

QtQuick Training Course



Module Qt Components

Objectives

1 Qt Components

What is Qt Components?

Setting up your environment

Basic interface elements

Style

Text input

Buttons

Menu

Dialog boxes

Tabs

Pages

Topics

1 Qt Components

2 Cheat Sheet

3 Questions

4 Lab

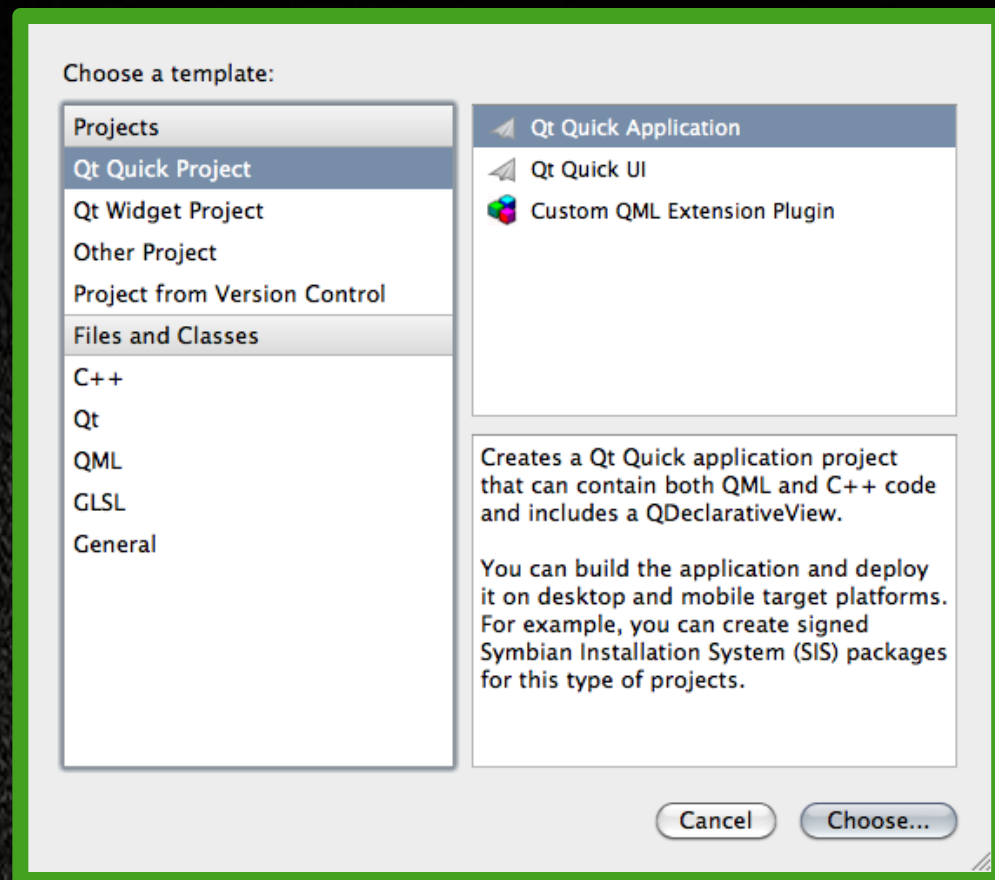
What is Qt Components?

