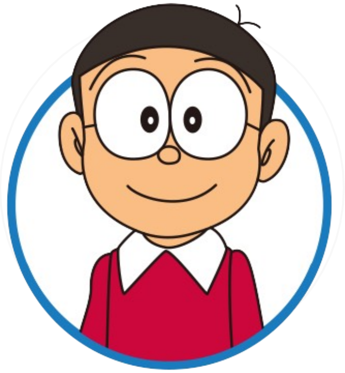
JavaFX 简明教程 2

INTRODUCTION



大熊

AUTHOR

https://qiaomenzhuan.com.au

Outline

Understand Top Level Containers

- Stage
- Scene

Understand Layout Containers

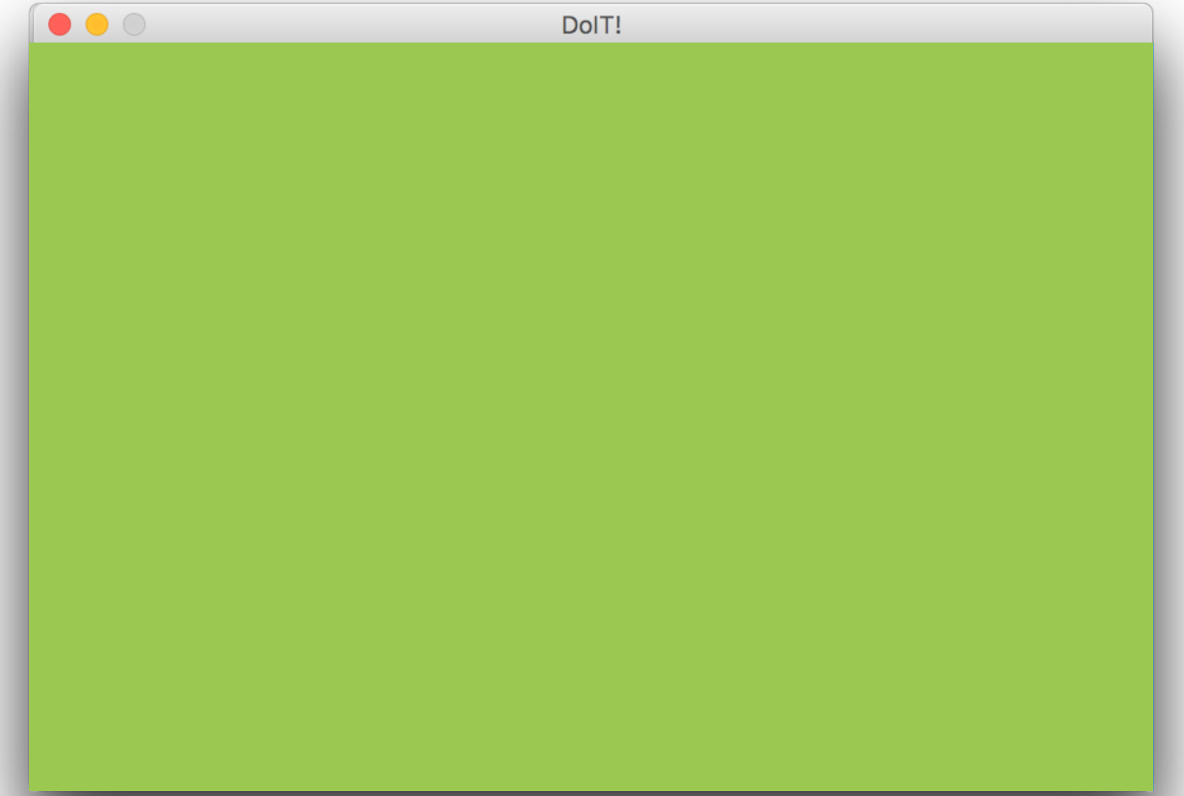
- Grid pane
- Border pane

Use various controls

- Textbox, Label
- Button, Combobox, Checkbox
- Table view

Stage & Scene

JavaFX App Components – Stage



Top Level Container

Defines a space for the app

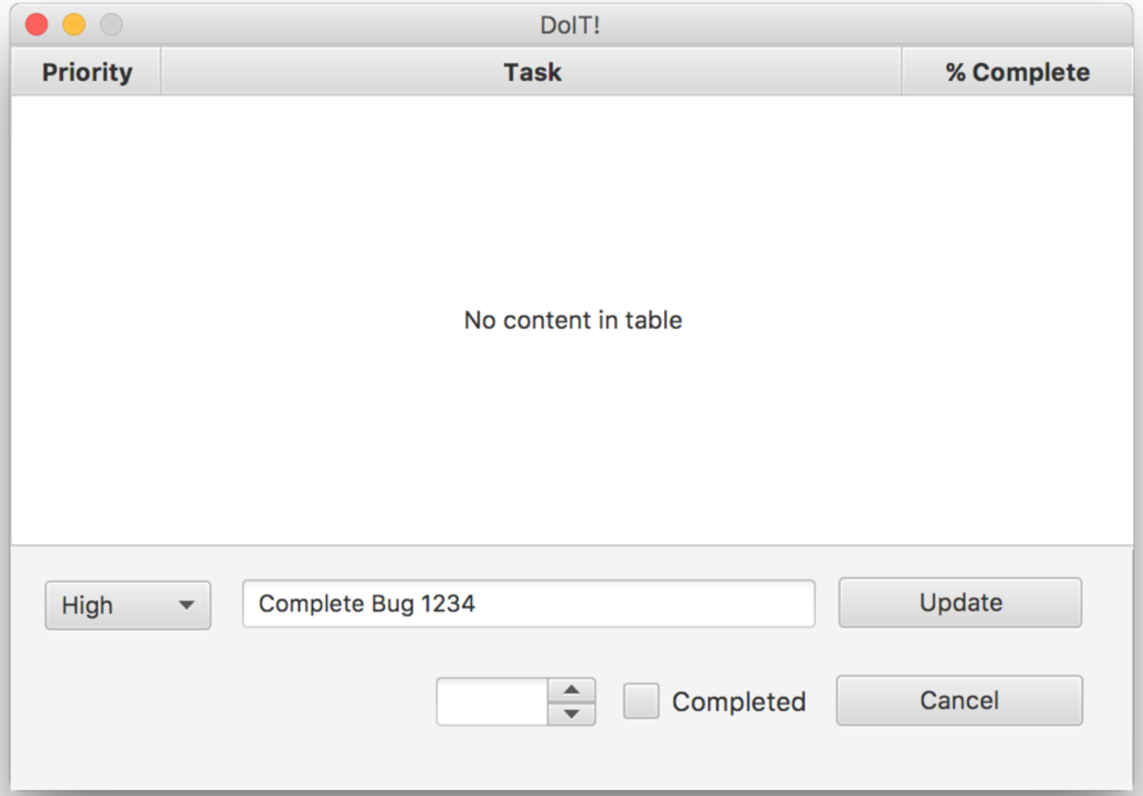
At least one stage, called primary stage

Primary stage is created by runtime, on startup

Container for scenes

Appearance and functionality varies based on platform

JavaFX App Components - Scene



DoIT!

