



Kunnskap for en bedre verden

TDT4105 Informasjonsteknologi, grunnkurs

Introduksjon til programmering i Matlab

Amanuensis Terje Rydland

Kontor: ITV-021 i IT-bygget vest (Gløshaugen)

Epost: terjery@idi.ntnu.no

Tlf: 735 91845

TDT4105 IT Grunnkurs

Forelesningsinfo

Tider	Mandag	Tirsdag	Onsdag	Torsdag	Fredag
0815 - 0900					
0915 - 1000					
1015 - 1100		Programmering M2 R1 (Anders)			
1115 - 1200		Øvingsforelesning M1 R7		Teori M1 R2 (Rune)	
1215 - 1300	Programmering M1 R1 (Rune)		Øvingsforelesning M3 R1	Øvingsforelesning M2 F1	
1315 - 1400					
1415 - 1500	Programmering M3 R1 (Terje)				
1515 - 1600					
1615 - 1700	Teori M3 R1 (Terje)				
1715 - 1800	Teori M2 R1 (Anders)				

Forelesningene skal være synkronisert - dere kan gå på de som passer best

Læringsmål og pensum

- Mål
 - Lære om programmering og hva er et program
 - Lære å designe et program
 - Lære om hvordan Matlab brukes
 - Lære om programmeringsomgivelse, skript, prompt, variabler
- Pensum
 - Matlab-boka, kapittel 1 (minus 1.5) Introduksjon til MATLAB
- Motivasjonsvideo: <http://code.org>

Hva er programmering?

- Å programmere er å fortelle en datamaskin hva den skal gjøre.
 - Vi bruker et programmeringsspråk (**f.eks. Matlab**) for å gjøre dette
- Før en datamaskin kan kjøre et program, må programmet **oversettes** til **maskinkode**.
 - Maskinkode er et språk som *prosessoren* kan utføre
 - Programmet som oversetter programmeringsspråket til maskinkode kalles kompilator, tolker eller oversetter.

Hva er et program?

- Et program er en oppskrift med instruksjoner som forteller en datamaskin hva den skal gjøre.
- Et program kan bestå av instruksjoner som:
 - **Oppretter** (deklarerer) og **gir verdier** (tilordne) til **variabler**
 - **Evaluerer** og **regner** på **variabler**
 - **Gjør valg**
 - Utfører ulike operasjoner (f.eks. Viser grafikk, spiller av lyd, tar imot informasjon fra brukeren)
 - Osv.

