



# Logo med FXML

## Introduksjon

En viktig del av det å lage app-er er å utforme skjerminnholdet, dvs. å lage et godt brukergrensesnitt. Formålet til denne leksjonen er å lære hvordan du kan bruke **FXML** for å lage innhold. I tillegg skal du lære hvordan du får app-en til å last inn og vise innholdet. Vi skal lage en komplisert, den skal bare vise frem logoene til Kodeklubben og NTNU.

### Kodeklubben

---



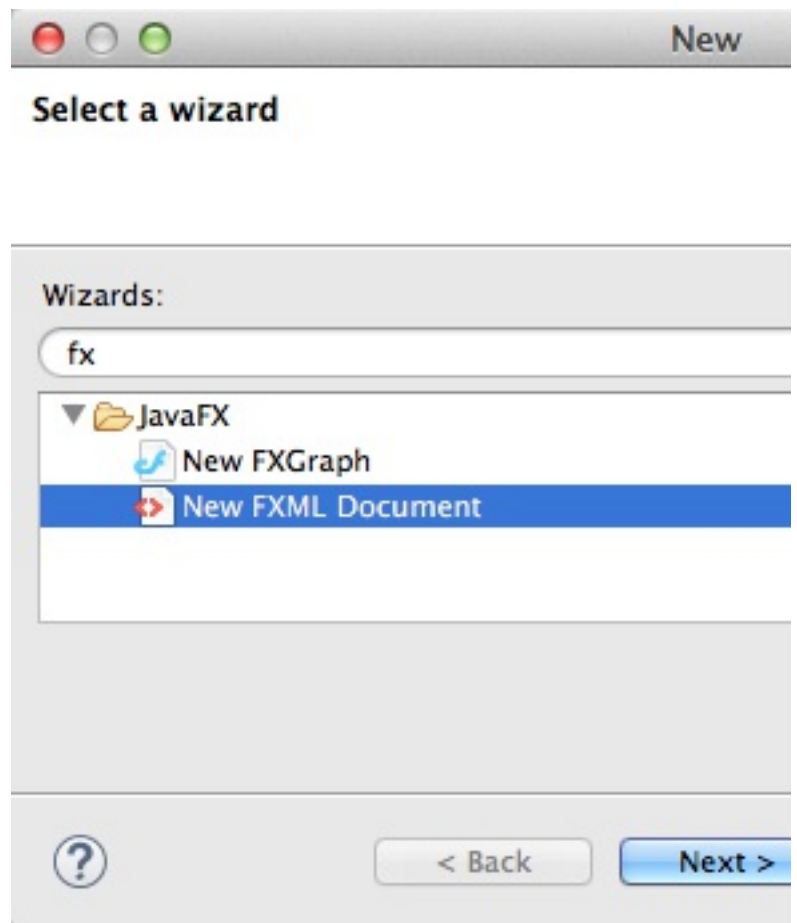
I leksjonen [Hello world](#) lærte du om hvordan du kan lage en app som viser innhold. Du skrev du Java-kode som *laget* tekst-objektet, *satte grafiske egenskaper* på objektet. Dette kan fort bli lang og uoversiktlig, og det kan være vanskelig å skrive kode som viser innholdet vil bli seende ut.

For å gjøre arbeidet med skjerminnhold enklere, så finnes det en eger som ligner på HTML ved at det bruker XML-elementer, f.eks. `<Text ...>` for å beskrive innholdet. Når en skal lage FXML-filer, kan en enten redigere eksisterende eller lage egen tegne-app som leser og skriver FXML-filer. Vi skal se på begge delene.

