



Kunnskap for en bedre verden

## TDT4105 Informasjonsteknologi, grunnkurs

**Mer om funksjoner. Logiske betingelser og betinget programutførelse (valg).**

Amanuensis Terje Rydland

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

Epost: [terjery@idi.ntnu.no](mailto:terjery@idi.ntnu.no)

Tlf: 735 91845

TDT4105 IT Grunnkurs

## Referansegruppe-2

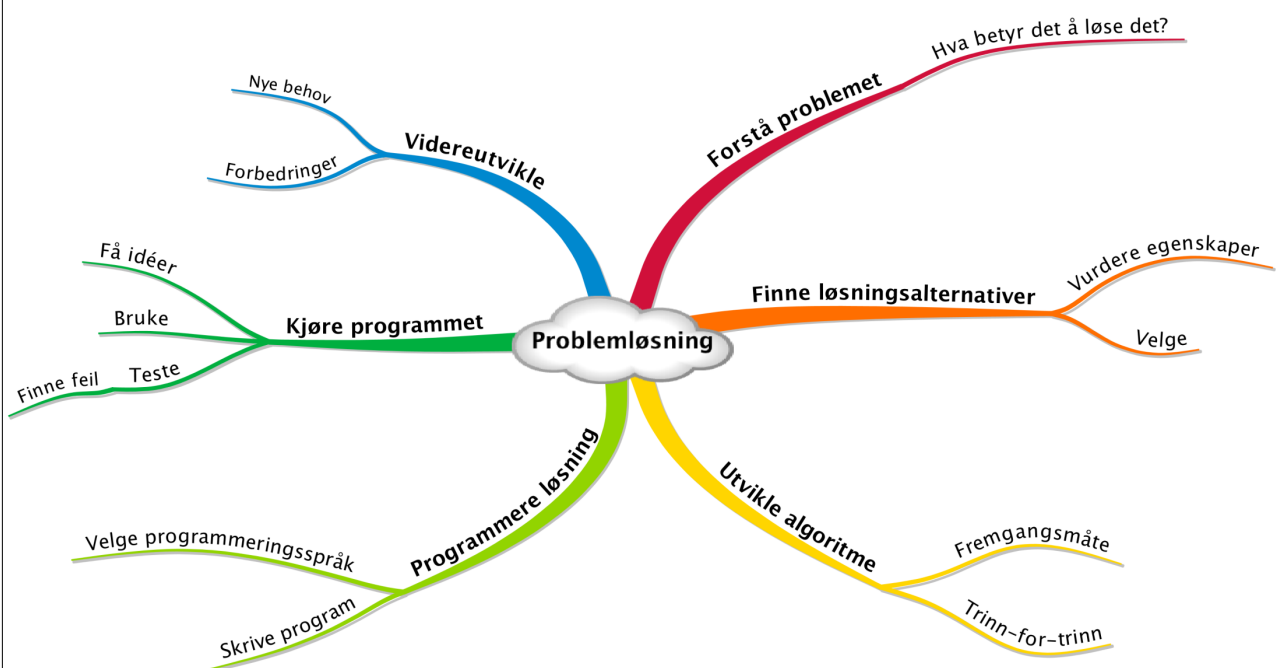
- Gi tilbakemelding på undervisning, øvinger, faginnhold, organisering, etc.
- Noen få timer i løpet av semesteret.
- Trenger 6 frivillige (ønsker spredning):
  - MTBYGG (Bygg- og miljøteknikk)
  - MTIØT (Industriell økonomi og teknologiledelse)
  - MTKJ (Industriell kjemi og bioteknologi)
  - MTMART (marin teknikk)
  - MTPROD (produktutvikling og produksjon)
  - MTMT (Materialteknologi)