Priority	Task	% Complete

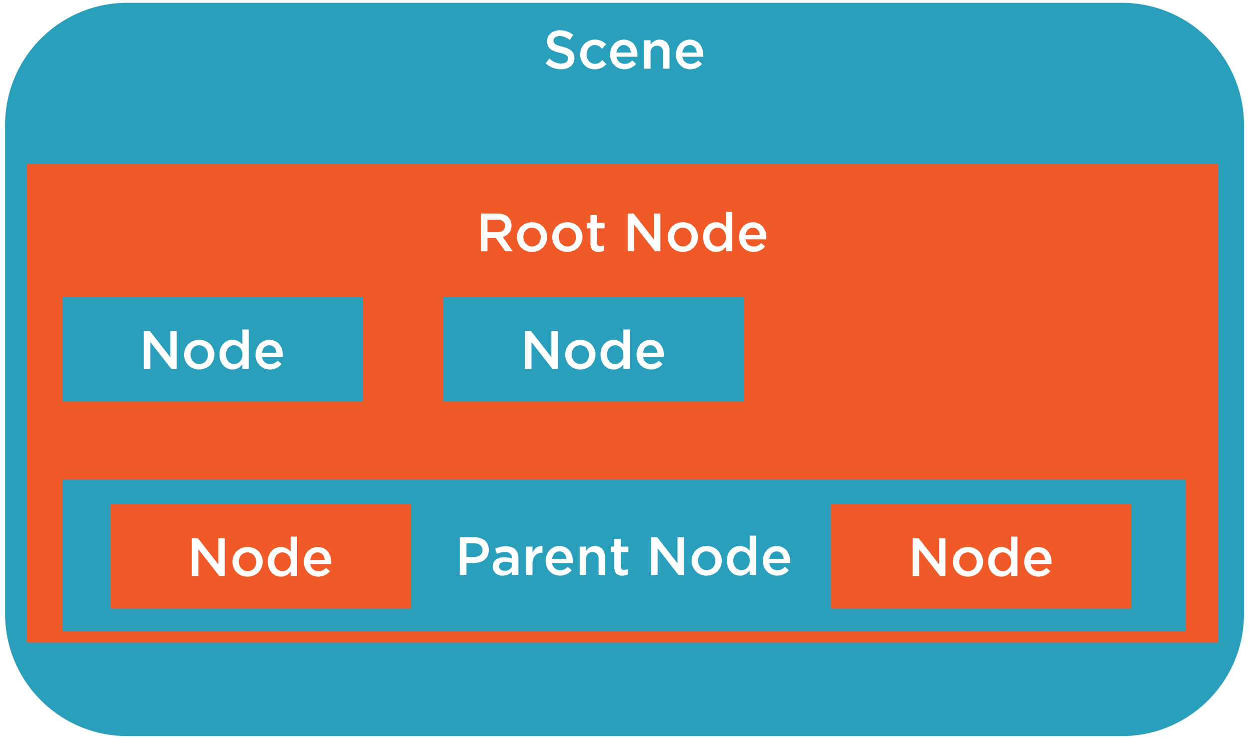
No content in table

Hosted by a stage

Container for elements that comprise the scene

Consists of a Scene Graph

Scene Graph



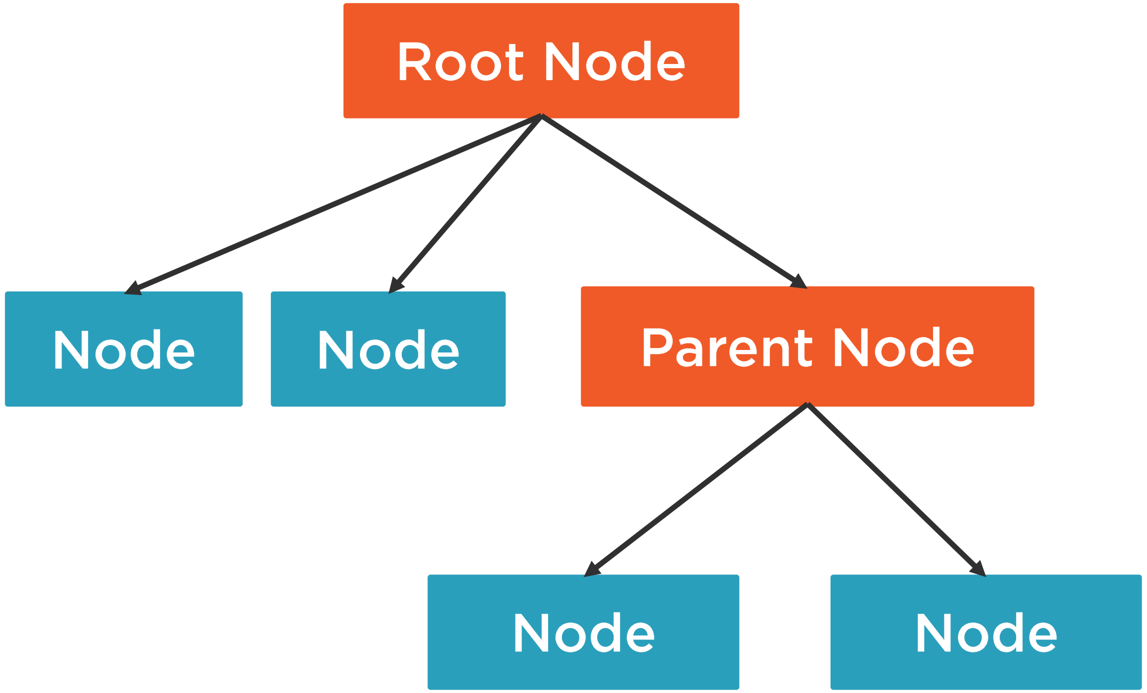
Elements of a scene are Nodes

Textbox, button, checkbox etc.,

Parent Nodes can contain other child nodes

All are subclasses of Node

Scene Graph



Elements of a scene are Nodes