Speed up Qt Quick development

Ready-made QML-based UI building blocks

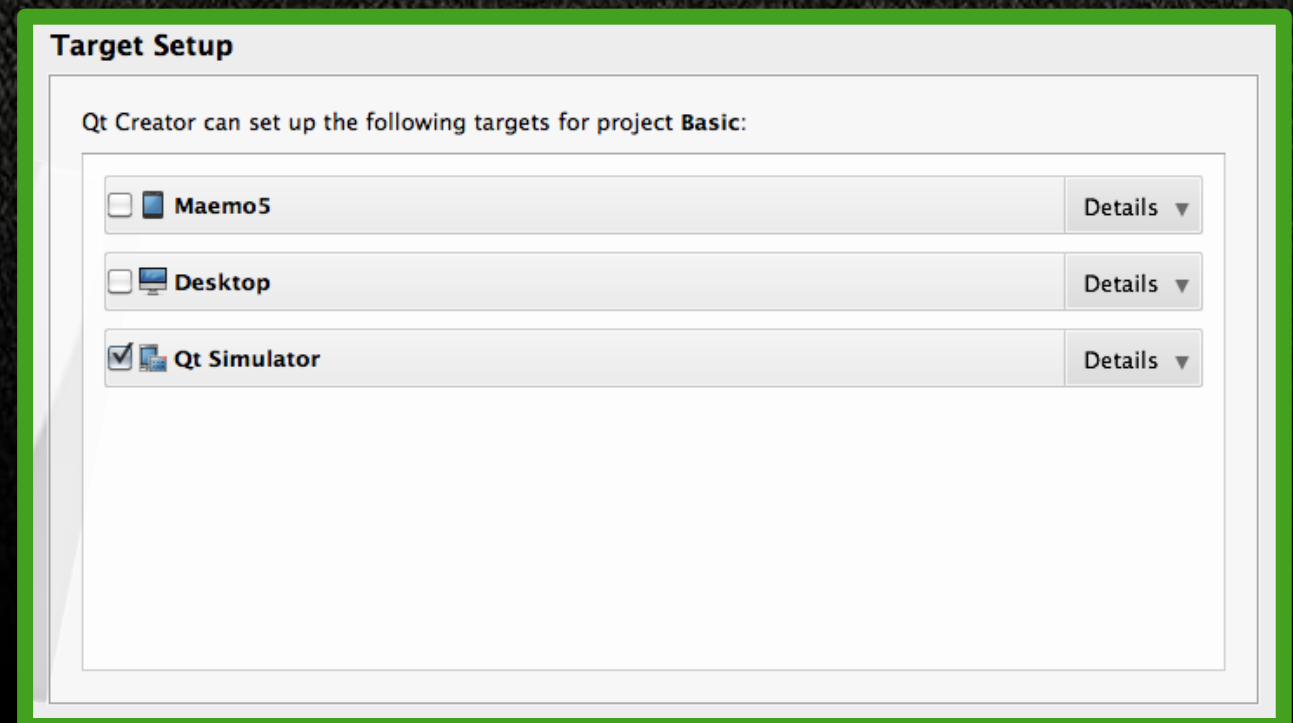
Functional Symbian components

Setting up your environment

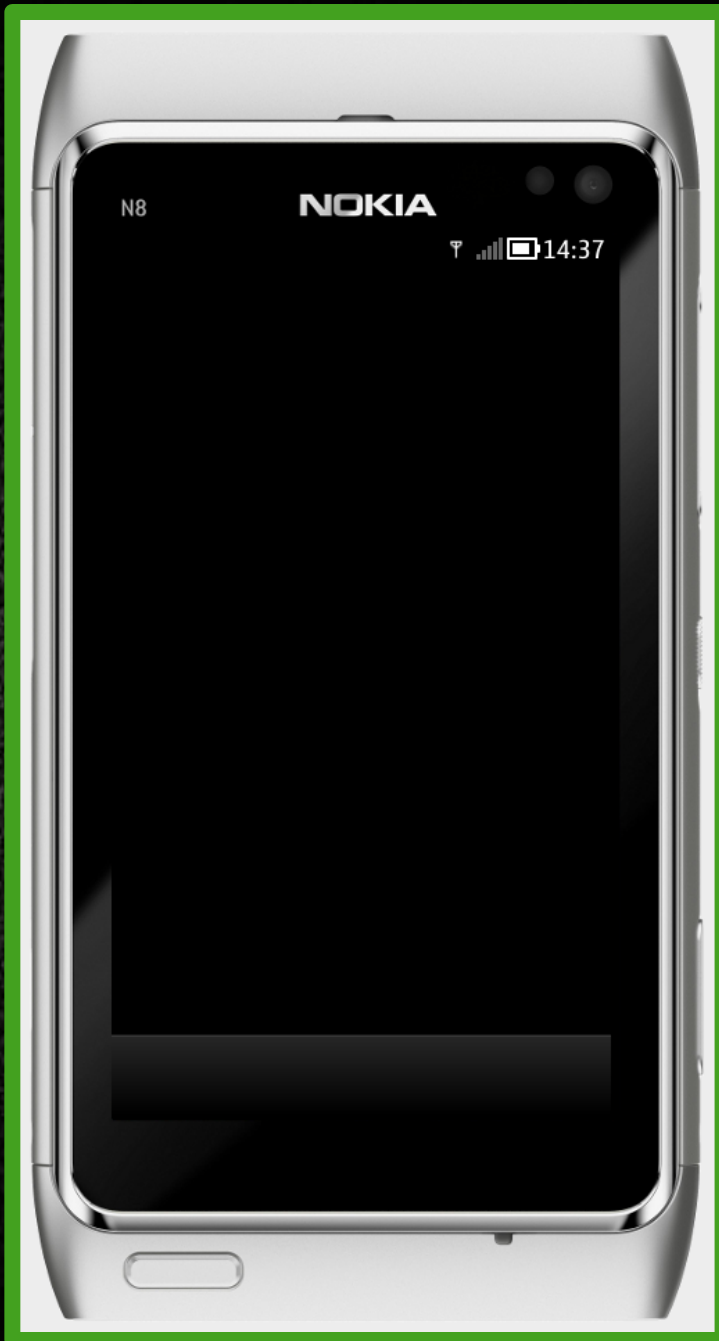


Create a Qt Quick Application project

Set the project up for Qt Simulator



Basic interface elements



First, import the Qt Components API

Window: the root component
(handles transitions when rotating the screen)

StatusBar: display device's informations
(connection, battery, time)

ToolBar: container for
quick access to tasks

```
import QtQuick 1.0
import com.nokia.symbian 1.1

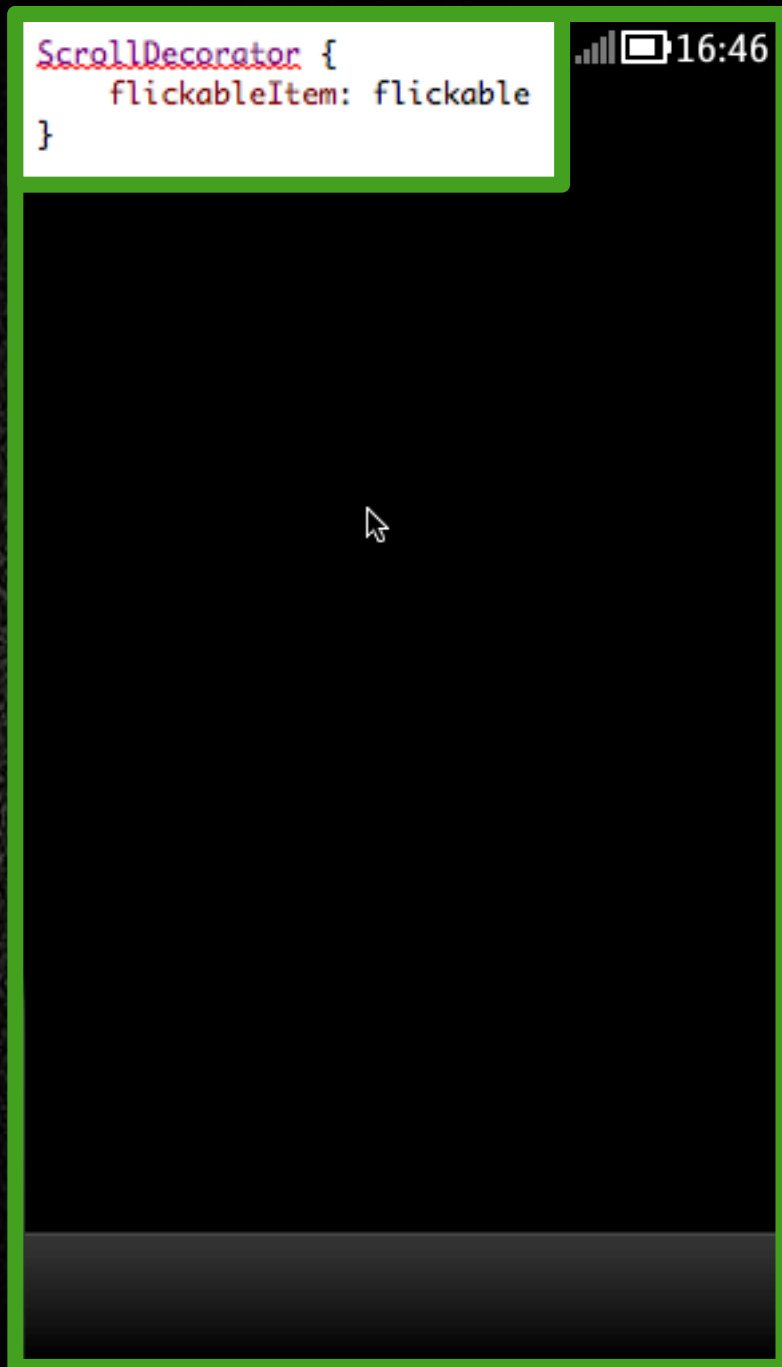
▼ Window {
    id: root

    ▼ StatusBar {
        id: statusBar
        anchors.top: root.top
    }

    ▼ ToolBar {
        id: toolBar
        anchors.bottom: root.bottom
    }
}
```

See example: [addon/module-010/examples/Basic/Basic.pro](#)

Basic interface elements

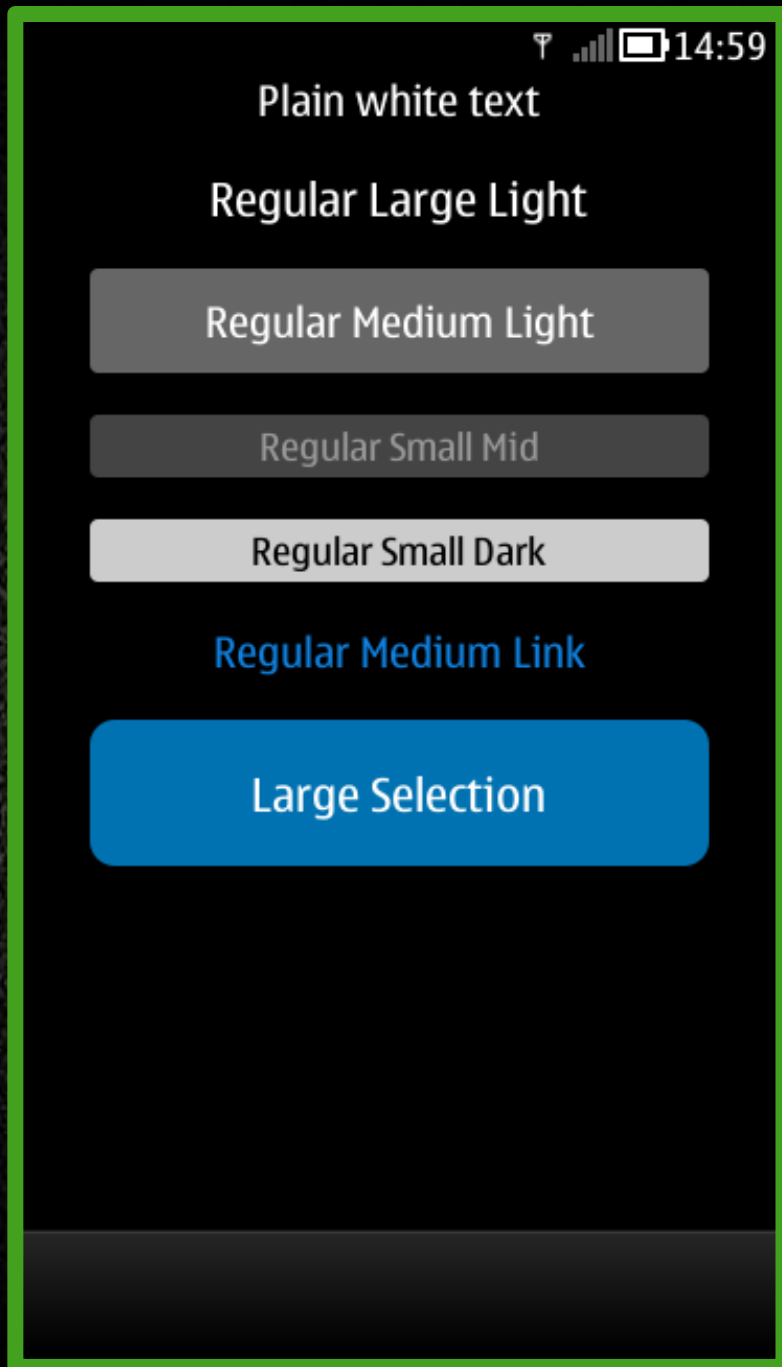


ScrollDecorator: sets a temporary scrollbar to a flickable content while it is being dragged.