**Ta kontakt i pausen. Viktig at vi kommer i gang med dette arbeidet!**

## Læringsmål

- Egne funksjoner, kalltrær og kontrollflyt
- Logiske uttrykk og logiske operatorer
- Operator-prioriteter
- Betinget programutførelse
  - IF
  - IF-ELSE
  - IF-ELSEIF-ELSE-END
- Kapittel 3.7 + 1.6 + 4.1-4.3

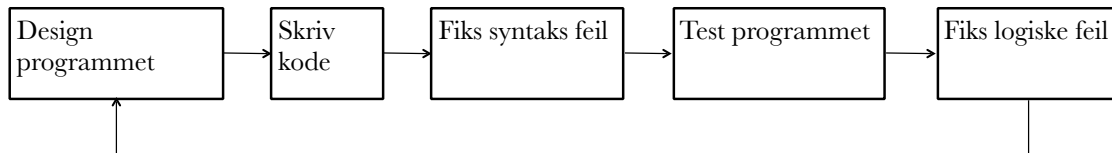
## Problemløsning



5

## 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

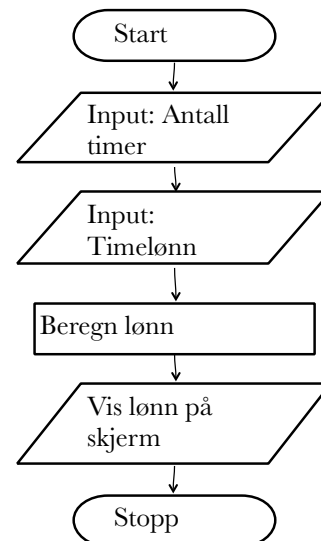


5

6

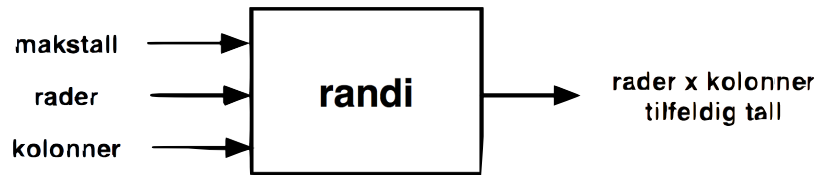
## Hjelp til å designe et program: Hvordan beskrive stegene?

- 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



6

## Sort boks tenking



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

## 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
>>
  
```

## 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

## Funksjoner

- Innebygde vs. Egne
- Innebygde

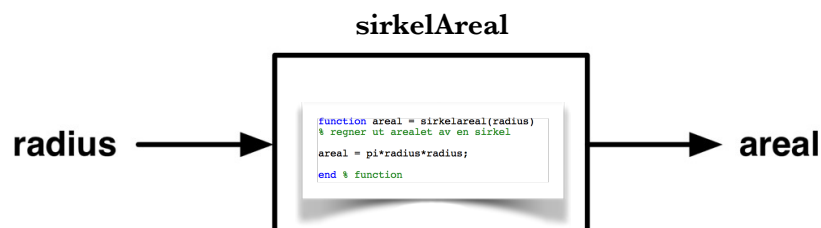
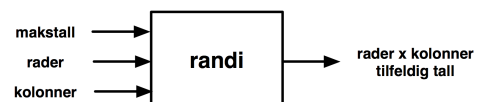
- Black box

- 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



11

## 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
>>
  
```

12

## Sirkelareal



```

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

areal = pi*radius*radius;

end % function
  
```

## Sirkelareal (forts.)

- Reserverte ord (kan ikke brukes til noe annet)
  - function, end
- Returvariabel (verdi tilbake)
  - areal
  - Det som returneres når funksjonen er ferdig (returnerer)
  - Ukjent utenom funksjonen
- Kommentarer
  - Alt etter % på en linje blir ignorert av Matlab

