

Ø02: ActionScript – MovieClip-objekter, eksterne bilder

Denne øvingen baserer seg også til en viss grad på ”les og trøkk”, så husk å bruk tid på å **reflektere** over hva koden ”gjør”, og hvordan du kan tilpasse den til å løse egne problemer.

Oppgave 1 – Programmering av MovieClip-objekter

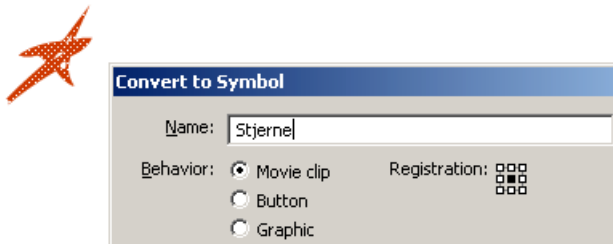
Bakgrunn: MovieClip-objektet

Grafikk som er gjort om til et MovieClip vil automatisk bli en instans av **MovieClip-klassen**. Instanser av MovieClip-klassen ”arver” alle **egenskaper**, **metoder** og **hendelseshåndterere** som denne klassen har innbygd. For å enkelt kunne referere til hver enkelt MovieClip-objekt bør de gis **instansnavn** (Instance Name); noe som kan gjøres i Properties-panelet.

Bakgrunn: Bruk av objektreferanser

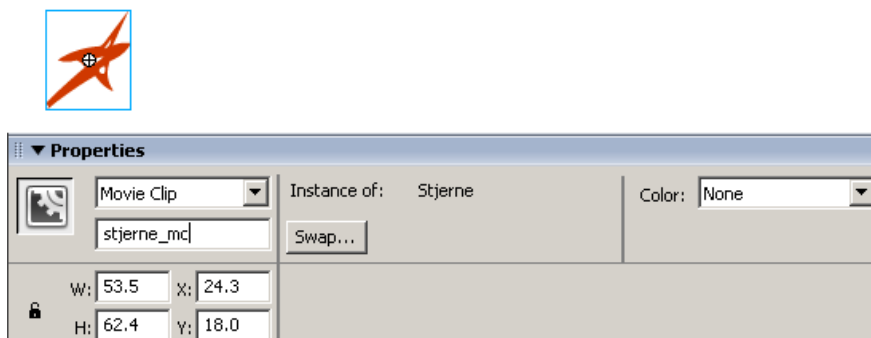
Her er et eksempel på hvordan man kan gjøre om grafikk i Flash til programmerbare objekter:

1. Opprett et nytt Flash-dokument.
2. Tegn et eller annet vha. tegneverktøyet i Flash (i Frame 1).
3. Merk det du har tegnet, og gjør det om til et Symbol (MovieClip) – Modify | Convert to Symbol... (F8). Gi symbolet et passende navn:

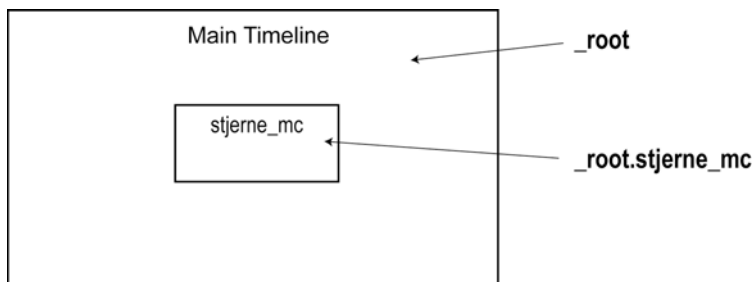


4. Det navnet du gir symbolet ved oppretting blir nærmest et klassenavn, og det nye symbolet ligger nå tilgjengelig i Library (Window | Library, Ctrl+L) slik at det enkelt kan opprettes nye instanser av symbolet. Hver instans kan gis egne instans-navn (Instance Name), og det er dette som blir det objektnavnet som kan benyttes for å referere til objektet i ActionScript-koden.