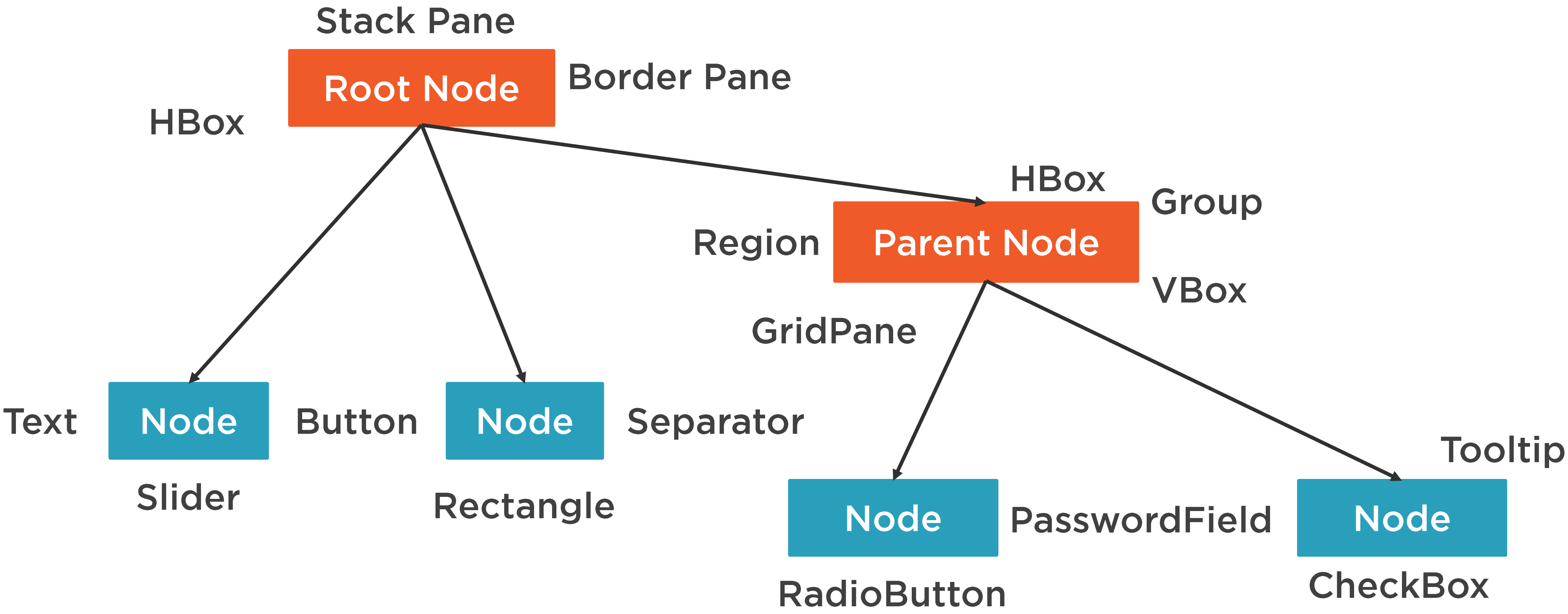
Textbox, button, checkbox etc.,

Parent Nodes can contain other child nodes

All are subclasses of Node

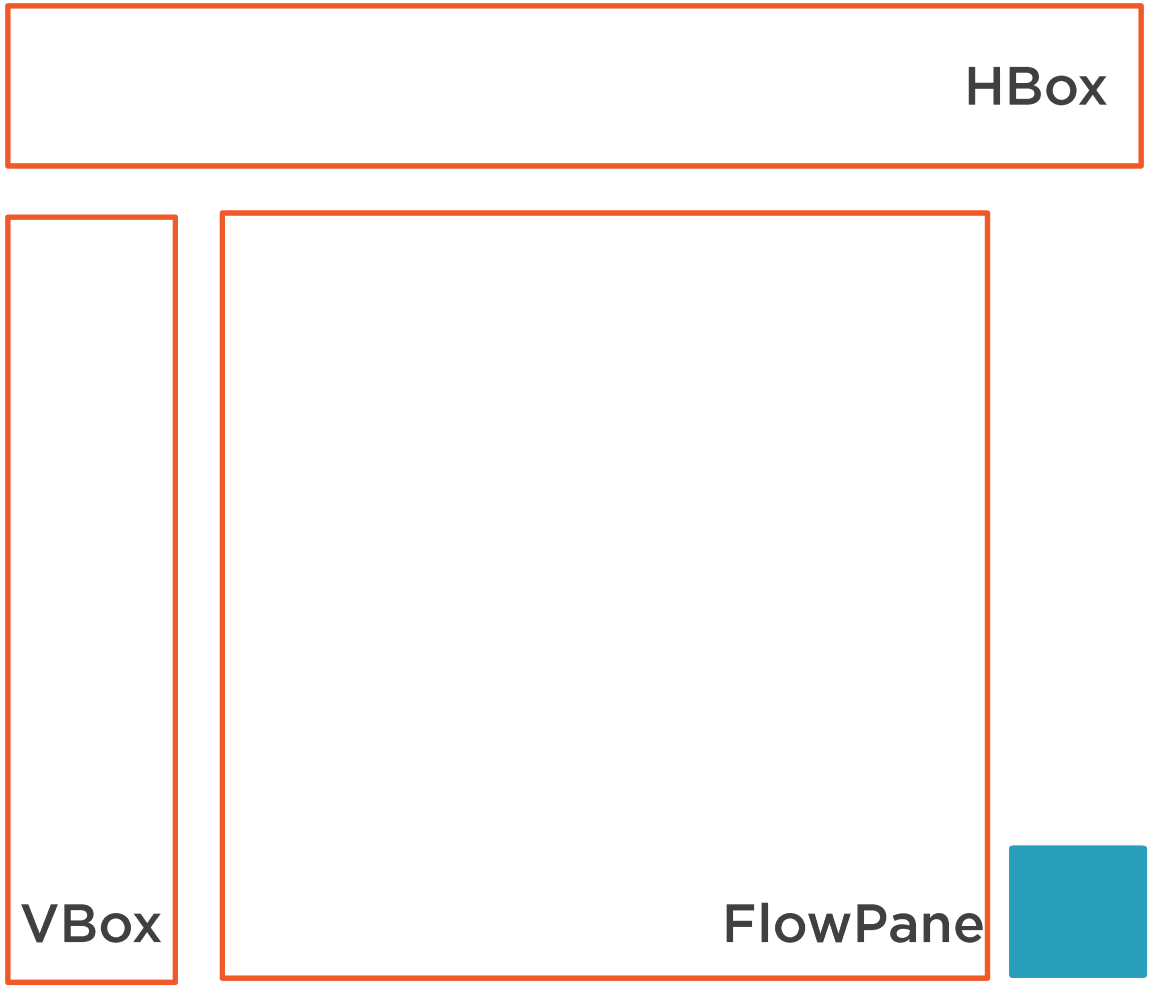
Nodes form a hierarchical data structure

Scene Graph


Stack Pane
Root Node
Border Pane
HBox
HBox
Group
Region
Parent Node
VBox
GridPane
Text
Node
Button
Node
Separator
Slider
Rectangle
Tooltip
Node
PasswordField
Node
RadioButton
CheckBox

