

# Skrive egen kode

*Skrevet av: Erik Aasmundrud*

*Kurs: Elm*

*Tema: Tekstbasert, Nettside*

*Fag: Matematikk, Programmering, Teknologi*

*Klassetrinn: 8.-10. klasse, Videregående skole*

## Introduksjon

Denne oppgaven har som hensikt i å lære å skrive sin egen kode, istedenfor å kopiere noe som allerede eksisterer.

Når man skal bli god til å programmere er det helt nødvendig å klare å skrive sin egen kode. Det er fristende å prøve å google problemet man har, eller se på noe som noen andre har gjort. Dette fører til at man ikke lærer like mye, så i denne oppgaven skal vi prøve å skrive vår helt egne kode.

## Steg 1: Få noe på skjermen

Mange utviklere synes det å starte på noe nytt er vanskelig. Programmering er ofte lettere når man ser hva man koder, selv om det ikke alltid er noe vi gjør. For å gjøre denne oppgaven litt lettere gir vi dere litt kode slik at dere får startet på oppgaven.

Nedenfor er utgangspunktet for dagens oppgave:

```
import Html exposing (text)

fire =
  4

ganger2 x =
  x * 2

main =
  text (toString [fire, ganger2 5])
```

 Gå til <http://elm-lang.org/try>

- ☐ Kopier inn koden som står ovenfor. Ikke trykk `Compile`
- ☐ Hva forventer du at skal stå dersom du trykker `Compile` ?
- ☐ Trykk `Compile` og se om du hadde rett

## Steg 2: Enkle funksjoner

La oss begynne i det små. Vi skal nå prøve å lage noen enkle matematiske funksjoner.

Prøv å lag følgende:

- ☐ En funksjon som tar inn to tall og plusser de sammen
- ☐ En funksjon som tar inn et tall og ganger det med 3
- ☐ En funksjon som tar inn et tall og ganger det med 10
- ☐ En funksjon som tar inn to tall og deler de på hverandre
- ☐ En funksjon som tar inn to tall og ganger tallene med hverandre
- ☐ En funksjon som tar inn tre tall og ganger tallene med hverandre
- ☐ En funksjon som tar inn åtte tall og ganger tallene med hverandre
- ☐ Test funksjonene dine med og se om du har gjort noe feil

Var det vanskelig? Overraskende mye programmering er å skrive funksjoner som gjør ting som dette.

## Steg 3: Vanskeligere funksjoner

Matte er gøy, så la oss fortsette å lage noen funksjoner som gjør enkle matematiske operasjoner. Siden vi allerede har gjort litt gange og dele, kan vi prøve oss på for eksempel kvadratroten. I elm tar man kvadratroten av et tall ved å skrive `sqrt` og deretter tallet. For eksempel `sqrt 16`.

Prøv å lage følgende:

- ☐ En funksjon som tar inn et tall og tar kvadratroten av tallet
- ☐ En funksjon som tar inn to tall og tar kvadratroten av det første tallet 4 ganger det andre tallet
- ☐ En funksjon som tar inn to tall og tar kvadratroten av det første tallet for deretter å gange dette med det andre tallet
- ☐ En funksjon som tar inn to tall og tar kvadratroten av det andre tallet for deretter å gange dette med det første tallet
- ☐ Test funksjonene dine med og se om du har gjort noe feil

## Steg 4: if-else-uttrykk

Når vi lager funksjoner er vi ofte interessert i å gjøre noe avhengig av hva det vi sender inn er. Dette kaller vi if-else-uttrykk. Dette har du sikkert vært borti flere ganger før, og nesten alle programmeringsspråk har dette.

I elm skriver man if-else-uttrykk på følgende måte:

```
sjekkTall tall =  
  if tall < 1 then  
    "tallet er mindre enn 1"  
  else if tall == 1 then  
    "tallet er 1"  
  else  
    "tallet er større enn 1"
```

Her sjekker vi om `tallet`, som er variabelen i funksjonen vår, er først mindre enn 1, så om den er lik 1. `else`-linjen til slutt er det som skjer dersom ikke noe av dette stemmer.

Spørsmål til funksjonen ovenfor:

- ☐ Hva slags type returnerer funksjonen?

Vi kan nå lage noen litt mer avanserte funksjoner. Prøv å lag følgende:

- ☐ En funksjon som tar inn et tall. Dersom tallet er større enn 9000, skal funksjonen returnere "Dette er over 9000!". Hvis det ikke er over 9000 skal funksjonen returnere "Prøv et høyere tall"
- ☐ Test funksjonen din og se om du har gjort noe feil
- ☐ En funksjon som tar inn to tall. Dersom tallene er like skal vi returnere "Tallene er like". Hvis tallene ikke er like skal vi returnere "Tallene er ikke like".

## Utfordring

- ☐ Lag dine helt egne funksjoner
- ☐ Klarer du å gjøre ukens mattelekse i elm?

## Absoluttverdi

Absoluttverdien til et tall er tallet uten fortegn. Her er noen eksempler:

```
> absoluttverdi 3
3 : number
> absoluttverdi -3
3 : number
> absoluttverdi 5
5 : number
> absoluttverdi -5
5 : number
> absoluttverdi 0
0 : number
```

# Utfordring

- ☐ Lag funksjonen `absoluttverdi` .

Hvordan lagde du funksjonen? Absoluttverdien til et tall kan vi regne ut på flere forskjellige måter. Jeg tipper at du ikke brukte ganger og kvadratrott! Prøv dette nå:

- ☐ Lag funksjonen `absoluttverdi` på nytt ved å bare bruke `*` og `sqrt` .

Lisens: CC BY-SA 4.0 (<http://creativecommons.org/licenses/by-sa/4.0/deed>)