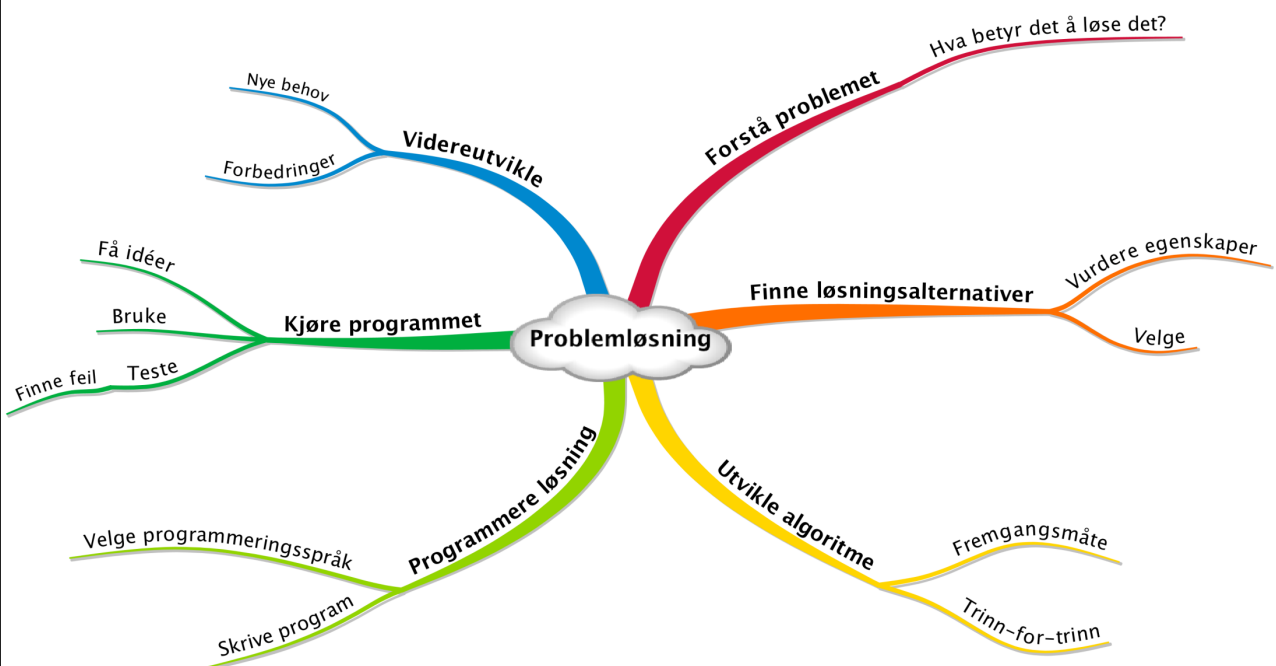
Hvordan et program fungerer

- Programmer utføres linje for linje - (altså ei linje av gangen).
- Dette er uhyre viktig å forstå for å klare å henge med etter hvert som vi lanserer avanserte programmer.
- Vi snakker gjerne om en **programpeker** som flytter seg nedover linje for linje i programmet.
 - Det som står øverst **skjer** først!

Illustrasjon av programkjøring

→ `x=123;`
 → `y=321;`
 → `sum=x+y;`
 → `disp(sum)`

Problemløsning



Generelt om Matlab

- Opprinnelig et dataprogram for matematiske beregninger, MathWorks Inc.
- Kan bruke det som en kalkulator
 - De 4 regnearterne: +, -, *, /
 - Elementære funksjoner som sin, cos osv.
 - Datatyper som vektorer og tabeller
- Avansert flerdimensjonal graf-plotting (med animering)
- Et programmeringsspråk
- Et prototypeverktøy for algoritmeutvikling
 - Ikke et produksjonsverktøy, er et interpretert språk
 - Ikke godt på parallel-beregning (flere-tråder)
 - Ikke godt for symbolsk matte (formel-regning)



Matlab desktop

De som ikke har installert enda: software.ntnu.no

The screenshot shows the MATLAB R2013a desktop environment. The interface is divided into several panes:

- Mappe (File Explorer):** Shows the current folder structure, including files like 'enterWords.m', 'main_yatzy.m', and 'yatz.m'.
- Editor:** The central workspace for writing code. It shows a function definition for 'enterWords' with 10 lines of code.
- Variabler (Workspace):** A table showing the current state of variables in the workspace.

Name	Value	Min	Max
x	[1 2 3 4 5 1]	1	10
y	[0.8415 0.9093 0.1411 -0.7568 -0.9589 -0.2794 0.6570 0.9894]	-0.9589	0.9894
- Kommandovindu (Command Window):** Shows the execution of commands and their output. It displays the output of the 'enterWords' function, including a warning about ignoring non-positive data in a pie chart.
- Kommando-historie (Command History):** A list of commands that have been executed in the Command Window.
- Fildetaljer (File Details):** A pane at the bottom left showing details for the selected file.

Programmering i Matlab

- Matlab kan brukes på følgende måter:
 - Interaktiv modus
 - Skrive en og en instruksjon inne i en tolker i Matlab-programmet
 - Lage Matlab-programmer (script)
 - Lage en tekstfil med Matlab-kommandoer ved hjelp av en teksteditor, og bruke Matlab-programmet til å oversette programmet til maskinkode og kjøre programmet.

Interaktiv modus

- Skriv en linje med en kommando og få direkte respons
- Fungerer nesten som en kalkulator
- I interaktiv modus huskes alle variabler som brukes
- Får feilmelding hvis du har skrevet noe feil
- Egner seg ikke til å skrive store programmer!

