



Kunnskap for en bedre verden

TDT4105 Informasjonsteknologi, grunnkurs

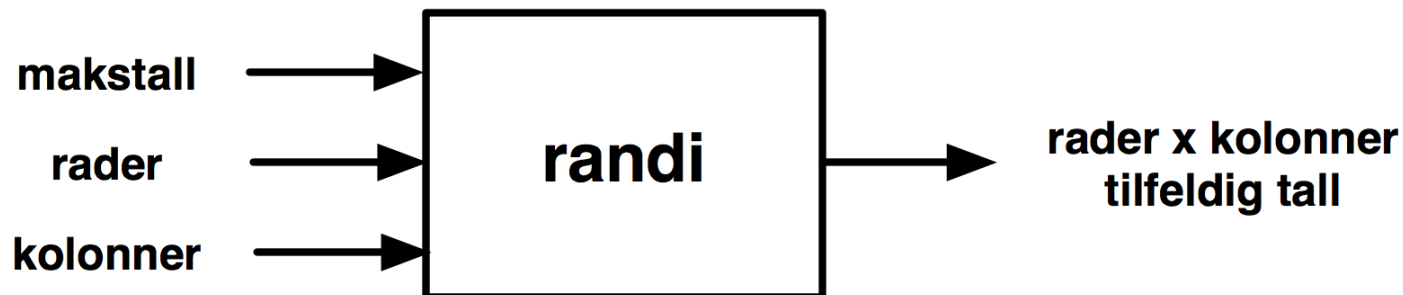
Matlab: Mer om funksjoner. Logiske betingelser og betinget programutførelse (valg).

Rune Sætre / Anders Christensen (satre / anders @idi.ntnu.no)

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

Sort boks tenking



- Trenger bare å vite:
 - Hva funksjonen gjør
 - Hvordan funksjonen tar innputt
 - 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



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

areal = pi*radius*radius;

end % function
```

Sirkelareal (forts.)

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

areal = pi*radius*radius;

end % function
```

- 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

Sirkelomkrets



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

omkrets = 2*pi*radius;

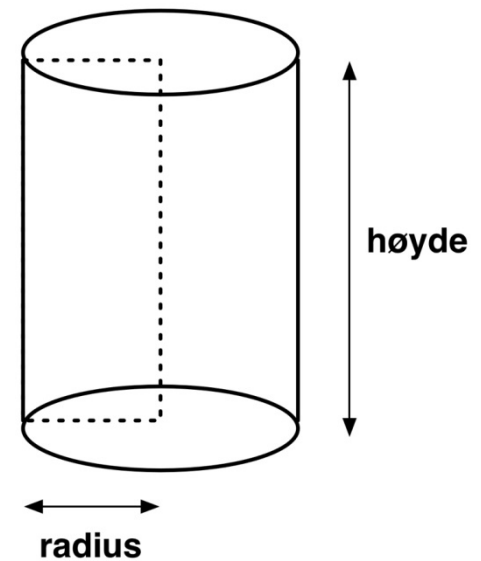
end % function
```

Bruk av egne funksjoner

- Som innebygde funksjoner

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


Sylindervolum.m



```
function volum = sylindervolum(radius, hoyde)
% beregner sylindervolum

volum = sirkelareal(radius)*hoyde;

