

TSKS08 Introduktionskurs i MATLAB

Examinator & föreläsare: Lasse Alfredsson
Datorseende, ISY, www.cvl.isy.liu.se
lasse.alfredsson@liu.se

Lab-assistenter:
Fredrik, Joakim, Joakim, Joel, & Marcus

Kurswebbsida:
www.cvl.isy.liu.se/education/undergraduate/TSKS08

Copyright © Lasse Alfredsson

MATLAB = MATrix LABoratory

- ♦ **Först:** Ett verktyg för beräkningar i linjär algebra.
- ♦ **Nu:** Verktyg för diverse beräkningar och simuleringar inom t.ex. statistik, signalbehandling, kommunikation, fysik, bildbehandling, ekonomi, mekanik, m.m., m.m. . .
- ♦ Används i ett stort antal kurser på LiTH och på många företag.
- ♦ Matlab med studentlicens: Ladda ned från Studentportalen!
- ♦ "Learn by doing" – bäst inläring genom att testa själv!
- ♦ **Matlab-introduktioner i text- och video** på kurswebbsidan!!
(Se länken "[Matlabresurser](#)")

Copyright © Lasse Alfredsson

MATLAB = MATrix LABoratory

1: v. 14

2: v. 17

3: v. 18

4: v. 19-20

5: v. 21

- **Ett laborationshäfte, 4 kapitel** – köp snarast häftet på Bokakademien:

Kap 1: Introduktion till Matlab (grundläggande funktionalitet, skript & enkla beräkningar)

Kap 2: Grafisk presentation av vektorer & matriser

Kap 3: Villkorssatser, loopar & funktioner

Kap 4: Funktioner – tillämpningsexempel

Arbetsform:

Klassvisa labbar i ISY:s datorsalar (Linux/Windows), 2 studenter/grupp.

Påbörja & helst avsluta Kap X på egen hand *innan* motsvarande lab-tillfälle.

Mål: Varje kapitel godkänd *senast* på resp. lab-tillfälle ovan.

Självverksamhet förutsätts!

2 hp \Leftrightarrow 53 tim,
men bara 2+10 tim schemalagt!

Om *ej* färdig/godkänd under lab-tillfället
 \Rightarrow bli godkänd fram till sid. x.
Gör sedan färdigt på egen hand och redovisa i början av nästa lab-tillfälle!

Copyright © Lasse Alfredsson

Återkoppling – kursutvärderingen 2014

Några frisvarskommentarer (bland de 27% som gjorde kursutvärderingen):

- ♦ "Kursupplägget gör att det är lätt att slinka igenom, om man vill, genom att skriva av varandra."
- ♦ "Se till att handledarna ställer högre krav på studenterna under redovisningen."
- ♦ "Assistenterna var olika hårda/noga vid redovisningarna."
- ♦ "Någon mer föreläsning, i samband med kapitel 3 och 4 tror jag skulle vara bra."

Nytt för i år:

- ♦ Ett nytt kösystem, med väntelista. Bra resultat i en annan kurs!
- ♦ 2 assistenter per pass redan från lab-pass 2 (Jfr. 2014: 2 ass/pass fr.o.m. lab-pass 3. 2013: 2 ass/pass fr.o.m. lab-pass 4).
- ♦ Assistenterna kommer överens om vad de kontrollerar och i vilken omfattning.
- ♦ En extra föreläsning (seminarium) införs i anslutning till kapitel 3 och (speciellt) 4. Jag besvarar då era frågor!