Layouts

JavaFX App Components - Layouts

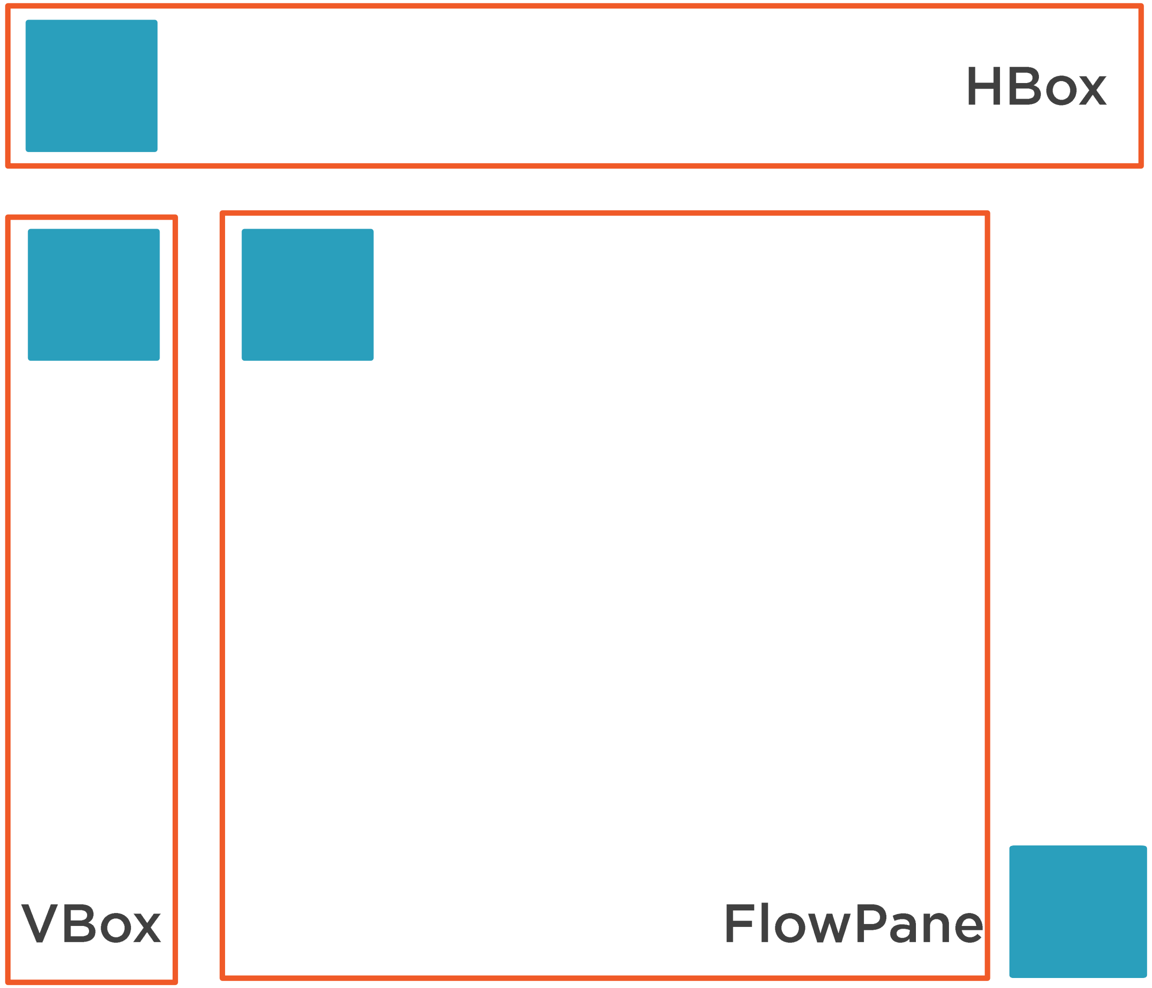


An easier option to place nodes

JavaFX has many default layouts

- HBox, VBox
- FlowPane
- BorderPane
- GridPane

JavaFX App Components - Layouts

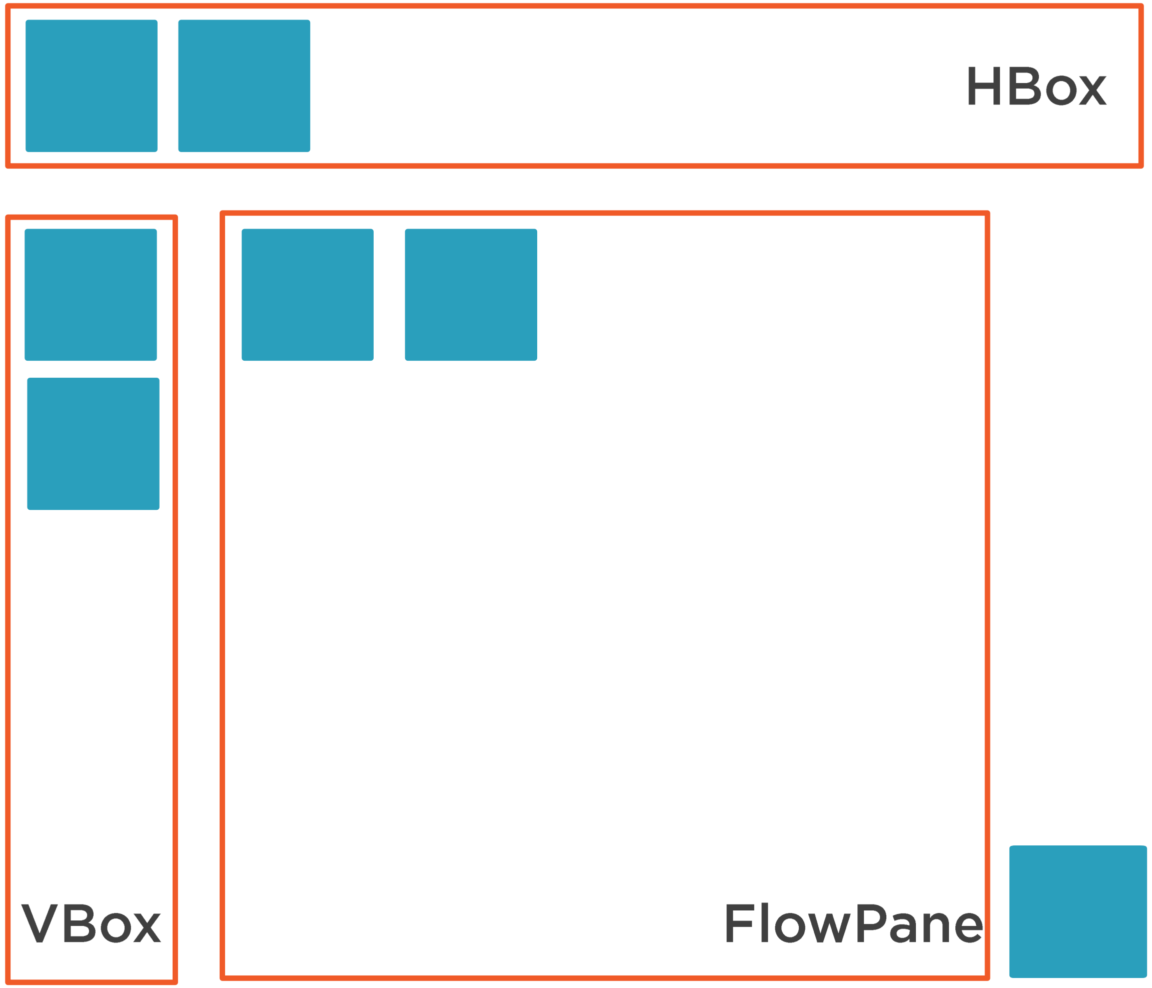


An easier option to place nodes

JavaFX has many default layouts

- HBox, VBox
- FlowPane
- BorderPane
- GridPane

JavaFX App Components - Layouts

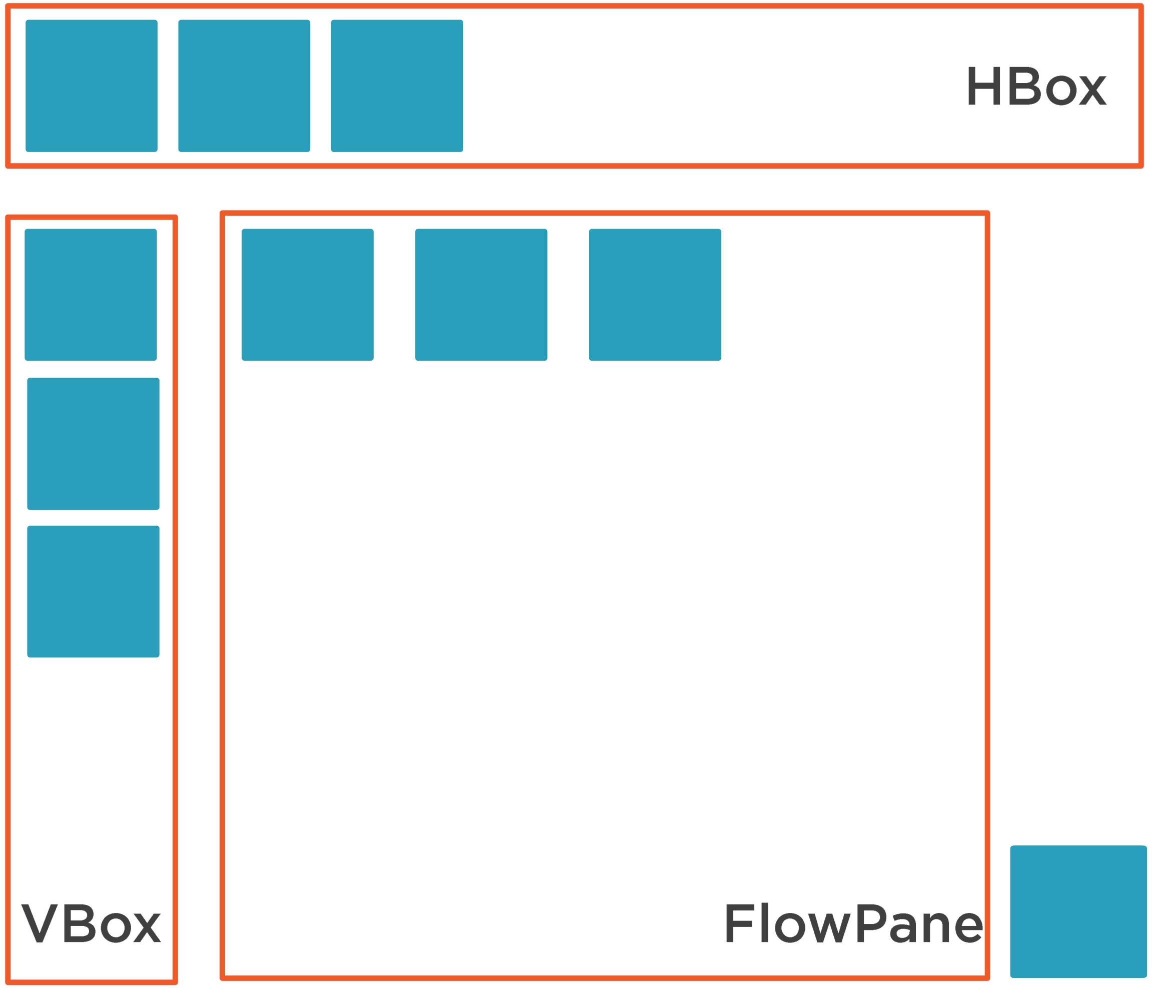


An easier option to place nodes