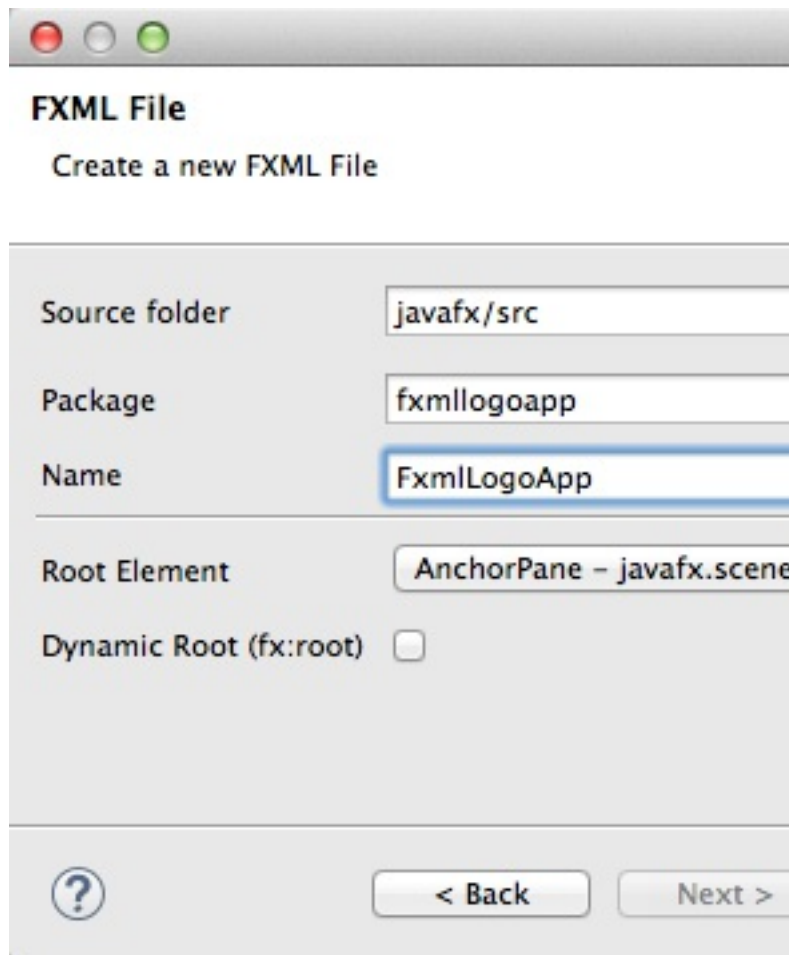
---

# Steg 1: Opprette app-pakke FXML-fil

- ☐ Lag først en ny app-pakke og app-klasse som forklart i [Hello world](#). Lag **FxmlLogoApp** som navn på app-klassen.
- ☐ Høyreklikk på **fxmlogo**-pakken og velg **New > Other...** så du kan velge en ny filtype. Vi skal bruke den som heter **New FXML Document**. Det enkleste utvalget:



Trykk **Next** for å velge veiviseren. Du vil da få opp følgende skjerm:



The image shows a dialog box titled "FXML File" with the subtitle "Create a new FXML File". It contains several input fields: "Source folder" with the value "javafx/src", "Package" with the value "fxmllogoapp", "Name" with the value "FxmlLogoApp" (highlighted with a blue border), "Root Element" with the value "AnchorPane - javafx.scene", and "Dynamic Root (fx:root)" with an unchecked checkbox. At the bottom, there is a question mark icon, a "< Back" button, and a "Next >" button.

|                        |                           |
|------------------------|---------------------------|
| Source folder          | javafx/src                |
| Package                | fxmllogoapp               |
| Name                   | FxmlLogoApp               |
| Root Element           | AnchorPane - javafx.scene |
| Dynamic Root (fx:root) | <input type="checkbox"/>  |

? < Back Next >

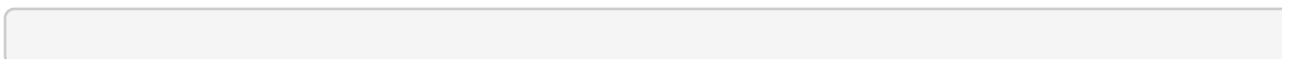
Fyll inn **FxmlLogoApp** og trykk **Finish**. Du vil da få opp en FXML



The image shows a code editor window with the file name "FxmlLogoApp.java". The code is as follows:

```
1 <?xml version="1.0" encoding="UTF-8" ?>
2
3 <?import javafx.scene.layout.AnchorPane ?>
4
5 <AnchorPane xmlns:fx="http://javafx.com/fxml" ?>
6     <!-- TODO Add Nodes -->
7 </AnchorPane>
8
```

- Denne FXML-en gir bare et tomt panel (av typen **AnchorPane**), teksten slik at den blir som følger:



An empty rectangular text box with a light gray border.