Copyright © Lasse Alfredsson

Hjälpmöjlighet – hjälp med funktioner/kommandon

- ◆ Hjälpfunktionen
- ◆ Hjälpfönstret
- ◆ Leta efter nyckelord

```

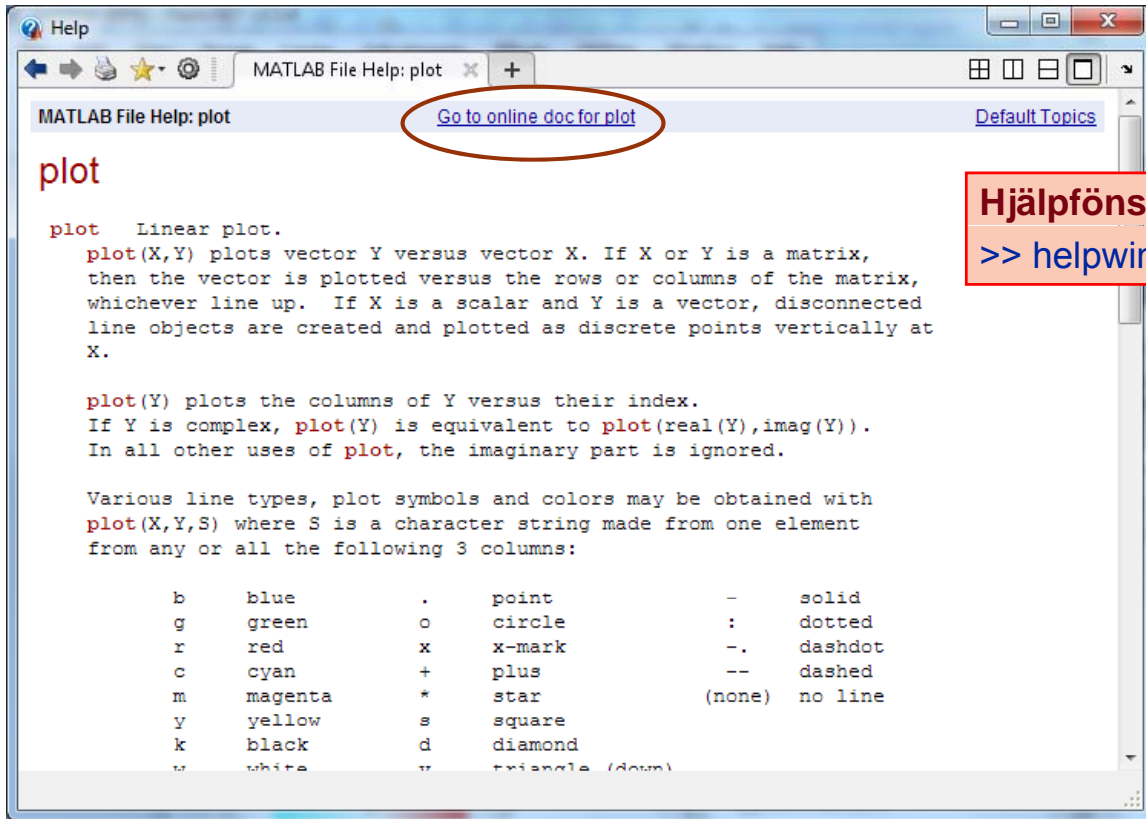
>> help
>> helpwin
>> lookfor
    
```