See video: [addon/module-010/videos/Scroll.mov](#)

See example: [addon/module-010/examples/Scroll/Scroll.pro](#)

Style



Use **platformStyle** on properties
(e.g. `font.family: platformStyle.fontFamilyRegular`)

fontFamilyRegular: default font family

fontSizeLarge: pixel size for a font
`fontSizeMedium / fontSizeSmall`

graphicSizeLarge: size of a graphical object
(Image)
`graphicSizeMedium / graphicSizeSmall / graphicSizeTiny`

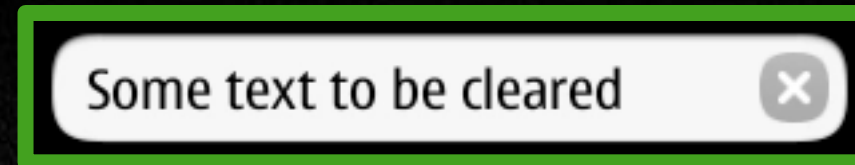
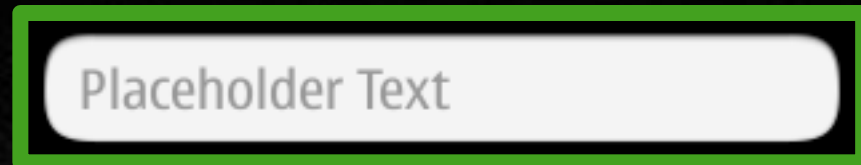
paddingLarge: size of padding, in pixels
`paddingMedium / paddingSmall`

colorNormalLight: default foreground color
`colorNormalMid / colorNormalDark / colorNormalLink`

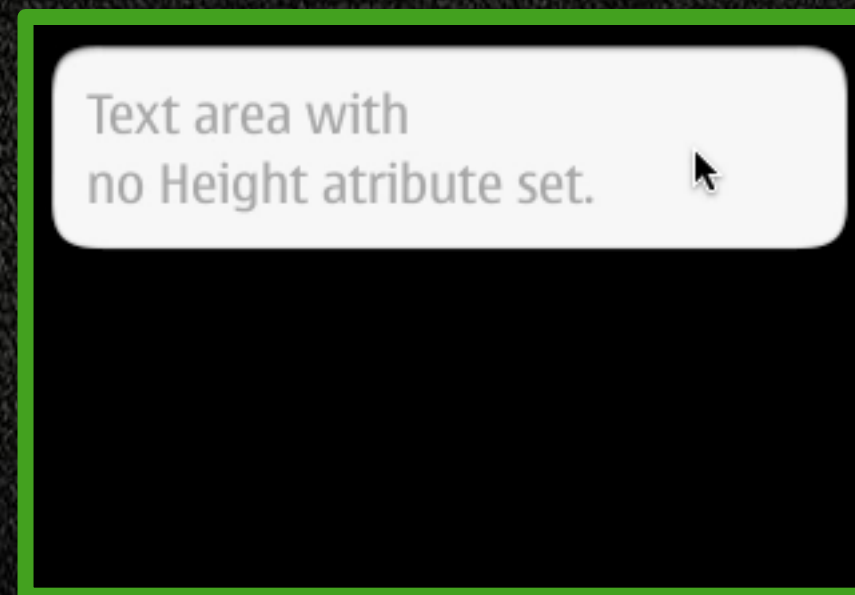
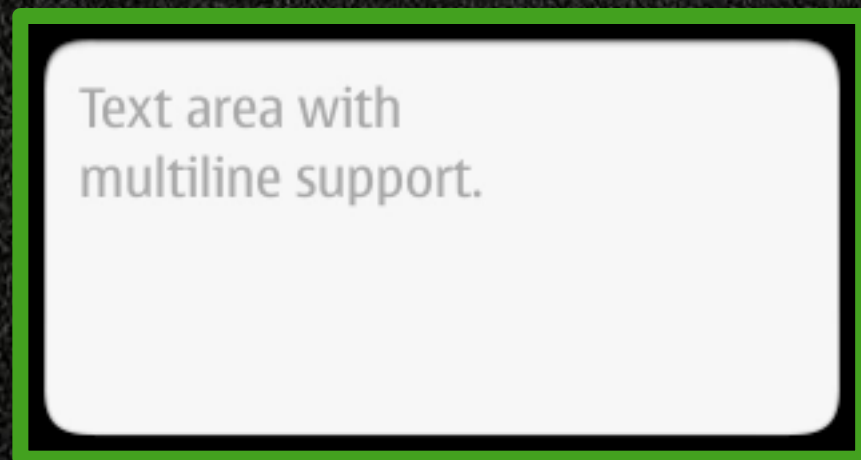
See the cheat sheet in the end of the module for more.

See example: `addon/module-010/examples/Style/Style.pro`

Text input



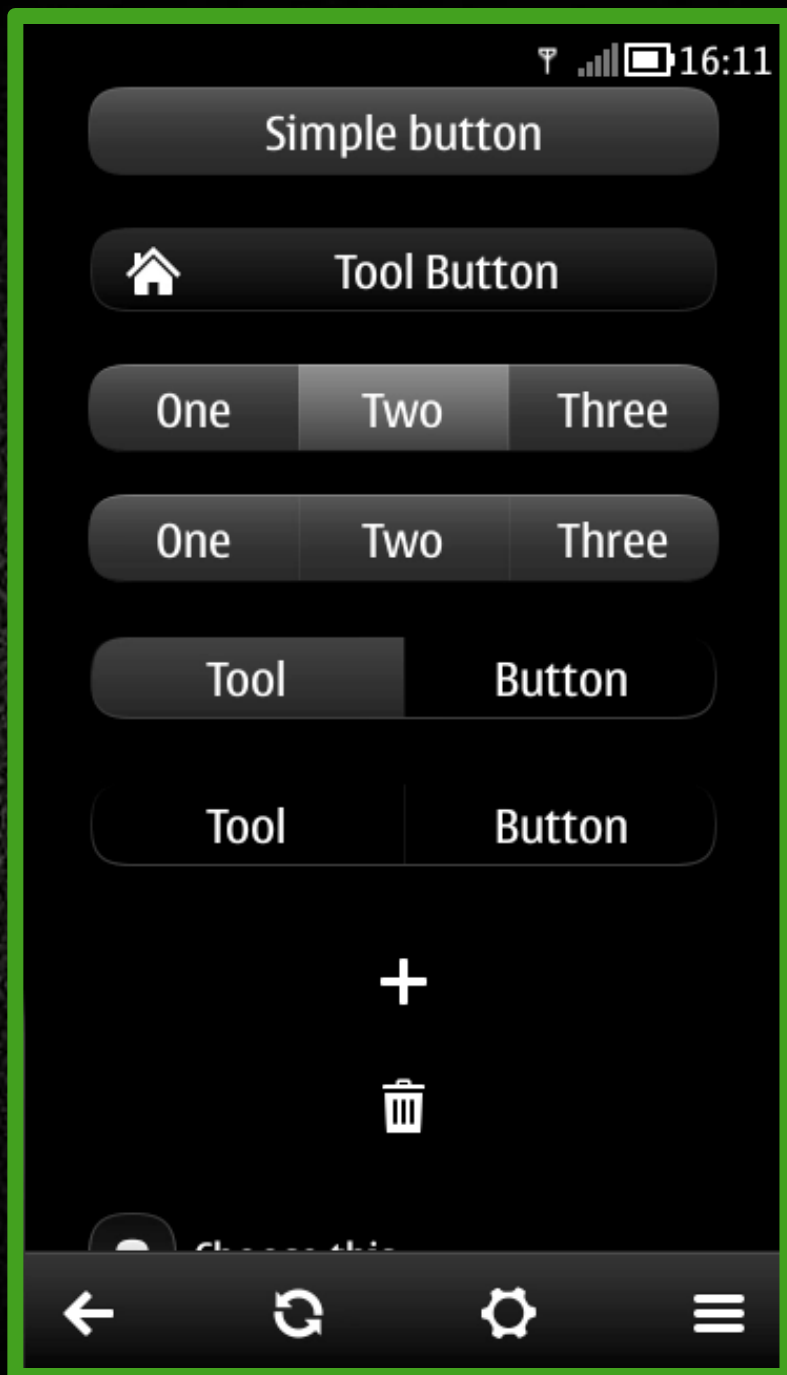
TextField can have **placeholderText**, integrated buttons and actions



