editor.  
It is a very powerful text editor.  
Especially for those who master it.  
~  
~
```

Slå ihop två rader, den nuvarande och nästa, genom att skriva ett stort **J** (**Shift+J**) :

```
J
```

Din skärm ska nu se ut ungefär så här:

```
Welcome to the vi editor.  
It is a very powerful text editor. Especially for those who  
master it.  
~  
~
```

Ångra den senaste åtgärden:

```
u
```

Din skärm ska nu se ut ungefär så här:



```
Welcome to the vi editor.  
It is a very powerful text editor.  
Especially for those who master it.  
~  
~
```

Kopiera (eller "yanka") det aktuella ordet:

```
yw
```

När du kopierar text händer inget synligt på skärmen.

Klistra in (eller "pasta") det kopierade ordet före den aktuella markören genom att skriva **Shift+p** :

```
P
```

Din skärm ska nu se ut ungefär så här:

```
Welcome to the vi editor.  
It is a very powerfulpowerful text editor.  
Especially for those who master it.  
~  
~
```

Ångra den senaste åtgärden:

```
u
```

Din skärm ska nu se ut ungefär så här:

```
Welcome to the vi editor.  
It is a very powerful text editor.  
Especially for those who master it.  
~  
~
```

Flytta till första raden, och slå ihop tre rader:

```
1G
```

```
3J
```

Din skärm ska nu se ut ungefär så här:

```
Welcome to the vi editor. It is a very powerful text editor. Esp  
~  
~
```

Ångra den senaste åtgärden:

```
u
```

Din skärm ska nu se ut ungefär så här:



```
Welcome to the vi editor.
It is a very powerful text editor.
Especially for those who master it.
~
~
```

Sök efter och ta bort ordet text (lägg till ett mellanslag efter ordet text):

```
:%s/text //g
```

Din skärm ska nu se ut ungefär så här:

```
Welcome to the vi editor.
It is a very powerful editor.
Especially for those who master it.
~
~
~
:%s/text //g
```

Gå till början av filen, tryck sedan på **i** för att gå in i insert-läge och lägga till text:

Tangent	Funktion
1G	Gå till början av filen (Shift+G)
i	Gå in i insättningsläge
Hello and	Lägg till text i dokumentet med mellanslag efter "och"

```
Hello and Welcome to the vi editor.
It is a very powerful editor.
Especially for those who master it.
~
~
```

Avsluta insättningsläget och gå tillbaka till kommandoläget genom att trycka på **Escape** -tangenten:

```
ESC
```

Flytta fram ett steg genom att trycka på lilla **l**, så att markören hamnar på **w**, och växla till liten bokstav genom att trycka på tilde (**~**):

Tangent	Funktion
l	Liten 'l' flyttas fram ett steg
~	Shift+` Ändrar bokstav till gemener

Din skärm ska nu se ut ungefär så här:

```
Hello and welcome to the vi editor.
It is a very powerful editor.
Especially for those who master it.
~
~
```



Spara filen. Tryck på **Esc** för att försäkra dig om att du är i kommandoläge. Skriv sedan **:w** och tryck på **Enter** :

```
:w
```

När du trycker på **Enter** för att bekräfta ändringen, lägg märke till meddelandet nere till vänster som visar att filen har sparats:

```
~
~
"myfile" 3 lines, 102 characters written
```

Navigera till mellanrummet mellan orden "powerful" och "editor" på andra raden, precis som visas på bilden nedan. Du kan trycka på **j** följt av **10l** eller använda piltangenterna.

Command Function/Keys

j	Flytta ner till andra linjen
10l	10 följt av gemener 'L'

Din skärm ska nu se ut ungefär så här:

```
Hello and welcome to the vi editor.
It is a very powerful editor.
Especially for those who master it.
~
~
```

Lägg till text till höger om markören genom att trycka på bokstaven **a**. Detta flyttar markören åt höger och aktiverar insättningsläge. Skriv ordet **text** följt av ett mellanslag, precis som visas på bilden nedan:

Kommando Funktion

a	Gå in i insättningsläget.
text	text följt av ett mellanslag

