

# GUI

Grafisk brukergrensesnitt i Java med  
JavaFX

# GUI-rammeverk i java

Awt

Swing

JavaFX



awt



- JavaFX er et svært rammeverk med mange, mange klasser
- Vi kan bare dekke en liten flik og må gå fort over detaljer
- Konsentrerer oss om å forstå noen basale begreper
- En helt annen måte å programmere på
- Noen viktige stikkord:

Scenegraf (Scene graph)

Node (grennode og løvnode)

Beholdere

Scener

Paneler

Stage

Scenegraf (Scene graph)

Node (grennode og løvnode)

Beholdere

Scener

Paneler

Stage

*Når du kan forklare disse begrepene og sammenhengen mellom dem, har du forstått mye !*

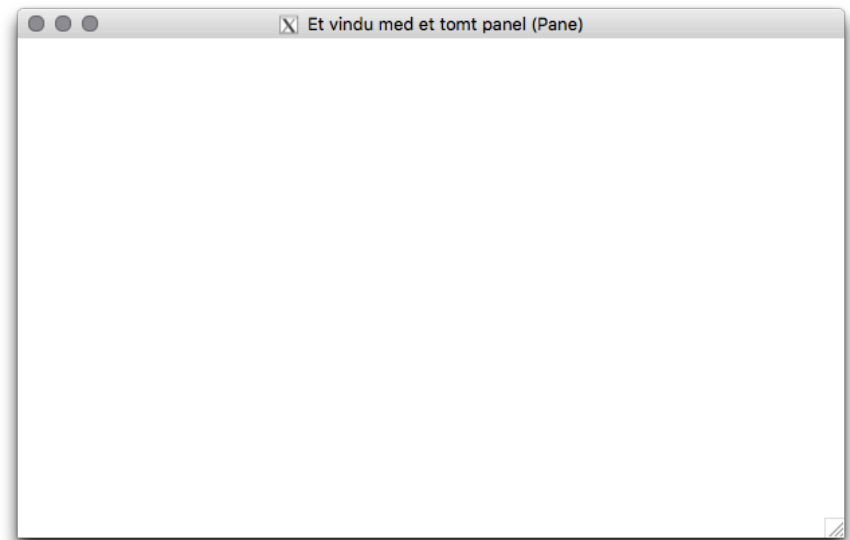
```
import javafx.application.Application;
import javafx.stage.Stage;
import javafx.scene.layout.Pane;
import javafx.scene.Scene;

public class Eks02 extends Application {

    @Override
    public void start(Stage vindu) throws Exception {

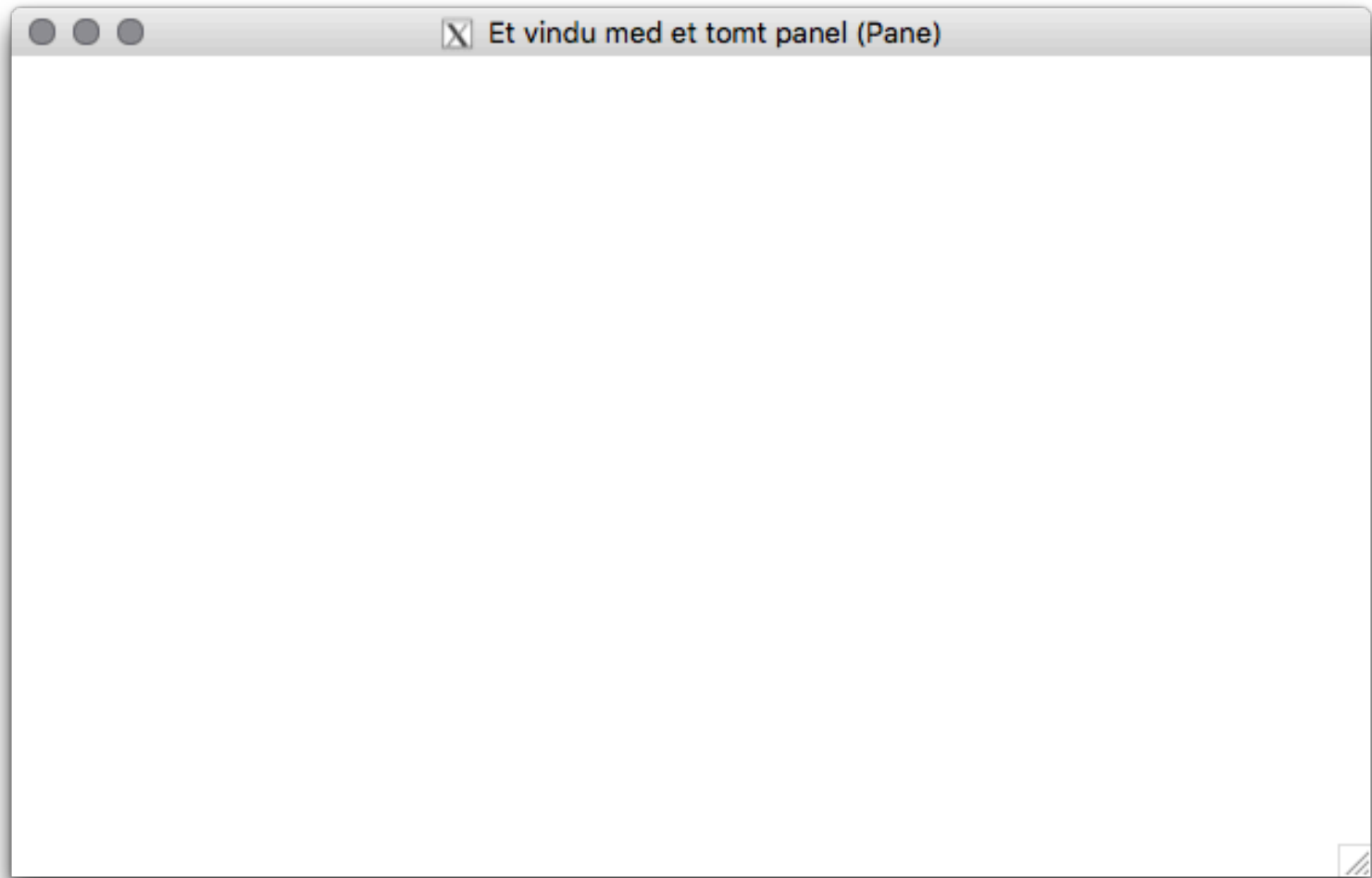
        Pane lerret = new Pane();
        Scene scene = new Scene(lerret, 900, 700);
        vindu.setScene(scene);
        vindu.show();
        vindu.setTitle("Et vindu med et tomt panel (Pane)");
    }

    public static void main(String[] a) {
        launch(a);
    }
}
```



1. Lag en klasse som er subklasse til `javafx.application.Application`
2. Redefiner (override) metoden `public void start(Stage vindu)` i denne subklassen
3. Lag et objekt av klassen `javafx.scene.layout.Pane`
4. Lag et objekt av klassen `javafx.scene.Scene` med `Pane`-objektet som parameter til konstr.
5. Sett `Scene`-objektet inn i vindusobjektet (type `Stage`) med kallet `vindu.setScene(scene)`;
6. Gjør vinduet synlig med kallet `vindu.show()`;

```
public class Eks02 extends Application {  
  
    @Override  
    public void start(Stage vindu) throws Exception {  
  
        Pane lerret = new Pane();  
        Scene scene = new Scene(lerret, 900, 700);  
        vindu.setScene(scene);  
        vindu.show();  
        vindu.setTitle("Et tomt 900 x 700-vindu");  
    }  
}
```

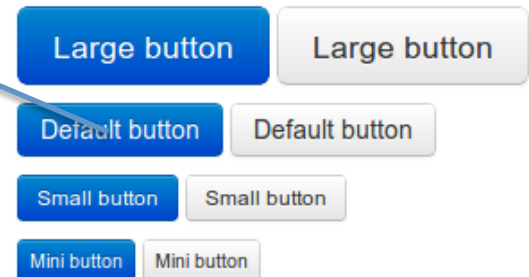


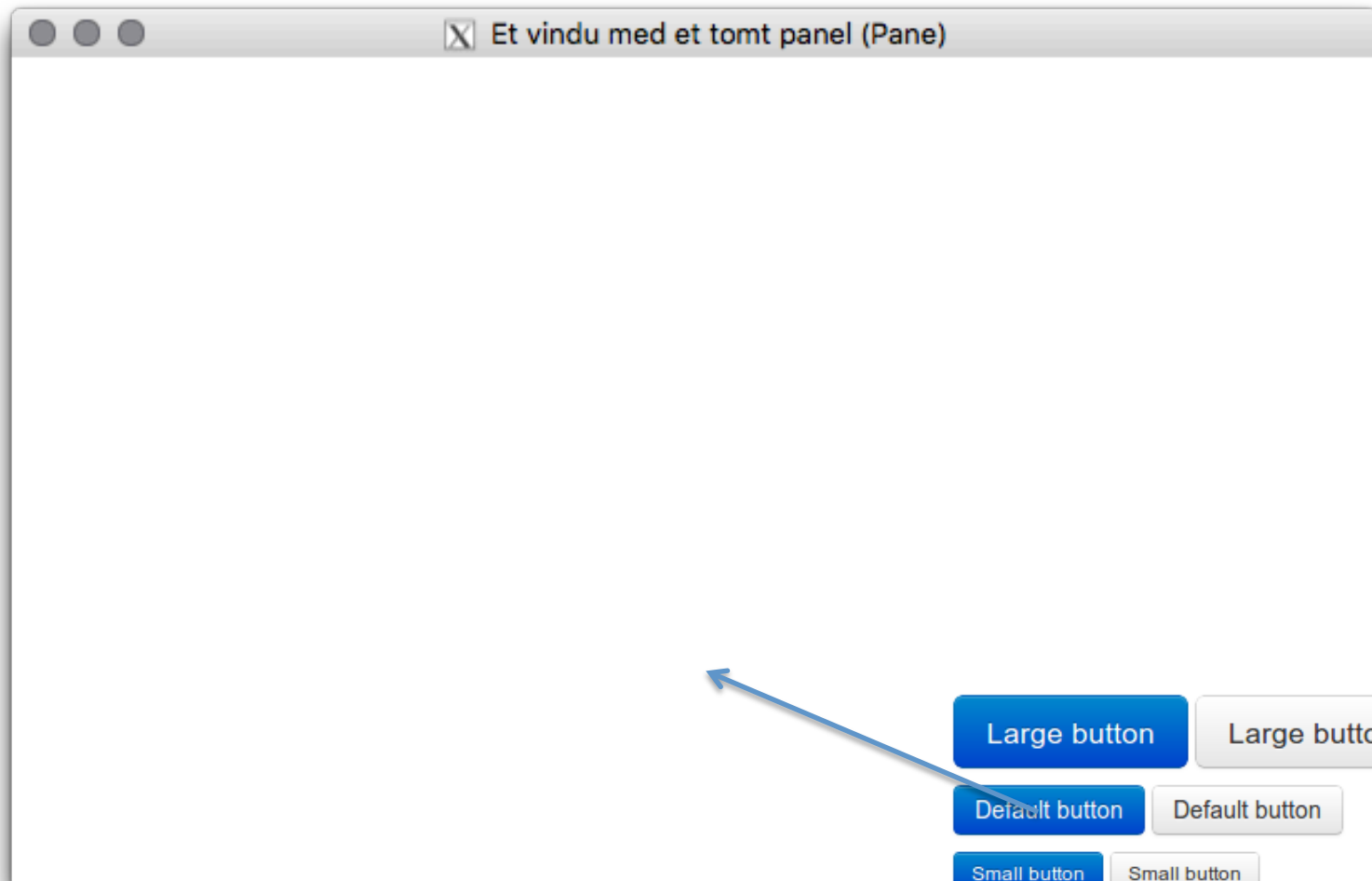




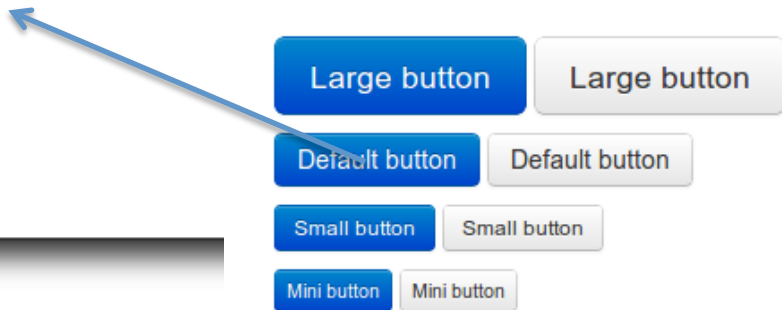
En tekst

		8	4		9			5
9		2		5	6			
4		5	7		2			6
	8	3		4	5		2	
			1		8			
	5		9	3		8	4	
3			5		1	7		9
			8	9		3		2
8			2		3	6		





