

FLEX - TEXTAREA CONTROL

http://www.tutorialspoint.com/flex/flex_textarea_control.htm

Copyright © tutorialspoint.com

Introduction

The TextArea control is a text-entry control that allows users enter and edit multiple lines of formatted text.

Class declaration

Following is the declaration for **spark.components.TextArea** class:

```
public class TextArea
    extends SkinnableTextBase
```

Public properties

S.N.	Properties & Description
1	content : Object This property is intended for use in MXML at compile time; to get or set rich text content at runtime, use the textFlow property instead.
2	heightInLines : Number The default height of the control, measured in lines.
3	textFlow : flashx.textLayout.elements.TextFlow The TextFlow representing the rich text displayed by this component.
4	widthInChars : Number The default width of the control, measured in em units.

Public methods

S.N.	Method & Description
1	TextArea Constructor.
2	getFormatOfRange <i>requestedFormats: Vector. < String >= null, anchorPosition: int = - 1, activePosition: int = - 1</i> :flashx.textLayout.formats.TextLayoutFormat Returns a TextLayoutFormat object specifying the computed formats for the specified range of characters.

3

scrollToRange*anchorPosition: int = 0, activePosition: int: void*

Scrolls so that the text range is visible in the container.

4

setFormatOfRange

format: flashx.textLayout.formats: TextLayoutFormat, anchorPosition: int = - 1, activePosition: int = - 1: void

Applies the specified formats to each element in the specified range that correspond to the given format.

Methods inherited

This class inherits methods from the following classes:

- spark.components.supportClasses.SkinnableTextBase
- spark.components.supportClasses.SkinnableComponent
- mx.core.UIComponent
- mx.core.FlexSprite
- flash.display.Sprite
- flash.display.DisplayObjectContainer
- flash.display.InteractiveObject
- flash.display.DisplayObject
- flash.events.EventDispatcher
- Object

Flex TextArea Control Example

Let us follow the following steps to check usage of TextArea control in a Flex application by creating a test application:

Step	Description
1	Create a project with a name <i>HelloWorld</i> under a package <i>com.tutorialspoint.client</i> as explained in the <i>Flex - Create Application</i> chapter.
2	Modify <i>HelloWorld.mxml</i> as explained below. Keep rest of the files unchanged.
3	Compile and run the application to make sure business logic is working as per the requirements.

Following is the content of the modified mxml file **src/com.tutorialspoint/HelloWorld.mxml**.

