# Lesson 13 – XUI Views

* What is a library.
* XUI library.
* Dialogs
* Templates

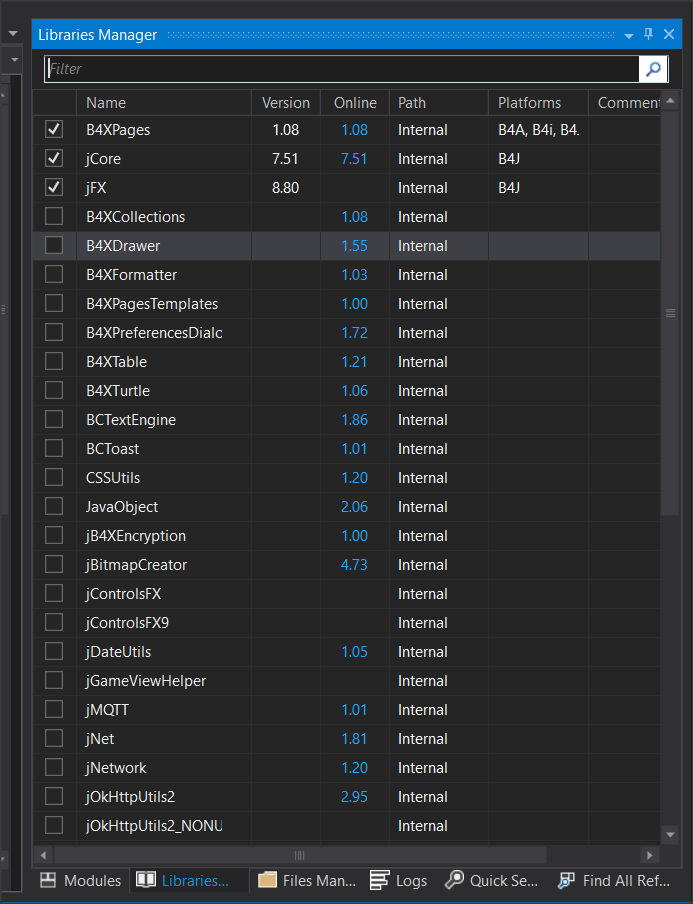
What students should know

**3h**

Each programming language has libraries. In general, a library includes collections of code, structures, classes, methods that can be used by developers to facilitate the process of developing a program.

## Libraries in B4X

Libraries are generally divided into internal libraries and are those that are installed together with the language and external ones created by the developer himself or found from other sources (e.g. Github). To use a library, you only need to select it from the corresponding Libraries list in B4X.



Picture Libraries

The list of libraries contains useful information. These are:

* the current version of the library that is installed,
* The latest version published in order to update our language.
* Whether it is an internal library or an external library.
* Which platforms it concerns.

To use a library, you must have been informed about the functions it performs as well as the data and methods it uses. For each library there is relevant information about its use through the language website at <https://www.b4x.com/android/documentation.html>.

External libraries should always be copied all in a specific folder on your computer. From Menu “Tools”, “Configure Paths”, “Additional Libraries” select the folder in which they will be stored.

**Teacher’s tip**

Below are some important views. Other very important ones are absent from this list, such as ComboBox, which we will refer to in more extensively in the next chapters.



## The XUI Views library

The purpose of the XUI Views library is to offer a common way to create applications for b4j, b4a and b4i.

This library is associated with creating forms and views that are included in them. Includes:

* **Views** (objects)
  + B4XComboBox - Cross platform ComboBox / Spinner / ActionSheet.
  + ScrollingLabel - A label that scrolls the text when it is wider than the label.
  + AnotherProgressBar - Vertical or horizontal animated progress bar.
  + B4XLoadingIndicator - 6 different animated loading indicators.
  + RoundSlider - A round slider.
  + SwiftButton - 3d button
  + AnimatedCounter
  + B4XFloatTextField - A TextField / EditText with a floating hint
  + B4XSwitch - Nice looking two state control.
  + B4XPlusMinus - Allows the user to select a number or item from a previously set list.
  + B4XBreadCrumb - Navigation control.
  + B4XSeekBar - Horizontal or vertical seek bar / slider.
  + MadeWithLove - Show your love to B4X :)
  + B4XImageView - ImageView with useful resize modes.
  + XUIViewsUtils - Static code module with various utility methods.
* **Dialogs** (Dialog Boxes)
  + A class that provides the features required to show a dialog. There are three methods to show dialogs: Show - Shows a simple dialog with text, ShowCustom - Allows you to pass a layout of your own and show it as a dialog, ShowTemplate - Shows a dialog based on a template class. See the source code for the template structure. It is quite simple.
* **Templates**
  + B4XDateTemplate - Based on AnotherDatePicker.
  + B4XColorTemplate - Color picker.
  + B4XLongTextTemplate - Scrollable text.
  + B4XListTemplate - A list of items. The user can choose one of the items.
  + B4XSignatureTemplate - Captures the user signature and adds a timestamp to the bitmap.
  + B4XInputTemplate - Template for text and numeric inputs.
  + B4XSearchTemplate - A list with a search field.
  + B4XTimedTemplate - A template that wraps other templates and creates a dialog that closes automatically after the set time with a nice, animated progress bar.

## Using XUI Views

To use the XUI Views first you need to check her name on the library tab. Then go to the Designer to create the objects you want.

**Example**

The following program shows the use of:

**ScrollingLabel**

**B4XFloatTextField**

**RoundSlider**

**AnotherProgressBar**

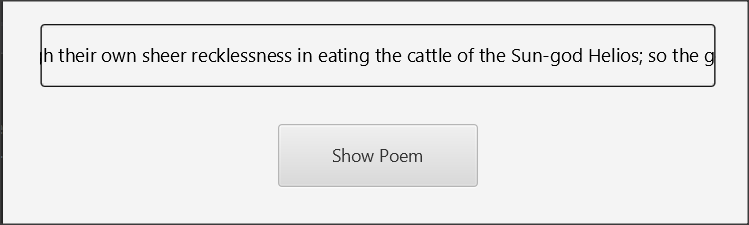
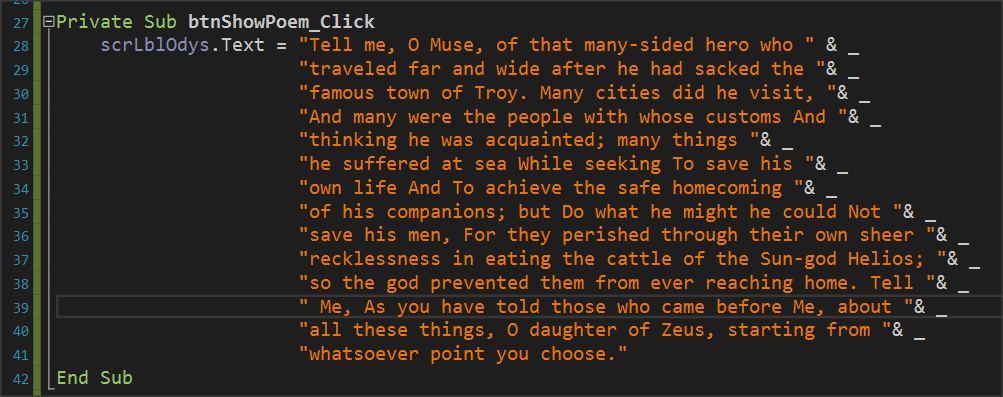
**B4XSwitch**

**B4XImageView**



### Scrolling Label

Scrolling label is a text label where it can move text so that it can appear in its entirety.

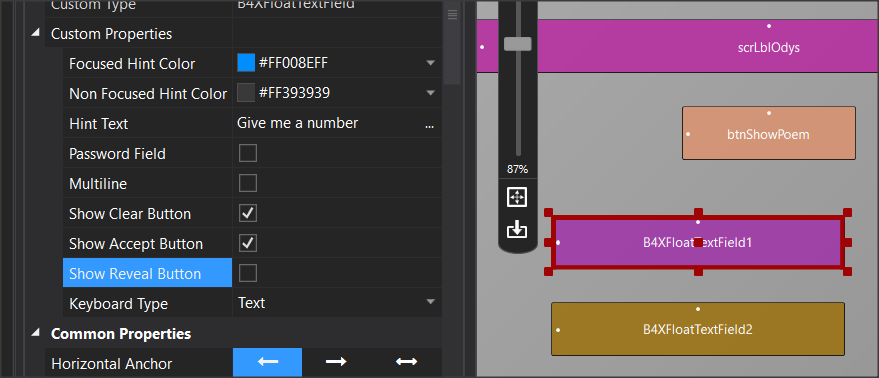


Picture Scrolling Label

It is mainly used for very large texts or to display a running message on the screen.