...

FXML-editoren har samme type kode-komplettering som Java-editorer ctrl-mellomrom og editoren vil både fylle ut resten av **Rectangle**-nav (tilsvarende import-setningen i Java):

```
<?import javafx.scene.shape.Rectangle?>
```

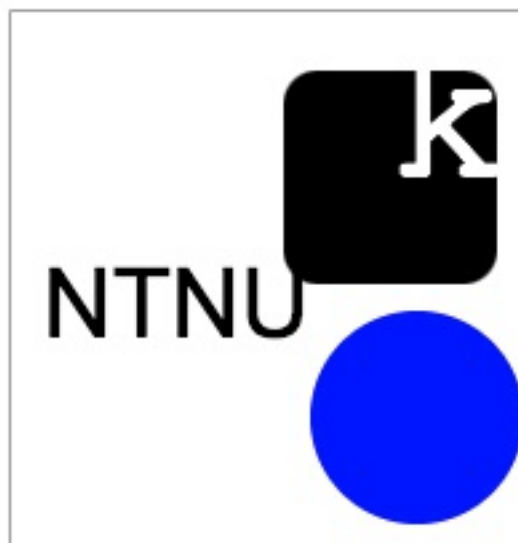
(Hvis du ikke bruker kode-kompletteringsfunksjonen, så må du skrive kode-komplettering på attributt-navnene **layoutX**, **layoutY**, **width**, **h**



Mens du redigerer FXML-koden, så kan det være kjekt å se hvor vises frem. Eclipse har et eget panel kalt **JavaFX preview** som frem innholdet. Dette åpnes ved å velge **Window > Show View..** filtrere lista (skriv 'pre', så panelet blir enklere å finne.

```
FxmlLogoApp.fxml  figurer1.fxml  FxmlLogoApp.fxml
1  <?xml version="1.0" encoding="UTF-8"?>
2
3  <?import javafx.scene.layout.AnchorPane?>
4  <?import javafx.scene.shape.Rectangle?>
5  <?import javafx.scene.text.Text?>
6  <?import javafx.scene.shape.Circle?>
7  <?import javafx.scene.text.Font?>
8
9  <AnchorPane xmlns:fx="http://javafx.com/fxml"
10      <Rectangle layoutX="100" layoutY="20" width="25" height="25"
11          arcWidth="25" arcHeight="25"
12      />
13      <Text layoutX="140" layoutY="60" text="K"
14          <font>
15              <Font name="Courier" size="72"/>
16          </font>
17      </Text>
18      <Circle layoutX="150" layoutY="150" radius="50" fill="blue"
19      <Text layoutX="10" layoutY="120" text="NTNU"
20          <font>
21              <Font name="Arial" size="36"/>
22          </font>
23      </Text>
24  </AnchorPane>
25
```

Problems @ Javadoc Declaration Console



Som du ser så inneholder FXML-fila allerede mange av de grafiske elementene, men ikke alle. Din jobb blir å justere på FXML-koden, så logoene

# Litt om FXML

Du har kanskje kjent igjen mange av ordene i FXML-fila fra Java-koder tilfeldig. Enkelt sagt så tilsvarer elementnavnene i FXML-koden (ordet klasser, og attributtnavnene tilsvarer egenskaper, f.eks. **layoutX**, **width** tilsvarer egenskaper, f.eks. **font**. Når FXML-koden blir lest inn, så vil h instans (et Java-objekt) av den klassen, og hvert attributt bli brukt til å **set**-metode. Følgende Java- og FXML-kode betyr omtrent det samme:

```
Circle circle = new Circle();
circle.setLayoutX(150);
circle.setLayoutY(150);
circle.setRadius(40);
circle.setFill(Color.BLUE);
```

```
<Circle layoutX="150" layoutY="150" radius="40" fill="blue"/>
```

Noen typer objekter, f.eks. paneler (typene **Pane**, **HBox**, **VBox** og **AnchorPane**) må en legge den ene til **children**-lista til den andre, mens en i FXML l

```
Pane pane = new Pane();
Circle circle = new Circle();
...
pane.getChildren().add(circle);
```