TextArea can either be scrollable on fixed width and height or resize to fit all input characters

See videos: [addon/module-010/videos/ClearableTextField.mov](#) ; [TextArea.mov](#) ;
[TextAreaNoHeight.mov](#)
See example: [addon/module-010/examples/Text/Text.pro](#)

Buttons



Button: standard button with a text

ToolButton: different button compatible with icons

ButtonRow: arranges a set of buttons in a row

ButtonColumn: arranges a set of buttons in a column

RadioButton / **CheckBox**: button variations for specific purposes

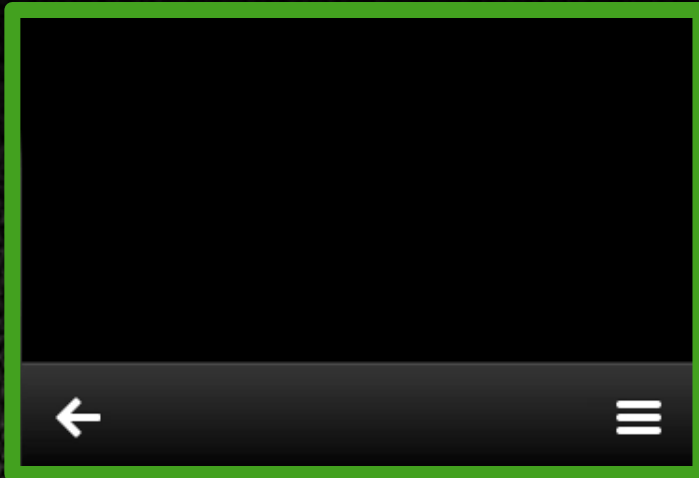
ToolBarLayout: automatically arranges items on a toolbar

See the cheat sheet in the end of the module for more.

See video: [addon/module-010/videos/Button.mov](#)

See example: [addon/module-010/examples/Button/Button.pro](#)

Menu



The Component element defines a QML component within the document, rather than a separate QML file

```
Component {  
    id: menuComponent  
    Menu {  
        content: MenuLayout {  
            MenuItem {  
                text: "Menu Item"  
            }  
            MenuItem {  
                text: "Quit"  
                onClicked: Qt.quit()  
            }  
        }  
    }  
}
```

The Menu component is hidden by default, therefore, you'll need to call it in an 'open' function.

MenuLayout automatically arranges the items in a Menu

See video: [addon/module-010/videos/Menu.mov](#)

See example: [addon/module-010/examples/Menu/Menu.pro](#)

Dialog boxes



CommonDialog and QueryDialog are variations of Dialog with subtle differences

CommonDialog: Dialog with support to title text

QueryDialog: CommonDialog with support to icon, accept and reject buttons, and message

SelectionDialog: Can work together with SelectionListItem, showing the current value

ContextMenu: Menu meets Dialog

See video: [addon/module-010/videos/Dialog.mov](#)

See example: [addon/module-010/examples/Dialog/Dialog.pro](#)

Tabs



TabBar: themeable background for TabBarLayout

TabBarLayout: positioner for the tab buttons

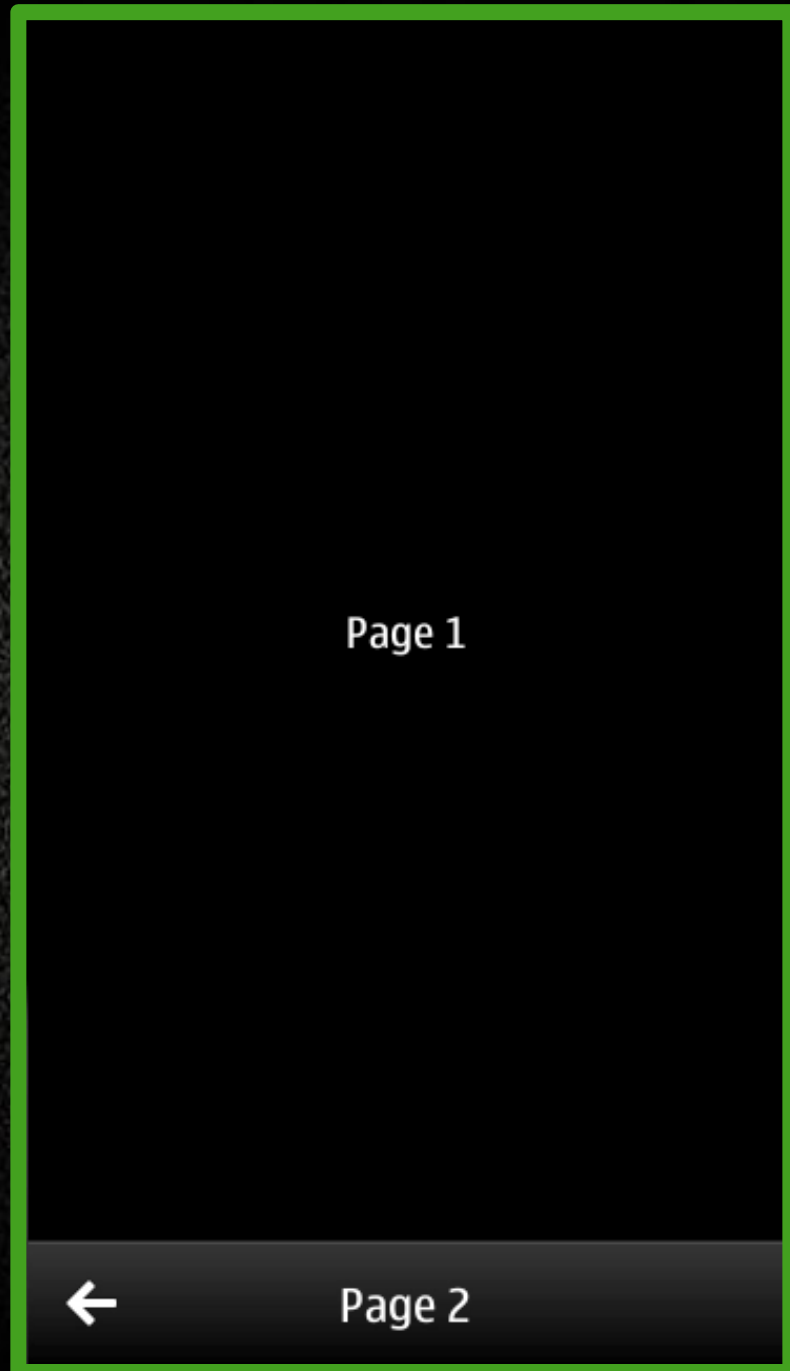
TabButton: has an image or label and can be clicked to activate a tab

TabGroup: set of pages for a tabbed interface

See video: [addon/module-010/videos/Tab.mov](#)

See example: [addon/module-010/examples/Tab/Tab.pro](#)

Pages



Page: defines one screen of user interface content

PageStack: defines a container for pages and a stack-based navigation model

See video: [addon/module-010/videos/Pages.mov](#)

See example: [addon/module-010/examples/Pages/Pages.pro](#)

Topics

1 Qt Components

2 Cheat Sheet

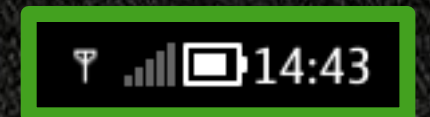
3 Questions

4 Lab

Window

The root component which handles transitions when rotating screen, maintaining a full screen view.