Matlab som kalkulator

- Skriv inn kommandoer etter prompt-et: >> ?
- Vanlig regning:
 - Vi har de 4 regneartene, + - * /
 - Eksponent, ^
 - Paranteser for å angi rekkefølge, ()
- Desimalskilletegn
 - Punktum (.) -- ikke komma (,)
- Konstanter
 - pi (3.141592653589793)
 - NaN (Not a Number, for eksempel 0/0)



Variabler og datastrukturer

- Et (brukerdefinert) navn på data
 - Refererer til et sted i minnet der verdien lagres
 - Starter med bokstav, fortsetter med bokstaver, tall eller understrek-tegn
 - Skiller mellom små og store bokstaver
- En enkelt verdi:
 - $A = 5$
- En vektor (liste):
 - $V = [1\ 2\ 3\ 4\ 5]$
- En matrise:
 - $M = [1\ 2\ 3; 4\ 5\ 6; 7\ 8\ 9]$

```
>> A = 5
A =
    5
>> V = [1 2 3 4 5]
V =
    1    2    3    4    5
>> M = [1 2 3; 4 5 6; 7 8 9]
M =
    1    2    3
    4    5    6
    7    8    9
>>
```

Datastrukturer

- Data som hører logisk sammen
- Alt i Matlab er tabeller (matriser)
 - Rader og kolonner
- Kan plukke ut enkeltelementer
 - $V(3)$ – tredje element
 - $M(1,2)$ – første rad, andre kolonne
 - $A(1)$ – Den ene verdien!
- Matriseoperasjoner
 - Transponering: M'
 - Matrisemultiplikasjon: $M*M$
 - Elementvis multiplikasjon: $M.*M$

```
>> M=[1 2 3; 4 5 6; 7 8 9]
M =
     1     2     3
     4     5     6
     7     8     9
>> M'
ans =
     1     4     7
     2     5     8
     3     6     9
>> M*M
ans =
    30    36    42
    66    81    96
   102   126   150
>> M.*M
ans =
     1     4     9
    16    25    36
    49    64    81
>>
```

Dat typer

- Alle uttrykk og variabler har en assosiert **type**
 - Må kjenne typen for å tolke (forstå) innholdet
- Matlab har mange typer (classes)

Reelle tall (float)

single, double

Logisk

logical (true, false)

Tegn

char

Heltall

Int8, int16, int32 ...

uint8, ... (uten fortegn, bare positive tall)

Tekst

String

- double er standard (default)
- Vi kommer tilbake til datatyper

Enkel generering av lister

- Liste = [startverdi:inkrement:sluttverdi]
 - A = [1:1:10] - liste med verdiene 1..10
 - B = [1:0.5:3] – liste med 1.0, 1.5, 2.0, 2.5, 3.0
- Det er ikke nødvendig med []
 - A = 1:1:10
- Kan lage matriser på samme måten
 - m = [1:5; 6:10; 11:15; 16:20; 21:25]

```
>> m = [1:5; 6:10; 11:15; 16:20; 21:25]
m =
     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
>>
```

Indeksring av matriseelementer

```
>> m = [1:5; 6:10; 11:15; 16:20; 21:25]
```

```
m =
```

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

m(:,4) points to column 4 (elements 9, 14, 19, 24)
 m(1,5) points to element 5 (row 1, column 5)
 m(:,1) points to column 1 (elements 1, 6, 11, 16, 21)
 m(2:4, 2:3) points to submatrix [7, 8; 12, 13; 17, 18]