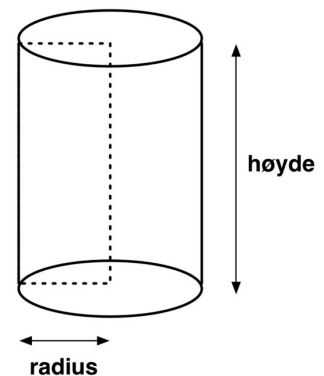
```
function areal = sirkelareal(radius)
% regner ut arealet av en sirkel

areal = pi*radius*radius;

end % function
```

## Sylindervolum.m

```
function volum = sylindervolum(radius, hoyde)
% beregner sylindervolum
volum = sirkelareal(radius)*hoyde;
end % function
```



15

## Sylinderoverflate

```
function areal = sylinderoverflate... (radius, hoyde)
% beregner sylinderoverflate
areal = 2*sirkelareal(radius)...
        + hoyde*sirkelomkrets(radius);
end % function
```



15

16

## Sylindereksempel.m

```
radius = input('Oppgi radius i sylinder: ');
hoyde = input('Oppgi hoyde i sylinder: ');

volum = sylindervolum(radius, hoyde);

overflate = sylinderoverflate(radius, hoyde);

fprintf('\n En sylinder med radius %5.2f og hoyde %5.2f\n',...
        radius, hoyde)
fprintf('Har volum lik %5.2f\n', volum)
fprintf('og overflateareal lik %5.2f\n', overflate)
```



16



## Kjøring

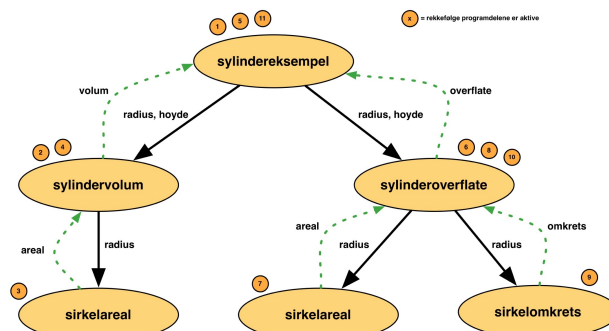
```
>> sylindereksempel
Oppgi radius i sylinter: 10
Oppgi hoyde i sylinter: 10

En sylinter med radius 10.00 og hoyde 10.00
Har volum lik 3141.59
og overflateareal lik 1256.64
>>
```

## Kalltre

```
function areal = sylinderoverflate(radius, hoyde)
% beregner sylinderoverflate
areal = 2*sirkelareal(radius) + hoyde*sirkelomkrets(radius);
end % function

radius = input('Oppgi radius i sylinter: ');
hoyde = input('Oppgi hoyde i sylinter: ');
volum = sylindervolum(radius, hoyde);
overflate = sylinderoverflate(radius, hoyde);
fprintf('\n En sylinter med radius %5.2f og hoyde %5.2f\n', radius, hoyde)
fprintf('Har volum lik %5.2f\n', volum)
fprintf('og overflateareal lik %5.2f\n', overflate)
```



## Operatorer

- Tabelloperatører
  - \* gir matrisemultiplikasjon mellom tabeller
  - .\* gir elementvis multiplikasjon
- Operand versus operator
  - m er operand (to ganger)
  - \* og .\* er operator

```
>> m=[1 2 3; 4 5 6; 7 8 9]
m =
     1     2     3
     4     5     6
     7     8     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
```

## Logiske utsagn og operatorer

- Sant / ikke sant (true-false, 1-0)
  - $4 < 5$
  - Radius  $> 0$
  - Student
- Ikke (not, negasjon): ~ (alt+~ på Mac)
  - ~Student
- Eller (or): || (alt+7)
  - Student || (Alder  $< 30$ )
- Og (and): &&
  - Student && (Alder  $< 30$ )

