

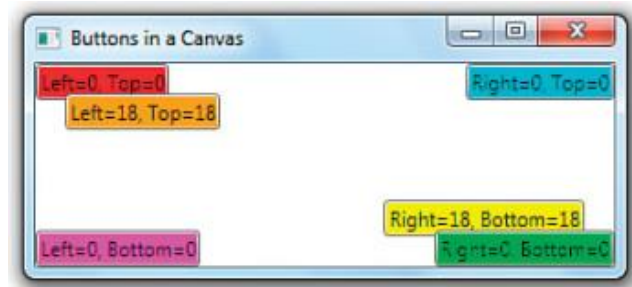
VISUAL PROGRAMMING

WPF Layouts

Zuhaib A. Shaikh,
Asst. Prof., CSE Deptt., QUEST
Web: zuhaib-shaikh.neocities.org

Introduction

- Several Layouts are available for forms
- Layout describe arrangement of elements.
- Helps to arrange GUI elements easily
- Canvas:
 - Basic layout
 - Elements can be attached from side of windows
 - Can only use two canvas properties



<Canvas>

<Button Background="Red">Left=0, Top=0</Button>

<Button Canvas.Left="18" Canvas.Top="18" Background="Orange">Left=18,
Top=18</Button>

<Button Canvas.Right="18" Canvas.Bottom="18" Background="Yellow">Right=18,
Bottom=18</Button>

<Button Canvas.Right="0" Canvas.Bottom="0" Background="Lime">Right=0,
Bottom=0</Button>

<Button Canvas.Right="0" Canvas.Top="0" Background="Aqua">Right=0, Top=0</Button>

<Button Canvas.Left="0" Canvas.Bottom="0" Background="Magenta">Left=0,
Bottom=0</Button>

</Canvas>

Introduction

- Stack Panel
 - Popular and simple
 - No attached properties
 - An element can be stretched to particular angle

<StackPanel>

<Label>Vertical stack elements from top to bottom</Label>

<Button Background="Red">Button 1</Button>

<Button Background="Orange">Button 2</Button>

<Button Background="Yellow">Button 3</Button>

<Button Background="Lime">Button 4</Button>

<Button Background="Aqua">Button 5</Button>

<Button Background="Magenta">Button 6</Button>

</StackPanel>

<StackPanel Orientation="Horizontal">

<Label>Horizontal stack elements from left to right</Label>

<Button Background="Red">Button 1</Button>

<Button Background="Orange">Button 2</Button>

<Button Background="Yellow">Button 3</Button>

<Button Background="Lime">Button 4</Button>

<Button Background="Aqua">Button 5</Button>

<Button Background="Magenta">Button 6</Button>

</StackPanel>

Vertical stacks elements from top to bottom.

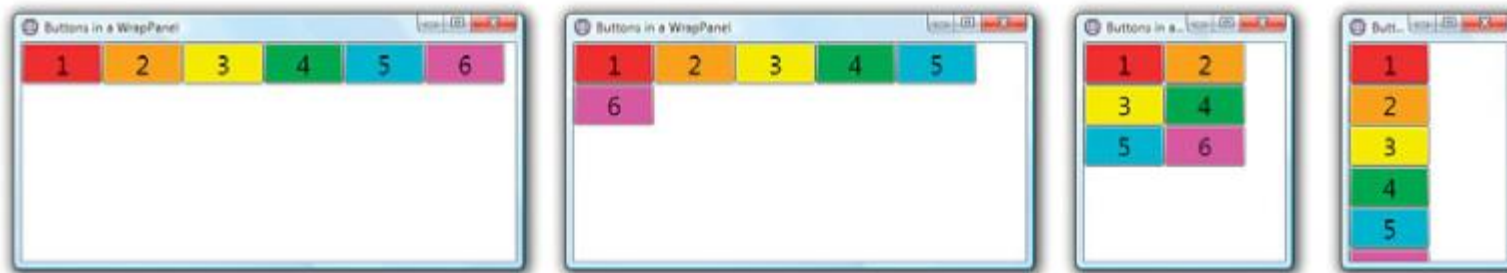


Horizontal stacks elements from left to right.