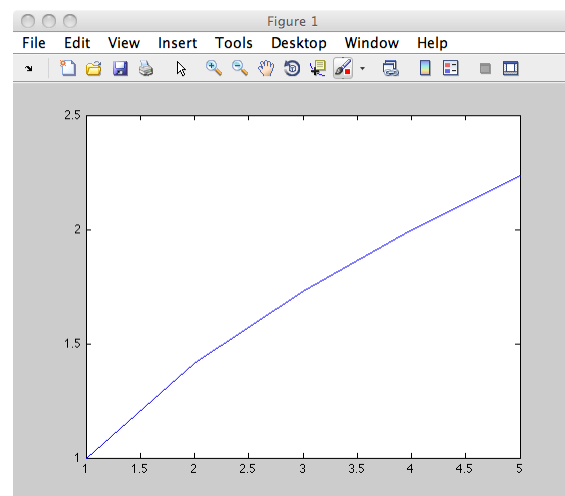
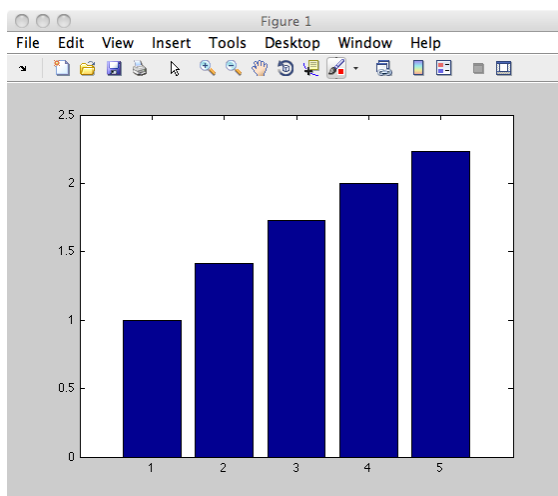
NB! Kolon (:) refererer til en hel rad eller kolonne.

Innebygde funksjoner, param

- Kvadratrot
 - `sqrt(x)`
- Tilfeldig tall
 - `rand(x)`
- Sinus
 - `sin(radianer)`
- Stolpediagram
 - `bar(sqrt(V))`
- Plotting
 - `plot(V, sqrt(V))`

```
>> sqrt(a)
ans =
    2.2361
>> sqrt(V)
ans =
    1.0000    1.4142    1.7321    2.0000    2.2361
>> sqrt(M)
ans =
    1.0000    1.4142    1.7321
    2.0000    2.2361    2.4495
    2.6458    2.8284    3.0000
>>
```

Resultat



Matlab hjelp

- Kommandoen *help* <nøkkelord> gir en kort forklaring på kommandoer og funksjoner
- Kommandoen *doc* <nøkkelord> åpner online- manualen (Octave sin er tekstbasert)
- Octaves HTML-manual
 - <http://www.gnu.org/software/octave/doc/interpreter>

Script

- Kjedelig å skrive alle kommandoer (på nytt, hver gang)
- Lagrer sekvenser av kommandoer i fil
 - <filnavn>.m
 - Ren tekst (NB! Ikke Word)
 - Kjører alle kommandoer på en gang (en etter en)
 - Kjør scriptet:
 - Skriv filnavnet i kommandovinduet (NB! Ofte mappe-relaterede problemer med å finne filen)
 - Fra den innebygde editoren
- Input
 - <variabelnavn> = input('Ledetekst')
 - Skriver ledetekst, venter på bruker-input, tilordner denne til variabel

```

--callsfaktorene i n
--11Faktorisering(n)
nesteFaktorNr = 1;
if n == 1
  v(nesteFaktorNr) = 1;
else
  v(nesteFaktorNr) = 1;
  restAvN = n;
  while restAvN > 1
    % tar vare på den største (gjenvarende faktoren)
    v(nesteFaktorNr) = maxPrimtallFaktor(restAvN);
    % oppdaterer det som står igjen av N
    restAvN = restAvN/v(nesteFaktorNr);
    % oppdaterer nr for neste faktor
    nesteFaktorNr = nesteFaktorNr + 1;
  end % while
  % if
  iffunction

```

Areal av sirkel

```
% Dette er en kommentar
% Beregner omkrets og areal av en sirkel

Radius = input('Oppgi radius: ');

Omkrets = 2*pi*Radius

Areal = pi*Radius^2
```