Et vindu med et tomt panel (Pane)



Large button

Large button

Default button

Default button

Small button

Small button

Mini button

Mini button

```

public class BareKnapp extends Application {
    @Override
    public void start(Stage vindu) throws Exception {

        Pane terret = new Pane();
        Scene scene = new Scene(terret, 400, 300);

        vindu.setScene(scene);
        vindu.show();
        vindu.setTitle("Hovedscene");

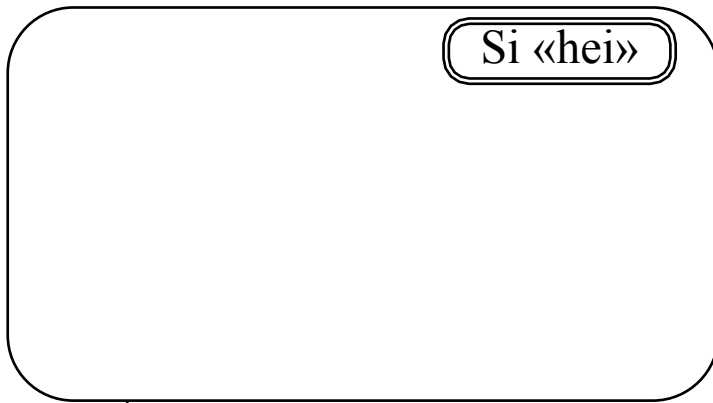
        Button knapp = new Button("Si «hei»");
        terret.getChildren().add(knapp);

        class MinLytterKlasse implements EventHandler<ActionEvent> {
            @Override
            public void handle(ActionEvent e) {
                System.out.println("Noen trykka på meg med musknappen!");
            }
        }

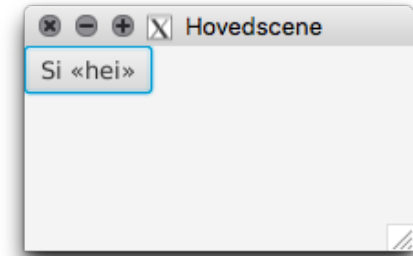
        knapp.setOnAction( new MinLytterKlasse() );
        // setOnAction skal ha en parameter av type EventHandler<ActionEvent>
    }

    public static void main(String[] a) {
        launch(a);
    }
}


```



*Dette objektet  
synes på skjermen  
(som en knapp)*

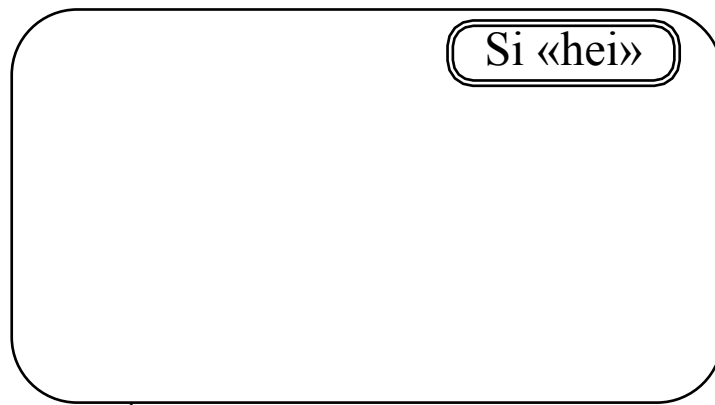


public void start (Stage hovedscene)

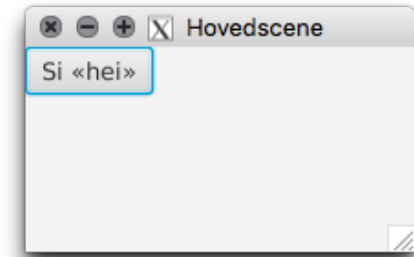
Button  
  
knapp

```
Button knapp = new Button("Si «hei»");
```

*Metoden start(Stage ..)  
i applikasjonsklassen*



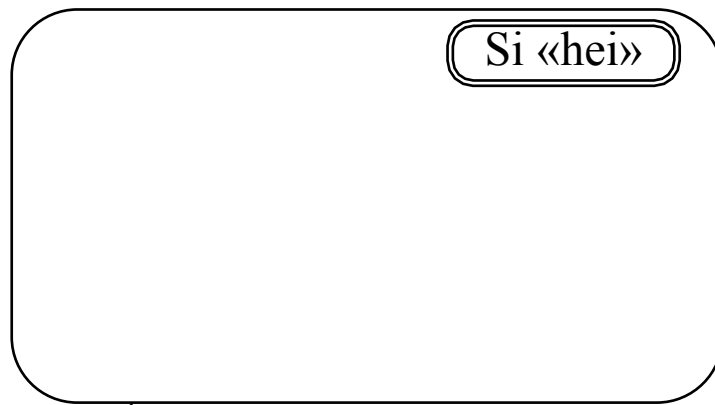
*Dette objektet  
synes på skjermen  
(som en knapp)*



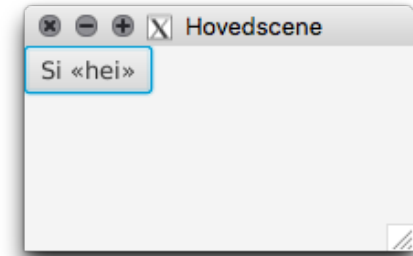
Button  
knapp

```
Button knapp = new Button("Si «hei»");
```

```
System.out.println("Skrives når hendelsen inntreffer..");
```



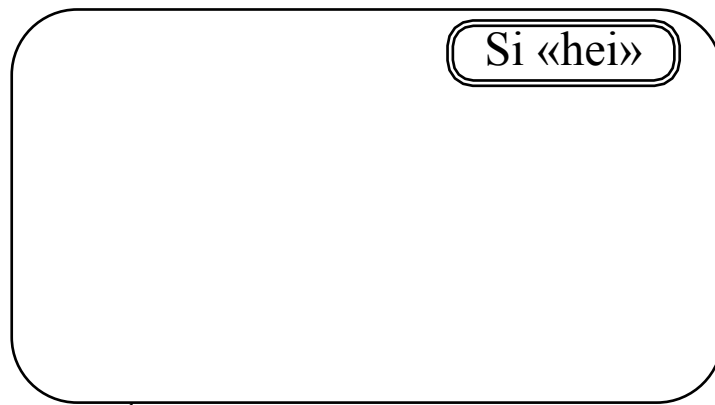
*Dette objektet  
synes på skjermen  
(som en knapp)*



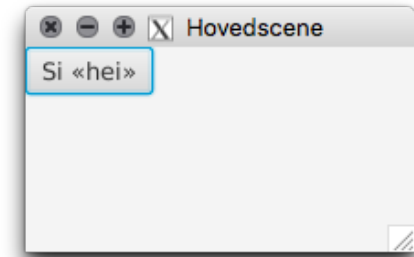
Button  
knapp

```
Button knapp = new Button("Si «hei»");
```

```
public void handle(ActionEvent ae) {  
    System.out.println("Skrives når hendelsen inntreffer..");  
}
```



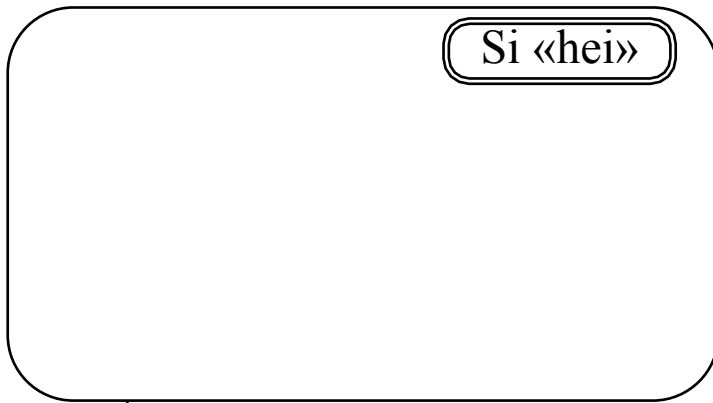
*Dette objektet  
synes på skjermen  
(som en knapp)*



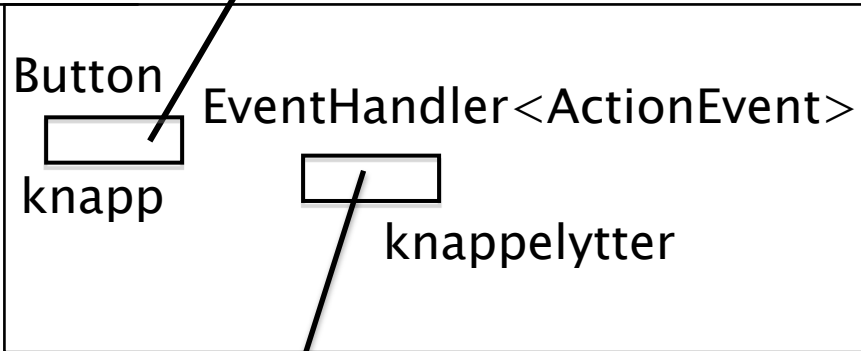
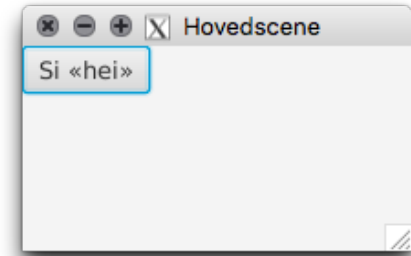
Button  
knapp

```
Button knapp = new Button("Si «hei»");
```

```
public class MinLytterKlasse implements EventHandler<ActionEvent> {  
  
    @Override  
    public void handle(ActionEvent ae) {  
        System.out.println("Skrives når hendelsen inntreffer..");  
    }  
}
```

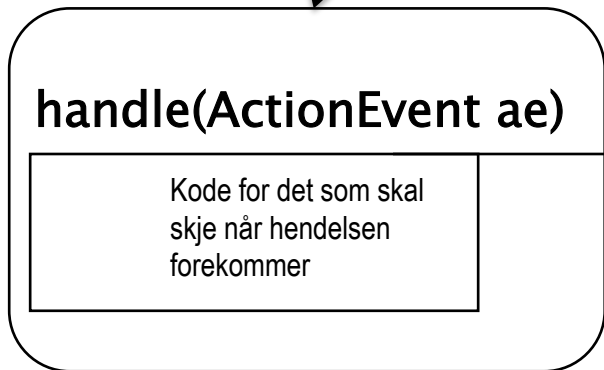


*Dette objektet synes på skjermen (som en knapp)*



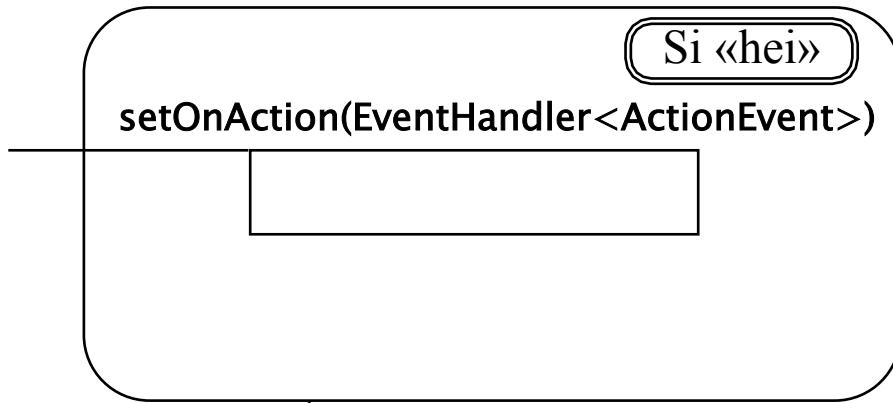
```
Button knapp = new Button("Si «hei»");
EventHandler<ActionEvent> knappelytter = new MinLyttterKlasse( ) ;
```