Introduction

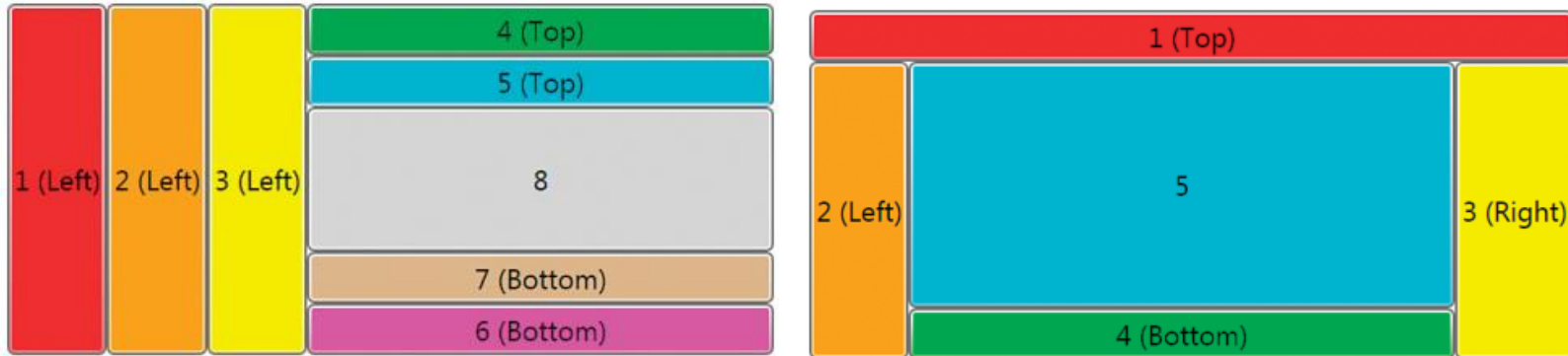
- Wrap Panel
 - Similar to Stack Panel, but have child elements in particular stack



```
<WrapPanel>  
  <Button Background="Red">Button 1</Button>  
  <Button Background="Orange">Button 2</Button>  
  <Button Background="Yellow">Button 3</Button>  
  <Button Background="Lime">Button 4</Button>  
  <Button Background="Aqua">Button 5</Button>  
  <Button Background="Magenta">Button 6</Button>  
</WrapPanel>
```

Introduction

- Dock Panel:
 - Provide dock panel to entire window
 - Elements can be attached from side of windows
 - Can only use two canvas properties



```

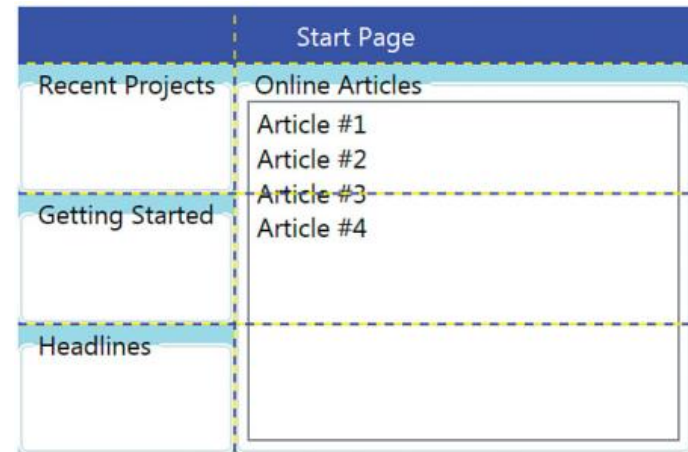
<DockPanel>
  <Button DockPanel.Dock="Top" Background="Red">1 (Top)</Button>
  <Button DockPanel.Dock="Left" Background="Orange">2 (Left)</Button>
  <Button DockPanel.Dock="Right" Background="Yellow">3 (Right)</Button>
  <Button DockPanel.Dock="Bottom" Background="Lime">4 (Bottom)</Button>
  <Button Background="Aqua">5</Button>
</DockPanel>

```

Introduction

- Grid Panel:
 - Default panel
 - divided into grid of rows and columns
 - Most versatile and often

```
<Grid Background="LightBlue">
<!-- Define four rows: -->
  <Grid.RowDefinitions>
    <RowDefinition/>
    <RowDefinition/>
    <RowDefinition/>
    <RowDefinition/>
  </Grid.RowDefinitions>
<!-- Define two columns: -->
<Grid.ColumnDefinitions>
  <ColumnDefinition/>
  <ColumnDefinition/>
</Grid.ColumnDefinitions>
<!-- Arrange the children: -->
<Label Grid.Row="0"
Grid.Column="0" Background="Blue"
Foreground="White"
HorizontalContentAlignment="Center"
">Start Page</Label>
```



```
<GroupBox Grid.Row="1" Grid.Column="0" Background="White"
Header="Recent Projects">...</GroupBox>
<GroupBox Grid.Row="2" Grid.Column="0" Background="White"
Header="Getting Started">...</GroupBox>
<GroupBox Grid.Row="3" Grid.Column="0" Background="White"
Header="Headlines">...</GroupBox>
<GroupBox Grid.Row="1" Grid.Column="1" Background="White"
Header="Online Articles">
<ListBox>
<ListBoxItem>Article #1</ListBoxItem>
<ListBoxItem>Article #2</ListBoxItem>
<ListBoxItem>Article #3</ListBoxItem>
<ListBoxItem>Article #4</ListBoxItem>
</ListBox>
</GroupBox>
</Grid>
```

Questions