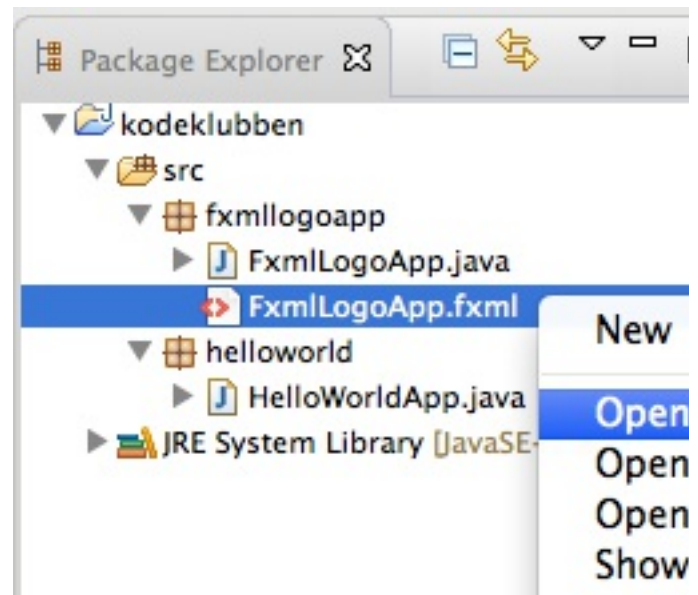
```
<Pane>
  <Circle layoutX="150" layoutY="150" radius="40" fill="blue" />
</Pane>
```

## Steg 2: Åpne og redigere F SceneBuilder

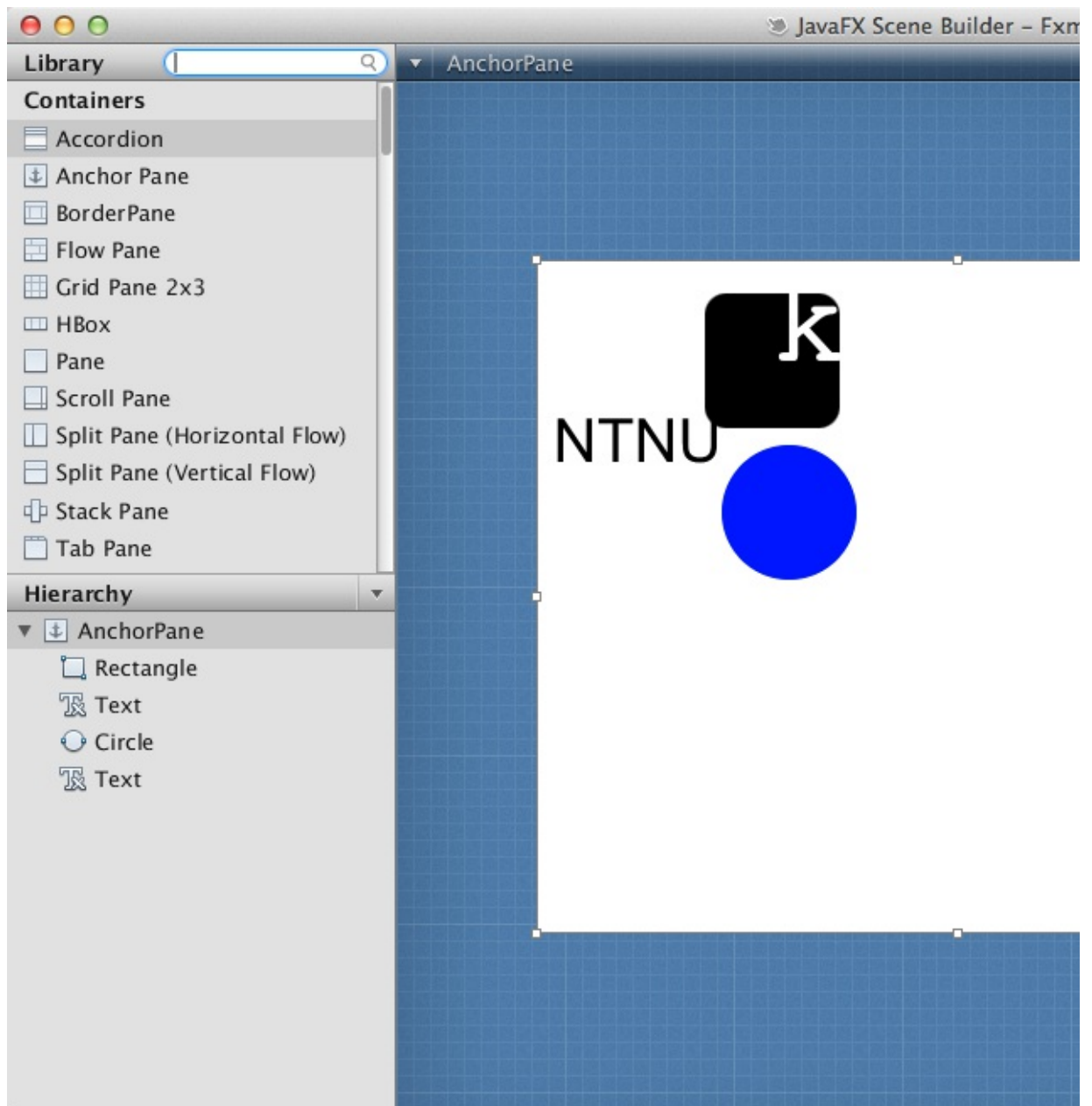
Det går greit å bruke FXML-editoren for enkel redigering (eller mer ko

