Variabler och konstanter

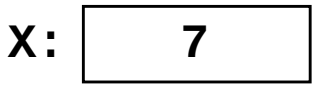
Deklareras automatisk när man stoppar in data i dem.

```
X = 7
Y = 'A'
Z = 'Kalle'
```

Definieras av att de har:

- **ett namn (X)**
- **en datatyp (Integer)**
- **ett värde (t.ex. 7)**

Lagras i datorns minne!



Om man stoppar in ett nytt värde omdeklareras automatiskt variabeln till den nya typ som behövs.

Det är alltså helt ok att utföra följande:

```
X = 7
X = 'Kalle'
```

Tilldelning

Generell form på en tilldelningssats:

```
Variabel = Uttryck
```

Exempel på ett antal tilldelningar i ett program.

```
antal = 7
antal = 3.14    % Lämpligt namn?
pi = 3.14       % Lämpligt att använda?
x = pi / 2
y = sin(x)
positive = abs(x * -y)

i = 3 / 2;
i = ceil(3 / 2);
i = floor(3.0 / 2.0);
i = mod(3, 2);
i = i + 1;
i = '2';
i = '0' + '1';  % Resultat?
b = true;
b = (I < 8);
```

Skillnaden mellan första gruppen av tilldelningar och den andra gruppen är att resultatet av beräkningen (det som kommer i variabeln) skrivs ut på skärmen i den första gruppen, men inte i den andra.

Man kan (läs SKA) alltid sätta dit ett semikolon för att undertrycka utskriften. Utskrifter gör vi normalt sett på annat sätt.

Operatorer

Aritmetiska operatorer:

```
+    -    *    /    ^    .+    .-    .*    ./    .^
```

Logiska operatorer:

```
and    &    or    |    not    ~    xor
```

Relationsoperatorer (också logiska operatorer):

```
==    ~=    <    >    <=    >=
```

OBS! Prioritetsordningen mellan operatorerna kan ändras med hjälp av parenteser precis som i matematiken.

OBS! När man använder flera relationsoperatorer i samma uttryck bör man (läs: skall man) använda parenteser för att man inte skall få problem att förstå vad som menas.

Sekvens

Ett antal satser/instruktioner utförs efter varandra.

```
Sats_1;
Sats_2;
Sats_3;
  ...
Sats_N;
```

OBS! Man brukar ofta ha semikolon, ”;”, för att slippa utskrifter av varje delresultat. Dessa behövs alltså inte, men som vi sa tidigare ...

In- och utmatning (eng: I/O) - 1 av 3

För att låta användaren få chansen att mata in saker till ditt program kan du använda följande instruktion:

```
x = input('Mata in ett tal: ');
```

Det som står innanför parentesen skrivs ut på skärmen och därefter förväntas användaren mata in ett data. Datat kommer sen att lagras i variabeln ”x”.

In- och utmatning - 2 av 3

Man kan utföra in och utmatning på många olika sätt.

```
x
x =
  10

disp(x);
10

disp(['x = ', num2str(x)]);
x = 10

fprintf('x = %03d\n', x);
x = 010
```

Utskrifterna är i exemplen ovan gjorda i magenta för att man skall se vad som är vad.

Man kan direkt observera att det inte är något semikolon efter första x:et, medan det är det i de andra fallen.

Vi kommer generellt att använda ”disp” när vi vill skriva ut saker. Dock kan det vara bra att ha tillgång till ”fprintf” om man t.ex. vill kunna skriva ut tabeller lite snyggare.

Den översta varianten använder man oftast när man vill testa saker eller skriva ut ”spårutskrifter” i sitt program.

In- och utmatning - 3 av 3

Man kan också ”formatera” utmatningen.

```
format compact
format loose

format long
format long e
format short
format short e

clc
```

Exakt hur dessa fungerar kan ni testa på laborationen.

Kommentarer

Det finns två sätt att sätta in kommentarer i MatLab.