Merk instansen (klikk) og skriv et passende ”Instance Name” i Properties-panelet:



5. Situasjonen kan nå illustreres slik:



På hovedtimeline ligger **stjerne_mc**. Hovedtimeline kan refereres til med navnet **_root**. Hvis du skriver kode på hovedtimeline (Frame 1) kan du enten referere til objektet "stjerne_mc" med en *relativ referanse* (**stjerne_mc** eller **this.stjerne_mc**), eller med en *absolutt referanse* (**_root.stjerne_mc**). Som om ikke det er nok så kan objekter også nås vha. en "assosiativ" array-notasjon: **_root["stjerne_mc"]**. Dette er nyttig i de tilfeller hvor vi er nødt til å lage objektnavnene dynamisk (for eksempel **bilde1**, **bilde2**, **bilde3**, ...; vha. "bilde" + i).

Klikk på Frame 1 i timeline, og åpne Actions-panelet (F9). Prøv deg fram, og sjekk at alle disse fire variantene for å endre objektets x-verdi vha. kode fungerer – prøv en om gangen, og test med Control | Test Movie (Ctrl+Enter):

```

stjerne_mc._x = 300;           // variant 1
this.stjerne_mc._x = 300;     // variant 2
_root.stjerne_mc._x = 300;    // variant 3
_root["stjerne_mc"]._x = 300; // variant 4
  
```

Bakgrunn: Egenskaper

I ActionScript-kode kan egenskapene til aktuelle MovieClip's leses av og (de fleste) settes direkte:

Eksempler på programmering av egenskaper:

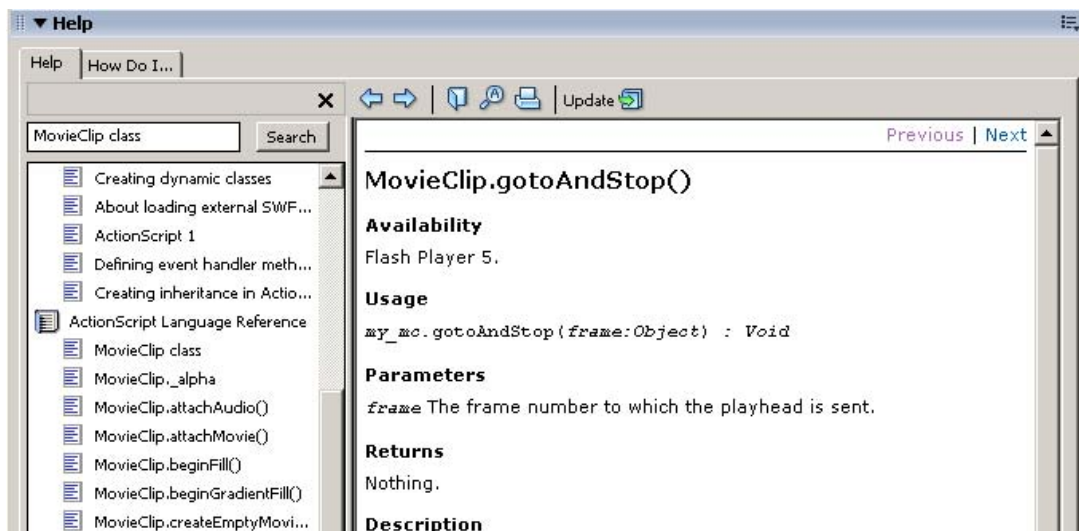
```

var xVerdien = firkant_mc._x;
var yVerdien = firkant_mc._y;
firkant_mc._x = 100;
firkant_mc._y = firkant_mc._y + 10;
  
```

Bakgrunn: Metoder

Metoder er funksjoner som inneholder et ukjent antall instruksjoner som vil resultere i at noe bestemt skjer. Metoder kan (men trenger ikke nødvendigvis) å returnere en verdi som kan "fanges opp" vha. en variabel (for eksempel returnerer metoden "getDepth()" et tall som indikerer dybden/stablingsrekkefølgen til et MovieClip).