*Et objekt av en klasse som implementerer EventHandler<ActionEvent>*

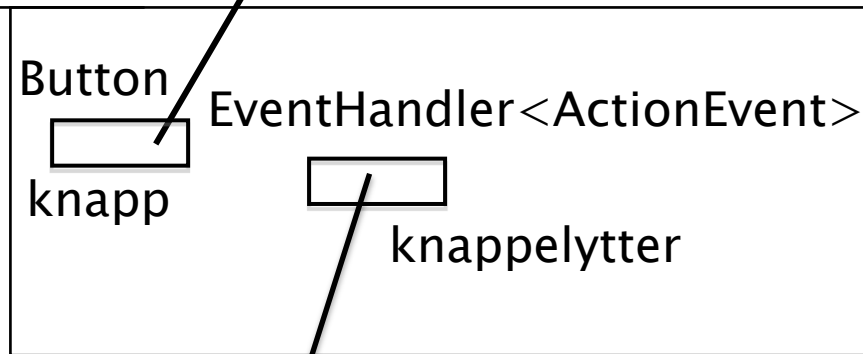
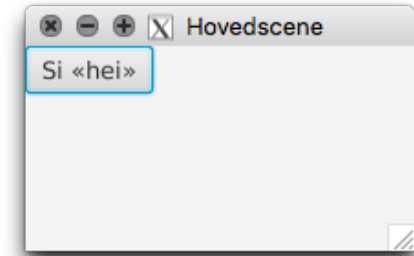


```
public class MinLyttterKlasse implements EventHandler<ActionEvent> {
    @Override
    public void handle(ActionEvent ae) {
        System.out.println("Skrives når hendelsen inntreffer..");
    }
}
```



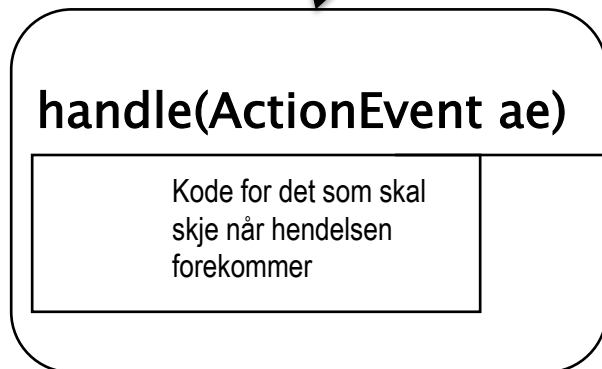


*Dette objektet synes på skjermen (som en knapp)*

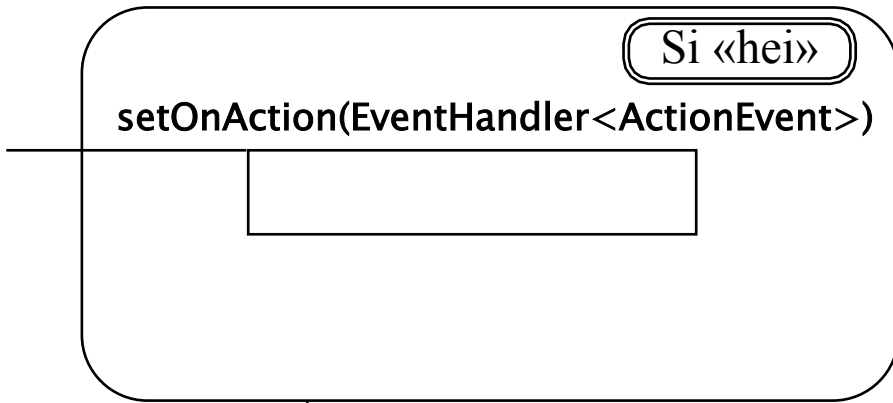


```
Button knapp = new Button("Si «hei»");
EventHandler<ActionEvent> knappelytter = new MinLyttterKlasse( ) ;
```

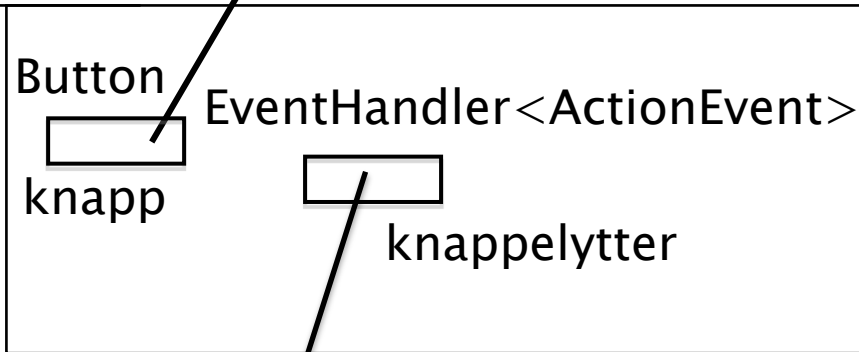
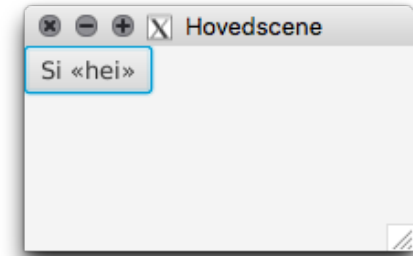
*Et objekt av en klasse som implementerer EventHandler<ActionEvent>*



```
public class MinLyttterKlasse implements EventHandler<ActionEvent> {
    @Override
    public void handle(ActionEvent ae) {
        System.out.println("Skrives når hendelsen inntreffer..");
    }
}
```



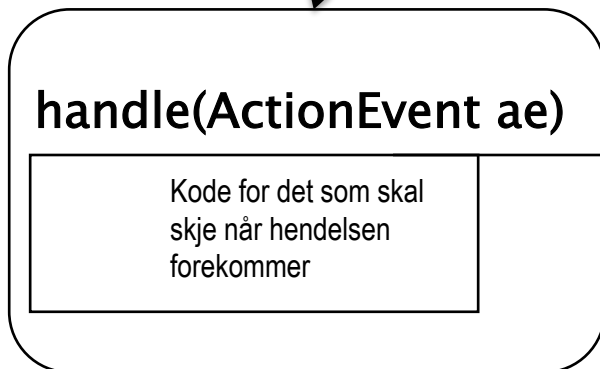
*Dette objektet synes på skjermen (som en knapp)*



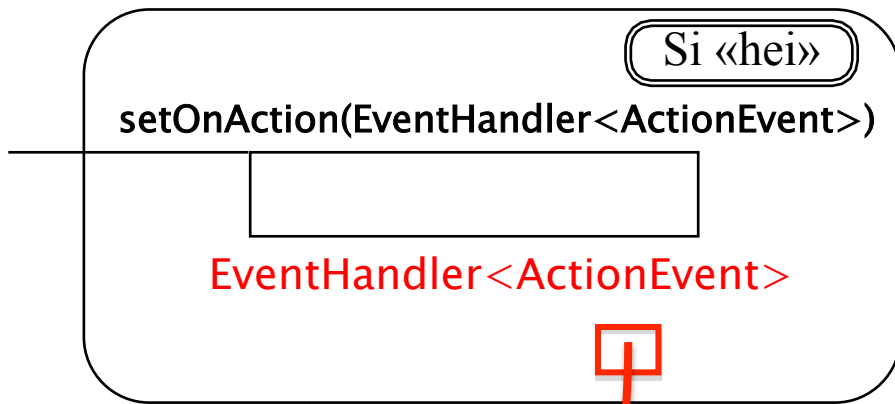
```
Button knapp = new Button("Si «hei»");
EventHandler<ActionEvent> knappelytter = new MinLytterKlasse( );
```

```
knapp.setOnAction(knappelytter);
```

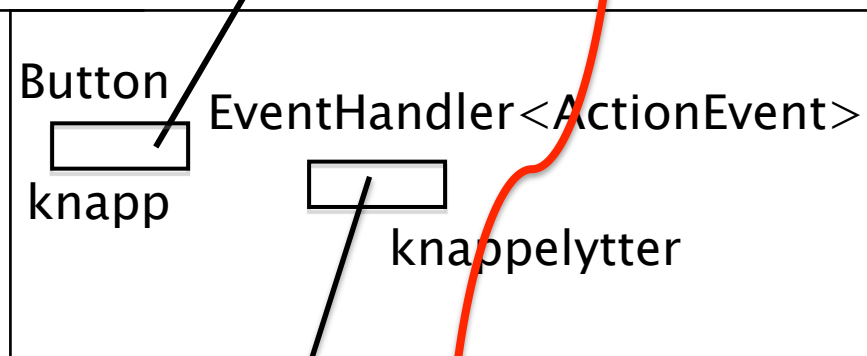
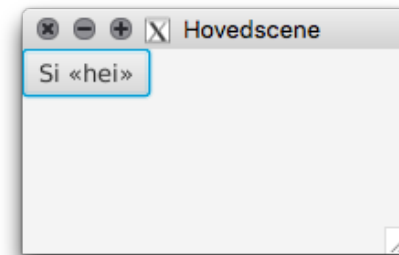
*Et objekt av en klasse som implementerer EventHandler<ActionEvent>*



```
public class MinLytterKlasse implements EventHandler<ActionEvent> {
    @Override
    public void handle(ActionEvent ae) {
        System.out.println("Skrives når hendelsen inntreffer..");
    }
}
```



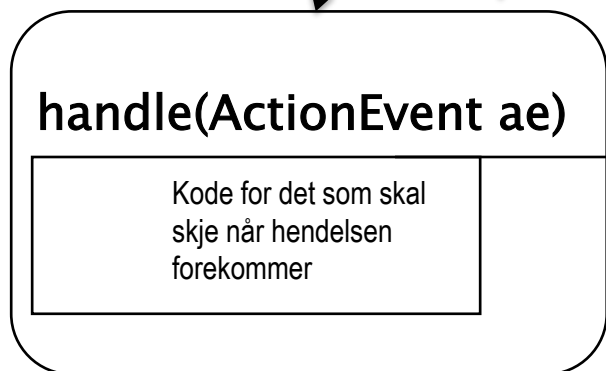
*Dette objektet synes på skjermen (som en knapp)*



```
Button knapp = new Button("Si «hei»");
EventHandler<ActionEvent> knappelytter = new MinLytterKlasse( ) ;
```

```
knapp.setOnAction(knappelytter);
```

*Et objekt av en klasse som implementerer EventHandler<ActionEvent>*



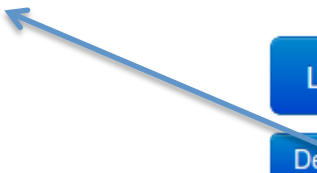
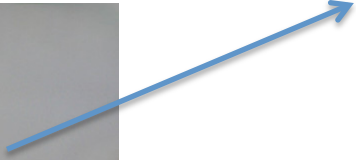
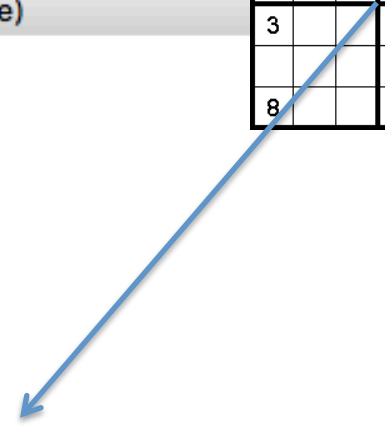
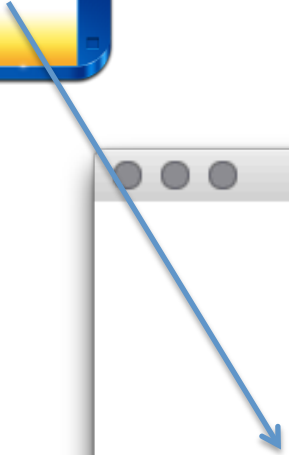
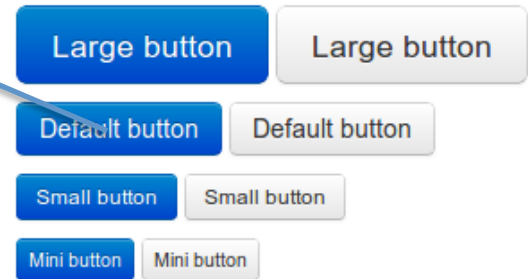
```
public class MinLytterKlasse implements EventHandler<ActionEvent> {
    @Override
    public void handle(ActionEvent ae) {
        System.out.println("Skrives når hendelsen inntreffer..");
    }
}
```