enklere alternativ, nemlig en grafisk editor kalt **SceneBuilder**. En kan også bruke Scene Builder (ofte bare kalt SceneBuilder), siden den bruker FXML so

- ☐ Høyreklikk på FXML-fila og velg **Open with SceneBuilder**



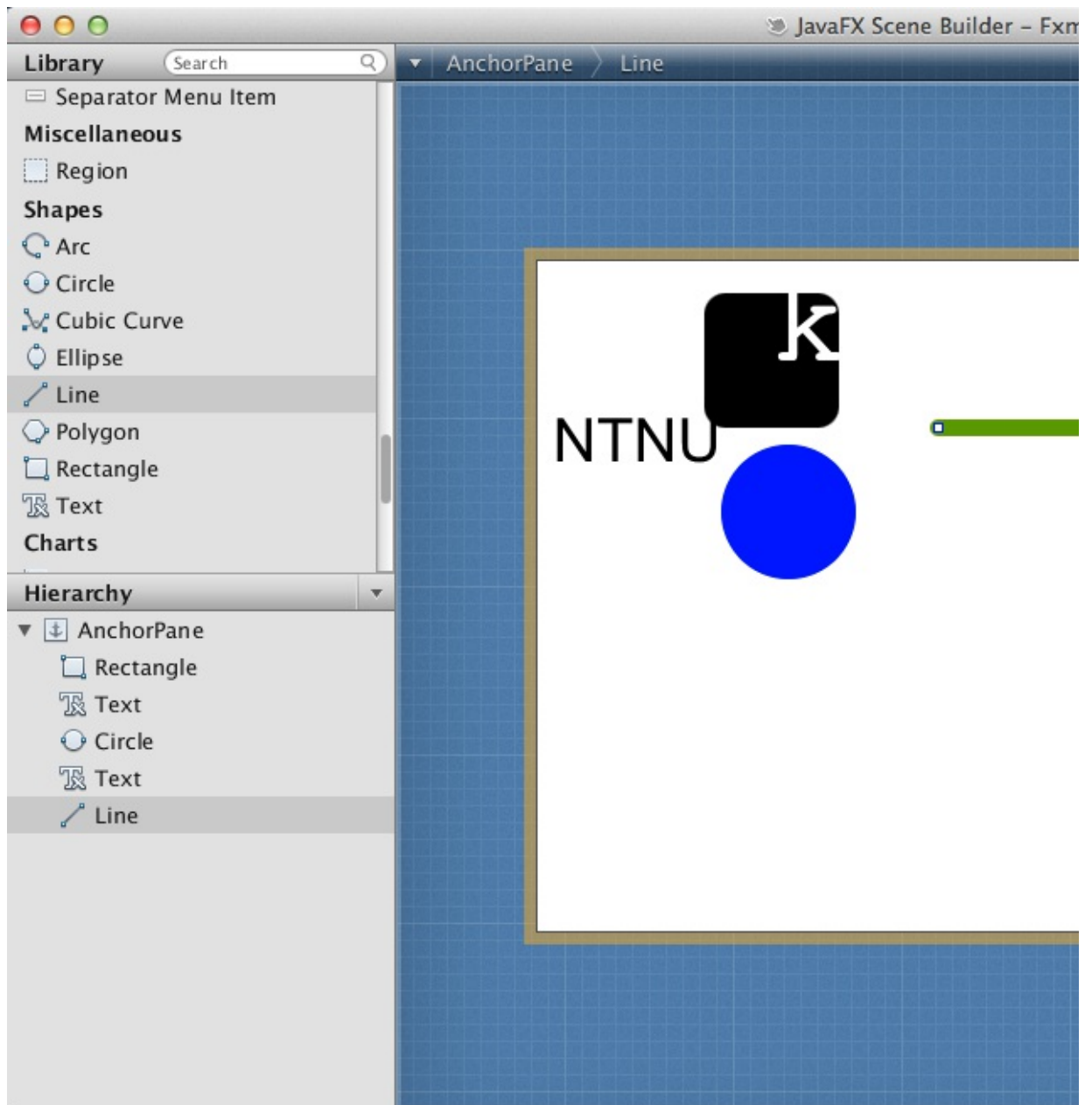
Hvis du har installer SceneBuilder riktig, så skal du etter litt ven