### B4XFloatTextField

B4XFolatTextField creates a text box on the screen to insert data. Its use is similar to the simple text box, but in addition it displays a label that helps the user identify the field without inserting an additional label before the field ( Hint Text).

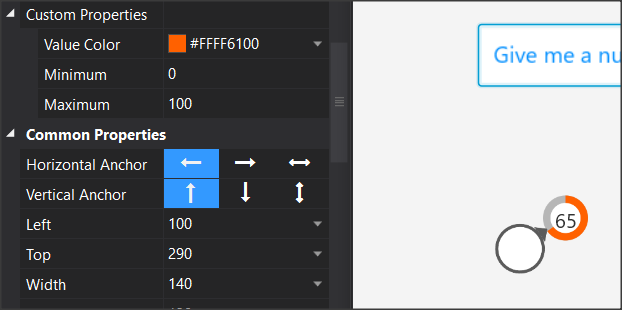


Picture B4XFloatTeextField

Additionally, it displays the appropriate controls to delete or enable this field (Show Clear Button, Show Accept button). Finally, it allows the Keyboard Type property to enter text, either integer numbers or decimal numbers.

### RoundSlider

RoundSlider is a controller that moves with the mouse around a circle. Each point in the cycle, it displays a value from a range of values that you have selected in the Minimum – Maximum properties in Designer.

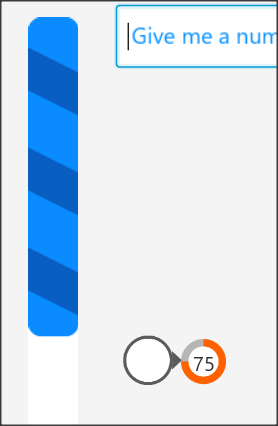
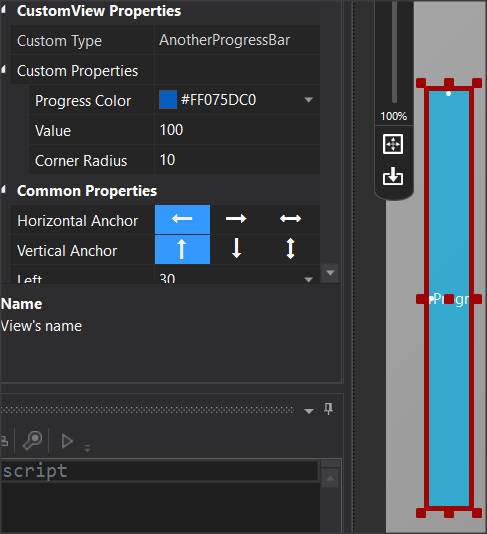
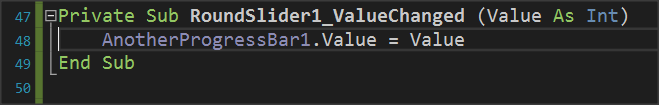


Picture RoundSlider

Each time the RoundSlider value changes, the "\_ValueChanged" event is triggered, and gives value of which the developer can use.

### AnotherProgressBar

AnotherProgressBar displays a bar that vary between 0 and 100 and is suitable for displaying percentages or other proportions. Using it is simple, after you declare the variable that corresponds to the item, use the property ".value" to set a value.

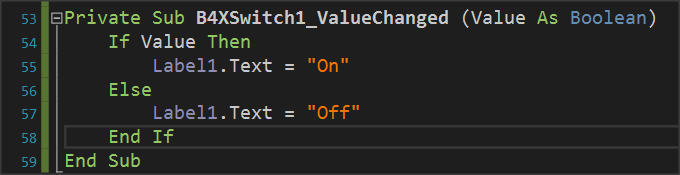
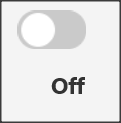


Picture AnotherProgressBar

In the example of the image, progressBar vary depending on the value it receives from RoundSlider.

B4XSwitch

B4XSwitch creates a sliding button on the screen.