Kode i sirkelscript_forste.m

```
>> sirkelscript_forste
Gi inn radius: 1
Omkrets =
    6.283185307179586
Areal =
    3.141592653589793
>> sirkelscript_forste
Gi inn radius: 2
Omkrets =
    12.566370614359172
Areal =
    12.566370614359172
>>
```

Output (disp og fprintf)

- Semikolon etter uttrykk hindrer output
- **disp(<uttrykk>)** er enklest, skriver ut verdien til uttrykk uten formatering
 - **disp(4*5)** skriver ut 20
- **fprintf** – formatert utskrift
 - **fprintf('<formatstreng>',** liste av verdier)
 - **%d** plassholder for heltall
 - **%f** plassholder for flyttall (med komma og desimaler)
 - **%c** plassholder for ett tegn ('a' etc.)
 - **%s** plassholder for tekststreng ('NTNU' ol.)
 - **\n** spesialtegn for linjeskift
 - **fprintf('Kommunenr %4d: %s \n', 1601, 'Trondheim')**

Sirkelscript_andre.m

- %8.2f betyr:
 - Feltbredde = 8
 - Antall desimaler = 2
 - Det kommer et flyttall

```
% Beregner omkrets og areal av en sirkel
Radius = input('Gi inn radius: ');
Omkrets = 2*Radius*pi; % pi er en forhåndsdeklarerert konstant
fprintf('Omkretsen er %8.2f\n', Omkrets)
Areal = pi*Radius^2;
fprintf('Arealet er %8.2f\n', Areal)
```

```
>> sirkelscript_andre
Gi inn radius: 10
Omkretsen er    62.83
Arealet er    314.16
>>
```

Renteregning.m

- Gjennomgås på tavla
- Se koden i **renteregning.m**

```
% Renteregning

Innskudd = input('Innskudd: ');
RenteFot = input('Rentefot i prosent: ');
AntallTerminer = input('Antall terminer: ');

% Gjør om fra prosent
r = RenteFot/100;

% Liste over terminer
Terminer = 0:1:AntallTerminer;

% Regner ut saldo for alle terminer
Saldo = Innskudd*(1+r).^Terminer;

% Tegner stolpediagram
bar(Terminer, Saldo, 'g');
grid on;
xlabel('Termin');
ylabel('Saldo');
title('Forrentning av innskudd');
```

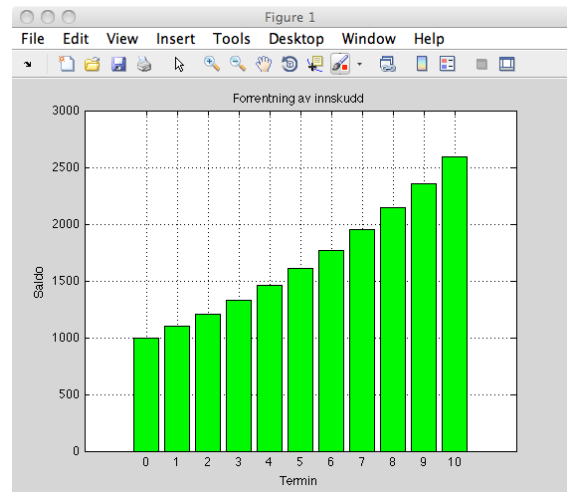
NB: vektor

NB: vektor

Renteregning (forts)

- Prøv **help bar** for å se hvordan plottet kan skreddersys.
- Legg merke til at Saldo blir en vektor med (AntallTerminer + 1) elementer.

```
>> renteregning
Innskudd: 1000
Rentefot i prosent: 10
Antall terminer: 10
>>
```