Alle metoder **skal** ha **parenteser** bak metodenavnet, og i enkelte tilfeller skal det inne i parentesene være en eller flere **parametere**. For å finne eventuelle parametere, og hva metoden eventuelt returnerer slår du selvfølgelig opp i hjelpefunksjonen til Flash (F1):



Eksempler på bruk av metoder:

```
brikke_mc.startDrag();  
brikke_mc.gotoAndStop(2);  
var plassering = brikke_mc.getDepth();
```

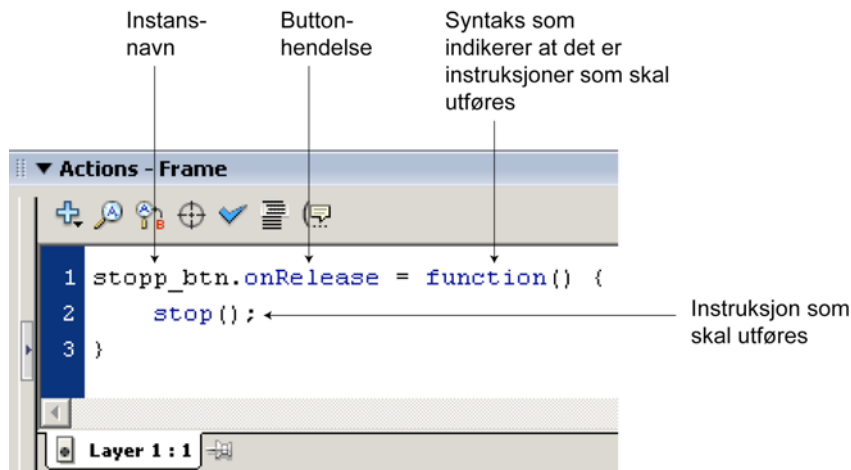
Bakgrunn: Hendelseshåndterere

Hendelseshåndterere (event handlers) er funksjoner som du kan programmere for å håndtere hendelser som et objekt utsettes for (for eksempel at noen klikker på det).

En utdypende forklaring: Dette dreier seg egentlig om hendelser som "kommer fra" det Windows-baserte operativsystemet (Windows XP, Macintosh, Linux, osv.), og som kan fanges opp av applikasjonene som kjører. Når hendelser blir sendt til en Flash Player, sender den informasjon om hendelsen videre til aktuelle objekter (Button- og MovieClip-objekter) i den kjørende Flash-applikasjonen.

Du kan skrive hendelseshåndterere på Button-objekter og MovieClip-objekter. **Igjen:** ved å slå opp i hjelpefunksjonen på "Button class" og "MovieClip class" finner du en oversikt over alle aktuelle hendelser objektene kan utsettes for.

Dere har allerede lært hvordan man kan skrive hendelseshåndterere "på" selve objektene (on (release) { ... }), men hendelseshåndterere kan også (med fordel) skrives direkte på timeline (Frame x). Du benytter da følgende syntaks:



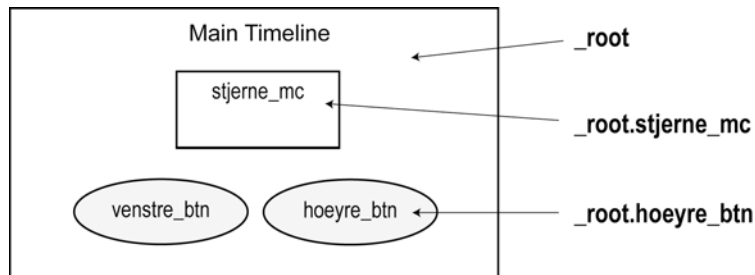
Eksempler:

```
stopp_btn.onRelease = function() {
    stop();
}

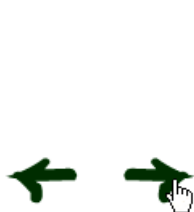
brikke_mc.onPress = function() {
    this.gotoAndStop(2);
    this.startDrag();
}
```

Oppgave: Programmert flytting av objekter

Lag et MovieClip, gi den et instansnavn (for eksempel **stjerne_mc**). Lag deretter to knapper (Button-objekter), og gi dem instansnavn (**venstre_btn** og **hoeyre_btn**):



Programmer hendelseshåndterere på knappene slik at når du klikker på "venstre_btn" så flyttes "stjerne_mc" 10 piksler til venstre (-), og når du klikker på "hoeyre_btn" så flyttes "stjerne_mc" 10 piksler til høyre (+):



Prøv gjerne ut andre egenskaper også (`_alpha`, `_xscale`, `_yscale`, ...).