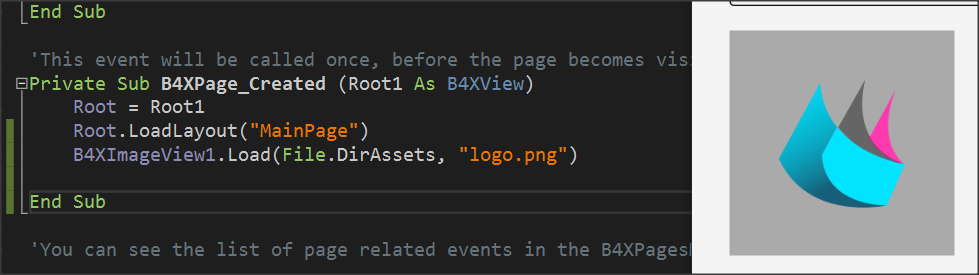
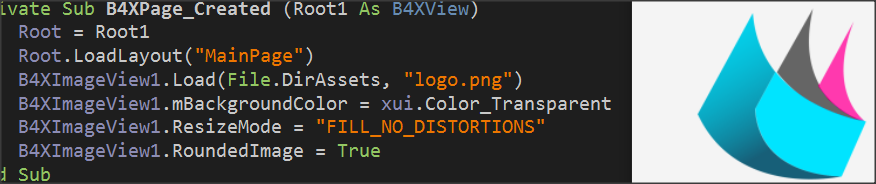


Picture B4XSwitch

In the example of the image when the returned value is true then a label displays the text "On" otherwise displays "Off".

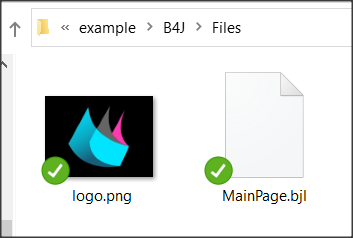
### B4XImageView

B4XImageView displays an image. Provides the ability to choose how the image is displayed either by code or by the object's options in Designer.



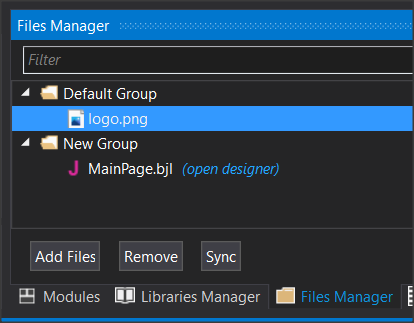
Picture B4XImageView

In the Load command, in addition to the image name, the folder where the image is stored must be declared first. In this case, the Files (Files.DirAsset) folder is declared as in the image below:



Picture Files Folder (DirAsset)

This folder is not accessible for writing new files when the application is created. In the course of the files, we will refer in greater detail to the folders.



Picture Files Manager

To display an image stored in the Files folder, it must also be declared on the File Manager tab of the language.

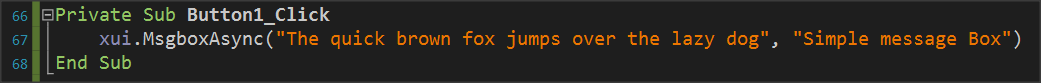
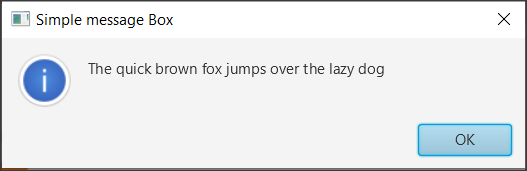
## B4XDialogs

Dialog boxes are screens that appear to inform the user of an event or to get data from users without the need to create another B4XPage.

### MsgBoxAsync

Already in previous courses you used the command xui.MSgBoxAsync which displays a simple message on the screen.