end % function
```

Sylinderoverflate

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

Sylindereksempel.m

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

volum = cylindervolum(radius, hoyde);

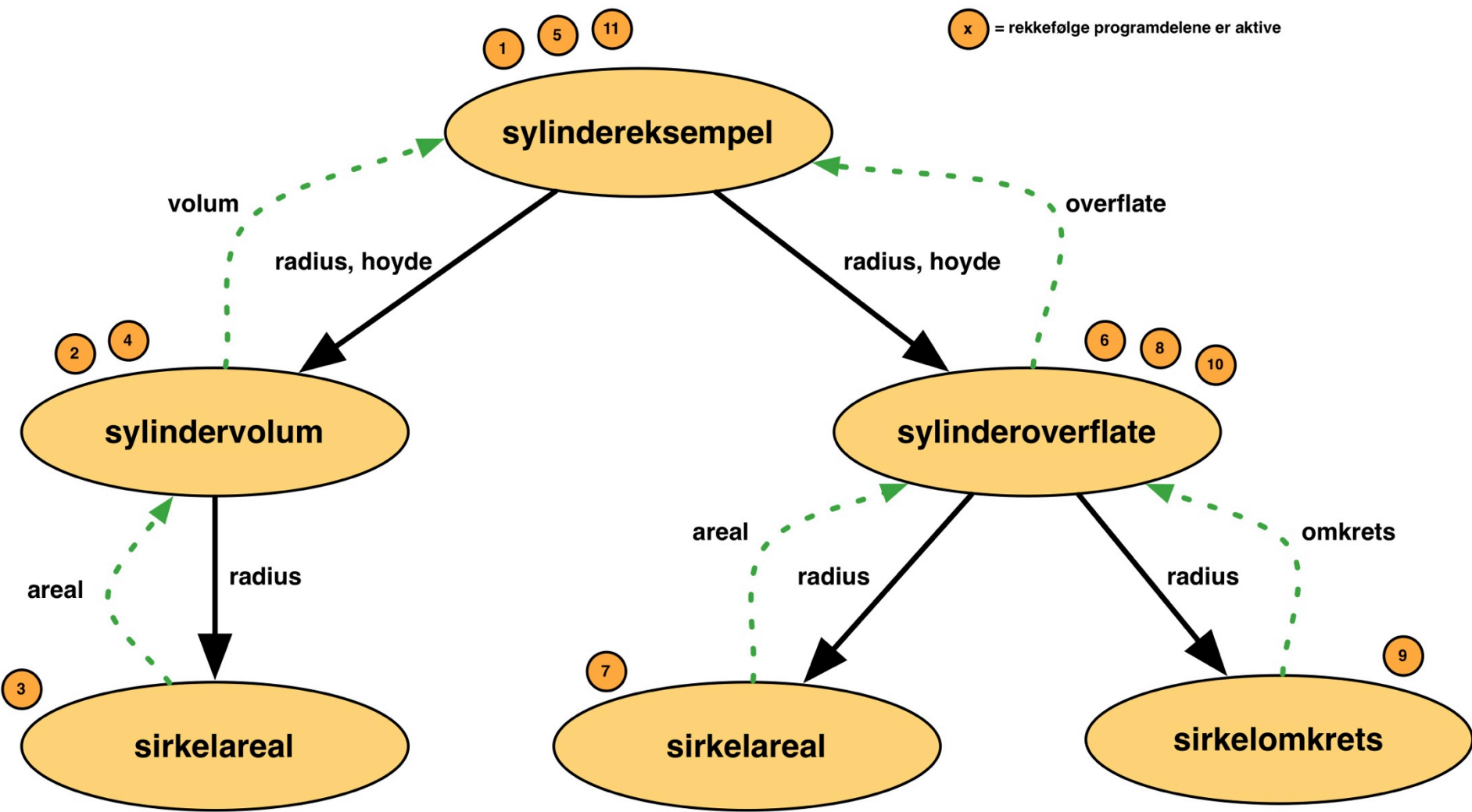
overflate = sylinderoverflate(radius, hoyde);

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)
```

Kjøring

```
>> sylindereksempel  
Oppgi radius i cylinder: 10  
Oppgi hoyde i cylinder: 10  
  
En cylinder med radius 10.00 og hoyde 10.00  
Har volum lik 3141.59  
og overflateareal lik 1256.64  
>>
```

Kaltre



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): \sim (alt+~ på Mac)
 - \sim Student
- Eller (or): \parallel (alt+7)
 - Student \parallel (Alder < 30)
- Og (and): $\&\&$
 - Student $\&\&$ (Alder < 30)

Sannhetstabell

A	B	$\sim A$	A B	A && B
F	F	T	F	F
F	T	T	T	F
T	F	F	T	F
T	T	F	T	T

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)
 - ()
 - ~
 - *, /
 - +, -
 - <, <=, >, >=, ==, ~=
 - &&
 - ||
- Samme nivå: Fra venstre mot høyre
 - `10 < speedAboveLimit < 25` [galt]
 - `true/false (0 eller 1) < 25` som alltid er sant!
- Moral: Bruk parenteser

Eksempler

- $3+3/2$
 - 3 eller 4.5?
- $4 < 7 \ \&\& \ \sim(3 > 1 \ || \ 8 \geq 9)$
 - Sant eller usant?