```
Hello and welcome to the vi editor.
It is a very powerful text editor.
Especially for those who master it.
~
~
```

Avsluta insättningsläget genom att trycka på **Esc** -tangenten.

Öppna en tom rad under den nuvarande raden genom att skriva en liten bokstav **o**:

```
o
```

Din skärm ska nu se ut ungefär så här:

```
Hello and welcome to the vi editor.
It is a very powerful text editor.
|
Especially for those who master it.
~
~
```



Mata in följande text:

```
This line was added by pressing lowercase o.
```

Din skärm ska nu se ut ungefär så här:

```
Hello and welcome to the vi editor.  
It is a very powerful text editor.  
This line was added by pressing lowercase o.  
Especially for those who master it.  
~  
~
```

Avsluta insättningsläget genom att trycka på **Esc**-tangentsen.

Öppna en tom rad ovanför den nuvarande raden genom att skriva stor bokstav **O**:

```
O
```

Din skärm ska nu se ut ungefär så här:

```
Hello and welcome to the vi editor.  
It is a very powerful text editor.  
  
This line was added by pressing lowercase o.  
Especially for those who master it.  
~  
~
```

Ange följande text:

```
You just pressed O to open a line above.
```

Din skärm ska se ut ungefär så här:

```
Hello and welcome to the vi editor.  
It is a very powerful text editor.  
You just pressed O to open a line above.█  
This line was added by pressing lowercase o.  
Especially for those who master it.  
~  
~
```

Avsluta insättningsläget genom att trycka på **Esc**-tangentsen.

Spara filen och stäng **vi**-redigeraren med något av följande sätt som *sparar* ändringarna:

Command	Function/Keys
:x	Kommer att spara och stänga filen.
:wq	Kommer att skriva för att lämna in och avsluta.
:wq!	Skriver till en skrivskyddad fil om möjligt och avslutar.
ZZ	Kommer att spara och stänga. Observera att ingen kolon : används i detta fall.
:q!	Avsluta utan att spara ändringar
:e!	Släng ändringar och ladda om filen
:w!	Skriv till skrivskyddad fil, om möjligt.



Öppna återigen `myfile` med `vi`-editorn:

```
vi myfile
```

```
sysadmin@localhost:~$ vi myfile
```

Gå till den tredje raden och ta bort tredje och fjärde raden:

```
3G
```

```
2dd
```

Din skärm ska se ut ungefär så här:

```
Hello and welcome to the vi editor.  
It is a very powerful text editor.  
Especially for those who master it.  
~  
~
```

Tryck på `Esc` för att bekräfta att du är i kommandoläge.

Avsluta `vi`-editorn utan att spara dina ändringar:

```
:q!
```

Öppna `myfile` med `vi`-editorn:

```
vi myfile
```

```
sysadmin@localhost:~$ vi myfile
```

Observera att rad 3 och 4 fortfarande finns kvar.

Sök framåt efter ordet `line`. Du märker att markören flyttas till början av första förekomsten av ordet `line`, som visas på bilden nedan:

```
/line
```

Din skärm ska se ut ungefär så här:

```
Hello and welcome to the vi editor.  
It is a very powerful text editor.  
You just pressed O to open a line above.  
This line was added by pressing lowercase o.  
Especially for those who master it.  
~  
~  
/ line
```

Sök efter nästa förekomst av ordet `line` genom att trycka på bokstaven `n`:

```
n
```

Din skärm ska se ut ungefär så här:



```
Hello and welcome to the vi editor.  
It is a very powerful text editor.  
You just pressed O to open a line above.  
This line was added by pressing lowercase o.  
Especially for those who master it.  
~  
~  
/ line
```

Sök bakåt efter ordet **line**. Du märker att markören flyttas till början av den föregående förekomsten av ordet **line**, som visas på bilden nedan:

```
?line
```

Din skärm ska se ut ungefär så här:

```
Hello and welcome to the vi editor.  
It is a very powerful text editor.  
You just pressed O to open a line above.  
This line was added by pressing lowercase o.  
Especially for those who master it.  
~  
~  
?line
```

Sök efter den föregående förekomsten av ordet **line** genom att trycka på bokstaven **n**. Eftersom det inte finns några i den riktningen kommer vi att gå runt i dokumentet:

```
n
```

Din skärm ska se ut ungefär så här:

```
Hello and welcome to the vi editor.  
It is a very powerful text editor.  
You just pressed O to open a line above.  
This line was added by pressing lowercase o.  
Especially for those who master it.  
~  
~  
search hit TOP, continuing at BOTTOM
```

Du kommer att ersätta ordet **line** med ordet **entry**. När du trycker på **cw** hamnar du i insert-läge och kan skriva över ordet **line**:

```
cw
```

```
entry
```

Din skärm ska se ut ungefär så här:



```
Hello and welcome to the vi editor.  
It is a very powerful text editor.  
You just pressed O to open a line above.  
This entry was added by pressing lowercase o.  
Especially for those who master it.  
~  
~
```

Tryck på **Esc** för att lämna insert-läget

Lägg till text i början av en rad. Gå in i insert-läge igen och lägg till en rad genom att trycka på stora **i**:

```
I
```

Din skärm ska se ut ungefär så här:

```
Hello and welcome to the vi editor.  
It is a very powerful text editor.  
You just pressed O to open a line above.  
Ithis entry was added by pressing lowercase o.  
Especially for those who master it.  
~  
~
```

Insert-lägen inkluderar: **i**, **I**, **a**, **A**, **o** och **O**.

Tryck på **Esc** för att återgå till kommandoläget.

Lägg till text i slutet av en rad (stora **A**). Flytta först till andra raden och lägg till frasen **Indeed!**:

```
2G  
A  
[Space]Indeed!
```

Press the **Esc** key to return to command mode.

Your screen should look similar to the following:

```
Hello and welcome to the vi editor.  
It is a very powerful text editor. Indeed!  
You just pressed O to open a line above.  
This entry was added by pressing lowercase o.  
Especially for those who master it.  
~  
~
```

Spara dina ändringar och avsluta **vi**:

```
:x
```



Grundläggande shell-skriptning

Shell-skriptning låter dig ta en komplex sekvens av kommandon, lägga dem i en fil och sedan köra filen som ett program. Det här sparar tid eftersom du slipper skriva in en lång rad kommandon varje gång du behöver göra något återkommande.

Den här övningen fokuserar på hur du skapar enkla shell-skript. Vi utgår ifrån att du redan vet hur man använder en textredigerare. Använd gärna den editor du själv föredrar: `vi`, `nano`, `gedit` eller någon annan du gillar.

För att skapa ett enkelt shell-skript behöver du bara skapa en textfil och lägga till kommandon. Skapa en fil som heter `sample.sh` och skriv in följande rader:

```
echo "Hello there! Here is the calendar for this month:"
cal
```

```
sysadmin@localhost:~$ vi sample.sh
```

```
echo "Hello there! Here is the calendar for this month:"
cal
~
~
```

För att göra det tydligt att detta är ett BASH-shellskript behöver du lägga till en speciell rad högst upp i filen som kallas för "shbang" (eller "shebang"). Denna rad börjar med `#!` och innehåller sedan sökvägen till BASH-shellens körbara fil. Lägg till följande rad överst i `sample.sh`-filen:

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

```
#!/bin/bash
echo "Hello there! Here is the calendar for this month:"
cal
~
~
```

Ett sätt att köra det här programmet är att skriva `bash` före filnamnet. Utför följande:

```
bash sample.sh
```

```
sysadmin@localhost:~$ bash sample.sh
Hello there! Here is the calendar for this month:
  March 2026
Su Mo Tu We Th 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 31

sysadmin@localhost:~$
```



Du kan slippa att behöva skriva bash före filnamnet genom att göra filen "körbar" för alla användare. Kör följande kommandon:

```
ls -l sample.sh
chmod a+x sample.sh
ls -l sample.sh
./sample.sh
```

```
sysadmin@localhost:~$ ls -l sample.sh
-rw-rw-r-- 1 sysadmin sysadmin 73 Dec 11 14:46 sample.sh
sysadmin@localhost:~$ chmod a+x sample.sh
sysadmin@localhost:~$ ./sample.sh
Hello there! Here is the calendar for this month:
  March 2026
Su Mo Tu We Th 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 31

sysadmin@localhost:~$
```

Kommandot `chmod` används för att ändra rättigheter på filen så att den kan köras.

En vanlig funktion som används i skript är "backquoting". Med den här tekniken kan du köra ett shellkommando "inuti" ett annat shellkommando. Resultatet från det interna kommandot returneras som ett argument till det yttre kommandot.

Lägg till följande längst ner i `sample.sh`-filen:

```
vi sample.sh
```

```
sysadmin@localhost:~$ vi sample.sh
```

```
echo "Today is" `date +%A`

#!/bin/bash
echo "Hello there! Here is the calendar for this month:"
cal
echo "Today is" `date +%A`_
~
~
```

Avsluta insert-läget genom att trycka på `Esc`, skriv sedan `:wq!` och tryck på `Enter` för att spara och avsluta filen.

Nu kan du köra `sample.sh`-filen:

```
cat sample.sh
```



```
./sample.sh

sysadmin@localhost:~$ cat sample.sh
#!/bin/bash
echo "Hello there! Here is the calendar for this month:"
cal
echo "Today is" `date +%A`
sysadmin@localhost:~$ ./sample.sh
Hello there! Here is the calendar for this month:
  March 2026
Su Mo Tu We Th 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 31

Today is Friday
sysadmin@localhost:~$
```

Du har använt `./` före filnamnet `sample.sh` för att visa att filen finns i den aktuella katalogen. Kör följande för att se hur skalet inte hittar filen om du inte använder `./`:

```
sample.sh
```

Din skärm ska se ut så här:

```
sysadmin@localhost:~$ sample.sh
-bash: sample.sh: command not found
sysadmin@localhost:~$
```

Kom ihåg att variabeln `$PATH` används för att söka efter kommandon som du skriver in. Kör följande kommando för att se `$PATH`-variabeln för sysadmin-kontot:

```
echo $PATH
```

```
sysadmin@localhost:~$ echo $PATH
/home/sysadmin/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games
sysadmin@localhost:~$
```

Observera att `/home/sysadmin/bin` är en av katalogerna i variabeln `$PATH`. Det är ett utmärkt ställe att lägga dina shellsript på.

```
mkdir bin
mv sample.sh bin
sample.sh
```



```

sysadmin@localhost:~$ mkdir bin
sysadmin@localhost:~$ mv sample.sh bin
sysadmin@localhost:~$ sample.sh
Hello there! Here is the calendar for this month:
  March 2026
Su Mo Tu We Th 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 31

Today is Friday
sysadmin@localhost:~$

```

Villkorlig och repetitiv exekvering

Observera att i det här avsnittet kommer exempel som är mer komplexa att visas. När dessa visas använder du en teknik för att beskriva vad som händer i programmet. Tekniken ser ut så här:

Skriv in denna kolumn i drive.sh	Den här kolumnen beskriver koden (ange inte i filen)
<code>echo "Please enter your age"</code>	# Skriv ut en prompt
<code>read age</code>	# läsa användarinmatning och placera i \$age variabel

När du följer instruktionerna ska du skriva in texten från vänstra kolumnen i den angivna filen (`drive.sh` i exemplet ovan). Den högra kolumnen används för att beskriva specifika rader i programmet. Tecknet `#` (hash) används eftersom du kan lägga in kommentarer i ett shellsript genom att använda `#`.

Mer avancerade script kan använda villkorlig exekvering. Ett villkorligt uttryck, som en `if`-sats, kan använda resultatet av ett kommando som heter `test`. `test`-satsen jämför två tal (eller två strängar) för saker som "lika med", "mindre än" osv.

Skapa filen `drive.sh` enligt nedan och gör den körbar för att se hur `if`- och `test`-satser fungerar.