En tekst

		8	4		9			5
9		2		5	6			
4		5	7		2			6
	8	3		4	5		2	
			1		8			
	5		9	3		8	4	
3			5		1	7		9
			8	9		3		2
8			2		3	6		

Et vindu med et tomt panel (Pane)



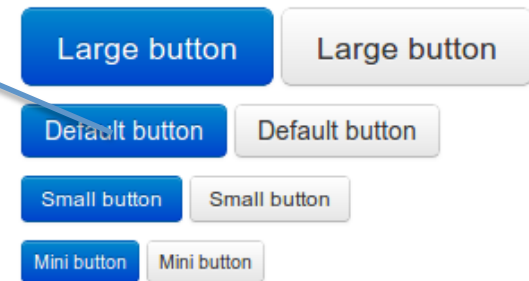


En tekst

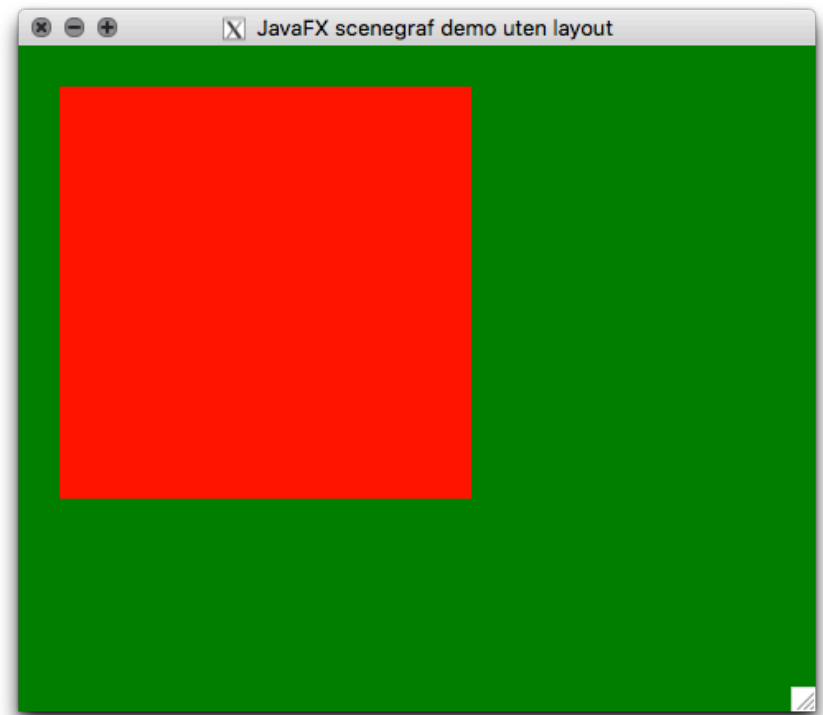
		8	4		9			5
9		2		5	6			
4		5	7		2			6
	8	3		4	5		2	
			1		8			
	5		9	3		8	4	
3			5		1	7		9
			8	9		3		2
8			2		3	6		

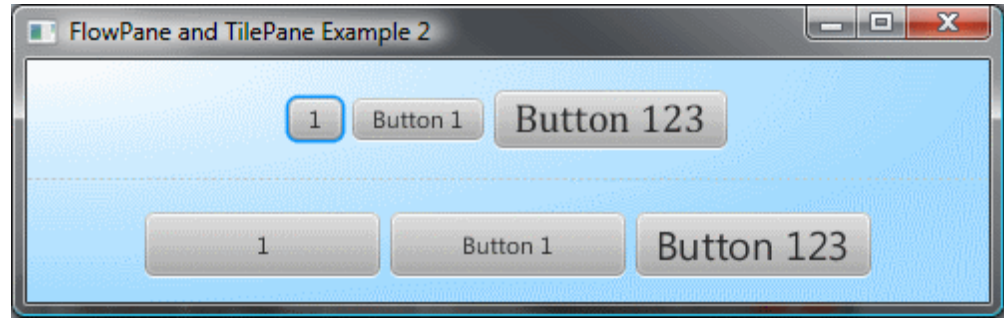
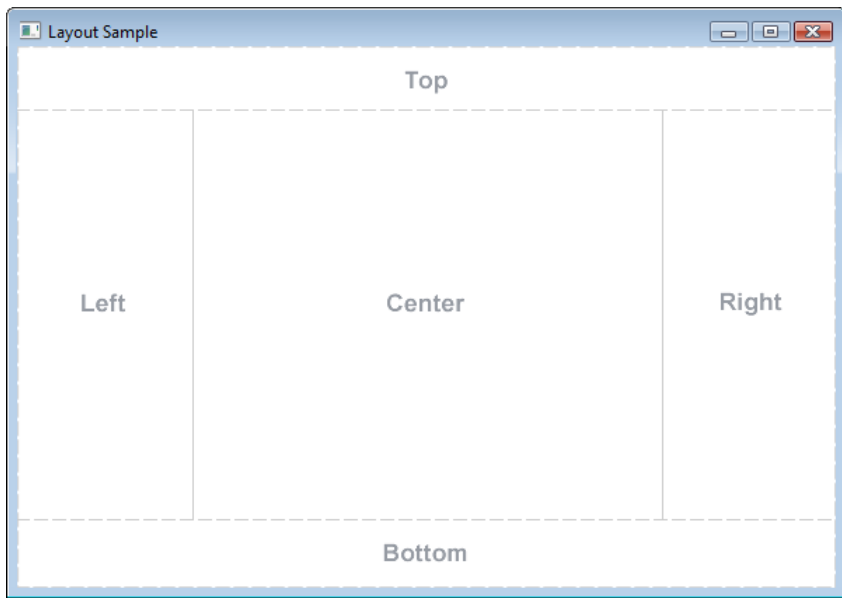



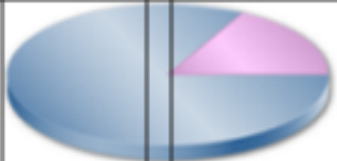
hvor ?

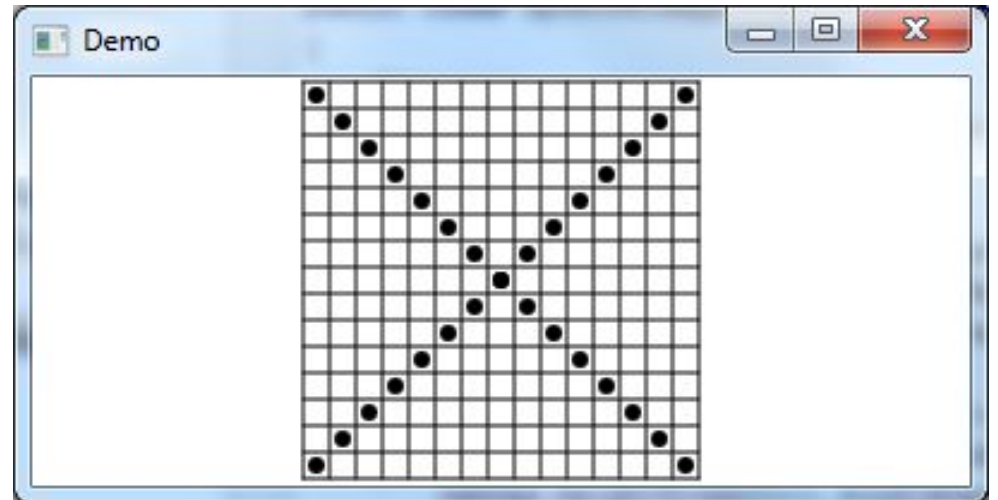


```
public class UtenLayout extends Application {  
  
    @Override  
    public void start(Stage stage) {  
        Group root = new Group();  
        Scene scene = new Scene(root, 500, 500, Color.GREEN);  
  
        Rectangle r = new Rectangle(25,25,250,250);  
        r.setFill(Color.RED);  
        root.getChildren().add(r);  
  
        stage.setTitle("JavaFX scenegraf demo uten layout");  
        stage.setScene(scene);  
        stage.show();  
    }  
  
    public static void main(String[] args) {  
        launch(args);  
    }  
}
```