StatusBar



Display the device's informations.

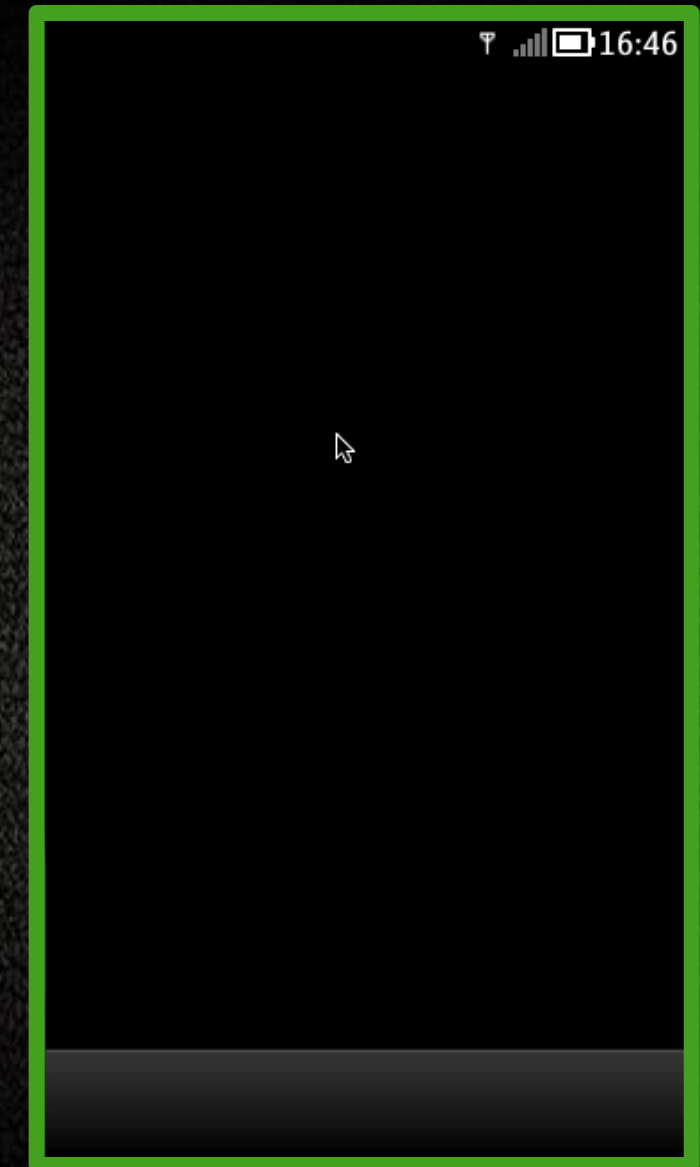
```
Ex.: Window {  
    id: root  
  
    StatusBar {  
        id: statusBar  
        anchors.top: root.top  
    }  
}
```


ScrollDecorator

Adds a visual indicator to a Flickable element to show the size relationship between the visible and the complete contents.

```
Ex.:  ListView {
        id: listView
        anchors.fill: parent
        focus: true
        clip: true
        model: listModel
        delegate: listDelegate
    }

    ScrollDecorator {
        id: scrollDecorator
        flickableItem: listView
    }
```



Style

Symbian platform specific style parameters for fonts, colors and layouts. Use `platformStyle` property.

fontFamilyRegular: default font family

fontSizeLarge: pixel size for a large font (also: `fontSizeMedium` / `fontSizeSmall`)

graphicSizeLarge: size of a large graphical object (also: `graphicSizeMedium` / `graphicSizeSmall` / `graphicSizeTiny`)

paddingLarge: size of a large padding in pixels (also: `paddingMedium` / `paddingSmall`)

borderSizeMedium: size for medium size border parts of a `BorderImage`, in pixels

colorNormalLight: default light foreground color (also: `colorNormalMid` / `colorNormalDark` / `colorNormalLink`)

colorPressed: foreground color for the pressed state (also: `colorChecked` / `colorHighlighted`)

colorDisabledLight: light foreground color for the disabled state (also: `colorDisabledMid` / `colorDisabledDark`)

colorTextSelection: background color for a selected item

```
Ex.:  Text {
      font.family: platformStyle.fontFamilyRegular
      font.pixelSize: platformStyle.fontSizeMedium
    }
```



Style

Image sources in the location “image:///theme/”.

 qtg_graf_popup_close_normal

 qtg_graf_popup_close_pressed


 qtg_graf_search_indicator

 qtg_graf_slider_decrement_normal



 qtg_graf_slider_decrement_pressed



 qtg_graf_slider_increment_normal



qtg_graf_slider_increment_pressed

 qtg_graf_textfield_add_normal



qtg_graf_textfield_add_pressed

qtg_graf_textfield_clear_normal



















qtg_graf_textfield_clear_pressed

Ex.:

```
Image {  
    source: "image:///theme/qtg_graf_popup_close_normal"  
}
```


Style

ToolBar icons.

	toolbar-add		toolbar-back
	toolbar-delete		toolbar-dialer
	toolbar-home		toolbar-list
	toolbar-mediacontrol-backwards		toolbar-mediacontrol-
	forward		
	toolbar-mediacontrol-pause		toolbar-mediacontrol-play
	toolbar-mediacontrol-stop		toolbar-menu
	toolbar-next		toolbar-previous
	toolbar-refresh		toolbar-search
	toolbar-settings		toolbar-share

Ex.: ToolBar {
 iconSource: "toolbar-home"
 }

TextField / TextArea

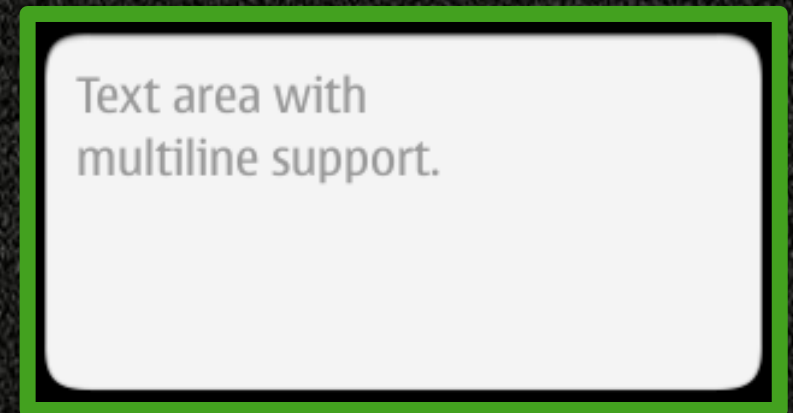
TextField and TextArea accept key inputs. TextArea is a multiline editor that resizes vertically to fit all characters.

font: font information for the text

placeholderText: text shown before the user enters any text

readOnly: if set to true, the field can't be edited, only viewed

text: already entered text



```
Ex.: TextField {
    width: parent.width – platformStyle.paddingLarge
platformStyle.paddingLarge
    font {
        family: fontFamilyRegular
lines\nof text"
        pixelSize: fontSizeMedium
    }
    placeholderText: "Type some text here"
    readOnly: false
    text: "This text is already entered"
}
```

```
TextArea {
    width: parent.width –
height: platformStyle.graphicSizeLarge
    placeholderText: "I accept many
    readOnly: true
text: "You cannot change me"
}
```