Plotte sinus og cosinus

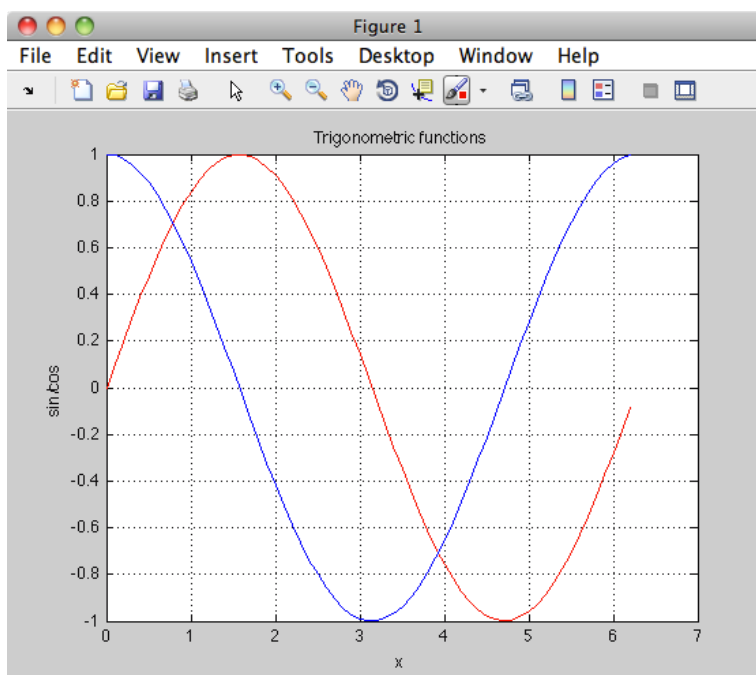
- Plotting av sinus (rødt) og cosinus (blått)
- Området som plottes bestemmes indirekte av de x- og y-verdiene som angis
- Mer informasjon om plot i:
 - help plot
 - doc plot for Matlab
 - Octaves HTML-manual
 - <http://www.gnu.org/software/octave/doc/interpreter>
- Koden blir lagt ut i sincos_plott.m

Plottescript

•% Plotter sin og cos

```
x = [0:0.1:2*pi];           % liste 0, 0.1, 0.2, ..., 2*pi
plot(x, sin(x), 'r')       % tegn sin med rød penn 'r'
grid on;                  % rutenett på
hold on;                  % plotter kurvene oppå hverandre
plot(x, cos(x), 'b')      % tegn cos med blå penn 'b'
xlabel('x');              % merker x-aksen
ylabel('sin/cos');        % merker y-aksen
title('Trigonometric functions'); % Overskrift
```

Resultat



BMI

- Body Mass Index (BMI)
 - Vekt i kilo / (høyde i m)²
- Spør bruker om:
 - Høyde
 - Vekt
- Beregner BMI
- Se kode i BMI_forste.m

```
Hoyde = input('Hoyde i cm: ');
Vekt = input('Vekt i kg: ');

bmi = Vekt / ((Hoyde/100)^2);

% utskrift med disp
disp(bmi)

% utskrift med fprintf
fprintf('\nDu har en BMI paa %5.2f\n', bmi)
```

BMI versjon 2

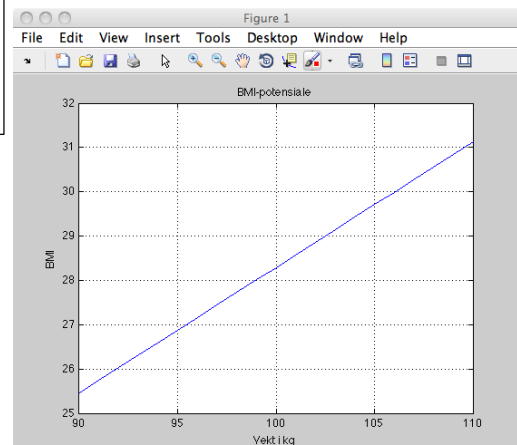
- Genererer BMI for et intervall rundt vekten brukeren oppgir
- Kode i BMI_andre.m

```
Hoyde = input('Hoyde i cm: ');
Vekt = input('Vekt i kg: ');

% Liste med vekter rundt brukerens oppgitte vekt
VektListe = [Vekt-10:1:Vekt+10];

% Regner ut BMI for et vektintervall
bmiListe = VektListe / ((Hoyde/100)^2);

% Plotter BMI-ene
plot(VektListe, bmiListe);
grid on;
xlabel('Vekt i kg');
ylabel('BMI');
title('BMI-potensiale');
```