Picture 10 MsgBoxAsync

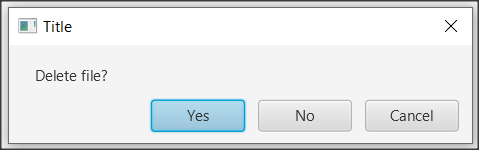
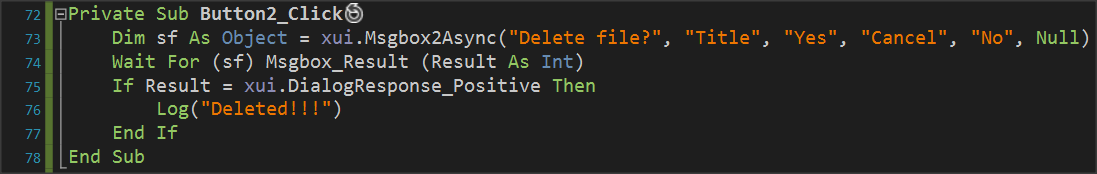


As shown in Picture 10 window. The command also displays an OK button to close the message and return to the program.

### MsgBox2Async

Using MsgBox2Async is more complex and allows the programmers to display a message and then choose between three different functions by pressing a corresponding button.

Picture 11 MsgBox2Async



MsgBox2Async returns an object (“sf” in the example) that waits for a response event from user.

* **DialogResponse\_Positive** (if “Yes” was pressed in the example)
* **DialogResponse\_ Negative** (if " No" key was pressed)
* **DialogResponse\_Cancel** (if "Cancel" was pressed)

The above values are controlled by an if command and appropriate actions are performed accordingly.

The words that appear on the action buttons can be changed but always have the same order as a “Positive”, “Cancel”, “Negative”.

You can skip a button simply by writing a blank string "" at the corresponding point when the button will not appear in the dialog box.

The last parameter allows you to display an icon to the left of the message, and Null displays nothing.

### Wait For

Wait For freezes the execution of an operation until an event is activated.

***Wait For*** (<sender>) <event signature>

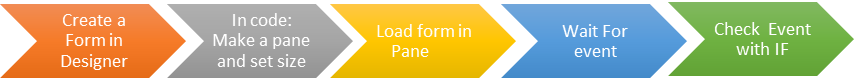
Where **sender** is the object for which an event is waiting to be activated and

**Event signature** the event that was activated where in the case of MsgBox2 Async are the DialogResponse\_Positive, DialogResponse\_Cancel, DialogResponse\_Negative.

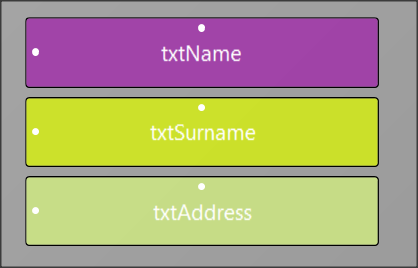
The event is then checked with an if command.

## CustomDialog

Creating a Custom dialog box includes several steps that start with the Designer.



Picture 12 Steps to create Custom Dialog.

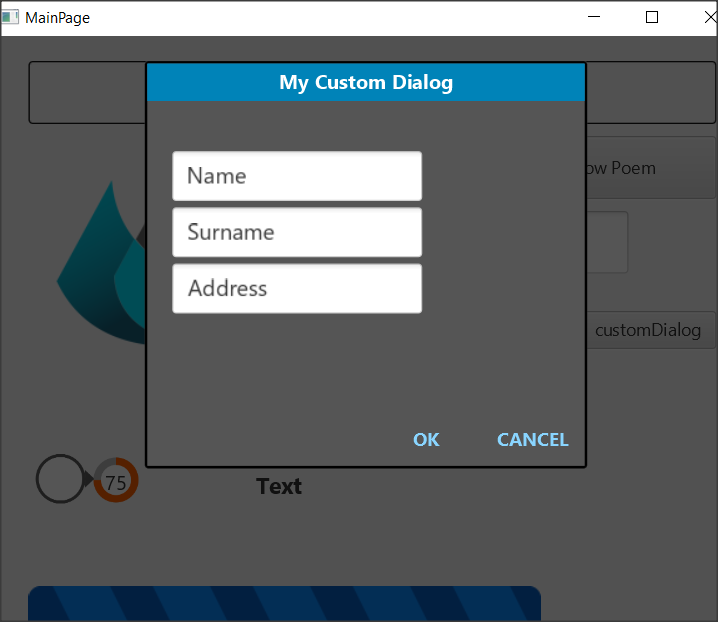
Continuing the first example create a new form in Designer and save under the name frmInsStudent. Also create corresponding variables for Text Fields from Generate Members.