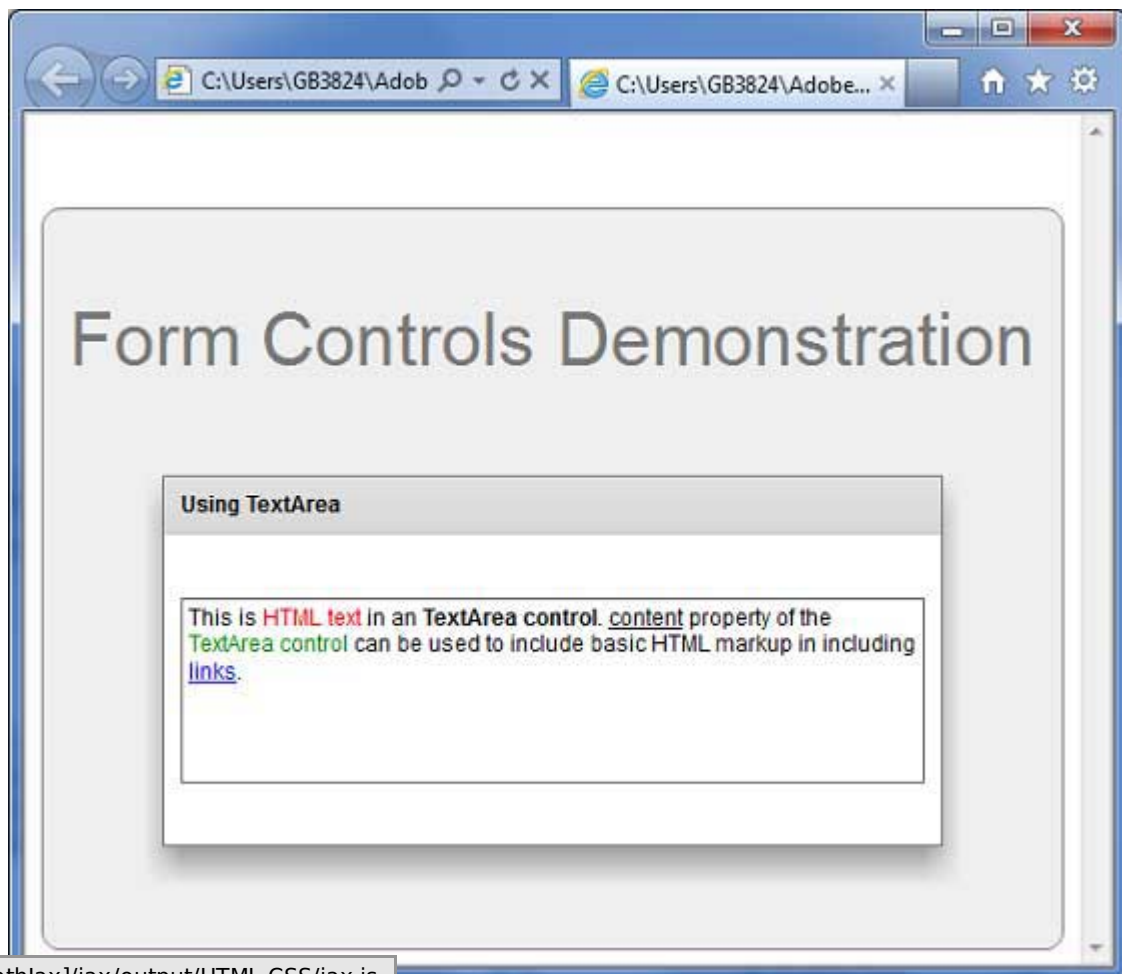
```
<?xml version="1.0" encoding="utf-8"?>
<s:Application xmlns:fx="http://ns.adobe.com/mxml/2009"
  xmlns:s="library://ns.adobe.com/flex/spark"
  xmlns:mx="library://ns.adobe.com/flex/mx"
  width="100%" height="100%" minWidth="500" minHeight="500"
  >
  <fx:Style source="/com/tutorialspoint/client/Style.css"/>
```

```

<s:BorderContainer width="550" height="400"
styleName="container">
  <s:VGroup width="100%" height="100%" gap="50"
horizontalAlign="center" verticalAlign="middle">
    <s:Label
fontSize="40" color="0x777777" styleName="heading"/>
    <s:Panel
width="420" height="200">
      <s:layout>
        <s:VerticalLayout gap="10" verticalAlign="middle"
horizontalAlign="center"/>
      </s:layout>
      <s:TextArea width="400" height="100">
        <s:content>
          This is <s:span color="#FF0000">HTML text</s:span>
          in an <s:span fontWeight="bold">TextArea control</s:span>.
          <s:span textDecoration="underline">content</s:span> property
          of the <s:span color="#008800">TextArea control</s:span>
          can be used to include basic HTML markup in including
          <s:a href="http://www.tutorialspoint.com"
target="_blank">links</s:a>.
        </s:content>
      </s:TextArea>
    </s:Panel>
  </s:VGroup>
</s:BorderContainer>
</s:Application>

```

Once you are ready with all the changes done, let us compile and run the application in normal mode as we did in [Flex - Create Application](#) chapter. If everything is fine with your application, this will produce following result: [[Try it online](#)]



Loading [Mathjax]/jax/output/HTML-CSS/jax.js