	<b>Sales: Current Year</b>		
	Goods and Services		
Goods 80%			Services 20%



HBox

Vbox

StackPane

GridPane

TilePane

BorderPane

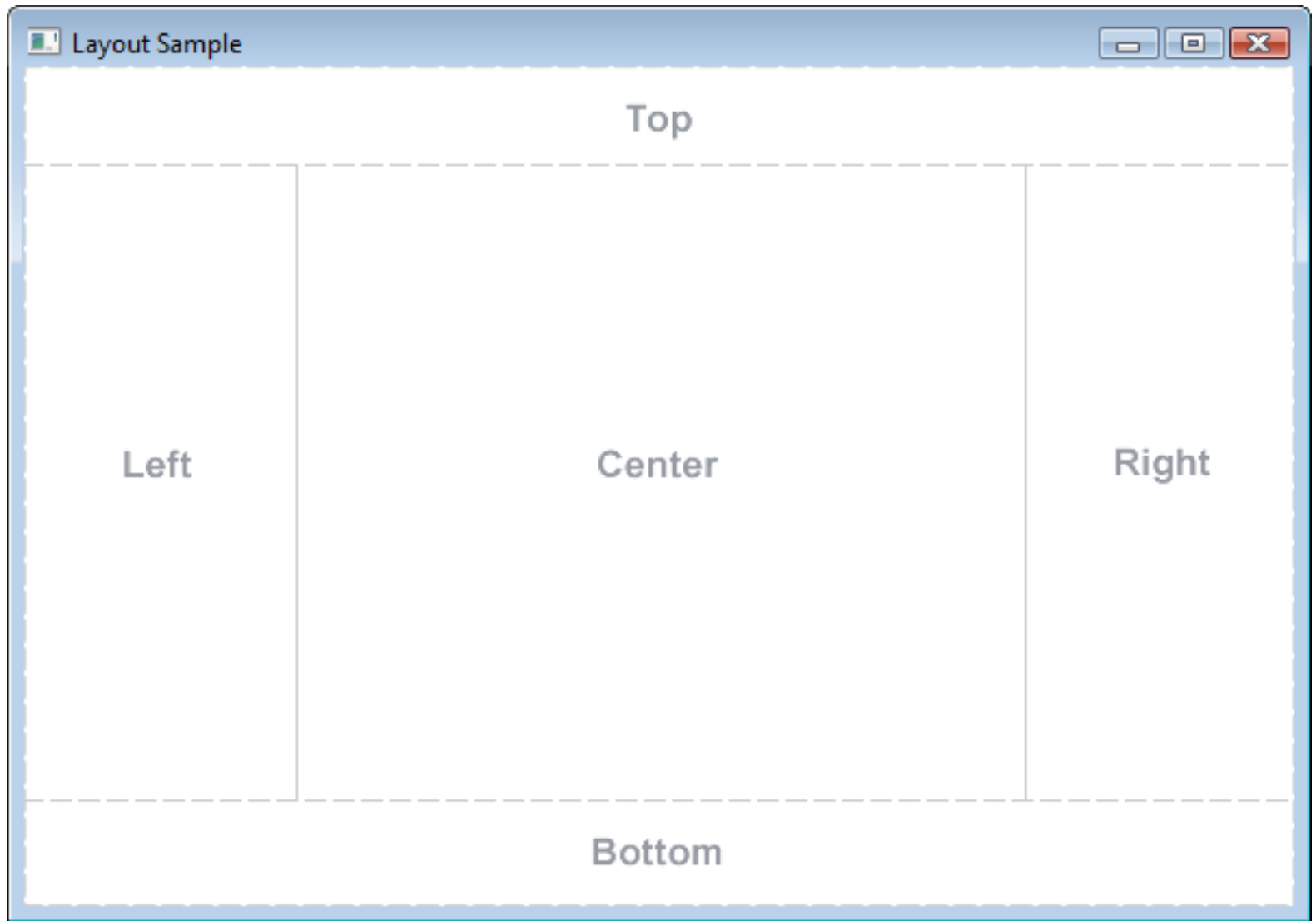
FlowPane

Pane

....







```

public class BorderP extends Application {

    @Override
    public void start(Stage vindu) throws Exception {

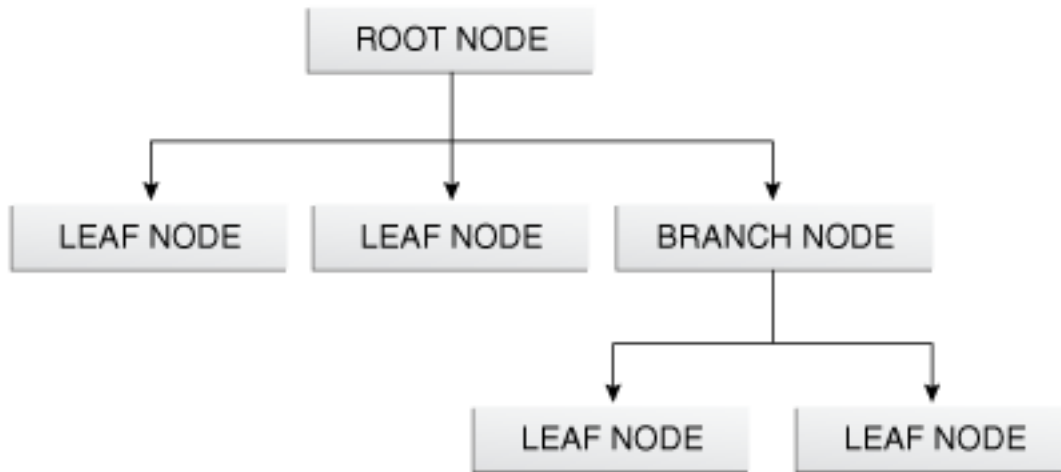
        BorderPane rot = new BorderPane(); // Rota til scenegrafen
        // Legg inn et tekstelement i hvert område av skjermen
        rot.setTop(new Text("top"));
        rot.setBottom(new Text("bottom"));
        rot.setLeft(new Text("left"));
        rot.setRight(new Text("right"));
        rot.setCenter(new Text("center"));

        Scene scene = new Scene(rot, 400, 700);
        vindu.setScene(scene);
        vindu.show();
        vindu.setTitle("Et BorderPane-vindu med tekstelementer");
    }

    public static void main(String[] a) {
        launch(a);
    }
}

```





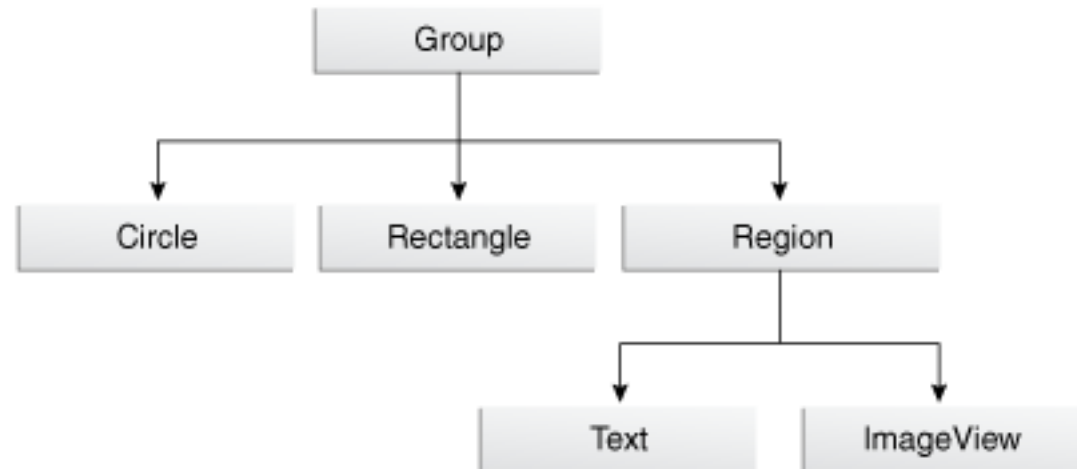
3 nodetyper:

rotnoder

grennoder eller beholdernoder

løvnoder eller endenoder

Eksempler på konkrete klasser av de forskjellige nodetyperne, et konkret scenetre eller scenegraf (scene graph)



*Bildene er hentet fra  
Oracles nettsider*

javafx.scene

## **Class Node**

java.lang.Object  
    javafx.scene.Node

### **All Implemented Interfaces:**

Styleable, EventTarget

### **Direct Known Subclasses:**

Camera, Canvas, ImageView, LightBase, MediaView, Parent, Shape, Shape3D, SubScene, SwingNode

---

```
@IDProperty(value="id")
public abstract class Node
extends Object
implements EventTarget, Styleable
```

---

javafx.scene

## **Class Parent**

java.lang.Object

    javafx.scene.Node

        javafx.scene.Parent

### **All Implemented Interfaces:**

Styleable, EventTarget

### **Direct Known Subclasses:**

Group, **Region**, WebView

---

```
public abstract class Parent
extends Node
```

javafx.scene.layout

## **Class Region**

java.lang.Object

javafx.scene.Node

javafx.scene.Parent

javafx.scene.layout.Region

### **All Implemented Interfaces:**

Styleable, EventTarget

### **Direct Known Subclasses:**

Axis, Chart, Control, Pane

---

```
public class Region
```

```
extends Parent
```

javafx.scene.layout

## **Class Pane**

java.lang.Object

javafx.scene.Node

javafx.scene.Parent

javafx.scene.layout.Region

javafx.scene.layout.Pane

### **All Implemented Interfaces:**

Styleable, EventTarget

### **Direct Known Subclasses:**

AnchorPane, BorderPane, DialogPane, FlowPane,  
GridPane, HBox, PopupControl.CSSBridge,  
StackPane, TextFlow, TilePane, VBox

---

```
@DefaultProperty(value="children")  
public class Pane  
extends Region
```