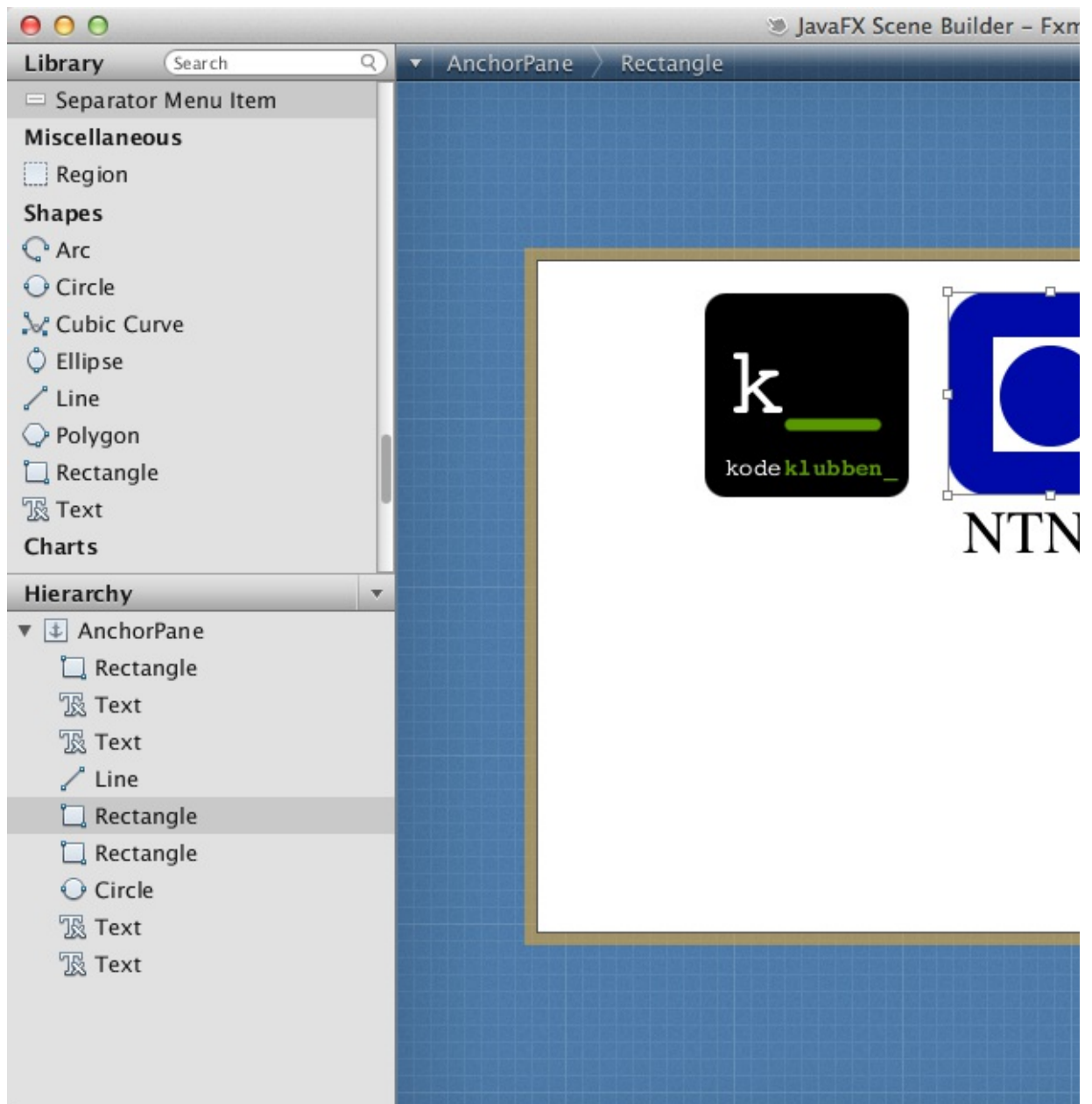
Som du ser så inneholder panelet i midten de samme grafiske elementer som du så i figuren tidligere. Du kan imidlertid *redigere* på dem, ved å velge, dra og slippe, som i vanlig grafisk design. Nederst til venstre ser du trestrukturen av objekter, med et panel (bilde) som inneholder et rektangel (**Rectangle**), en sirkel (**Circle**) og to tekst-objekter (**Text**).