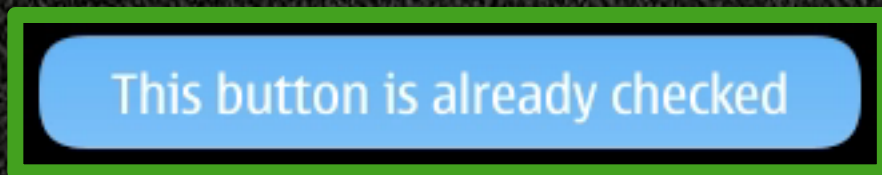

Button

checkable: if true, it becomes a toggle button with two states, 'on' and 'off'

checked: if true, it is toggled 'on'

font: font information for the text

text: the button label



Ex.:

```
Button {  
    width: parent.width - platformStyle.paddingLarge  
    anchors.horizontalCenter: parent.horizontalCenter  
    font {  
        family: fontFamilyRegular  
        pixelSize: fontSizeMedium  
    }  
    checkable: true  
    checked: true  
    text: "This button is already checked"  
}
```

See video: addon/module-010/videos/CS_Button.mov

ToggleButton

Different button, compatible with icons.

checkable: if true, it becomes a toggle button with two states, 'on' and 'off'

checked: if true, it is toggled 'on'

checkedIconSource: icon source for the checked state

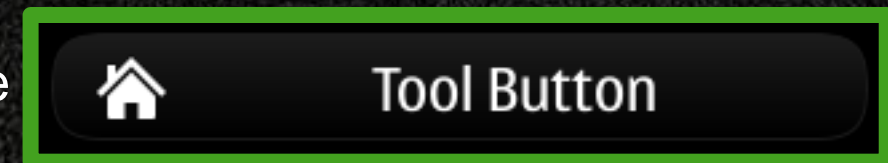
enabled: if false, the button is disabled

flat: if true, it has a flat look

iconSource: icon source for the normal state

pressedIconSource: icon source for the pressed state

text: the button label



Ex.:

```
ToggleButton {  
  width: parent.width - platformStyle.paddingLarge  
  anchors.horizontalCenter: parent.horizontalCenter  
  iconSource: "toolbar-home"  
  text: "Tool Button"  
}
```


ButtonRow / ButtonColumn

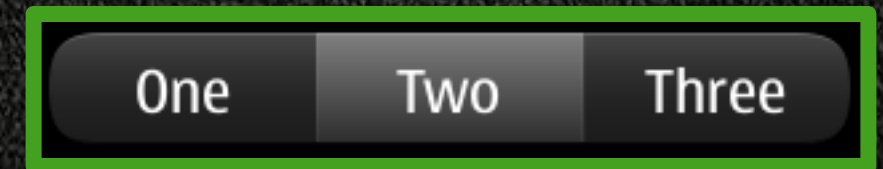
Button positioners.

checkedButton: defines which button is checked

exclusive: if true, only one button at a time is toggled

Ex.:

```
ButtonRow {  
  width: parent.width - platformStyle.paddingLarge  
  anchors.horizontalCenter: parent.horizontalCenter  
  checkedButton: b2  
  exclusive: true  
  
  Button { text: "One" }  
  Button { id: b2; text: "Two" }  
  Button { text: "Three" }  
}
```



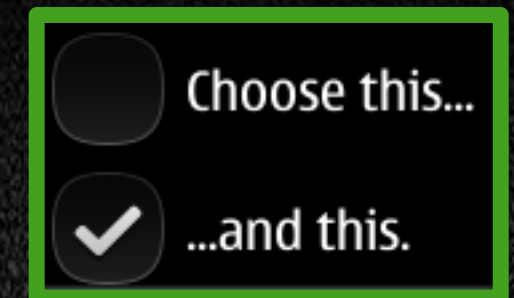
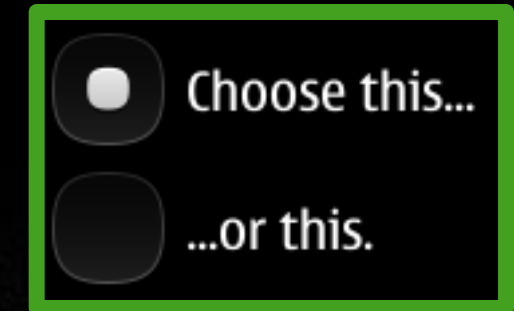
RadioButton / CheckBox

Button variations for specific purposes.

checked: if true, it is toggled 'on'

text: text shown beside the button

platformExclusiveGroup: exclusive group the RadioButton belongs to
(can be replaced with ButtonRow or ButtonColumn)



```
Ex.: CheckableGroup { id: group }
      Row {
        RadioButton {
          text: "Choose this..."
          platformExclusiveGroup: group
        }
        RadioButton {
          text: "...or this."
          platformExclusiveGroup: group
        }
      }
    }
```

```
ButtonColumn {
  spacing: 10
  exclusive: false

  CheckBox { text: "Choose this..." }
  CheckBox {
    text: "...and this."
    checked: true
  }
}
```


ToolBar



Container for quick access to common tasks

tools: the tools that make up the content of the toolbar

transition: type of transition (set / push / pop / replace)

ToolBarLayout

Automatically arranges items on the ToolBar

backButton: if set to 'false', the leftmost space will be empty

```
Ex.: ToolBar {  
    anchors.bottom: root.bottom  
    tools: ToolBarLayout {  
        ToolButton { iconSource: "toolbar-back" }  
        ToolButton { iconSource: "toolbar-settings" }  
        ToolButton { iconSource: "toolbar-menu" }  
    }  
}
```


Menu / ContextMenu

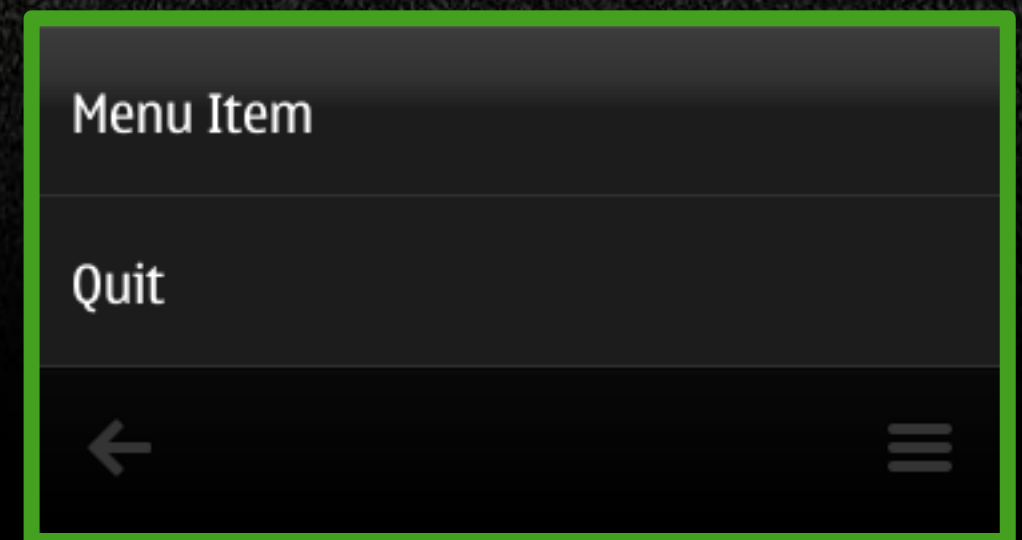
Provides a list of options the user can select from.
ContextMenu opens in a popup dialog.

content: where you call for the content

MenuLayout

Automatically arranges the MenuItem's in a Menu

```
Ex.: Menu {  
    content: MenuLayout {  
        MenuItem { text: "Menu Item" }  
        MenuItem {  
            text: "Quit"  
            onclicked: Qt.quit()  
        }  
    }  
}
```