javafx.scene.layout

## **Class GridPane**

java.lang.Object

    javafx.scene.Node

        javafx.scene.Parent

            javafx.scene.layout.Region

                javafx.scene.layout.Pane

                    javafx.scene.layout.GridPane

### **All Implemented Interfaces:**

Styleable, EventTarget

---

```
public class GridPane
```

```
extends Pane
```

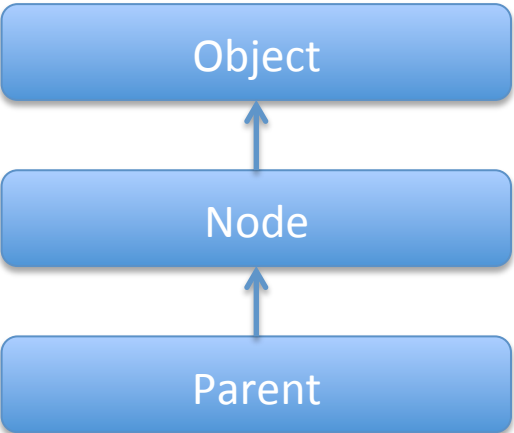


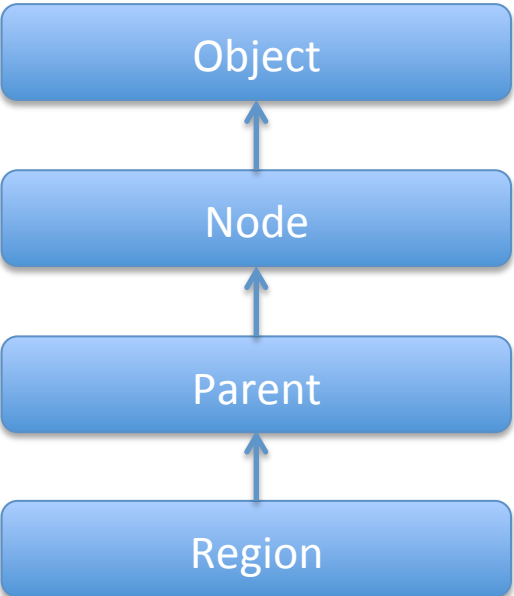
Object

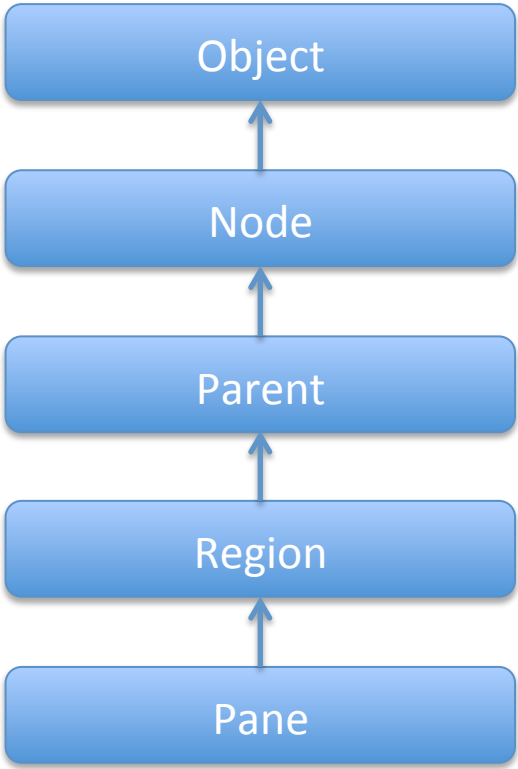
Object

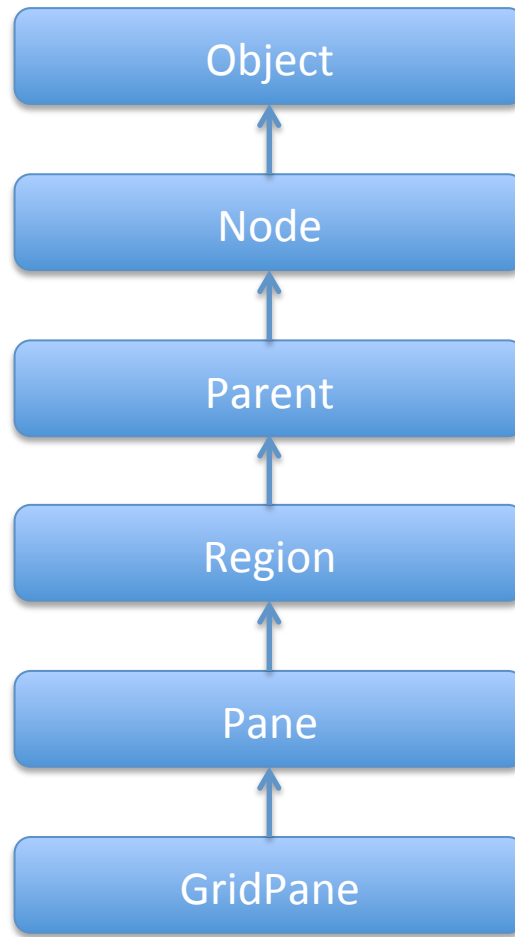


Node

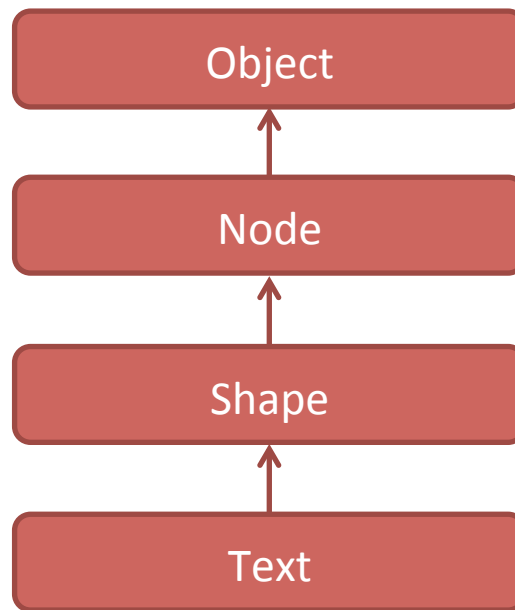




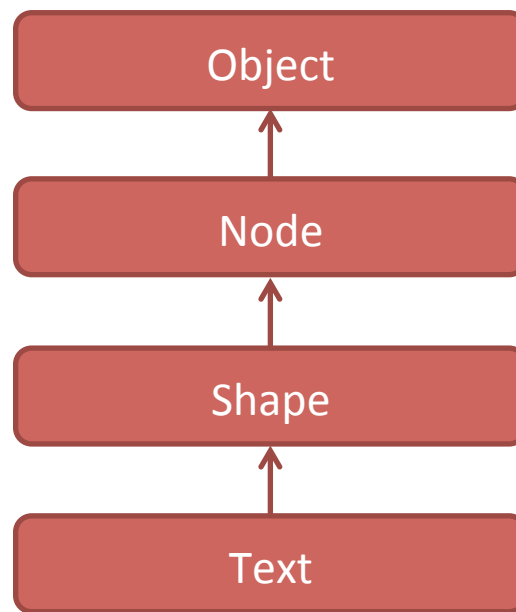
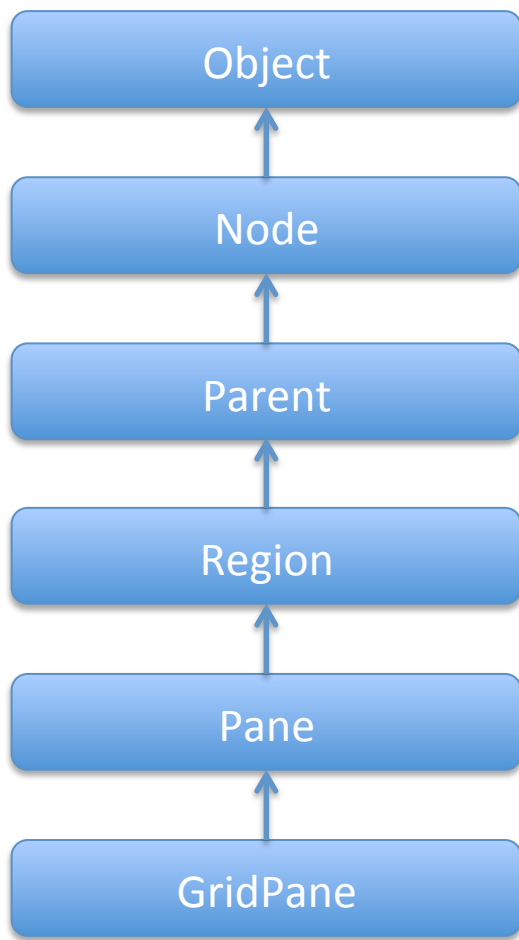




Et objekt av GridPane er en grennode (Branch node) som kan inneholde andre noder. Grennoder og løvnoder.

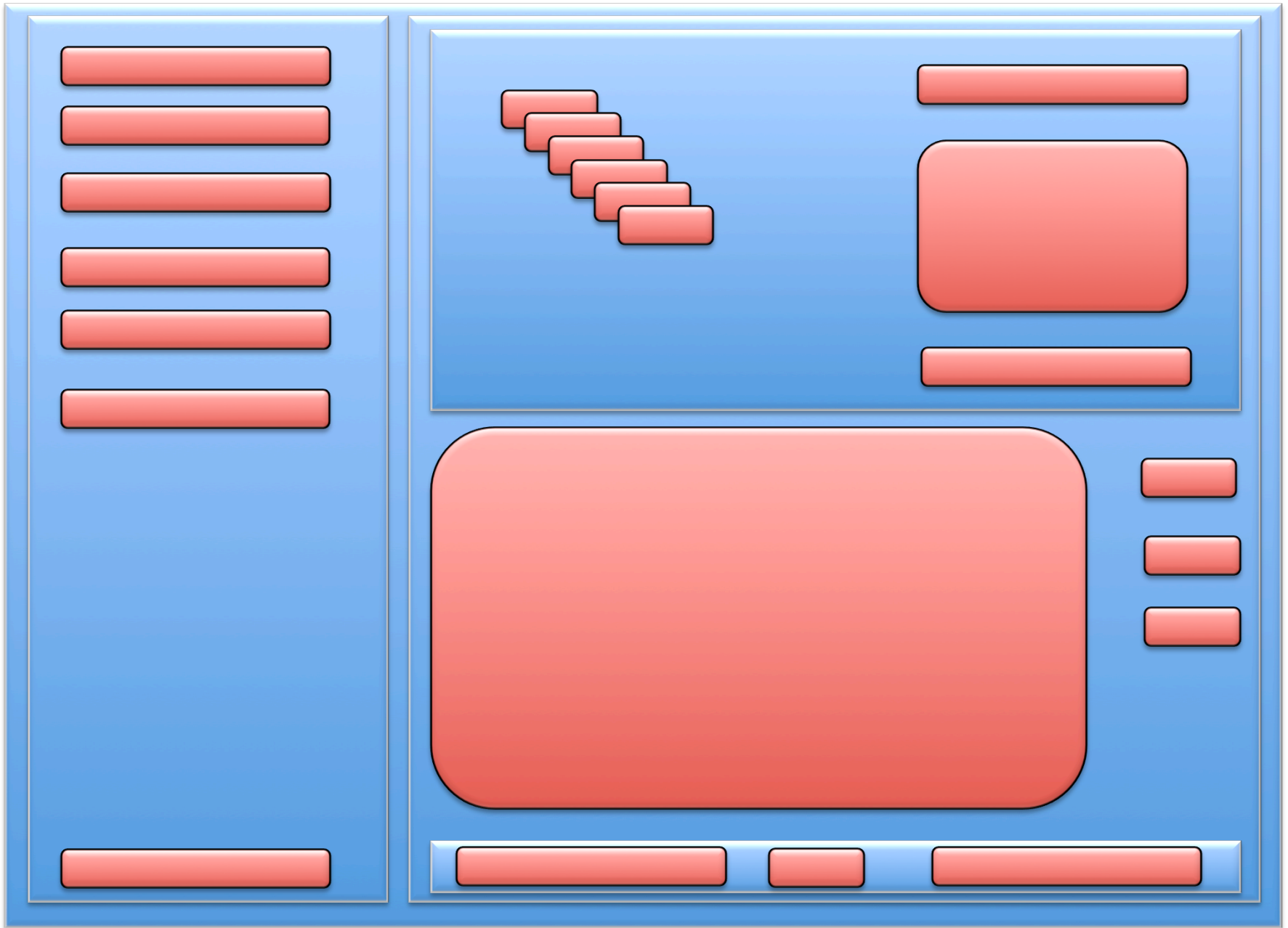


Et objekt av Text er en løvnode (Leaf node) som **ikke** kan inneholde andre noder.



- Grennoder (Branch nodes) er objekter av klasser som er subclasser til Parent
- Løvnoder (Leaf nodes) har ikke Parent som superklasse
- Rotnoder må være en grennode, dvs være (av en) subclasse til Parent





# Konstruktørene til Scene

## Constructor Summary

### Constructors

#### Constructor and Description

**Scene**(**Parent** root)

Creates a Scene for a specific root Node.

**Scene**(**Parent** root, double width, double height)

Creates a Scene for a specific root Node with a specific size.

**Scene**(**Parent** root, double width, double height, boolean depthBuffer)

Constructs a scene consisting of a root, with a dimension of width and height, and specifies whether a depth buffer is created for this scene.

**Scene**(**Parent** root, double width, double height, boolean depthBuffer, **SceneAntialiasing** antiAliasing)

Constructs a scene consisting of a root, with a dimension of width and height, specifies whether a depth buffer is created for this scene and specifies whether scene anti-aliasing is requested.

**Scene**(**Parent** root, double width, double height, **Paint** fill)

Creates a Scene for a specific root Node with a specific size and fill.

**Scene**(**Parent** root, **Paint** fill)

Creates a Scene for a specific root Node with a fill.

## Noen av metodene til BorderLayout, som vi kommer til å bruke i eksemplene

`void`                    **setBottom(Node value)**  
Sets the value of the property bottom.

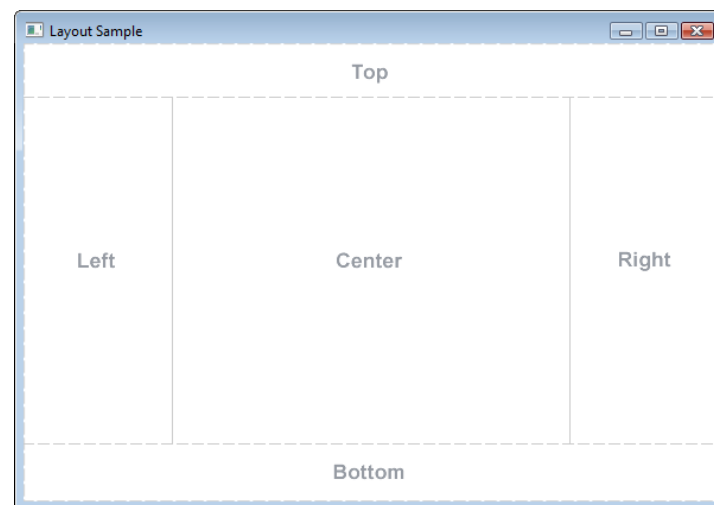
`void`                    **setCenter(Node value)**  
Sets the value of the property center.

`void`                    **setLeft(Node value)**  
Sets the value of the property left.

`static void`            **setMargin(Node child, Insets value)**  
Sets the margin for the child when contained by a border pane.

`void`                    **setRight(Node value)**  
Sets the value of the property right.

`void`                    **setTop(Node value)**  
Sets the value of the property top.



```

public class Eks04 extends Application {

    @Override
    public void start(Stage vindu) throws Exception {

        BorderPane border = new BorderPane();
        HBox hbox = nyHBox();

        border.setTop(hbox);

        border.setLeft(vBoxMedTekst());

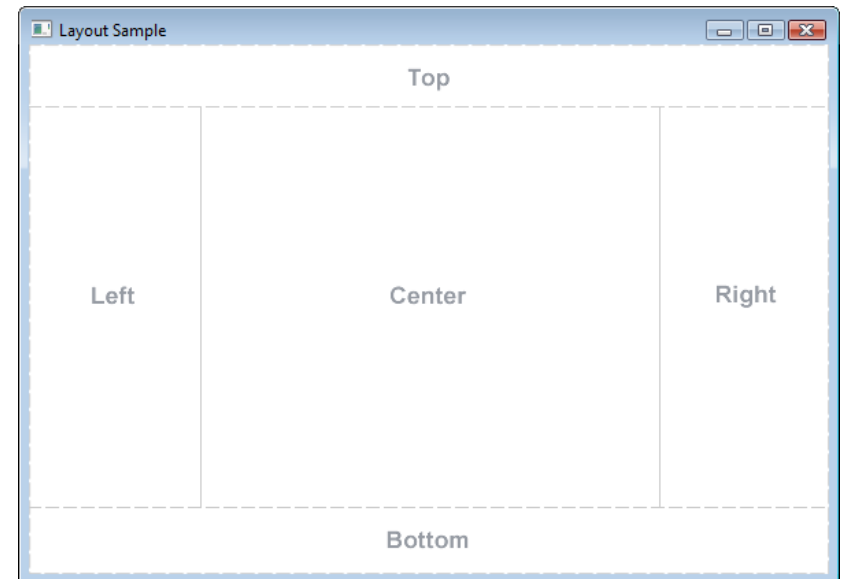
        border.setCenter(minGridPane());

        Scene scene = new Scene(border, 400, 700);
        vindu.setScene(scene);
        vindu.show();
        vindu.setTitle("Eksempel på bruk av Border- og GridPane, HBox og VBox");
        vindu.setScene(scene);
    }

    public static void main(String[] a) {
        launch(a);
    }

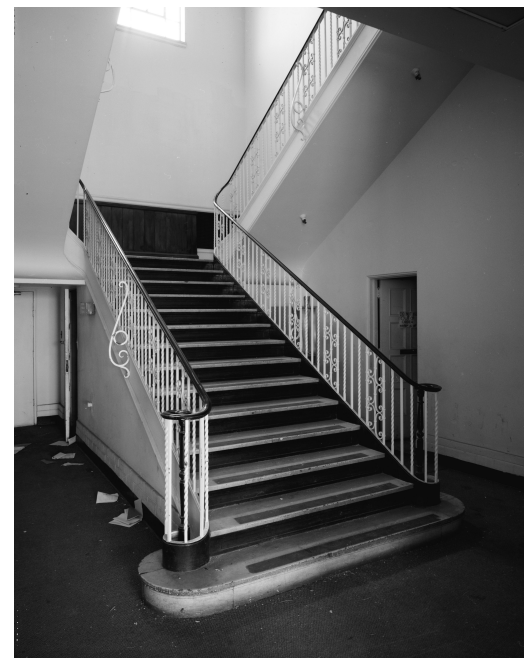
    private HBox nyHBox() { ... }
    private VBox vboxMedTekst() { ... }
    private GridPane minGridPane() { ... }
    . . .
}