Picture containing text, black, screenshot, silver

Automatically created description

Picture 13 Custom Dialog

1. Set an object named dialog type B4XDialog within globalSub, and then use the command “dialog.Initialize(Root)” to initialize it. “dialog” sets a title in the dialog box that you are about to create.
2. Set a pane in which to display the items on the form you designed and set its dimensions in pixels.
3. Load your form with the “p.LoadLayout” command where “p” the pane you set before and then set to appear at the top above all other windows.
4. Wait for a button to click and check the results.



Picture 14 The dialog box

## Templates

The B4X has ready-made templates for creating dialog boxes, some of which are:

* B4XDateTemplate
* B4XColorTemplate
* B4XLongTextTemplate

**Teacher’s** **tip.**

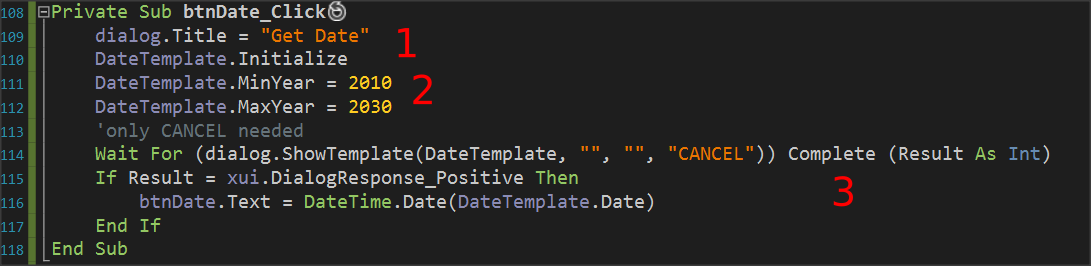
**Additional functions will be used in next chapters as concepts such as lists, dictionaries, etc. have not yet been discussed.**



### B4XDateTemplate

Creates a Dialog box to select a date. This box only needs a cancel button because when a date is selected it automatically returns the relevant value to the program. When the form opens, it has already marked the current date in a different color.

B4XDateTemplate based on a dialogue template that should be declared and initialised as in the Custom Dialogs



Picture 15 Create Date Dialog Box

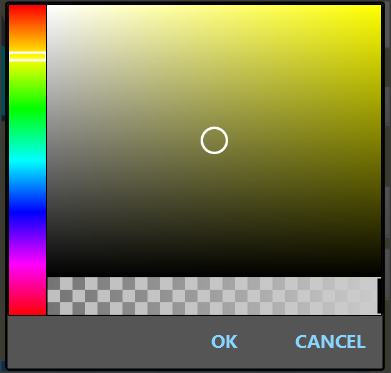
1. Initialize and set a title in the dialog. In the example initialization done in Class\_Globals Sub.
2. Specify whether you want years limits.
3. Wait for the date to be selected, and if selected, display it as a name on the “btnDate” or use it according to your needs.

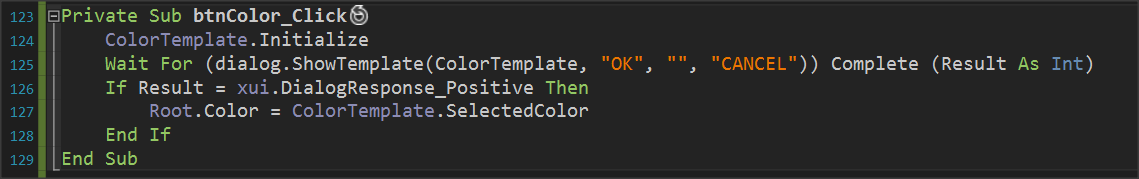
**Remember**

The date is returned to Ticks which represent milliseconds from 1/1/1970. These milliseconds are converted to a date with the “DateTime.Date” command.



### B4XColorTemplate

B4XColorTamplate is similar to that of the day. Once again the code is waiting with the Wait For command to make a color selection, and when pressed OK returns a color number.