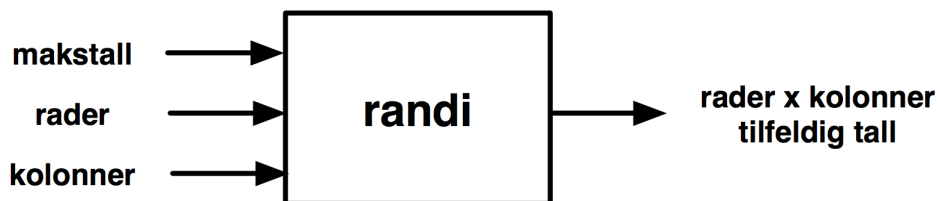


Innebygde funksjoner

- Kaller funksjonen
 - Kjører funksjonen uten å kjenne koden
 - Inn-parametere (fra ingen til mange slike)
- Resultat ut, bruker dette videre
- Eks: `lottoTall = randi(34,1,7)`
 - Tre parametere
 - Største heltall som kan trekkes
 - Antall rader
 - Antall kolonner

```
>> lottoTall = randi(34,1,7)
lottoTall =
    19    33    33     6    34    33    17
>>
```

Sort boks tenking



- Trenger bare å vite:
 - Hva funksjonen gjør
 - Hvordan funksjonen tar input
 - Hvordan funksjonen returnerer resultat
- Kan ignorere realiseringen (implementasjonen)
- Ingen sideeffekter

Egne funksjoner

```
function <returvariabel> = <funksjonsnavn>(<parametere>)
    <setninger>
    <returvariabel> = <tilordningsverdi>
end
```

- Lagres i filen **funksjonsnavn.m**
- Kalles som innebygde funksjoner
 - <funksjonsnavn>(<aktuelle parametere>)
- Lokale variable
 - Ukjent utenfor funksjonen

Sirkelareal og -omkrets



```
function areal = sirkelareal(radius)
% regner ut arealet av en sirkel

areal = pi*radius*radius;
end % function
```

```
function omkrets = sirkelomkrets(radius)
% regner ut omkretsen av en sirkel

omkrets = 2*pi*radius;
end % function
```

Som en innebygd funksjon

```
>> radius = 10;
>> areal = sirkelareal(radius)
areal =
    3.141592653589793e+02
>> omkrets = sirkelomkrets(radius)
omkrets =
    62.831853071795862
>>
```

Funksjoner

- Innebygde vs. Egne

- Innebygde

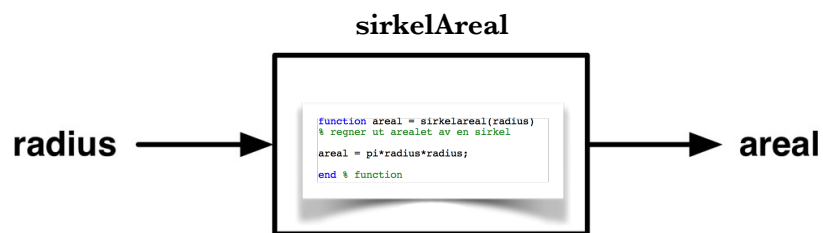
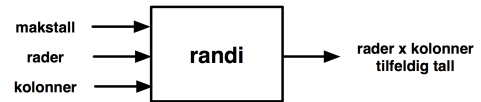
- Black box

- vet hva som sendes inn og hva som kommer ut
- vet ikke hva som skjer inne i boksen