JavaFX has many default layouts

- HBox, VBox
- FlowPane
- BorderPane
- GridPane

JavaFX App Components - Layouts

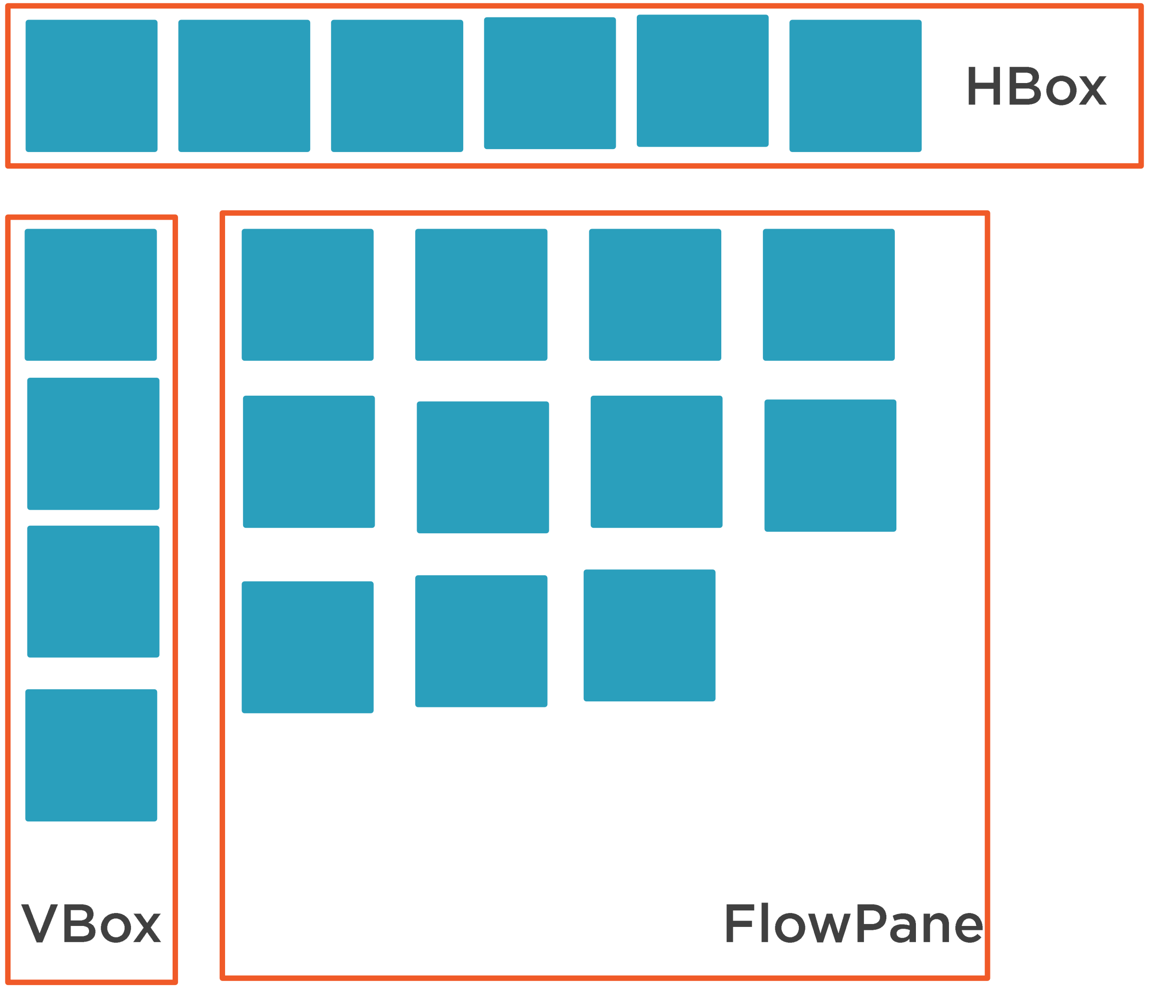


An easier option to place nodes

JavaFX has many default layouts

- HBox, VBox
- FlowPane
- BorderPane
- GridPane

JavaFX App Components - Layouts

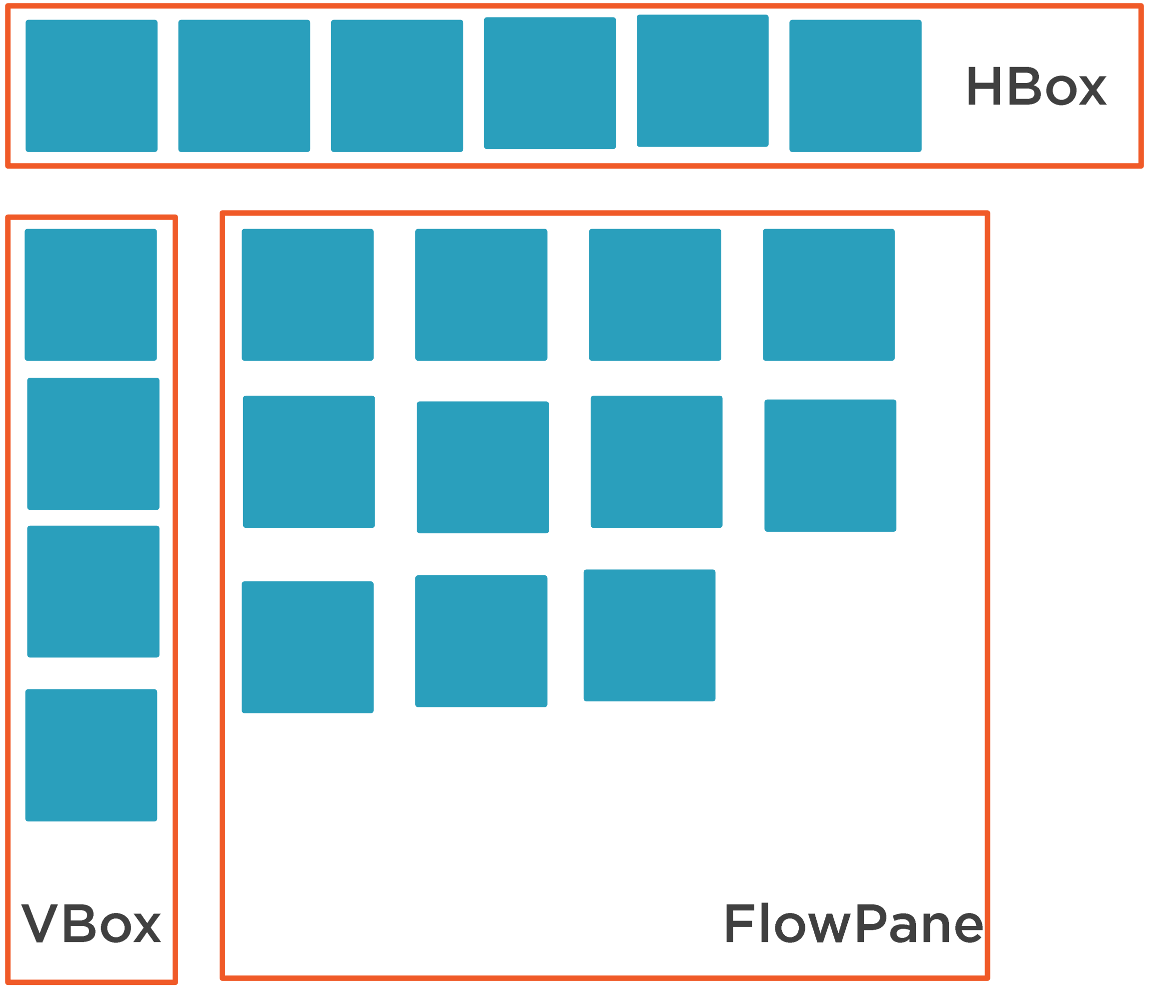


An easier option to place nodes

JavaFX has many default layouts

- HBox, VBox
- FlowPane
- BorderPane
- GridPane

JavaFX App Components - Layouts



An easier option to place nodes

JavaFX has many default layouts

- HBox, VBox
- FlowPane
- BorderPane
- GridPane

Layouts can contain other layouts

```
HBox hbox = new HBox();
hbox.setSpacing(10);

Button button1 = new Button(”First");
button1.setPrefSize(80, 20);

Button button2 = new Button(”Second");
button2.setPrefSize(80, 20);

hbox.getChildren()
    .addAll(button1,button2);

Scene scene = new Scene(hbox);
primaryStage.setScene(scene);
```