◆ Ex: `>> help sin`

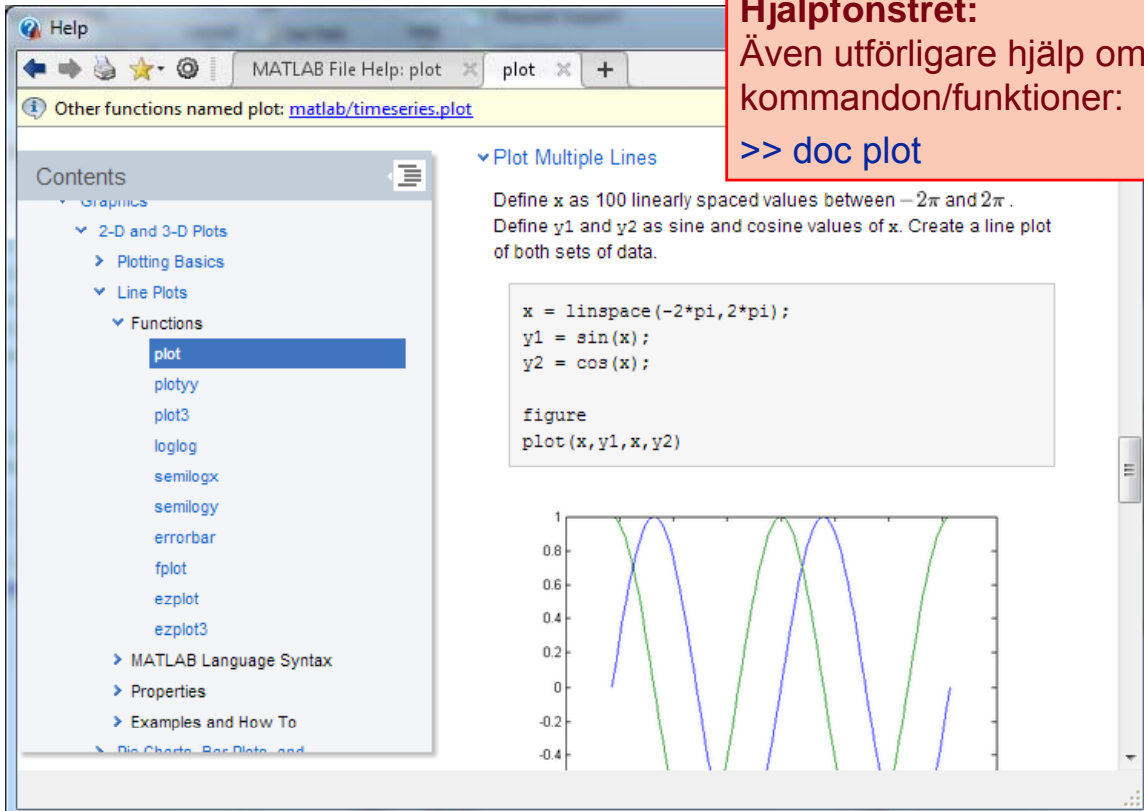
SIN Sine of argument in radians.
 SIN(X) is the sine of the elements of X.

...

- `>> helpwin plot` ⇒ hjälptexten för "plot" i hjälpfönstret
- `>> doc plot` ⇒ fylligare dokumentation om "plot"



Hjälpfönstret:
 >> helpwin plot



Hjälpfönstret:
 Även utförligare hjälp om olika kommandon/funktioner:
 >> doc plot

Några intressanta länkar:

- Flera **videor & översikter** finns på kurwebbsidan, under Matlabresurser, www.cvl.isy.liu.se/education/undergraduate/TSKS08/matlab-1/resurser , t.ex:
 - Interactive Matlab Tutorial: www.mathworks.se/academia/student_center/tutorials/mltutorial_launchpad.html
 - Videodemonstrationer, Matlab ver. 8: www.mathworks.se/products/matlab/videos.html
- Matlabs **Toolboxar**: www.mathworks.se/products
 - Matlab Finance Toolbox: www.mathworks.se/products/finance/
 - Matlab Econometrics Toolbox: www.mathworks.se/products/econometrics
 - m.fl. ...
- **Tillämpningsexempel** – lösa Sudoku med Matlab och en webbkamera: www.youtube.com/watch?v=SFBKXcASkEA
- **Matlab is Cool** rap: www.youtube.com/watch?v=hyzwxD33Lcl

Copyright © Lasse Alfredsson

De flesta av de efterföljande powerpointbilderna visas *inte* på föreläsningen, men kan vara av intresse ändå.

De utgör en inledande presentation av Matlab och ni kan gå igenom materialet själva.

Copyright © Lasse Alfredsson

Matlab som "räknedosa" – Variabler

```
>> 5*3 + pi/2
ans =
16.5708
```

```
>> sqrt(2)*cos(pi/3)
ans =
0.7071
```

```
>> 2*exp(j*pi/4)
ans =
1.4142 + 1.4142i
```

Det senaste resultatet sparas i "ans" (answer), om ingen annan variabel specificeras

$i, j = \sqrt{-1}$

```
>> a=2
a =
2
>> b=pi/4;
>> c=a*tan(b)
c =
2.0000
>> c=c*5
c =
10.0000
```

Heltal

Semikolon

⇒ resultatet skrivs inte ut

$c \leftarrow c*5$

Flyttal

Matriser – Matlabs huvudsakliga dataobjekt

Kolumner
(n)