$4 < 7 \ \&\& \ \sim(\text{True} \ || \ \text{False})$

$4 < 7 \ \&\& \ \sim(\text{True})$

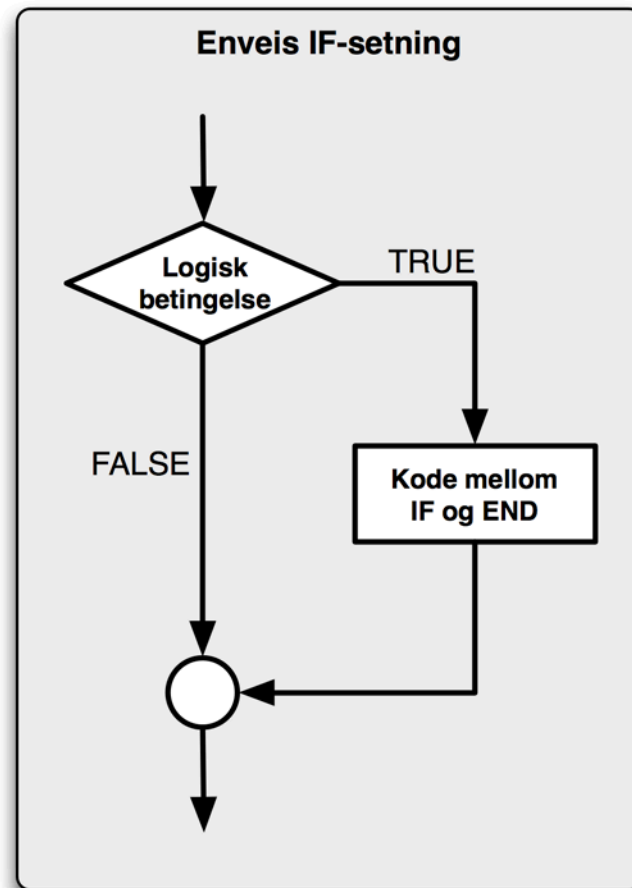
$4 < 7 \ \&\& \ \text{False}$

$\text{True} \ \&\& \ \text{False}$

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 IKKE er sann
- Frivillig å ha med ELSE