## Sannhetstabell

<b>A</b>	<b>B</b>	<b><math>\sim A</math></b>	<b><math>A \parallel B</math></b>	<b><math>A \&amp;\&amp; B</math></b>
<b><i>F</i></b>	<b><i>F</i></b>	<b><i>T</i></b>	<b><i>F</i></b>	<b><i>F</i></b>
<b><i>F</i></b>	<b><i>T</i></b>	<b><i>T</i></b>	<b><i>T</i></b>	<b><i>F</i></b>
<b><i>T</i></b>	<b><i>F</i></b>	<b><i>F</i></b>	<b><i>T</i></b>	<b><i>F</i></b>
<b><i>T</i></b>	<b><i>T</i></b>	<b><i>F</i></b>	<b><i>T</i></b>	<b><i>T</i></b>

## Relasjonsoperatorer

- < mindre enn
- > større enn
- <= mindre enn eller lik
- >= større enn eller lik
- == lik (NB! Ikke = som er tilordning)
- ~= ulik

## Sammensatte uttrykk

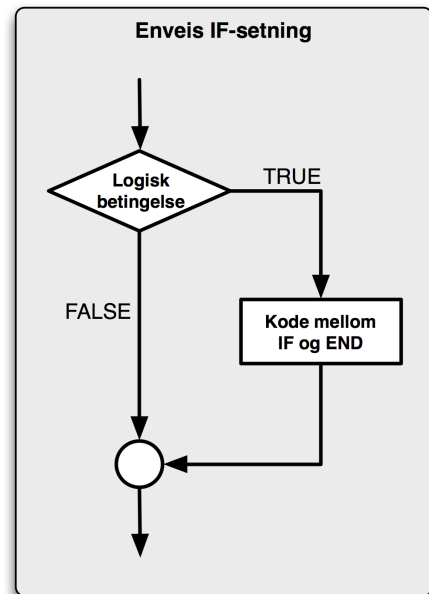
- Operatorpresedens (etter synkende presedens)
  - $()$
  - $\sim$
  - $*, /$
  - $+, -$
  - $<, <=, >, >=, ==, \sim =$
  - $\&\&$
  - $||$
- Samme nivå: Fra venstre mot høyre
  - $10 < \text{speedAboveLimit} < 25$  [galt]
  - $\text{true/false}$  (0 eller 1)  $< 25$  som alltid er sant!
- Moral: Bruk parenteser

## Eksempler

- $3+3/2$ 
  - 3 eller 4.5?
- $4 < 7 \ \&\& \ \sim(3 > 1 \ || \ 8 >= 9)$ 
  - Sant eller usant?
- $4 < 7 \ \&\& \ \sim(\text{True} \ || \ \text{False})$
- $4 < 7 \ \&\& \ \sim(\text{True})$
- $4 < 7 \ \&\& \ \text{False}$
- $\text{True} \ \&\& \ \text{False}$
- $\text{False}$

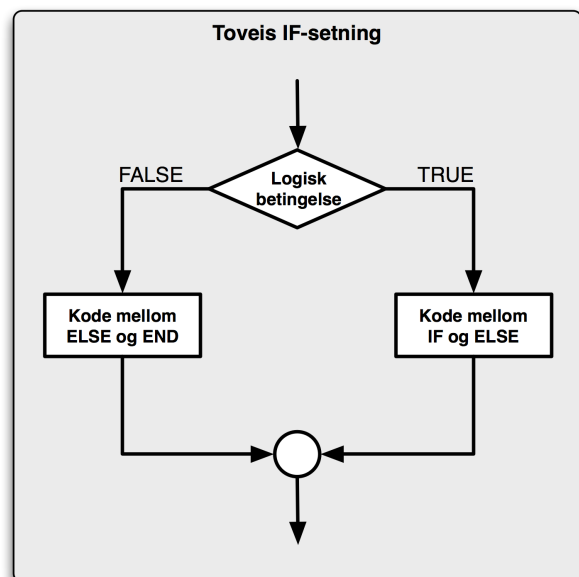
## IF-setningen

- Programutførelsen styres av en logisk betingelse
- IF <logisk betingelse>  
  <programkode>  
  END
- Programkoden utføres bare hvis betingelsen er sann
- IF Student
- Rabatt = 0.25;
- END