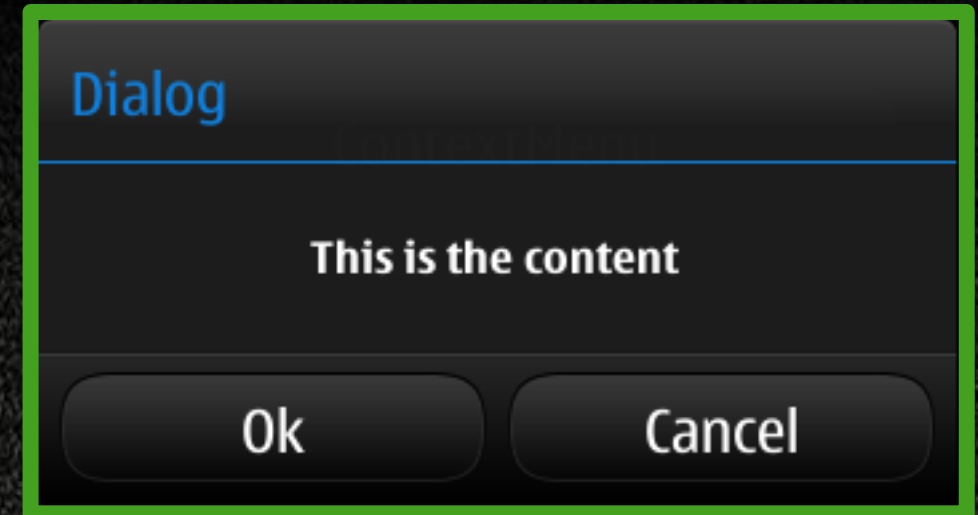
Dialog

Popup dialogs that float above the top layer of the application view

title: items for the title area of the dialog

content: items for the content area of the dialog

buttons: items for the buttons area of the dialog



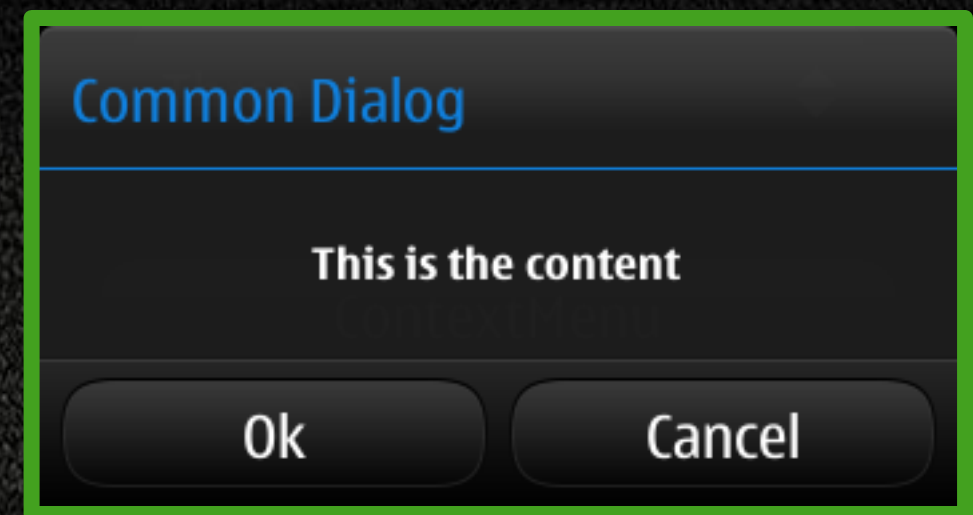
```
Ex.: Dialog {
    title: Text {
        anchors { verticalCenter: parent.verticalCenter; left: parent.left; leftMargin: 10 }
        text: "Dialog"
        font { family: platformStyle.fontFamilyRegular; pixelSize: platformStyle.fontSizeLarge }
        color: platformStyle.colorNormalLink
    }
    buttons: ToolBar {
        tools: Row {
            ToolButton { text: "Ok" }
            ToolButton { text: "Cancel" }
        }
    }
    content: Text {
        text: "This is the content"
        color: platformStyle.colorNormalLight
        font.pixelSize: 16
    }
}
```


CommonDialog

Dialog with predefined layout for the title

`titleText`: string for predefined title layout

```
Ex.: CommonDialog {
    titleText: "Common Dialog"
    buttons: ToolBar {
        width: parent.width
        tools: Row {
            anchors.center: parent
            spacing: platformStyle.paddingMedium
            ToolButton { text: "Ok" }
            ToolButton { text: "Cancel" }
        }
    }
    content: Text {
        text: "This is the content"
        color: platformStyle.colorNormalLight
        font.pixelSize: 16
        anchor.fill: parent
        horizontalAlignment: Text.AlignHCenter
        verticalAlignment: Text.AlignVCenter
    }
}
```



QueryDialog

CommonDialog with predefined layout for the icon, accept and reject buttons, and content message.

icon: predefined icon layout, beside the title

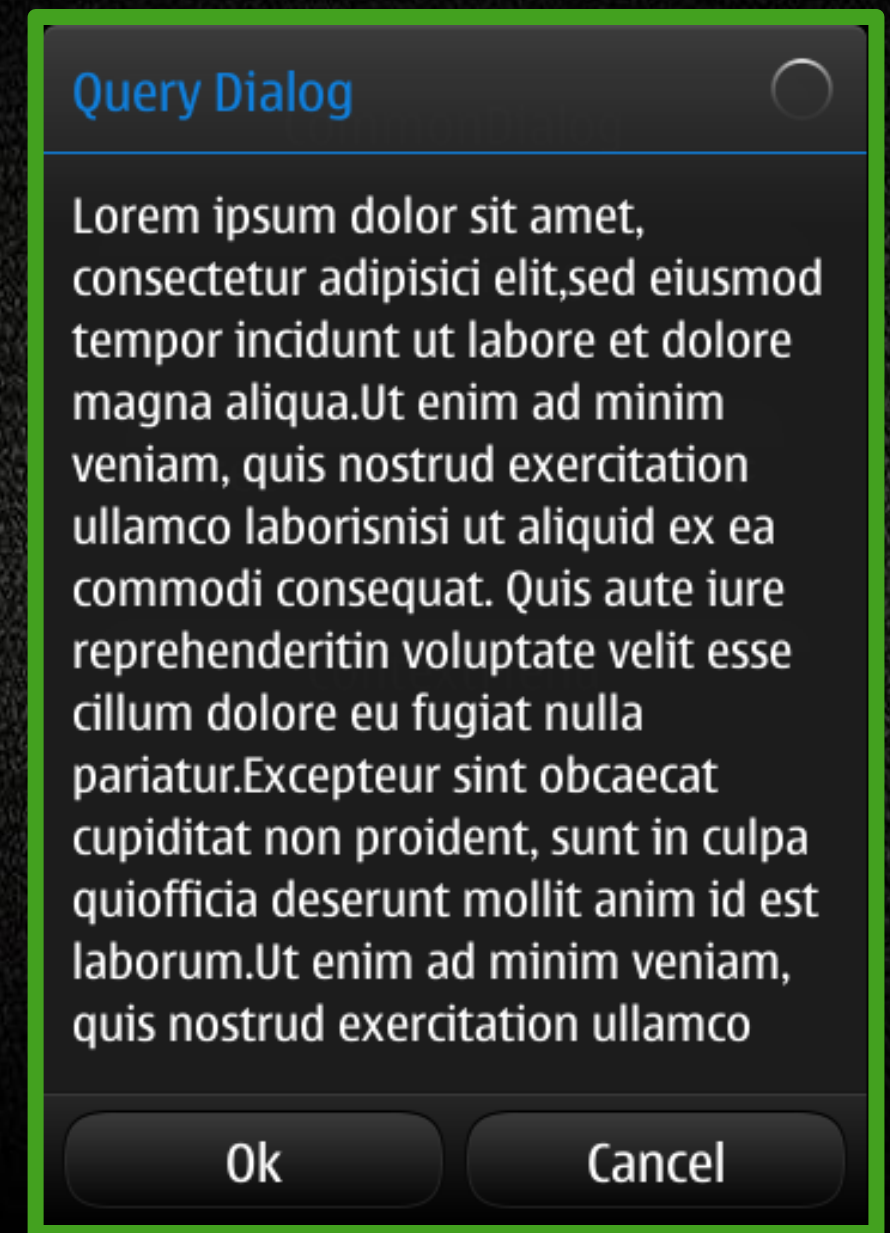
message: predefined content message layout

acceptButtonText: predefined button layout (left)

rejectButtonText: predefined button layout (right)