Øverst til venstre ser du en kategorisert liste med elementer som du kan dra og slippe i midten. I figuren vises bare såkalte **Containers**, som inneholder andre elementer. Du finner du bl.a. **Controls** (interaktive elementer) og **Shapes** (grafiske elementer). Hvis du slipp den litt til høyre for det svarte rektangelet. Hvis du vil endre egenskaper i panelet til høyre. Prøv f.eks. å gjøre streken tykkere (**Stroke** = #569900) og enden til avrundet (**Stroke Line Cap** =





Dette er omtrent de grafiske elementen du trenger for å lage Kc redigeringsmulighetene til å få tegningen til å bli seende omtrent

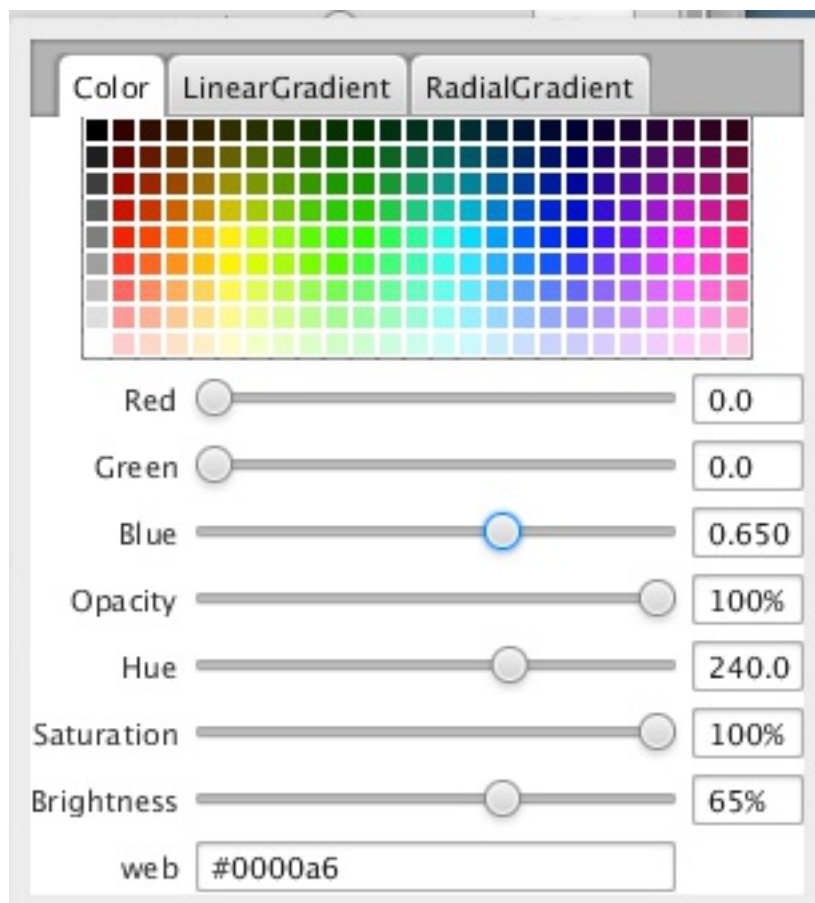


Her er noen tips om nyttige redigeringsfunksjoner som du kan bruke:

- ☐ Du kan raskt lage en kopi (duplisere) et element med **Edit > Duplicate**
- ☐ Du kan runde av hjørnene på et rektangel ved å endre **AnchorPane > Corner Radius**
- ☐ Når du klikker på **Stroke**- og **Fill**-egenskapene, så kommer du til et valgmeny med flere alternativer (se under til venstre)
- ☐ Tilsvarende kan du stille inn navn og størrelsen på font-en (se under til høyre)

## Valg av farge

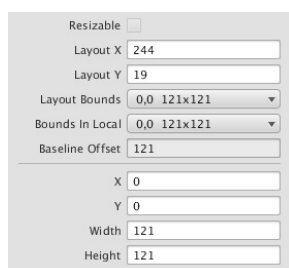
---



Spesielt fin-innstillingen av plassering og dimensjoner kan være du egenskaper med kategorien **Layout**, bl.a. koordinater og størrelse avhenger av hvilket grafisk element som er valgt. I figuren under

## Layout-egenskaper

---



Det er egentlig disse egenskapene du setter for elementene og de små hjørne-boksene som er presist.

# Steg 3: Få appen til å vise FXML-fila

Det siste trinnet i denne leksjonen er å få app-en til å vise frem det gr redigere koden i app-klassen din, som du laget i trinn én og heter **Fxm**

- ☐ Husk å lagre FXML-fila i SceneBuilder før du går tilbake til Eclips