	1	2	3	4	5
1	4 ¹	10 ⁶	1 ¹¹	6 ¹⁶	2 ²¹
2	8 ²	1.2 ⁷	9 ¹²	7 ¹⁷	25 ²²
3	7.2 ³	5 ⁸	7 ¹³	1 ¹⁸	11 ²³
4	0 ⁴	0.5 ⁹	4 ¹⁴	5 ¹⁹	56 ²⁴
5	23 ⁵	83 ¹⁰	13 ¹⁵	0 ²⁰	10 ²⁵

Rader (m)

A (2,4)

A (17)

Matriselementen kan antingen vara tal eller tecken/symboler

Rektangulär matris:
Skalär: 1x1-tabell
Vektor: mx1-tabell
 1xn-tabell
Matris: mxn-tabell

Numeriska matriser

```
>> A = [1 2 3; 4 5 6]
```

```
A =
```

```
1 2 3
4 5 6
```

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix}$$

- 2x3-tabell (2x3-matris)
- Använd hakparanteser [...] vid definition av tabeller/matriser.
- Semikolon är radseparator.
- Matlab skiljer på versaler och gemener. Exempel: $A \neq a$

```
>> b = [-5.6, sqrt(-7), (11+8)/5]
```

```
b =
```

```
-5.6 0+2.6458i 3.8
```

$$b = (-5.6 \quad j2.6458 \quad 3.8)$$

- 1x3-tabell = vektor av längd 3
- Kommatecken eller mellanslag separerar elementen i en rad.

Copyright © Lasse Alfredsson

Numeriska matriser och vektorer

```
>> A = [1 2 3; 4 5 6];
```

```
>> A(3,5)=round( sqrt(163) * real( exp(j*pi/6) ) )
```

```
A =
```

```
1 2 3 0 0
4 5 6 0 0
0 0 0 0 11
```

Matriser måste vara rektangulära.
Icke-definierade matris-element sätts till 0 (noll).

Nästlade funktionsanrop är tillåtna!

Skapa talsekvenser/-vektorer med kolon ":"

```
>> x = 0 : 5;
```

```
>> y = 8 : 0.3 : 9;
```

```
>> z = 1 : -0.5 : -1;
```

```
>> w = linspace(1,3,5);
```

$$x = (0 \ 1 \ 2 \ 3 \ 4 \ 5)$$

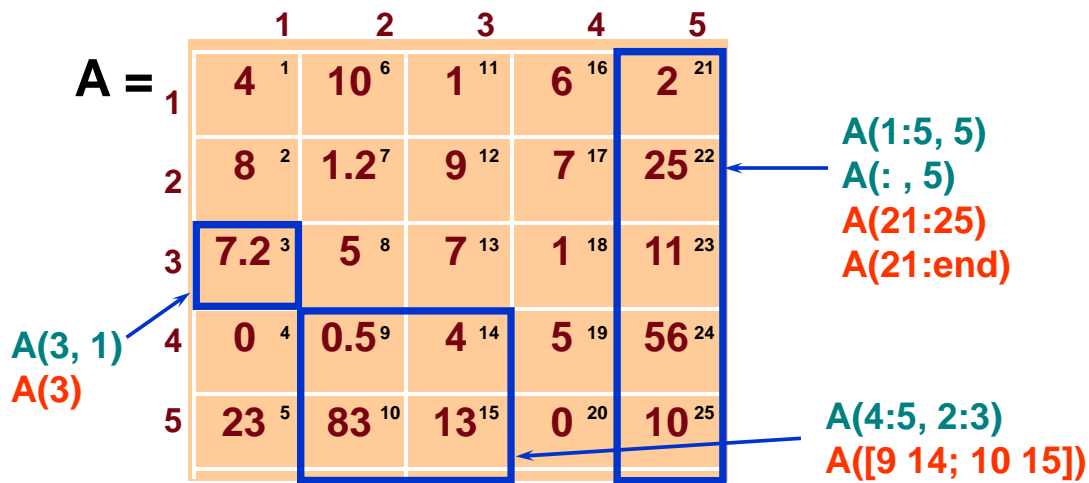
$$y = (8.0 \ 8.3 \ 8.6 \ 8.9)$$

$$z = (1 \ 0.5 \ 0 \ -0.5 \ -1)$$

$$w = (1 \ 1.5 \ 2 \ 2.5 \ 3)$$

Copyright © Lasse Alfredsson

Adressering av matriser



- Använd parenteser (...) för att adressera/specificera matrisindex
- Kolonoperatoren (:) specificerar ett *intervall* eller *alla* element i raden/kolumnen
- Använd [...] för att adressera delmatriser