◀ Creation of a HBox

◀ Customize HBox

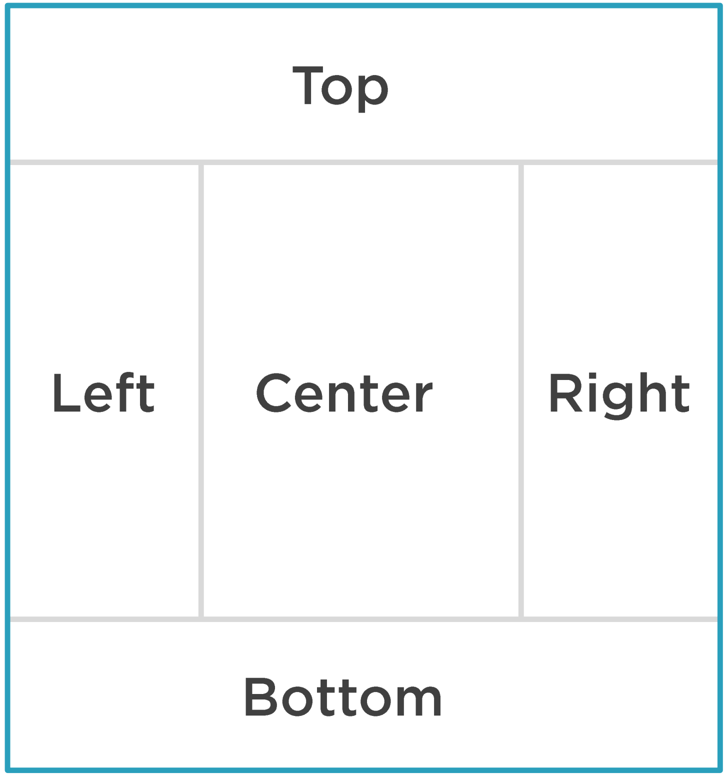
◀ Creating a button

◀ One more button

◀ Buttons are added to hbox

◀ Add the hbox to scene

Border Pane



Top

Left

Center

Right

Bottom

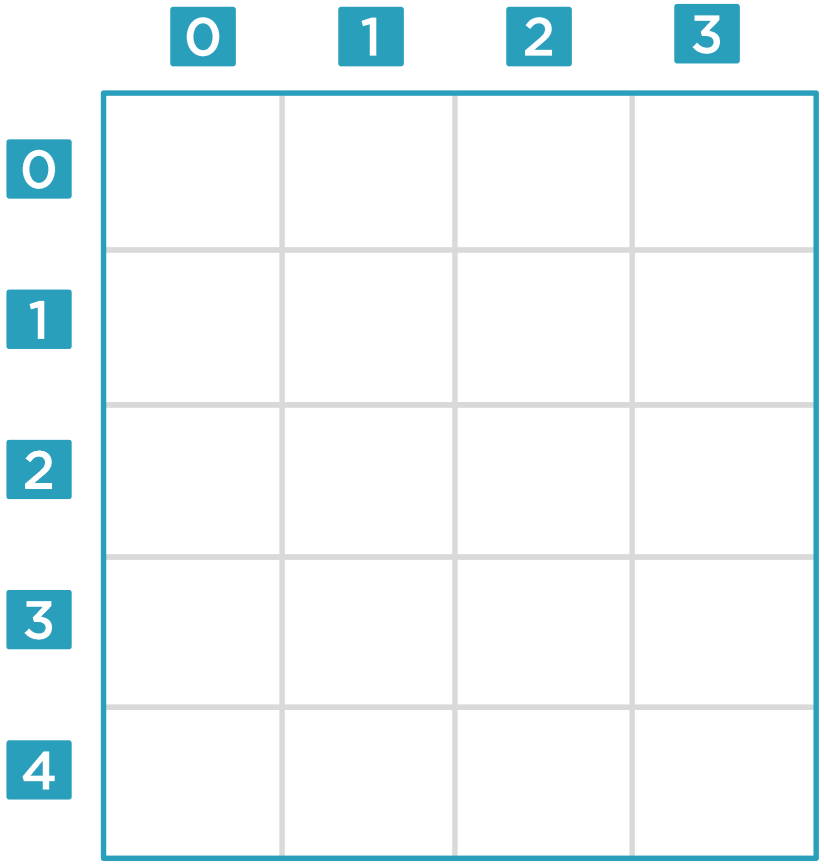
Splits into 5 regions

- Top, Bottom, Right, Left, Center

Each region

- Is optional
- Can be of any size
- Can only have 1 node

Grid Pane



Divides area into cells

Nodes can be placed in any cell

Constraints can be added

Very good for form layouts

Other Layouts

TilePane

StackPane

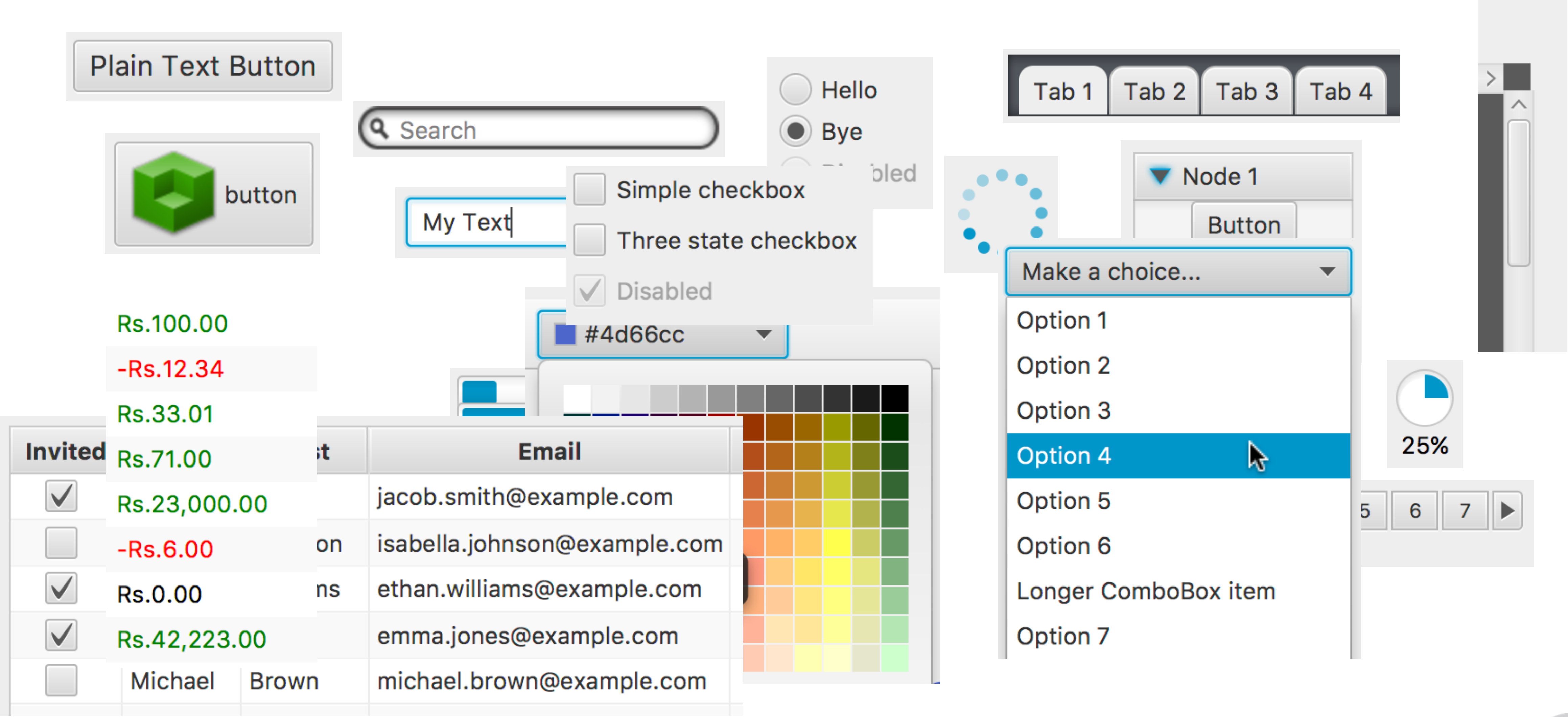
AnchorPane

Controls

Controls

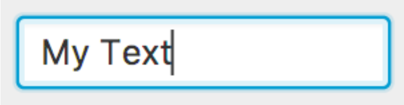
Specialized nodes that are suited for reuse in different application contexts and mainly intended for interacting and communicating with Users.

Examples



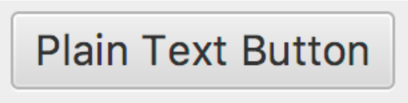
Invited		t	Email
☑			jacob.smith@example.com
□		on	isabella.johnson@example.com
☑		ns	ethan.williams@example.com
☑			emma.jones@example.com
□	Michael	Brown	michael.brown@example.com