```
vi drive.sh
```

```
sysadmin@localhost:~$ vi drive.sh
```

Börja med att placera följande i `drive.sh`:

Skriv in denna kolumn i drive.sh	Denna kolumn beskriver koden (ange inte filen i filen)
<code>#!/bin/bash</code>	
<code>echo "Please enter your age"</code>	# Skriv ut en prompt
<code>read age</code>	# läsa användarinmatning och placera i \$age variabeln
<code>if test \$age -lt 16</code>	# test \$age -lt 16 returnerar "sann" om \$age numeriskt är mindre än 16



then	
echo "You are not old enough to drive."	# kör när test är sant
else	Detta avslutar if-satsen
echo "You can drive!"	# utförs när testet är falskt
fi	# This ends the if statement

Gör sedan filen körbar och kör den:

```
cat drive.sh
chmod a+x drive.sh
./drive.sh
```

Din skärm ska se ut så här:

```
sysadmin@localhost:~$ cat drive.sh
#!/bin/bash
echo "Please enter your age"
read age
if test $age -lt 16
then
    echo "You are not old enough to drive."
else
    echo "You can drive!"
fi
sysadmin@localhost:~$ chmod a+x drive.sh
sysadmin@localhost:~$ ./drive.sh
Please enter your age
14
You are not old enough to drive.
sysadmin@localhost:~$
```

Om man läser **if**-satsen verbalt kan man säga *Om \$age är mindre än 16, då echo 'Du är inte gammal nog för att köra', annars echo 'Du får köra!'*. **fi** avslutar **if**-satsen.

Varning

\$age måste vara ett heltal. Om det inte är det kommer programmet att krascha.

Test-satsen körs automatiskt när du placerar dess argument inom hakparenteser [] och ser till att det finns mellanslag runt dem. Ändra **if**-raden i **drive.sh** så att den ser ut så här:

```
if [ $age -lt 16 ]
```

Kör sedan programmet igen:

```
cat drive.sh
./drive.sh
```

Skärmen ska se ut så här:



```
sysadmin@localhost:~$ cat drive.sh
#!/bin/bash
echo "Please enter your age"
read age
if [ $age -lt 16 ]
then
    echo "You are not old enough to drive."
else
    echo "You can drive!"
fi
sysadmin@localhost:~$ ./drive.sh
Please enter your age
21
You can drive!
sysadmin@localhost:~$
```

För att se en komplett lista med testvillkor, kör kommandot `man test`.

Viktigt

Det måste finnas mellanslag runt hakparenteserna. `[$age -lt 16]` fungerar inte, men `[$age -lt 16]` fungerar.

Du kan också använda resultatet av andra shell-kommandon, eftersom de alla returnerar "framgång" eller "misslyckande". Till exempel, skapa och kör följande program, som kan användas för att ta reda på om ett användarkonto finns på det här systemet.

skapa en fil som heter `check.sh`:

```
vi check.sh
```

```
sysadmin@localhost:~$ vi check.sh
```

Lägg till följande i `check.sh`:

```
#!/bin/bash
echo "Enter a username to check: "
read name
if grep $name /etc/passwd > /dev/null
then
    echo "$name is on this system"
else
    echo "$name does not exist"
fi
```

Kör sedan följande kommandon:

```
cat check.sh
chmod a+x check.sh
```



```
./check.sh
```

När du blir tillfrågad om ett användarnamn, ange värdet "root". Utför kommandot igen (./check.sh) och ange värdet "bobby". Din skärm ska se ut så här:

```
sysadmin@localhost:~$ cat check.sh
#!/bin/bash
echo "Enter a username to check: "
read name
if grep $name /etc/passwd > /dev/null
then
    echo "$name is on this system"
else
    echo "$name does not exist"
fi
sysadmin@localhost:~$ chmod a+x check.sh
sysadmin@localhost:~$ ./check.sh
Enter a username to check:
root
root is on this system
sysadmin@localhost:~$ ./check.sh
Enter a username to check:
bobby
bobby does not exist
```

Ett annat vanligt villkorsuttryck är så kallad while-loop. Den används för att köra kod upprepade gånger så länge en villkorskontroll ger "sant". Börja med att skapa en fil som heter num.sh:

```
vi num.sh
```

```
sysadmin@localhost:~$ vi num.sh
```

Lägg sedan in följande i filen num.sh:

Lägg in detta i num.sh filen	Beskrivning (Lägg ej till i filen)
#!/bin/bash	
echo "Please enter a number greater than 100"	
read num	
while [\$num -le 100]	
do	# Exekvera kod från do till done om testvillkoret är "true"
echo "\$num is NOT greater than 100."	
echo "Please enter a number greater than 100"	
read num	
done	# Detta avslutar while-påståendet
echo "Finally, \$num is greater than 100"	



Gör filen körbar och kör den:

```
cat num.sh
chmod a+x num.sh
./num.sh
```

När du blir tillfrågad om ett nummer, ange **25**. När du blir tillfrågad igen, ange **99**. Slutligen, ange **101** när du blir tillfrågad en tredje gång. Din skärm ska då se ut så här:

```
sysadmin@localhost:~$ cat num.sh
#!/bin/bash
echo "Please enter a number greater than 100"
read num
while [ $num -le 100 ]
do
    echo "$num is NOT greater than 100."
    echo "Please enter a number greater than 100."
    read num
done
echo "Finally, $num is greater than 100"
sysadmin@localhost:~$ chmod a+x num.sh
sysadmin@localhost:~$ ./num.sh
Please enter a number greater than 100
25
25 is NOT greater than 100.
Please enter a number greater than 100.
99
99 is NOT greater than 100.
Please enter a number greater than 100.
101
Finally, 101 is greater than 100
sysadmin@localhost:~$
```

Om villkorskontrollen för **while**-satsen (`[$num -le 100]`) ger sant, så körs satserna mellan **do** och **done**.

När dessa satser har körts, kontrolleras villkorset för **while**-satsen igen. Om det fortfarande ger sant, körs satserna mellan **do** och **done** på nytt.

Det här fortsätter att upprepas tills **while**-villkoret ger falskt, alltså när värdet är större än **100**.

Scriptkod är en del av BASH-skalet, vilket innebär att du kan använda dessa satser direkt i kommandoraden precis som du gör i ett shellscript. Det kan vara smidigt för satser som **for**-satsen, en sats som tilldelar en lista av värden till en variabel – ett i taget. På så sätt kan du utföra en uppsättning operationer på varje värde. Till exempel, kör följande på kommandoraden:

```
for name in /etc/passwd /etc/hosts /etc/group
do
wc $name
done
```



Din skärm ska se ut så här:

```
sysadmin@localhost:~$ for name in /etc/passwd /etc/hosts /etc/group
> do
>   wc $name
> done
 24  30 1001 /etc/passwd
  7  15  161 /etc/hosts
 46  46  561 /etc/group
sysadmin@localhost:~$
```

Observera att `wc`-kommandot kördes tre gånger: en gång för `/etc/passwd`, en gång för `/etc/hosts` och en gång för `/etc/group`.

OBS

`for`-loopen i BASH påminner mer om en klassisk `foreach`-loop i C, C++, C# och Java språken.

Ofta används kommandot `seq` tillsammans med `for`-satsen. `seq`-kommandot kan generera en lista med heltalsvärden, till exempel från 1 till 10. Om du till exempel kör följande på kommandoraden skapas 12 filer med namnen `test1`, `test2`, `test3` och så vidare (upp till `test12`):

```
ls
for num in `seq 1 12`
do
    touch test$num
done
ls
```

```
sysadmin@localhost:~$ ls
Desktop  Downloads  Pictures  Templates  check.sh  num.sh
Documents Music      Public    Videos     drive.sh
sysadmin@localhost:~$ for num in `seq 1 12`
> do
>   touch test$num
> done
sysadmin@localhost ls
Desktop Music Templates drive.sh test10 test2 test5 test8
Documents Pictures Videos num.sh test11 test3 test6 test9
Downloads Public check.sh test1 test12 test4 test7
sysadmin@localhost:~$
```