Copyright © Lasse Alfredsson

Matris- och vektoroperationer

```

>> A = [1 2 3; 4 5 3; 7 8 9];
>> x = [8 4 2]';
>> b = A*x;

>> % Antag A och b kända.
>> % Beräkna x:
>> A\b      % Alt. inv(A)*b
ans =
     8
     4
     2
  
```

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 3 \\ 7 & 8 & 9 \end{pmatrix} \quad x = \begin{pmatrix} 8 \\ 4 \\ 2 \end{pmatrix}$$

Apostrof \Leftrightarrow matristransponat:

$$x = (8, 4, 2)' = \begin{pmatrix} 8 \\ 4 \\ 2 \end{pmatrix}$$

$$b = Ax \Rightarrow x = A^{-1}b = \frac{b}{A}$$

Copyright © Lasse Alfredsson

Matrismultiplikation vs. elementvis multiplikation

```
>> A = [1 2 3; 4 5 3; 7 8 9];
```

```
>> B = [2 3 1; 5 3 2; 5 9 2];
```

```
>> C = A*B
```

```
C =
```

```
27 36 11
```

```
48 54 20
```

```
99 126 41
```

Matris-
multiplikation

```
>> D = A.*B
```

```
D =
```

```
2 6 3
```

```
20 15 6
```

```
35 72 18
```

Elementvis
multiplikation

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 3 \\ 7 & 8 & 9 \end{pmatrix}, \quad B = \begin{pmatrix} 2 & 3 & 1 \\ 5 & 3 & 2 \\ 5 & 9 & 2 \end{pmatrix}$$

$$C = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 3 \\ 7 & 8 & 9 \end{pmatrix} \cdot \begin{pmatrix} 2 & 3 & 1 \\ 5 & 3 & 2 \\ 5 & 9 & 2 \end{pmatrix}$$

$$D = \begin{pmatrix} 1 \cdot 2 & 2 \cdot 3 & 3 \cdot 1 \\ 4 \cdot 5 & 5 \cdot 3 & 3 \cdot 2 \\ 7 \cdot 5 & 8 \cdot 9 & 9 \cdot 2 \end{pmatrix}$$

OBS: .* ./ .\ .^

Copyright © Lasse Alfredsson

Exempel på matris- och vektoranvändning

Rita sinussignalen $x(t) = 3\sin\left(2\pi 2t + \frac{\pi}{4}\right)$ i intervallet $0 \leq t \leq 1$ sek

samt motsvarande samplade signal $x[n] = x(nT_s)$

```
>> Ts = 0.01;
```

```
>> t = 0:Ts:1;
```

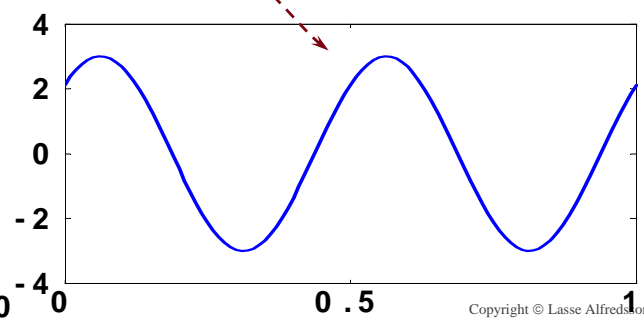
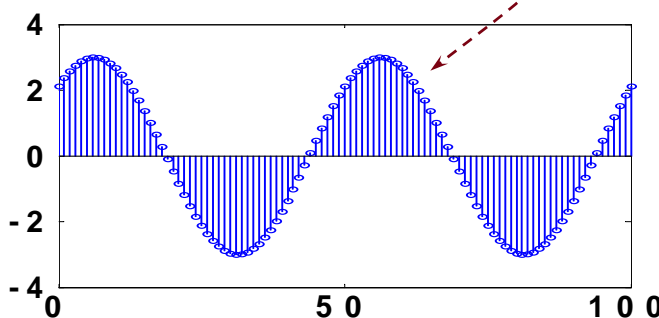
```
>> x = 3*sin(2*pi*2*t+pi/4);
```

```
>> plot(t,x)
```

```
>> n = 0:100; stem(n,x)
```