Examples

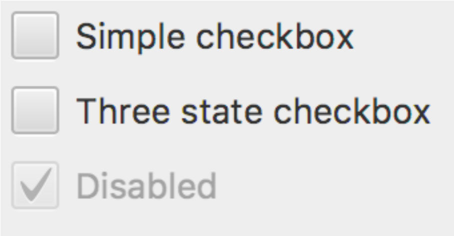


My Text

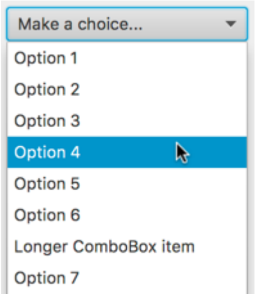
TextField



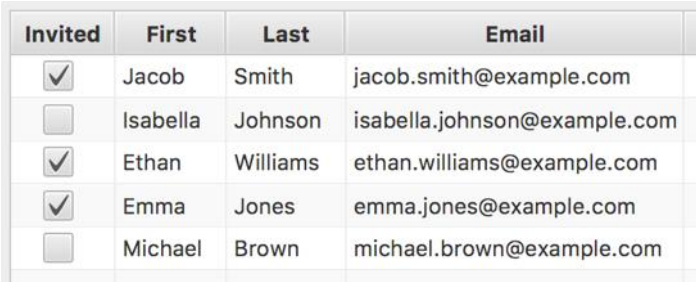
Button



Checkbox

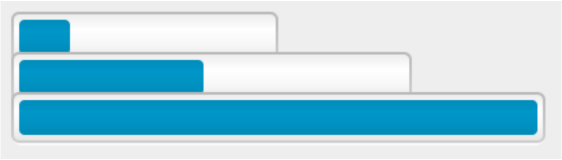


ComboBox



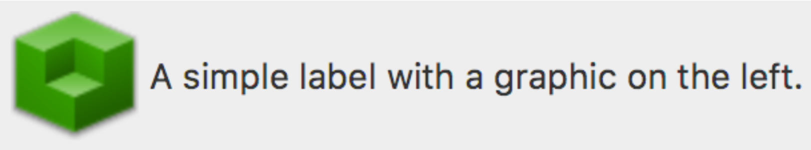
Invited	First	Last	Email
☑	Jacob	Smith	jacob.smith@example.com
☐	Isabella	Johnson	isabella.johnson@example.com
☑	Ethan	Williams	ethan.williams@example.com
☑	Emma	Jones	emma.jones@example.com
☐	Michael	Brown	michael.brown@example.com

Table



ProgressBar

Label



A simple label with a graphic on the left.

Class

- javafx.scene.control.Label

User Interaction

- Non editable content

Accessing from code

- setText() to change label text
- setCursor() to set cursor icon
- setGraphic() to use an image as label

```
Label l1 = new Label();

Label l2 = new Label("Search");

Image image = new Image(getClass()

    .getResourceAsStream(”pic.jpg"));

l1.setGraphic(new ImageView(image));
```

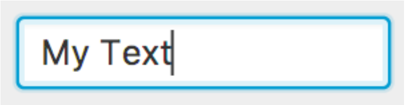
◀ Create label without text

◀ Create label with text

◀ Prepare image to use with label

◀ Attach image to the label

Text Field



My Text

Class

- javafx.scene.control.TextField

User Interaction

- Accepts textual input from user

Accessing from code

- getText() method returns text
- setText() to set value programatically
- We can write validators

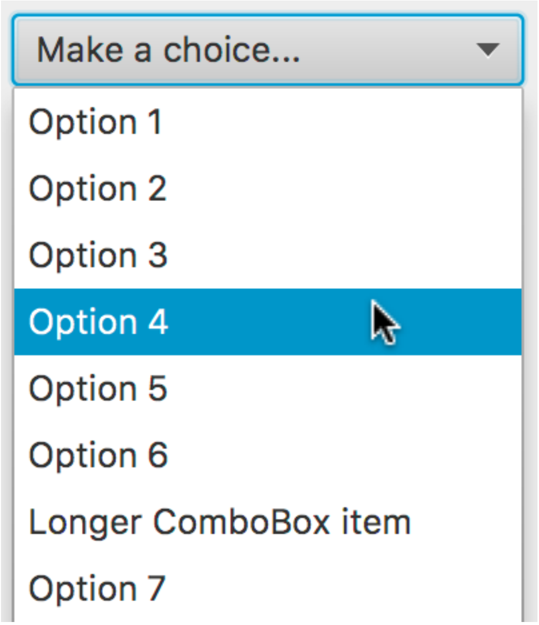
Variations

- PasswordField

```
TextField name = new TextField();

name.setPromptText("Enter name.");

hbox.getChildren().add(name);


name.setText(”My Text”);
```

- Creates a text field
- Text when field is empty
- Can be added to layout
- Setting value programatically

ComboBox



Make a choice...

Option 1

Option 2

Option 3

Option 4

Option 5

Option 6

Longer ComboBox item

Option 7

Class

- javafx.scene.control.ComboBox

User Interaction

- Lets user choose from list of values

Accessing from code

- getItems().addAll() to add options
- setValue() to selects a value
- getValue() returns selected value

Variations

- ChoiceBox

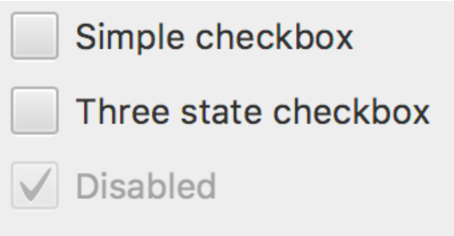
```
ComboBox comboBox = new ComboBox();
priorityComboBox.getItems().addAll(

    "Highest",

    "High",

    "Normal",

    "Low");

comboBox.setValue("Normal");
```

◀ Creates a Combo box

◀ Adds list of values

◀ Selecting from code

CheckBox



□ Simple checkbox

□ Three state checkbox

☑ Disabled

Class

- javafx.scene.control.CheckBox

User Interaction

- Deselect or select

Accessing from code

- setSelected() to change state
- getSelected() to get state
- setIndeterminate(true) for 3 state checkbox

```
CheckBox cb1 = new CheckBox();


CheckBox cb2 = new CheckBox(“1”);

cb1.setText(“2”);

cb1.setSelected(true);
```

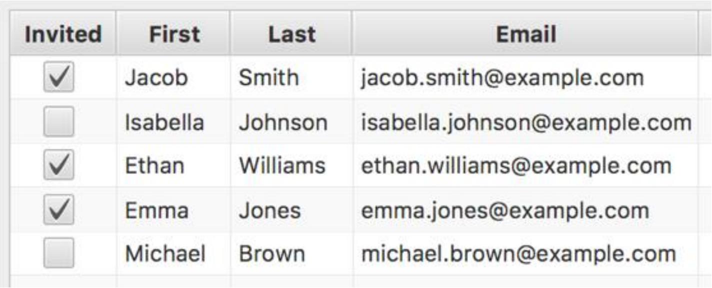
◄ Checkbox without label

◄ Checkbox with label

◄ Setting label

◄ Selects the checkbox

TableView



Invited	First	Last	Email
☑	Jacob	Smith	jacob.smith@example.com
☐	Isabella	Johnson	isabella.johnson@example.com
☑	Ethan	Williams	ethan.williams@example.com
☑	Emma	Jones	emma.jones@example.com
☐	Michael	Brown	michael.brown@example.com

Class

- javafx.scene.control.TableView

User Interaction

- Select rows, cells, reorder, sort and many more

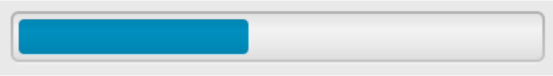
Accessing from code

- setEditable() to change read only status
- getColumns().addAll() to add columns
- setCellFactory() for complicated cells

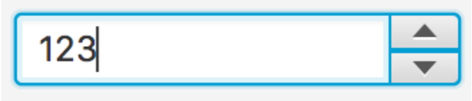
```
TableView table = new TableView();

table.setEditable(true);

TableColumn firstNameCol = new
TableColumn("First Name");
TableColumn lastNameCol = new
TableColumn("Last Name");



table.getColumns().addAll(firstName
Col, lastNameCol);
```

- Creation of TableView
- Make the table editable
- Creation of columns
- Add the columns to table

Other Controls



ProgressBar



Spinner

Demo

Demonstrate usage of following controls

- Label
- TextField
- ComboBox
- Spinner
- CheckBox
- TableView

Add these controls to GridPane

Summary

Stage & Scene

Following layouts have been explained

- GridPane
- BorderPane
- HBox
- VBox

Many controls have been demonstrated