

FLEX - BORDERCONTAINER

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

Copyright © tutorialspoint.com

Introduction

The BorderContainer class defines a set of CSS styles that control the appearance of the border and background fill of the container.

Class declaration

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

```
public class BorderContainer
    extends SkinnableContainer
```

Public properties

S.N.	Property & Description
1	backgroundFill : IFill Defines the background of the BorderContainer.
2	borderStroke : IStroke Defines the stroke of the BorderContainer container.

Public methods

S.N.	Method & Description
1	BorderContainer Constructor.

Methods inherited

This class inherits methods from the following classes:

- spark.components.SkinnableContainer
- spark.components.supportClasses.SkinnableContainerBase
- 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 BorderContainer Example

Let us follow the following steps to check usage of BorderContainer 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"/>
  <fx:Script>
    <![CDATA[
      private function applyBorderContainerProperties():void
      {
        borderContainer.setStyle("borderWeight",weightStepper.value);
        borderContainer.setStyle("borderColor",colorPicker.selectedColor);
        borderContainer.setStyle("dropShadowVisible",chkShadow.selected);
      }
    ]]>
  </fx:Script>
  <s:BorderContainer width="630" height="480"
    styleName="container">
    <s:VGroup width="100%" height="100%" gap="50"
      horizontalAlign="center" verticalAlign="middle">
      <s:Label
        fontSize="40" color="0x777777" styleName="heading"/>
      <s:Panel
        title="Using BorderContainer" width="500"
        height="300" includeInLayout="true" visible="true">
        <s:layout>
          <s:VerticalLayout gap="10" verticalAlign="middle"
            horizontalAlign="center"/>
        </s:layout>
        <s:layout>
          <s:HGroup>
            <s:VGroup>
              <s:HGroup verticalAlign="middle" >
                <s:Label text="Border Weight:" width="100"/>
                <s:NumericStepper />
              </s:HGroup>
              <s:HGroup verticalAlign="middle">
                <s:Label text="Border Color:" width="100"/>
                <mx:ColorPicker />
              </s:HGroup>
            <s:HGroup verticalAlign="middle">
```

```

        <s:Label text="Drop Shadow:" width="100"/>
        <s:CheckBox />
    </s:HGroup>
    <s:Button label="Apply Properties"
        click="applyBorderContainerProperties()"/>
</s:VGroup>
<s:BorderContainer
    backgroundColor="0xff99ff">
    </s:BorderContainer>
</s:HGroup>
</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](#)]