$$t = [0 \quad T_s \quad 2T_s \quad 3T_s \quad \dots \quad 1]$$

$$x = [0 \quad x(T_s) \quad x(2T_s) \quad x(3T_s) \quad \dots \quad x(1)]$$



Copyright © Lasse Alfredsson

Variabelhantering

- ◆ Visa aktuella/använda variabler: `>> who`
`>> whos`
- ◆ Spara variabler i datafil (*filnamn.mat*): `>> save filnamn`
- ◆ Rensa arbetsminnet från alla variabler: `>> clear`
- ◆ Hämta/ladda variabler från datafil: `>> load filnamn`

Skript och funktioner

Om man vill "köra" ett antal Matlabkommandon flera gånger, t.ex. med olika värden på några variabler, kan man skriva dessa i ett s.k. **skript**, som sedan anropas.

Ett skript är en **textfil** med filändelsen **.m** (liksom alla andra Matlab-funktioner). Alla variabler i ett skript finns i "workspace".

Ex: `ritasignal.m` (Anropas i kommandofönstret som "ritasignal")

```
t = linspace(0,10,200); % 200 sampelvärden, t=0->10
Fkn = input('Ange x(t) som en sträng: ');
x = eval(Fkn); % Evaluera Fkn => vektorn x[n]
plot(t,x) % Rita funktionen

% Ha för vana att alltid kommentera koden väl!
```

Skript och funktioner

Funktioner används istället för skript om man vill skicka med invariabler och erhålla andra variabler i retur. Variabler i funktionerna är vanligen lokala, och kan inte nås från "workspace".

Ex: ritasignaler.m (Anropas som "[t,x,y] = ritasignaler(xstr,ystr,tmax)")

```
function [t, x, y] = ritasignaler(xstr, ystr, tmax)
t = linspace(0,tmax,200); % 200 sampelv. t=0->tmax
x = eval(xstr);          % Evaluera xstr => vektorn x[n]
y = eval(ystr);          % Evaluera ystr => vektorn y[n]
subplot(2,1,1), plot(t,x) % Rita funktionen x(t)
subplot(2,1,2), plot(t,y) % Rita funktionen y(t)
```

Copyright © Lasse Alfredsson

Kontrollstrukturer – if, switch, for, while

if-satser

```
if villkor
    kommandon
end
```

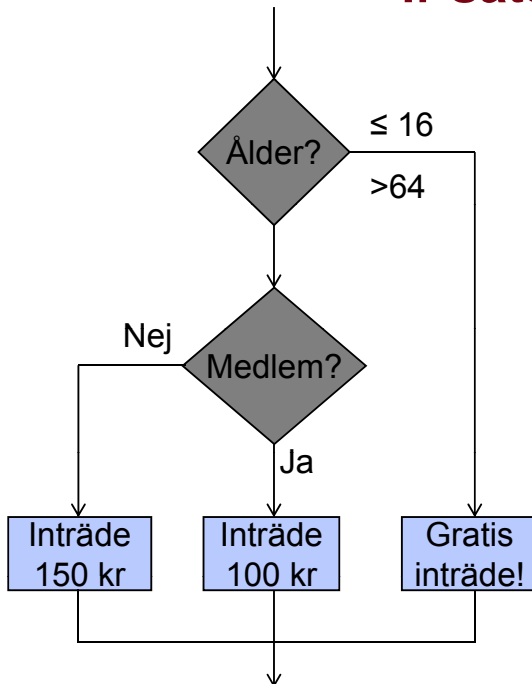
```
if villkor
    kommandon1
else
    kommandon2
end
```

```
if villkor1
    kommandon1
elseif villkor2
    kommandon2
else
    kommandon3
end
```

Copyright © Lasse Alfredsson

Kontrollstrukturer – if, switch, for, while

if-sats, exempel



```

if alder<=16 | alder>64
    disp('Gratis inträde!')
elseif medlem==1
    disp('Inträde 100 kr')
else
    disp('Inträde 150 kr')
end
  