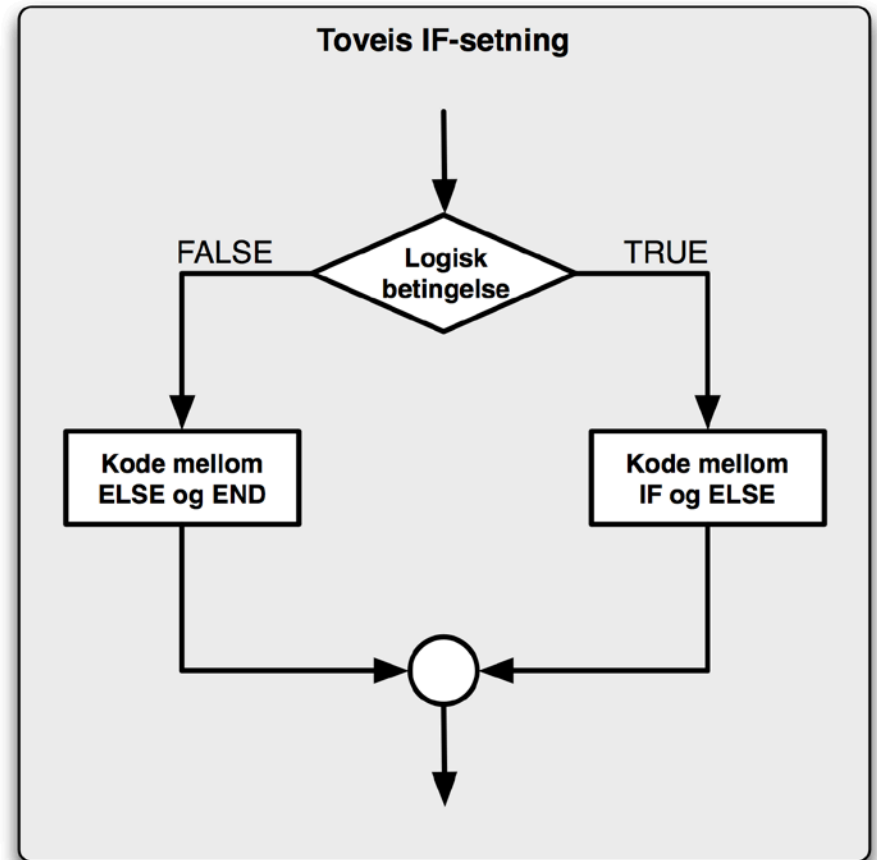
IF Student

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!');
```

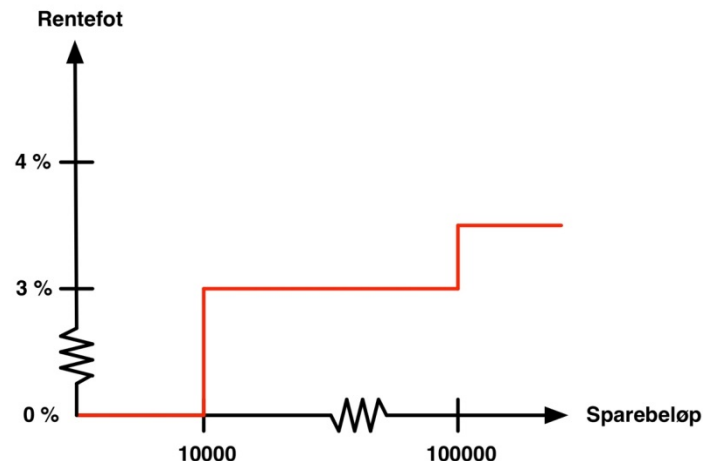
```
end
```

Programkjøring

```
>> 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!
>>
```

- Sjekk gyldig innputt
 - Garbage in, Garbage out

Sparing



- Rentefot avhengig av sparebeløp
- Beregne terminrente og sluttbeløp
- Lager skript for brukerdiallog og å sette delene sammen
 - terminRente(belop)-funksjon for å beregne terminrenten
 - 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)

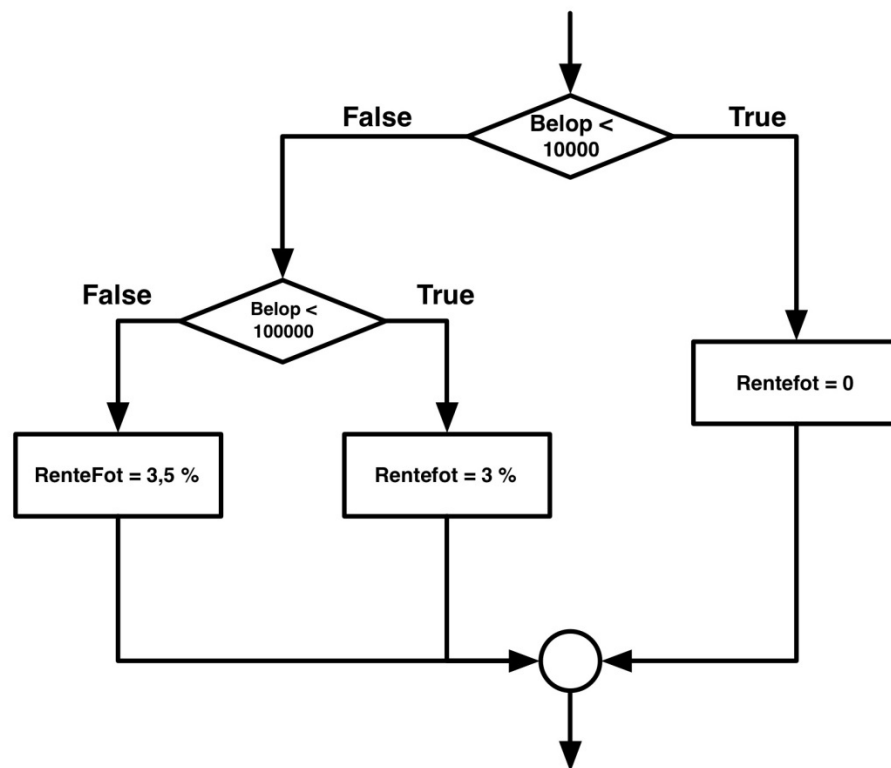
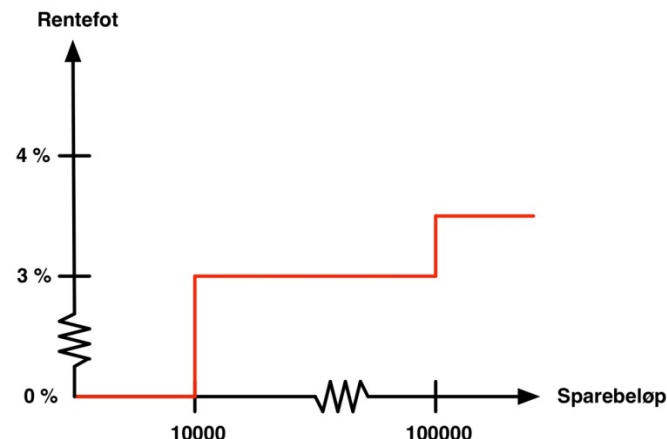
```
function tr = terminRente(belop)
% beregner tjent rente ut fra belop