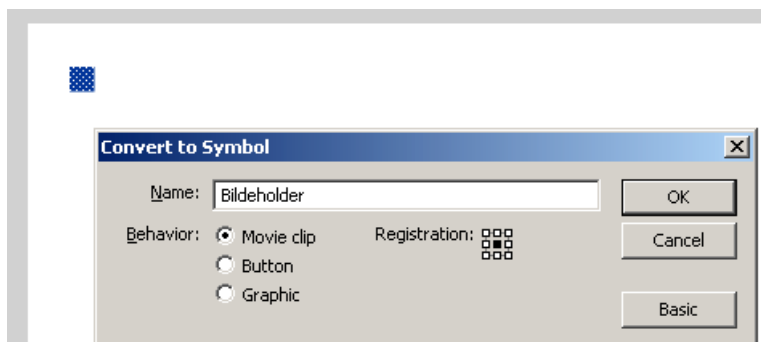
Oppgave 2 – Dynamisk innlasting av bilder

Bakgrunn: loadMovie

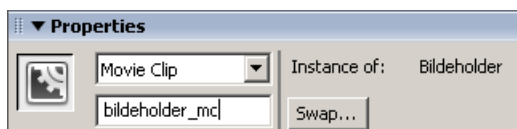
I denne oppgaven skal vi se på muligheten vi har til å laste inn bilder dynamisk til Flash. For å gjøre dette kan vi benytte MovieClip-metoden "loadMovie". Denne metoden kan benyttes for å laste inn eksterne Flash-movies (.swf) eller eksterne JPEG-bilder (.jpg). SWF-fila eller JPG-bildet vil da **erstatte** det MovieClip'et som metoden brukes på.

Nedenfor vises et eksempel på dette. Vi tegner et grafisk element i Flash, gjør det om til et MovieClip-objekt, og erstatter dette objektet med et eksternt bilde:

1. Opprett et nytt blankt dokument.
2. Tegn et eller annet vha. tegneverktøyet i Flash (for eksempel en firkant). Merk grafikken og gjør den om til et symbol (MovieClip) – Modify | Convert to Symbol... (F8):



3. Gi det nye symbolet instansnavnet **bildeholder_mc**:



4. Test Flash-applikasjonen med Control | Test Movie (Ctrl+Enter), og sjekk at du får fram (den opprinnelige) grafikken til "bildeholder_mc".
5. Nå skal du benytte MovieClip-objektet "bildeholder_mc" for å laste inn et eksternt bilde (*merk* at hvis du **ikke** er koblet til internett må du finne et JPEG-bilde selv og kopiere det inn i den samme mappen som Flash-prosjektet – benytt da **kun** navnet på JPEG-filen i stedet for den fulle URL-adressen som er brukt nedenfor). Skriv følgende kode i Frame 1 i Timeline:

```
_root.bildeholder_mc.loadMovie  
("http://multimedia.hint.no/mmt2004/nmstafetter05/images/janette1.jpg");
```

6. Test Flash-applikasjonen igjen (Ctrl+Enter), og sjekk at du får fram et bilde i stedet for den opprinnelige grafikken til "bildeholder_mc". Voila!!!!

Oppgave: Knapper for å bytte bilde

Lag to eller flere knapper som kan benyttes for å bytte ut bildet i "bildeholder_mc" ovenfor. **knapp1_btn** benyttes for å laste inn bilde 1, **knapp2_btn** benyttes for å laste inn bilde 2, osv.