Ex.:

```
QueryDialog {
    titleText: "Query Dialog"
    message: "Lorem ipsum dolor sit amet, consectetur adipisicing "
            + "elit, sed eiusmod tempor incididunt ut labore et..."
}
acceptButtonText: "Ok"
rejectButtonText: "Cancel"
icon: "image:///theme/qtg_anim_spinner_large_1"
}
```



SelectionDialog

Provides a list of options the user can choose from

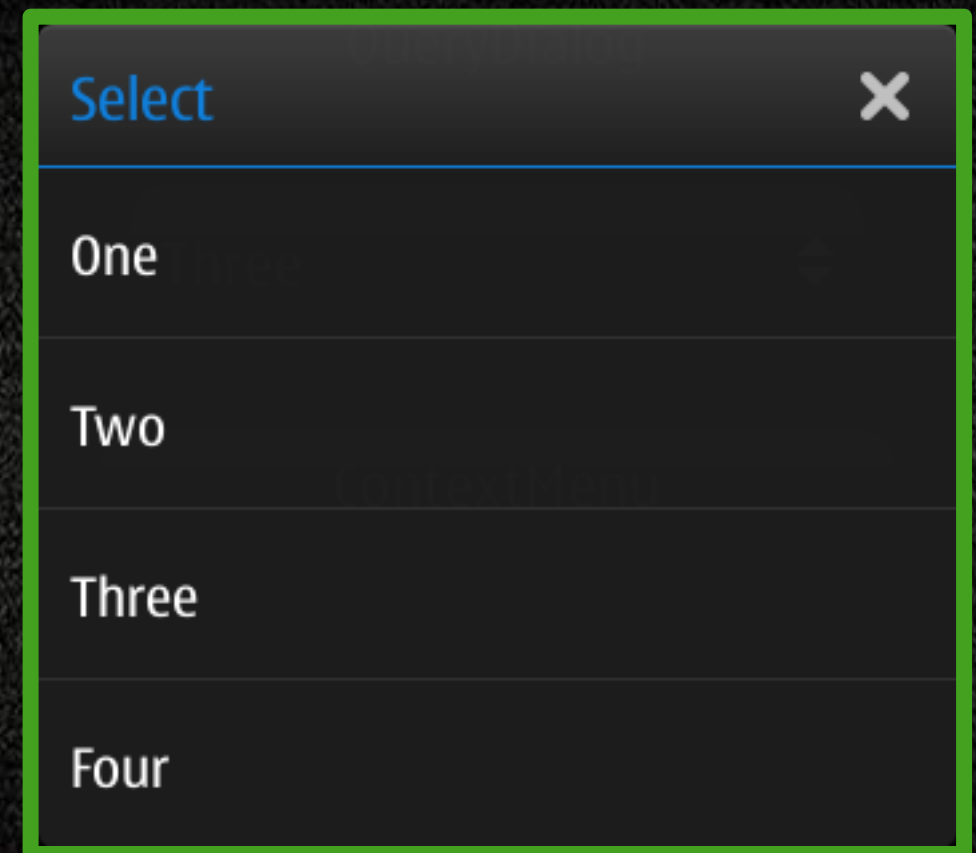
titleText: predefined title layout

selectedIndex: selected option

model: options

Ex.:

```
SelectionDialog {
  titleText: "Select"
  selectedIndex: 2
  model: ListModel {
    ListElement { name: "One" }
    ListElement { name: "Two" }
    ListElement { name: "Three" }
    ListElement { name: "Four" }
    ListElement { name: "Five" }
    ListElement { name: "Six" }
    ListElement { name: "Seven" }
    ListElement { name: "Eight" }
    ListElement { name: "Nine" }
  }
}
```



TabBarLayout

Positioner for Tab Buttons



TabBar

Functions the same way as TabBarLayout, but is themed in a Symbian device style



```
Ex.: TabBarLayout {
    anchors.bottom: root.bottom
    width: parent.width
    TabButton { text: "1" }
}

    TabButton { iconSource: "toolbar-home" }
    TabButton { text: "3" }
}
```

```
TabBar {
    anchors.bottom: root.bottom
    TabButton { text: "1" }
    TabButton { iconSource: "toolbar-home" }

    TabButton { text: "3" }
}
```


TabButton

Has an image or label and can be clicked to activate a tab

checked: if button corresponds to the activated tab

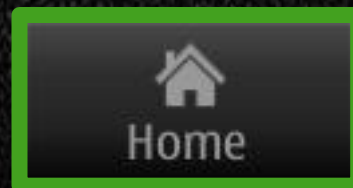
iconSource: image shown on the button

tab: indicates the tab content (child of TabGroup) that is activated when the TabButton is clicked

text: text to be displayed on the button

Ex.:

```
TabButton {  
    tab: tab1Content  
    text: "Home"  
    iconSource: "toolbar-home"  
    checked: false  
}
```



TabGroup

A set of pages for a tabbed interface

```
Ex.:  TabGroup {
      anchors: { top: toolBar.bottom; bottom: tabBar.top; left: parent.left; right: parent.right }

      Button {
        id: tab1content
        text: "tab1"
      }
      Text {
        id: tab2content
        text: "tab2"
        horizontalAlignment: "AlignHCenter"
        verticalAlignment: "AlignVCenter"
        color: platformStyle.colorNormalLight
      }
      Page {
        id: tab3content
        CheckBox {
          anchors.centerIn: parent
          text: "tab3"
        }
      }
    }
```


Page

Defines one screen of the interface content

tools: defines the tools for the page

```
Ex.: Page {
    Text {
        id: label
        anchors.centerIn: parent
        text: "Page 1"
        color: "white"
    }

    tools: ToolBarLayout {
        ToolButton {
            flat: true
            iconSource: "toolbar-back"
        }
        ToolButton {
            flat: true
            text: "Page 2"
        }
    }
}
```


PageStack

Defines a container for pages and a stack-based navigation model

currentPage: the page in the stack that is currently visible

toolBar: toolbar container for the tools associated with each page (if toolbar is null, no tools are shown to the user even if a page has tools)

```
Ex.: PageStack {
    id: pageStack
    anchors { left: root.left; top: root.top; right: root.right; bottom: mainToolBar.top }
    toolBar: mainToolBar
}

ToolBar {
    id: mainToolBar
    anchor.bottom: parent.bottom
}
```


Topics

1 Qt Components

2 Cheat Sheet

3 Questions

4 Lab

Questions

How can you retrieve Symbian platform specific styles?

How does Menu work?

What's the difference between TabBar and TabBarLayout?

What's the difference between Page and Component?

When would you use QueryDialog, CommonDialog and Dialog?

Topics

1 Qt Components

2 Cheat Sheet

3 Questions

4 Lab

Lab

Build a 3-tabbed app using the provided cheat sheet as reference

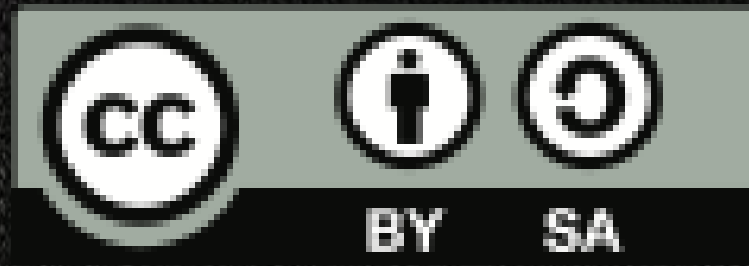


See video: [addon/module-010/videos/lab.mov](#)

See lab: [addon/module-010/labs/lab-10](#)

(c) 2011 Nokia Corporation and its Subsidiary(-ies).

The enclosed Qt Training Materials are provided under the Creative Commons Attribution ShareAlike 2.5 License Agreement.



The full license text is available here: <http://creativecommons.org/licenses/by-sa/2.5/legalcode>

Nokia, Qt and the Nokia and Qt logos are the registered trademarks of Nokia Corporation in Finland and other countries worldwide.