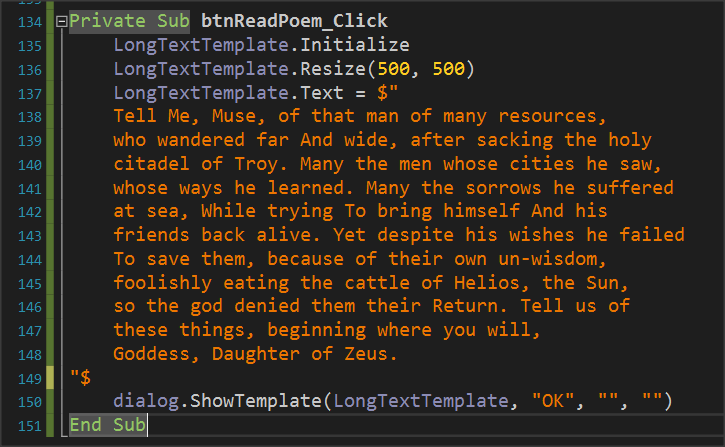
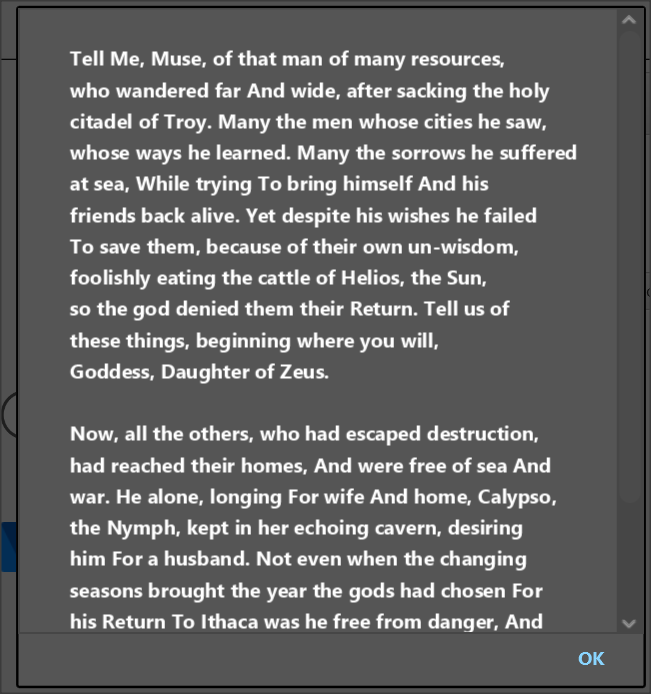


Picture Create Color Template

The example uses the color to change the background of the home page with the “Root.Color”.

### B4XLongTextTemplate

B4XLongTextTemplate displays a window on the screen to display long text. Also, it displays a navigation bar to scroll the text.



Picture LongTextTemplate

As in previous templates, set a variable of type B4XLongTextTemplate, and then after you initialize it you can set a size for the window that you are going to create with the “Resize” command.

Finally, set the Text property to the text you want to display and call “ShowTemplate”.

## Exercises

1. Create a program that asks for two dates between 2000 and 2021 and shows the distance between them in days, months and years.

Tip:

DateUtils.PeriodBetween(date1, date2).Days

DateUtils.PeriodBetween(date1, date2).Months

DateUtils.PeriodBetween(date1, date2).Years

1. Create a program that reads the following customer information:
   1. Name
   2. Surname
   3. Phone
   4. Phone Account (in € )

Then create a button on the form that says payment and if pressed display the message "Do you want to pay?" If answered yes set 0 to the variable corresponding to the Phone Account.

1. Create a program that for 5 books displays a summary of them in an appropriate window. Books must be selected from an equal number of buttons with book’s images on them. The summaries and images of 5 books can be found in the supporting material of the exercise.
2. Create a form that contains the following items:
   1. Name
   2. Surname
   3. Phone

It also includes a switch that when the form is open has a light\_Gray background color with text fields in White while when closed it has dark gray background with light\_Gray text Fields. Also include a label to say day or night with black or white letters each.

*Tip: Use* ***JFX library*** *to set floatTextFields colors*

**Private** fx **As** JFX ‘ *in Class\_Globals*

lblDayNight.TextColor = fx.Colors.Black ‘ *Where you want to change color*

1. A water tank is 5 meters height and has a base of 3x3 meters. Make a program that,
2. continues reads the level height in meters and display a bar, represent the height. Text field will not allow any texts but only decimals values.
3. If water height is above 5 meters show message "Danger Water Leak!".
4. If the level Is less than 0.5 meters display a message. "Warning there is not enough water in the tank.
5. In every change show the total amount of water in cubic meters.