```



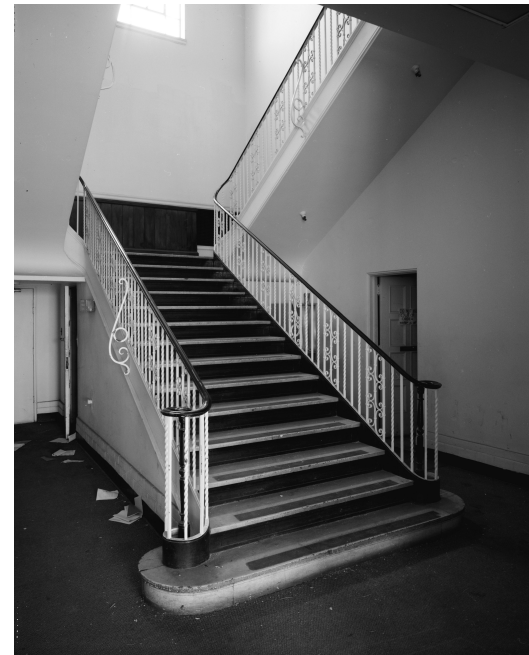
```
public void start(Stage vindu) throws Exception {  
    BorderPane border = new BorderPane();  
    HBox hbox = nyHBox();  
    border.setTop(nyHBox);  
    border.setLeft(vBoxMedTekst());  
    Scene scene = new Scene(border, 400, 700);  
    vindu.setScene(scene);  
    vindu.show();  
}
```

Minner om trinnene:



Minner om trinnene:

1. Oppretter en rotnode/grennode (subklasse av Parent)



Minner om trinnene:

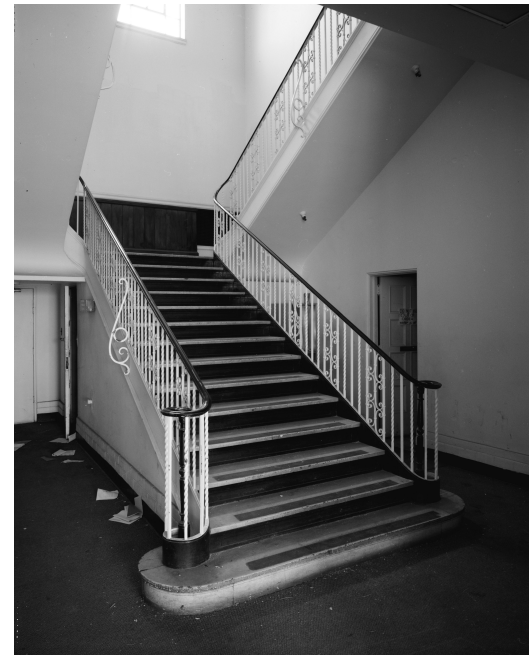
1. Oppretter en rotnode/grennode (subklasse av Parent)
2. Bygger opp et tre fra denne bestående av grennoder (beholdere) og løvnoder (enkeltelementer)





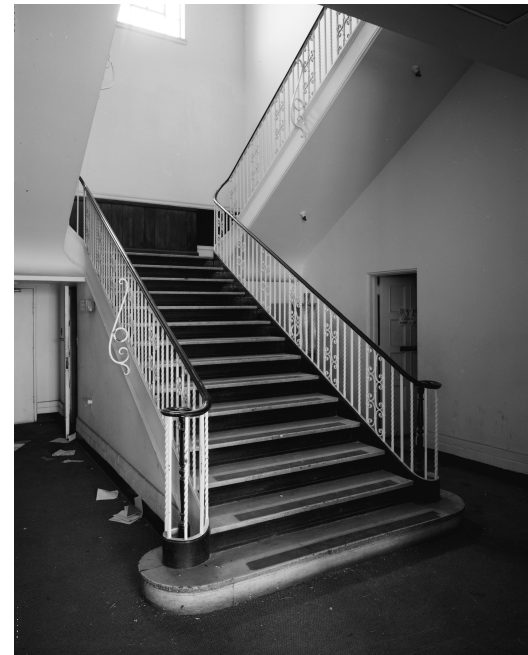
Minner om trinnene:

1. Oppretter en rotnode/grennode (subklasse av Parent)
2. Bygger opp et tre fra denne bestående av grennoder (beholdere) og løvnoder (enkeltelementer)
3. Legger scenegrafen/rotnoden til treet inn i et sceneobjekt med konstruktøren til sceneobjektet



Minner om trinnene:

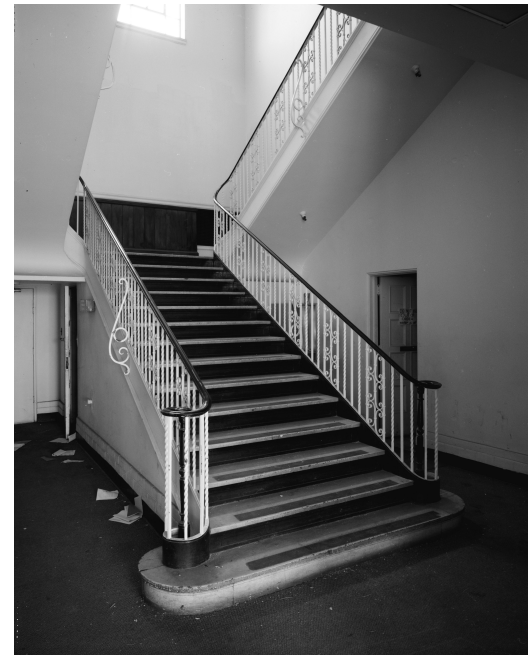
1. Oppretter en rotnode/grennode (subklasse av Parent)
2. Bygger opp et tre fra denne bestående av grennoder (beholdere) og løvnoder (enkeltelementer)
3. Legger scenegrafen/rotnoden til treet inn i et sceneobjekt med konstruktøren til sceneobjektet
4. Legger sceneobjektet inn i vindusramma med metoden `Stage.setScene()`



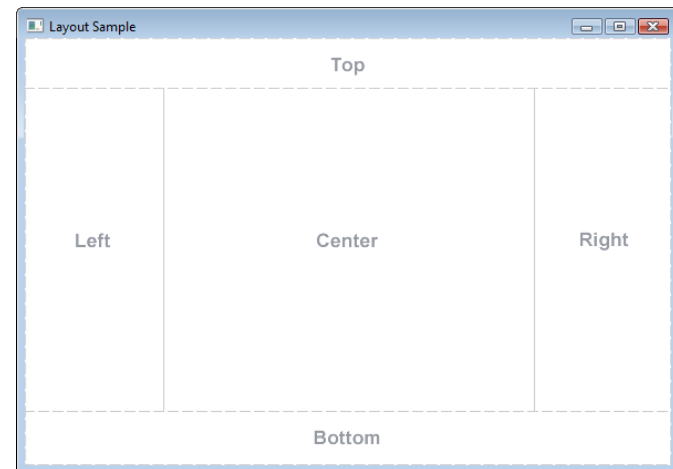
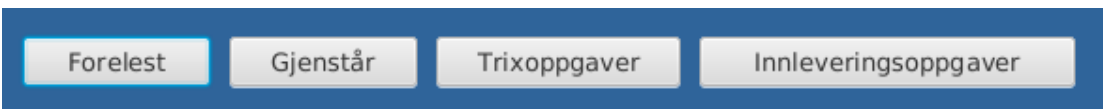
Minner om trinnene:

1. Oppretter en rotnode/grennode (subklasse av Parent)
2. Bygger opp et tre fra denne bestående av grennoder (beholdere) og løvnoder (enkeltelementer)
3. Legger scenegrafen/rotnoden til treet inn i et sceneobjekt med konstruktøren til sceneobjektet
4. Legger sceneobjektet inn i vindusramma med metoden Stage.setScene()

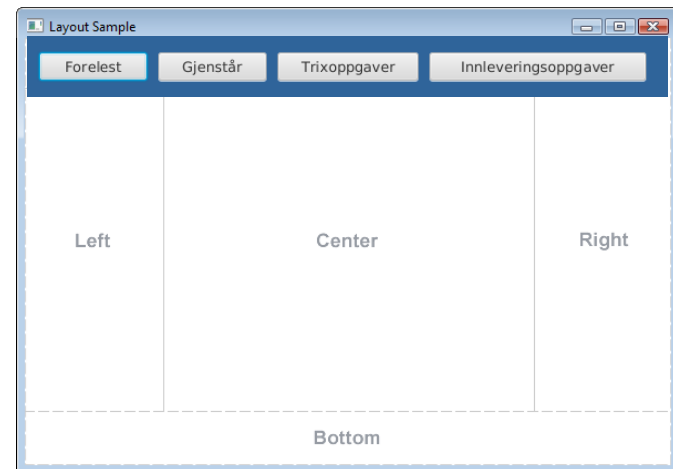
```
public void start(Stage vindu) throws Exception {  
  
    BorderPane border = new BorderPane();  
    border.setTop( nyHBox() );  
    border.setLeft( vboxMedTekst() );  
    Scene scene = new Scene(border, 400, 700);  
    vindu.setScene(scene);  
    vindu.show();  
}
```



```
private HBox nyHBox() {  
  
    HBox hbox = new HBox();  
    hbox.setPadding(new Insets(15, 12, 15, 12)); // t, r b, l  
    hbox.setSpacing(10); // Avstand mellom nodene  
    hbox.setStyle("-fx-background-color: #336699;");  
  
    Button knapp1 = new Button("Forelest");  
    knapp1.setPrefSize(100, 20);  
    Button knapp2 = new Button("Gjenstår");  
    knapp2.setPrefSize(100, 20);  
    Button knapp3 = new Button("Trixoppgaver");  
    knapp3.setPrefSize(130, 20);  
    Button knapp4 = new Button("Innleveringsoppgaver");  
    knapp4.setPrefSize(200, 20);  
  
    hbox.getChildren().addAll(knapp1, knapp2, knapp3, knapp4);  
  
    return hbox;  
}
```



```
private HBox nyHBox() {  
  
    HBox hbox = new HBox();  
    hbox.setPadding(new Insets(15, 12, 15, 12)); // t, r b, l  
    hbox.setSpacing(10); // Avstand mellom nodene  
    hbox.setStyle("-fx-background-color: #336699;");  
  
    Button knapp1 = new Button("Forelest");  
    knapp1.setPrefSize(100, 20);  
    Button knapp2 = new Button("Gjenstår");  
    knapp2.setPrefSize(100, 20);  
    Button knapp3 = new Button("Trixoppgaver");  
    knapp3.setPrefSize(130, 20);  
    Button knapp4 = new Button("Innleveringsoppgaver");  
    knapp4.setPrefSize(200, 20);  
  
    hbox.getChildren().addAll(knapp1, knapp2, knapp3, knapp4);  
  
    return hbox;  
}
```



```

public class Eks04 extends Application {

    @Override
    public void start(Stage vindu) throws Exception {

        BorderPane border = new BorderPane();
        HBox hbox = nyHBox();

        border.setTop(hbox);

        border.setLeft(vBoxMedTekst());

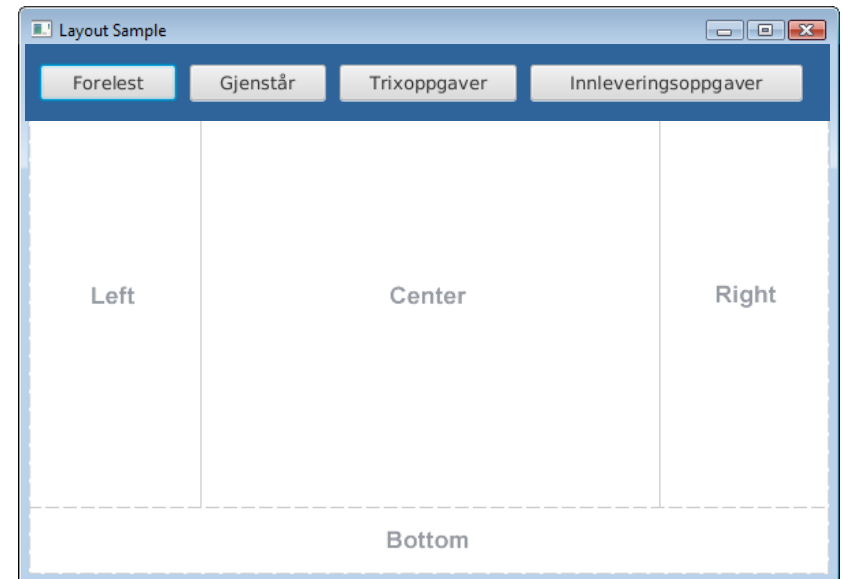
        border.setCenter(minGridPane());

        Scene scene = new Scene(border, 400, 700);
        vindu.setScene(scene);
        vindu.show();
        vindu.setTitle("Eksempel på bruk av Border- og GridPane, HBox og VBox");
        vindu.setScene(scene);
    }

    public static void main(String[] a) {
        launch(a);
    }

    private HBox nyHBox() { ... }
    private VBox vboxMedTekst() { ... }
    private GridPane minGridPane() { ... }
    . . .
}