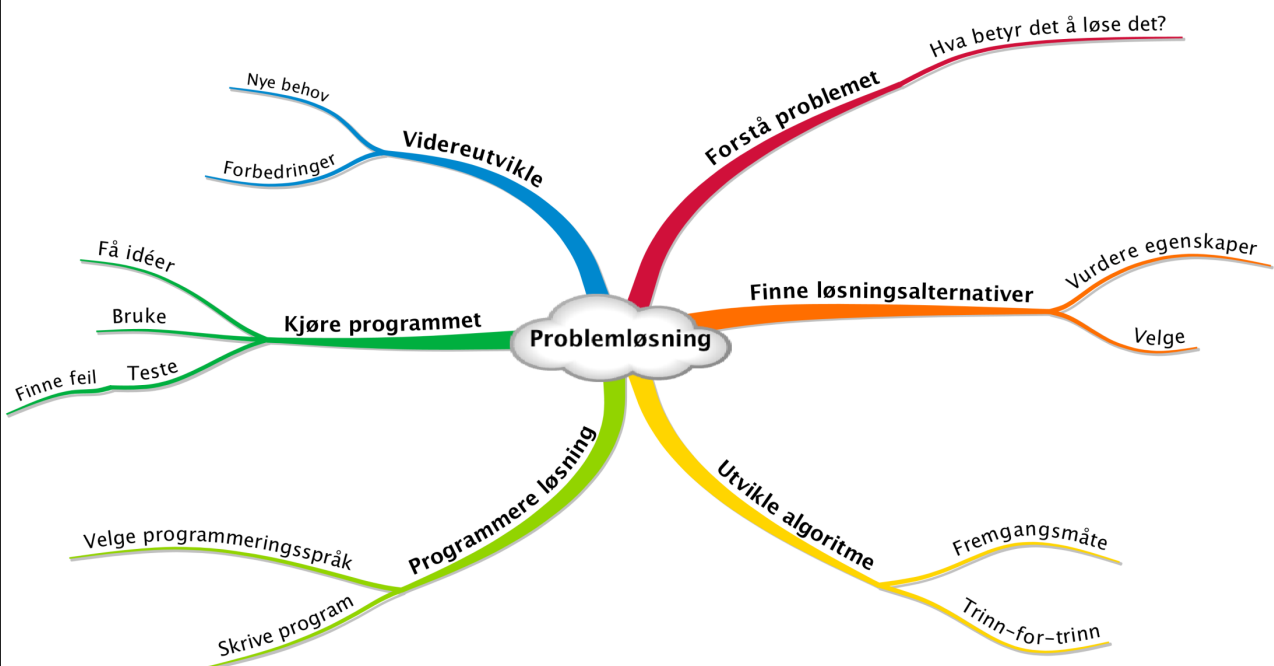
- Egne

- White box

- vet hva som sendes inn, hva som kommer ut, og hvordan den fungerer inni

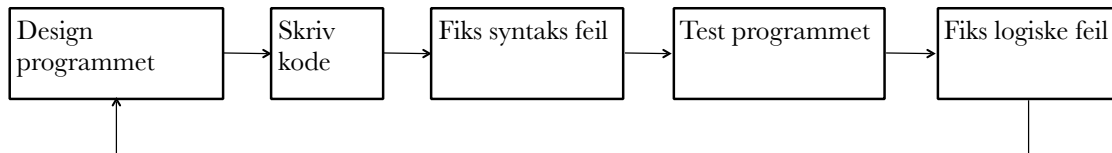


Problemløsning



Design et program: Programutviklingssyklus

- Design programmet:
 - Forstå oppgaven
 - Bestem steg for å utføre oppgaven
- Skriv kode:
 - Velg et språk som passer
- Fiks syntaks feil:
 - Skrivefeil i koden
- Test programmet:
 - Tester programmet for å lete etter logiske feil (feil oppførsel)
- Fiks logiske feil:
 - Debugge kode for å finne ut logiske feil



Hjelp til å designe et program: Hvordan beskrive stegene?

- Pseudokode:
 - Kode som kan skrives, leses og forstås av mennesker men ikke av en maskin
 - Trenger ikke å tenkte på at koden inneholder skrivefeil
- Flytskjema:
 - En grafisk framstilling av stegene i et program:
 - Oval: Start og slutt i programmet
 - Parallelogram: input og output
 - Rektangler: Prosessering av data