- ☐ Klikk i FXML-editoren. Den skal da vise frem det nye innholdet, s SceneBuilder.
- ☐ Det er greit å gjøre endringer rett i FXML-koden i editor. Når du samme endringene der. Hvis ikke så må du kanskje lukke den o
- ☐ Velg editoren med **FxmlLogoApp**-klassen i (eller åpne den ved slik at du får følgende kode:

```
package fxmllogoapp;

import java.io.IOException;

import javafx.application.Application;
import javafx.fxml.FXMLLoader;
import javafx.scene.Scene;
import javafx.scene.layout.Pane;
import javafx.stage.Stage;

public class FxmlLogoApp extends Application {

    public void start(Stage primaryStage) throws IOExcepti
        FXMLLoader fxmlLoader = new FXMLLoader(FxmlLogoApp.c
        Pane root = fxmlLoader.load();
        primaryStage.setScene(new Scene(root));
        primaryStage.show();
    }
```

```
public static void main(String[] args) {  
    launch(FxmlLogoApp.class, args);  
}  
}
```

Trikset her er å lage et **\*\*FXMLLoader\*\***-objekt (første linje) med **load**-metoden (andre linje).

+ Kjør **app-en** ved å **høyre-klikke** på fila eller i editoren og etter litt venting) dukke opp et vindu som dette:



# Hva har du **l**ært?

- + lage **en** FXML-fil med Eclipse sin veiviser
- + redigere FXML i Eclipse sin FXML-editor.
- + **se** forhåndsvisning av FXML-innholdet i JavaFX Preview-panel
- + åpne FXML-fila i JavaFX Scene Builder med **`Open** with SceneB
- + legge nye elementer til FXML-fila ved å dra og slippe fra S
- + redigere grafiske egenskaper i SceneBuilder
- + skrive **app**-kode som laster inn og viser frem FXML-fila i et

I leksjonen KalkulatorFX (to be continued) vil du **l**ære hvordan e **app-en** levende (interaktiv)!

**Lisens:** [CC BY-SA 4.0](#)