## ELSE-del

- Programkode som utføres når betingelsen
- Frivillig å ha med ELSE
- 
- IF Student
- Rabatt = 0.25;
- ELSE
- Rabatt = 0.1;
- END



## Sirkelomkrets og -areal

- Sjekker at det er en sirkel (radius > 0)
- Kode i sirkelscript\_tredje.m

```
% Omkrets og areal av en sirkel

radius = input('Gi inn radius: ');

if radius > 0
    omkrets = sirkelomkrets(radius);
    areal = sirkelareal(radius);
    fprintf('Omkrets = %3.2f og areal = %3.2f\n', omkrets, areal)
else
    display('Ikke en sirkel!');
end
```

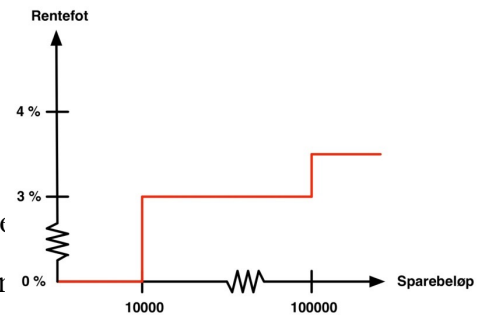
## Programkjøring

- Sjekk gyldig innputt
  - Garbage in, Garbage out

```
>> sirkelscript_tredje
Gi inn radius: 200
Omkrets = 1256.64 og areal = 125663.71
>> sirkelscript_tredje
Gi inn radius: 0
Ikke en sirkel!
>> sirkelscript_tredje
Gi inn radius: -10
Ikke en sirkel!
>>
```

## Sparing

- Rentefot avhengig av sparebeløp
- Beregne terminrente og sluttbeløp
- Lager skript for brukerdialog og å sette delene sammen
  - terminRente(belop)-funksjon for å beregne termin
  - renteFot(belop)-funksjon for å bestemme rentefoten



## sparing.m

```
% brukerinnputt
innskudd = input('Hvor mye vil du spare? ');

% beregninger
tjentRente = terminRente(innskudd);
sluttBelop = innskudd + tjentRente;

% utputt
fprintf('%1.0f kroner gir %1.0f i rente og %1.0f på konto\n', ...
        innskudd, tjentRente, sluttBelop)
```

```
>> sparing
Hvor mye vil du spare? 100000
100000 kroner gir 3500 i rente og 103500 på konto
>>
```

## terminRente(belop)

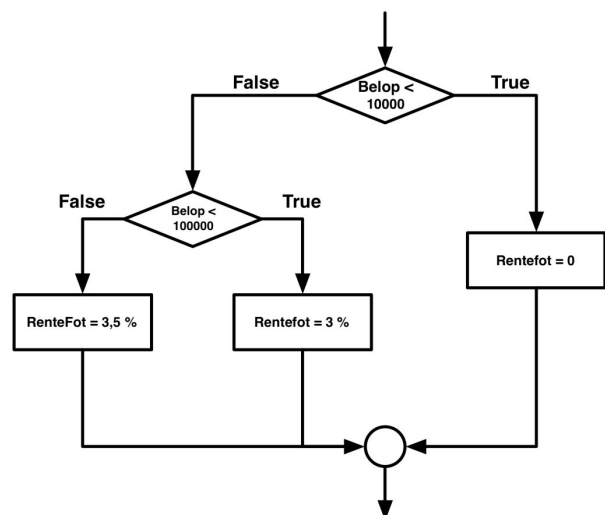
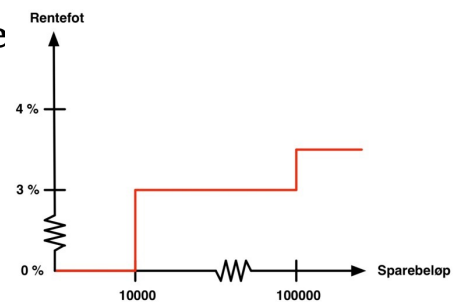
```
function tr = terminRente(belop)
% beregner tjent rente ut fra belop

    tr = belop*renteFot(belop);

end % function
```

