

# □ Kontroller med musen

↓ LAST NED PDF

## Introduksjon

Denne oppgaven viser hvordan vi kan bruke musen til å kontrollere elm-programmene våre.

## Steg 1: Vise en posisjon

Vi starter ut med å vise en sirkel på en gitt posisjon. Her må vi kombinere to ting som har vært i tidligere oppgaver:

1. Vise en sirkel med musen (Tegning med SVG)
2. Bruke `Html.program` i stedet for `Html.beginnerProgram` (Tell sekunder)

Vi trenger `Html.program` fordi vi skal lytte på musebevegelser på samme måte som vi tidligere lyttet på tid.

Først setter vi opp SVG til å vise en ball vi kan styre posisjonen til:

```
import Svg exposing (svg, circle, rect)
import Svg.Attributes exposing (width, height, viewBox, cx, cy, r, fill, x, y, width, height)

import Mouse exposing (Position)

p = Position 10 20

model =
  { position = p
  }

main =
  view model

view model =
  svg
    [ width "500", height "500", viewBox "0 0 200 200" ]
    [ circle [ cx (toString model.position.x)
              , cy (toString model.position.y)
              , r "10", fill "blue" ] [ ]
    ]
```

Endre tallene 10 og 20 som inngår i verdien `p`.

- ☐ Hva gjør det første tallet?

- ☐ Hva gjør det andre tallet?

## Steg 2: Bruke `Html.program`

Her er et eksempel på hva `Html.program` trenger som input:

```
main =
  Html.program
    { init = init
    , view = view
    , update = update
    , subscriptions = subscriptions
    }
```

`init` er tilstanden til programmet vårt når vi starter opp.

`view` er hvordan vi viser tilstanden til programmet vårt.

`update` er hvordan tilstanden til programmet vårt reagerer på nye hendelser.

`subscriptions` er hvilke hendelser programmet vårt skal reagere på.

Verdien vi kalte `p` i forrige avsnitt kan vi bruke direkte som `init.view` har vi allerede skrevet. La oss fylle inn `update` og `subscriptions` som ikke gjør noe.

Vi legger inn en variant hvor `update` gir tilbake modellen uendret, og `subscriptions` ignorerer alt som kommer inn:

```
import Svg exposing (svg, circle, rect)
import Svg.Attributes exposing (width, height, viewBox, cx, cy, r, fill, x, y, width, height)

import Mouse exposing (Position)
import Html

p = Position 10 20

init =
  ( { position = p
    }
  , Cmd.none
  )

main =
  Html.program
    { init = init
    , view = view
    , update = update
    , subscriptions = subscriptions
    }

view model =
  svg
    [ width "500", height "500", viewBox "0 0 200 200" ]
    [ circle [ cx (toString model.position.x)
              , cy (toString model.position.y)
              , r "10", fill "blue" ] [ ]
    ]
```

```

type Msg =
    NoChange

type alias Model =
    { position : Position
    }

update : Msg -> Model -> (Model, Cmd Msg)
update msg model =
    (model, Cmd.none)

subscriptions model =
    Sub.batch - [ ]

```

## Steg 3: Koble på musen

Vi kobler på mus:

```

import Svg exposing (svg, circle, rect)
import Svg.Attributes exposing (width, height, viewBox, cx, cy, r, fill, x, y, width, height)

import Mouse exposing (Position)
import Html

p = Position 10 20

init =
    ( { position = p
      }
    , Cmd.none
    )

main =
    Html.program
        { init = init
        , view = view
        , update = update
        , subscriptions = subscriptions
        }

view model =
    svg
        [ width "500", height "500", viewBox "0 0 200 200" ]
        [ circle [ cx (toString model.position.x)
                  , cy (toString model.position.y)
                  , r "10", fill "blue" ] [ ]
        ]

type Msg =
    MouseAt Position

type alias Model =

```

```
{ position : Position  
}
```

```
update : Msg -> Model -> (Model, Cmd Msg)  
update msg model =  
  case msg of  
    MouseAt pos ->  
      ( { model | position = pos }  
      , Cmd.none)
```

```
subscriptions model =  
  Sub.batch [Mouse.moves MouseAt]
```

Her er det noe rart! Ballen er ikke samme sted som pekeren. Hvorfor?

- ☐ Prøv å endre viewboxen i SVG:

```
[ width "500", height "500", viewBox "0 0 200 200" ]
```

- ☐ Hvordan henger dette sammen?

Lisens: CC BY-SA 4.0