```



```

private VBox vboxMedTekst() {

    VBox vbox = new VBox();
    vbox.setPadding(new Insets(10)); // Setter alle sider til 10
    vbox.setSpacing(8);             // Avstand mellom nodene (elementene)

    Text title = new Text("Hva er JavaFX ?");
    title.setFont(Font.font("Arial", FontWeight.BOLD, 14));
    vbox.getChildren().add(title);

    Text t = new Text("JavaFX is a software platform");
    vbox.getChildren().add(t);
    t = new Text("for creating and delivering desktop");
    vbox.getChildren().add(t);
    t = new Text("applications, as well as rich internet");
    vbox.getChildren().add(t);
    t = new Text("applications (RIAs) that can run");
    vbox.getChildren().add(t);
    t = new Text("across a wide variety of devices.");
    vbox.getChildren().add(t);
    t = new Text("JavaFX is intended to replace");
    vbox.getChildren().add(t);
    t = new Text("Swing as the standard GUI library");
    vbox.getChildren().add(t);
    t = new Text("for Java SE, but both will be included");
    vbox.getChildren().add(t);
    t = new Text("for the foreseeable future.");
    vbox.getChildren().add(t);

    return vbox;
}

```

### Hva er JavaFX ?

JavaFX is a software platform for creating and delivering desktop applications, as well as rich internet applications (RIAs) that can run across a wide variety of devices. JavaFX is intended to replace Swing as the standard GUI library for Java SE, but both will be included for the foreseeable future.

```

private VBox vboxMedTekst() {

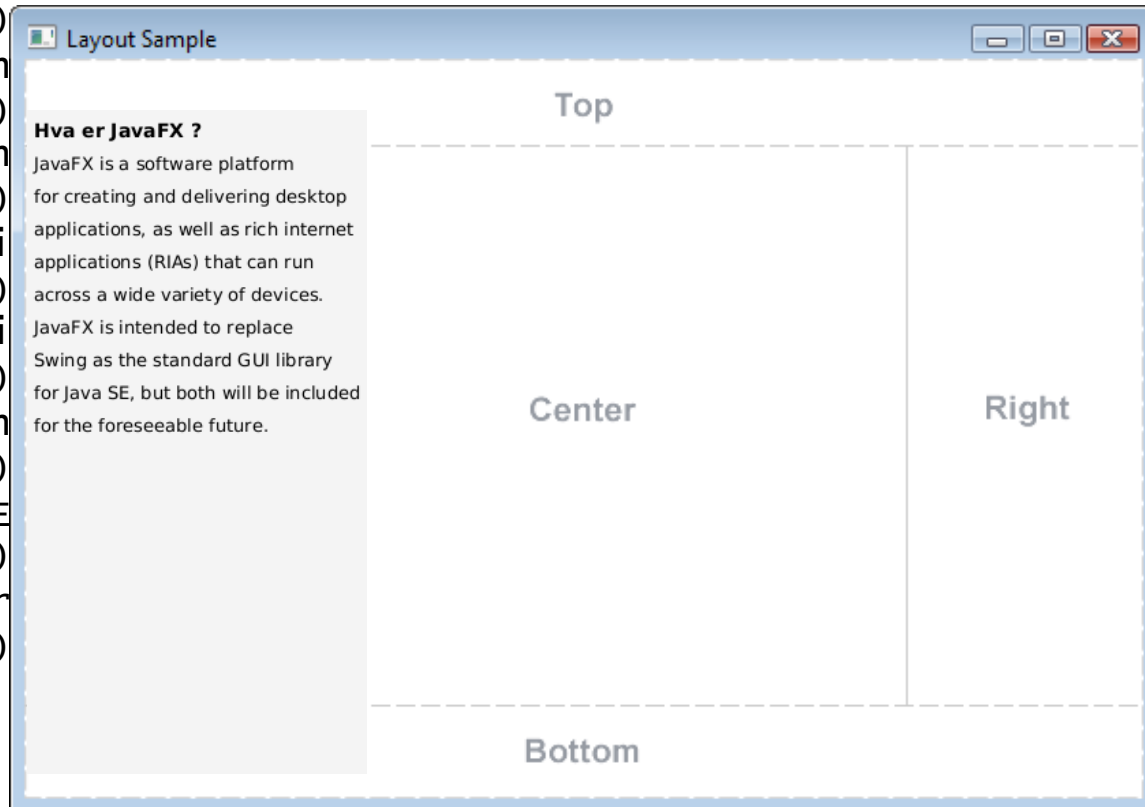
    VBox vbox = new VBox();
    vbox.setPadding(new Insets(10)); // Setter alle sider til 10
    vbox.setSpacing(8);             // Avstand mellom nodene (elementene)

    Text title = new Text("INF1010 temaer");
    title.setFont(Font.font("Arial", FontWeight.BOLD, 14));
    vbox.getChildren().add(title);

    Text t = new Text("JavaFX is a software platform");
    vbox.getChildren().add(t);
    t = new Text("for creating and delivering desktop");
    vbox.getChildren().add(t);
    t = new Text("application");
    vbox.getChildren().add(t);
    t = new Text("application");
    vbox.getChildren().add(t);
    t = new Text("across a wide variety of devices");
    vbox.getChildren().add(t);
    t = new Text("JavaFX is intended to replace");
    vbox.getChildren().add(t);
    t = new Text("Swing as the standard GUI library");
    vbox.getChildren().add(t);
    t = new Text("for Java SE, but both will be included");
    vbox.getChildren().add(t);
    t = new Text("for the foreseeable future.");
    vbox.getChildren().add(t);

    return vbox;
}

```





```

public class Eks04 extends Application {

    @Override
    public void start(Stage vindu) throws Exception {

        BorderPane border = new BorderPane();
        HBox hbox = nyHBox();

        border.setTop(hbox);

        border.setLeft(vBoxMedTekst());

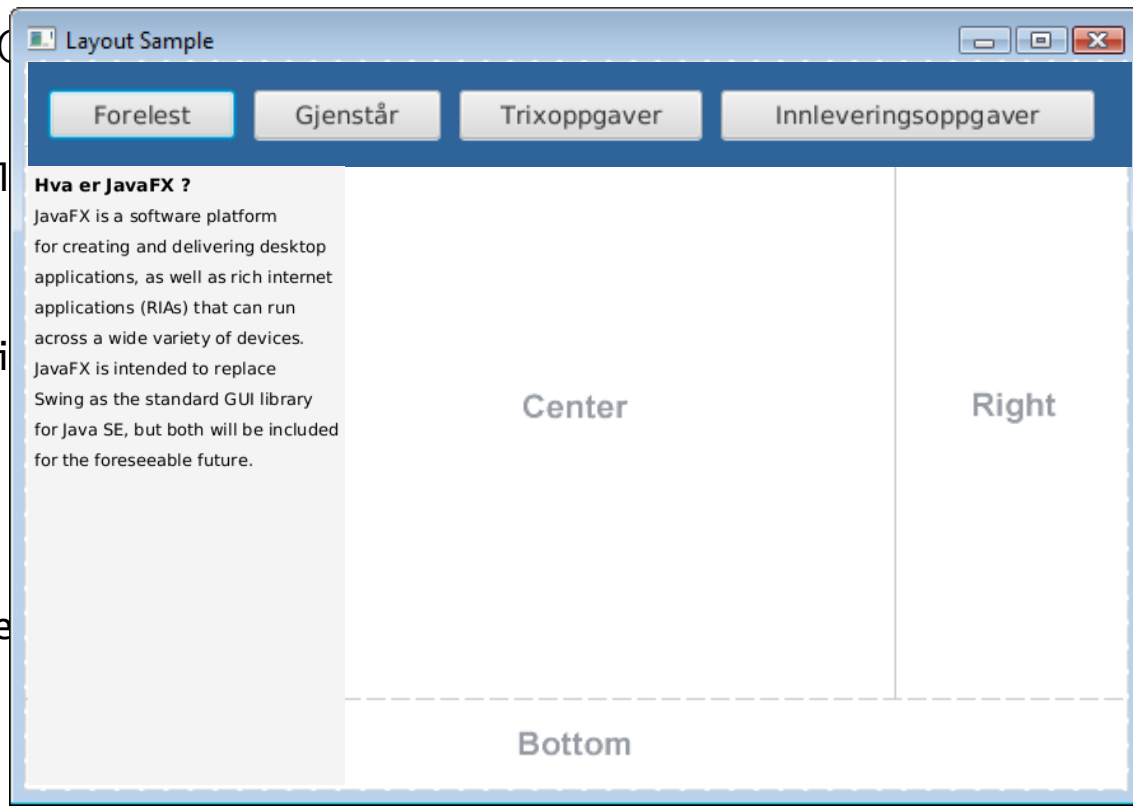
        border.setCenter(minGridPane());

        Scene scene = new Scene(
vindu.setScene(scene);
vindu.show();
vindu.setTitle("Eksempel");
vindu.setScene(scene);
    }

    public static void main(String
        launch(a);
    }

    private HBox nyHBox() { ... }
    private VBox vBoxMedTekst()
    private GridPane minGridPane
        . . .
    }

```



```

public class Eks04 extends Application {

    @Override
    public void start(Stage vindu) throws Exception {

        BorderPane border = new BorderPane();
        HBox hbox = nyHBox();

        border.setTop(hbox);

        border.setLeft(vBoxMedTekst());

        border.setCenter(minGridPane());

        Scene scene = new Scene(border);
        vindu.setScene(scene);
        vindu.show();
        vindu.setTitle("Eksempel på bruk av Border- og GridPane, HBox og VBox");
        vindu.setScene(scene);
    }

    public static void main(String[] args) {
        launch(a);
    }

    private HBox nyHBox() { ... }
    private VBox vboxMedTekst() { ... }
    private GridPane minGridPane() { ... }
}

```