```

Copyright © Lasse Alfredsson

Kontrollstrukturer – if, switch, for, while

switch-sats istället för flera elseif-satser

```

switch a
  case 12
    disp('Fall 1')
  case 'Lasse'
    disp('Fall 2')
  case {'Anneli', 'Tom'}
    disp('Fall 3')
  case {17, 21, 43}
    disp('Fall 4')
  otherwise
    disp('Fall 5')
end
  
```

```

if a==12
    disp('Fall 1')
elseif strcmp(a,'Lasse')
    disp('Fall 2')
elseif strcmp(a,'Anneli') ...
    | strcmp(a,'Tom')
    disp('Fall 3')
elseif a==17 | a==21 | a==43
    disp('Fall 4')
else
    disp('Fall 5')
end
  
```

Copyright © Lasse Alfredsson

Kontrollstrukturer – if, switch, for, while

Slingor – upprepa kommandon flera gånger

for-sats

```
for vektordefinition
kommandon
end
```

for - end:

Kontrollerad uppräknig av en variabel för användning vid upprepning av ett antal kommandon.

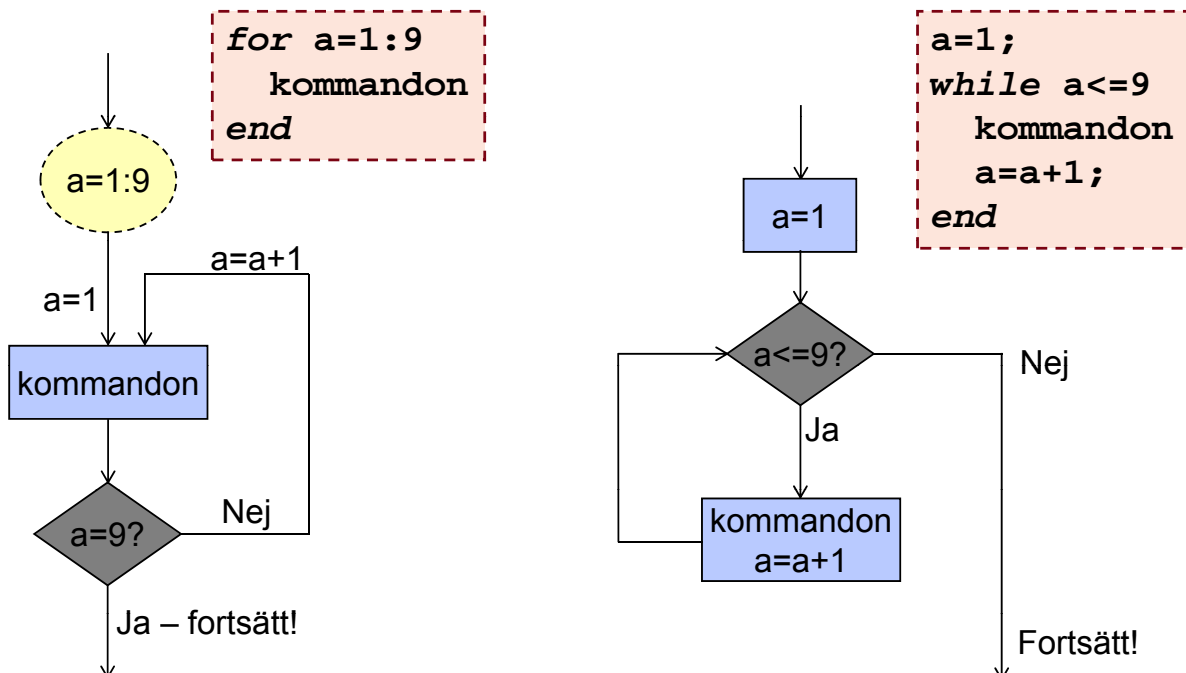
while-sats

```
while villkor
kommandon
end
```

while - end:

Används vanligen då man inte vet hur många gånger man vill upprepa ett antal kommandon.

Exempel, ekvivalent for-sats & while-sats





**LYCKA TILL MED
LABORATIONSUPPGIFTERNA!**