Du kan for eksempel benytte noen av bildene du finner i listen nedenfor (listen inneholder URL-adressene til bildene som befinner seg på internettensiden <http://multimedia.hint.no/mmt2004/nmstafetter05/>):

<http://multimedia.hint.no/mmt2004/nmstafetter05/images/eirik1.jpg>
<http://multimedia.hint.no/mmt2004/nmstafetter05/images/eirik2.jpg>
<http://multimedia.hint.no/mmt2004/nmstafetter05/images/janette1.jpg>
<http://multimedia.hint.no/mmt2004/nmstafetter05/images/janette2.jpg>
<http://multimedia.hint.no/mmt2004/nmstafetter05/images/severin.jpg>
<http://multimedia.hint.no/mmt2004/nmstafetter05/images/fredrik.jpg>
http://multimedia.hint.no/mmt2004/nmstafetter05/images/tom_rolf.jpg
http://multimedia.hint.no/mmt2004/nmstafetter05/images/stine_kristin.jpg
<http://multimedia.hint.no/mmt2004/nmstafetter05/images/ta.jpg>
<http://multimedia.hint.no/mmt2004/nmstafetter05/images/tom.jpg>
<http://multimedia.hint.no/mmt2004/nmstafetter05/images/julian3.jpg>
<http://multimedia.hint.no/mmt2004/nmstafetter05/images/monitorer.jpg>
<http://multimedia.hint.no/mmt2004/nmstafetter05/images/gunnar1.jpg>
<http://multimedia.hint.no/mmt2004/nmstafetter05/images/kontrollrom1.jpg>
<http://multimedia.hint.no/mmt2004/nmstafetter05/images/kontrollrom5.jpg>
http://multimedia.hint.no/mmt2004/nmstafetter05/images/martin_gunnar1.jpg
http://multimedia.hint.no/mmt2004/nmstafetter05/images/martin_gunnar2.jpg
<http://multimedia.hint.no/mmt2004/nmstafetter05/images/kontrollrom4.jpg>
<http://multimedia.hint.no/mmt2004/nmstafetter05/images/kontrollrom6.jpg>
<http://multimedia.hint.no/mmt2004/nmstafetter05/images/kontrollrom2.jpg>
<http://multimedia.hint.no/mmt2004/nmstafetter05/images/julian1.jpg>
<http://multimedia.hint.no/mmt2004/nmstafetter05/images/julian2.jpg>
<http://multimedia.hint.no/mmt2004/nmstafetter05/images/alex.jpg>
<http://multimedia.hint.no/mmt2004/nmstafetter05/images/kaos2.jpg>
<http://multimedia.hint.no/mmt2004/nmstafetter05/images/kaos3.jpg>
<http://multimedia.hint.no/mmt2004/nmstafetter05/images/lydmixer.jpg>
<http://multimedia.hint.no/mmt2004/nmstafetter05/images/joar.jpg>
<http://multimedia.hint.no/mmt2004/nmstafetter05/images/kontrollrom3.jpg>
<http://multimedia.hint.no/mmt2004/nmstafetter05/images/haavard1.jpg>
<http://multimedia.hint.no/mmt2004/nmstafetter05/images/skisse.jpg>
<http://multimedia.hint.no/mmt2004/nmstafetter05/images/webtv.jpg>
http://multimedia.hint.no/mmt2004/nmstafetter05/images/oppsummering_dag1.jpg
http://multimedia.hint.no/mmt2004/nmstafetter05/images/oppsummering_dag2.jpg

Listen ovenfor (pluss en ”semikolondelt” liste) finner du også her: <http://multimedia.hint.no/filer/MMT230/bildeliste.html>

Her er et eksempel på hvordan resultatet kan se ut:



Oppgave: Inn- og ut-zooming

Prøv i tillegg å legge til knapper for inn- og ut-zooming av bildet. **Tips:** Egenskapene **bildeholder_mc._xscale** og **bildeholder_mc._yscale** brukes for å sette skalering i prosent. Sett gjerne skaleringen til 50% i utgangspunktet (ved innlasting).