1) Enstaka rad

```
% Kommentar
```

2) Kommentar över flera rader

```
%{
  Kommentar på flera rader.

  Måste ha enter efter inledande %{,
  men måste inte ha enter före
  avslutningsmarkeringen.
%}
```

En anledning till ovanstående kan vara att följande skall kunna göras:

```
switch a
  case 2 %{2,3,4}
end
```

Val (selektion) - 1 av 2

Man kan välja att utföra en av en eller flera sekvenser av satser beroende på olika villkor eller att inte utföra någon.

```
if Villkor
  Sekvens_Av_Satser
end
```

```
if Villkor
  Sekvens_Av_Satser
else
  Sekvens_Av_Satser
end
```

```
if Villkor
  Sekvens_Av_Satser
elseif villkor
  Sekvens_Av_Satser
end
```

```
if Villkor
  Sekvens_Av_Satser
elseif villkor
  Sekvens_Av_Satser
else
  Sekvens_Av_Satser
end
```

Val (selektion) - 2 av 2

Man kan välja att utföra en av en eller flera sekvenser av satser beroende på ett uttryck.

```
switch Uttryck
  case Uttryck
    Sekvens_Av_Satser
  case Uttryck
    Sekvens_Av_Satser
  case {Uttryck, Uttryck}
    Sekvens_Av_Satser
  otherwise
    Sekvens_Av_Satser
end
```

Repetition (iteration)

Att upprepa något ett antal gånger. Två olika sätt att göra detta i Matlab.

```
for Styrvariabel = Intervall
  Sekvens_Av_Satser
end
```

```
while Villkor
  Sekvens_Av_Satser
end
```

I båda dessa varianter kan man utföra en speciell sats för att avbryta repetitionen (`break`) om man vill.

Man kan också hoppa över restan av satserna i sekvensen som utförs inne i repetitionen (`continue`). Denna variant går vidare i nästa varv i repetitionen direkt.

Ett intervall skrivs på ett av följande sätt:

```
N:M
N:S:M
```

Den första varianten ger alla heltal i intervallet [N, M].

Den andra varianten ger alla tal N, N+S, N+2S, ... i intervallet [N, M]. Observera att det inte alltid är så att M kommer med. Observera att man kan ange negativa S.

Underprogram (funktioner)

Följande frågor är intressanta när de gäller underprogram. I MatLab finns det bara en typ av underprogram och det är funktioner.

- Hur ser det ut när man anropar ett underprogram?

- Vad betyder "skriver ut" i samband med underprogram?

- Vad betyder "returnerar" / "får tillbaka" / "ger tillbaka" när man säger det i samband med underprogram?

Normalfallet är att man anropar en funktion och "får tillbaka" ett (eller flera) värde(n).

- Lokala variabler?

- Globala variabler?

- Parametrar. Antal, ordning (och typ).

- Indata kontra utdata.

- Lokala funktioner (filmässigt).

Några exempel på underprogram

Två funktioner som varken tar emot några data eller returnerar något:

```
function main

  disp('Hello world!');
  welcome;
end

function welcome

  disp('Welcome to the world.');
end
```

En funktion som tar emot ett data, men inte returnerar något:

```
function print_message(message)

  disp(message);
end
```

En funktion som endast returnerar ett värde:

```
function y = random

  y = rand
end
```

En funktion med både indata och utdata:

```
function sum = add(a, b)

  sum = a + b;
end
```

Anrop till underprogram

De underprogram som finns på föregående OH anropas på följande sätt (det röda är utskrifterna som fås):

```
main;
Hello world!
Welcome to the world.

welcome;
Welcome to the world.

print_message('Vi älskar MatLab!');
Vi älskar MatLab!

print_message(209);
209

random
y =
    0.1988
ans =
    0.1988

a = random;
disp(a);
0.9649

add(10, 7)
ans =
    17

add(a, 1);

b = add(random, random);
```

Kommentarer i underprogram

Antag att vi har följande program:

```
function y = random
  % RANDOM returnerar ett slumptal.
  % Slumptalet ligger i intevallet
  % [0.0, 1.0]

  y=rand;
end
```

I MatLab kan man få fram information om funktioner genom att använda följande kommandon:

```
help "funktionsnamn"
   Ger hela den kommentar som står
   direkt efter funktionshuvudet

Exempel:
  help random
  RANDOM returnerar ett slumptal.
  Slumptalet ligger i intevallet
  [0.0, 1.0]


lookfor "funktionsnamn"
   Ger bara den första raden i
   kommentaren

Exempel:
  lookfor random
  RANDOM returnerar ett slumptal.
```