## renteFot-funksjone

- Valg mellom 3 alternativer
- IF-setningen kan nøstes:
- Gir 3 veier gjennom koden
- Tips til testing:
  - Kjør alle veier
  - Test grenseverdiene





## renteFot(belop)

```
function rf = renteFot(belop)
% bestemmer retedefot ut fra belop

    if belop < 10000
        rf = 0;
    else
        if belop < 100000
            rf = 0.03;
        else
            rf = 0.035;
        end % if
    end % if

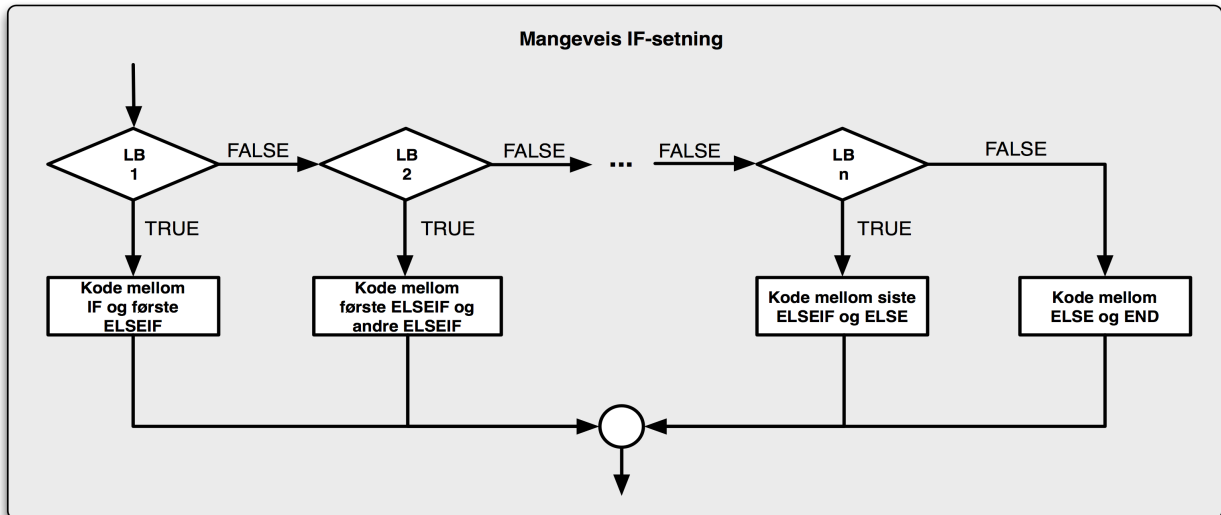
end % function
```

## Testing

```
>> sparing
Hvor mye vil du spare? 9999
9999 kroner gir 0 i rente og 9999 på konto
>> sparing
Hvor mye vil du spare? 10000
10000 kroner gir 300 i rente og 10300 på konto
>> sparing
Hvor mye vil du spare? 99999
99999 kroner gir 3000 i rente og 102999 på konto
>> sparing
Hvor mye vil du spare? 100000
100000 kroner gir 3500 i rente og 103500 på konto
>>
```

## IF-ELSEIF-ELSE-END

- Valg mellom **flere alternativer**, flerveis-if
- Alternativ til nøsting (kan kombineres med nøsting)



## renteFot2(belop)

```

function rf = renteFot2(belop)
% bestemmer retefot ut fra belop

    if belop < 10000
        rf = 0;
    elseif belop < 100000
        rf = 0.03;
    else
        rf = 0.035;
    end % if

end % function
  
```