    tr = belop*renteFot(belop);

end % function
```

renteFot-funksjonen

- 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 retefot ut fra belop

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

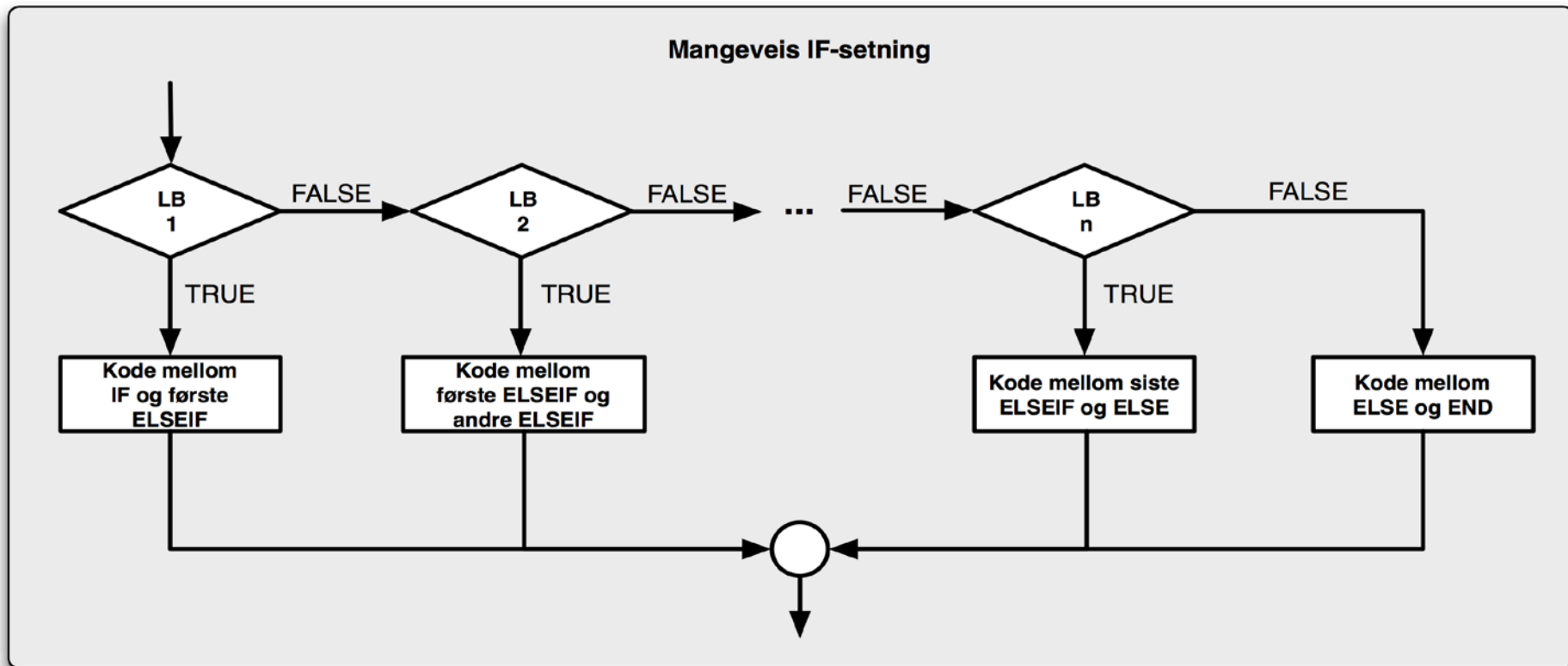
end % function
```

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
```