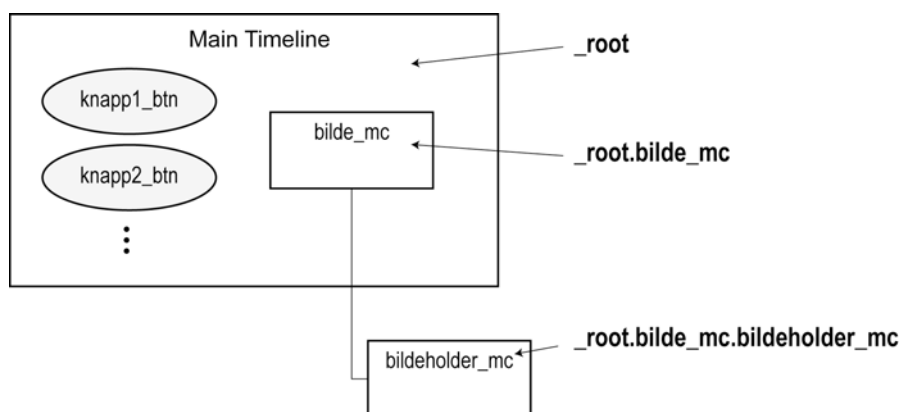
Her er et eksempel på hvordan resultatet kan se ut:



Oppgave 3 – Interaktive bilder

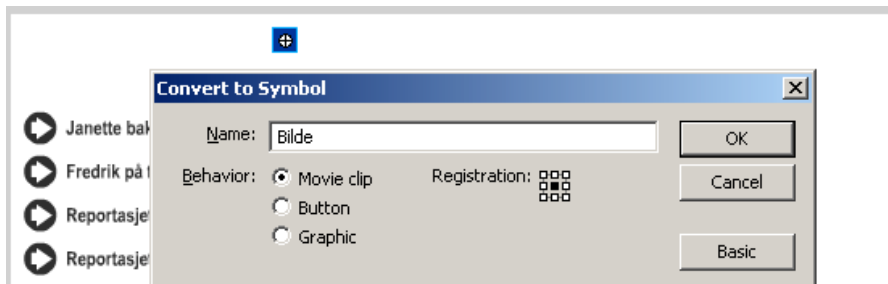
Bakgrunn: MovieClip i MovieClip (objekthierarki)

Det er ikke så enkelt å gjøre bilder direkte interaktive; dvs. å gjøre "bildeholder_mc" klikkbar/flyttbar (startDrag) etc. Et triks for å få til dette er å legge "bildeholderen" inn et annet MovieClip; for eksempel **bilde_mc**. Objekthierarkiet ser da slik ut:



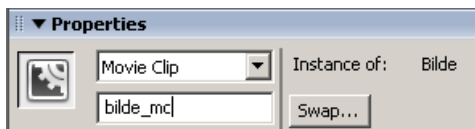
Illustrasjonen ovenfor beskriver ønsket situasjon. Siden vi allerede har MovieClip'et "bildeholder_mc" på hovedtimelinen (**_root.bildeholder_mc**) vil det være en smal sak å legge dette inn i et nytt MovieClip ("bilde_mc"):

1. Merk bildeholder-objektet (klikk på det), og gjør den *nok en gang* om til et symbol – Modify | Convert to Symbol... (F8):



”bildeholder_mc” befinner seg nå på timeline til det nye MovieClip’et i stedet for på hovedtimeline...

2. Gi det nye symbolet instansnavnet **bilde_mc**:



3. Dobbeltklikk på symbolet to ganger (helt til du ”kommer inn til grafikken”) for å kontrollere at Bildeholder-objektet virkelig ligger inne i Bilde-objektet:



(Utfordrende?) Oppgave: Drag and Drop

Gjør nå bildet flyttbart ved å benytte `startDrag()`- og `stopDrag()`-funksjonene på ”bilde_mc”-objektet.

Tips 1: I forhold til forrige oppgave må du nå gjøre om referansen til bildeholder-objektet (find/replace...) – ”_root.bildeholder_mc” skal gjøres om til ”_root.bilde_mc.bildeholder_mc”.

Tips 2: Når du trykker ned museknappen på ”bilde_mc” (onPress) skal du starte å dra objektet, og når du slipper museknappen igjen skal du slutte å dra objektet.