Mastering XAML Layouts

Thomas Claudius Huber

SOFTWARE DEVELOPER

@thomasclaudiush   www.thomasclaudiushuber.com

Module Outline

Know the XAML Layout Panels

Build a layout with the Grid

Set Attached Properties in

- XAML

- C#

**Use layout properties
to position elements**

Know the XAML Layout Panels

StackPanel   Grid   VariableSized WrapGrid   Relative Panel   Canvas

```xml
<StackPanel>
  <StackPanel.Children>
    <Rectangle Fill="LightBlue"
      Height="20" Margin="2"/>
    <Rectangle Fill="LightBlue"
      Height="20" Margin="2"/>
  </StackPanel.Children>
</StackPanel>
```
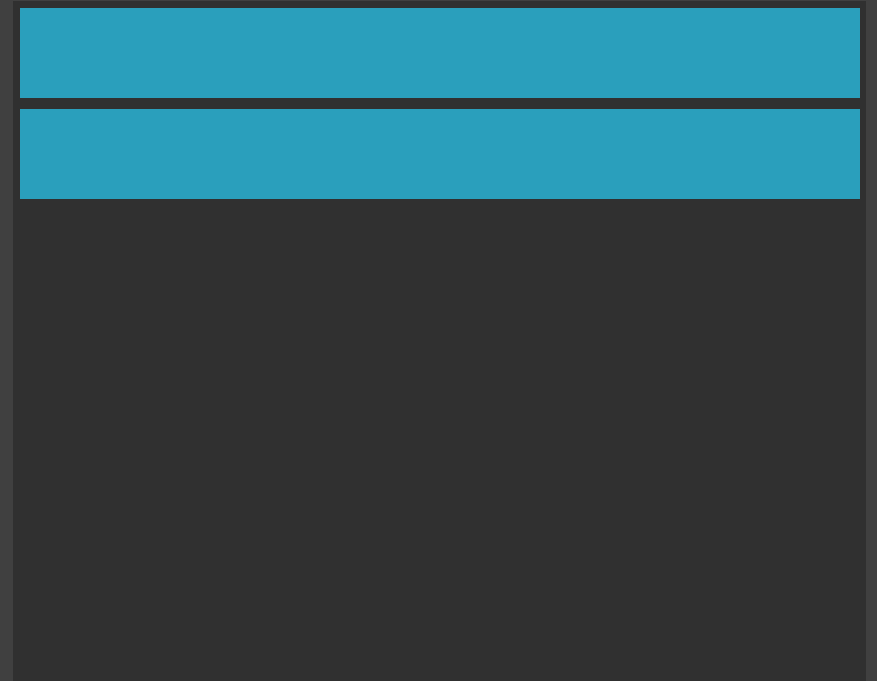
Know the XAML Layout Panels

StackPanel   Grid   VariableSized WrapGrid   Relative Panel   Canvas

```xml
<StackPanel>
  <Rectangle Fill="LightBlue"
    Height="20" Margin="2"/>
  <Rectangle Fill="LightBlue"
    Height="20" Margin="2"/>
</StackPanel>
```
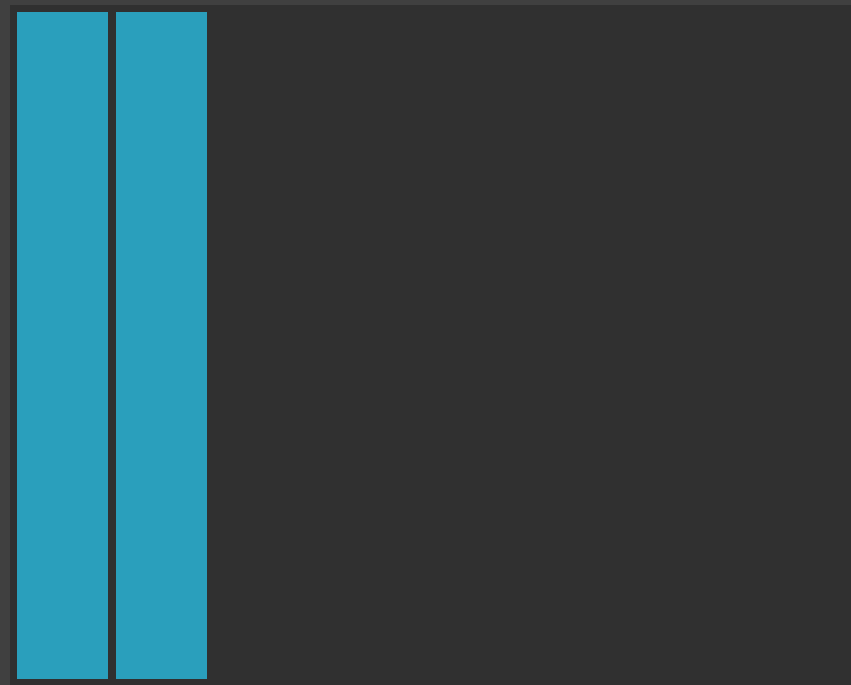
Know the XAML Layout Panels

StackPanel Grid VariableSized WrapGrid Relative Panel Canvas

```xml
<StackPanel Orientation="Horizontal">
  <Rectangle Fill="LightBlue"
    Width="20" Margin="2"/>
  <Rectangle Fill="LightBlue"
    Width="20" Margin="2"/>
</StackPanel>2
```

Know the XAML Layout Panels

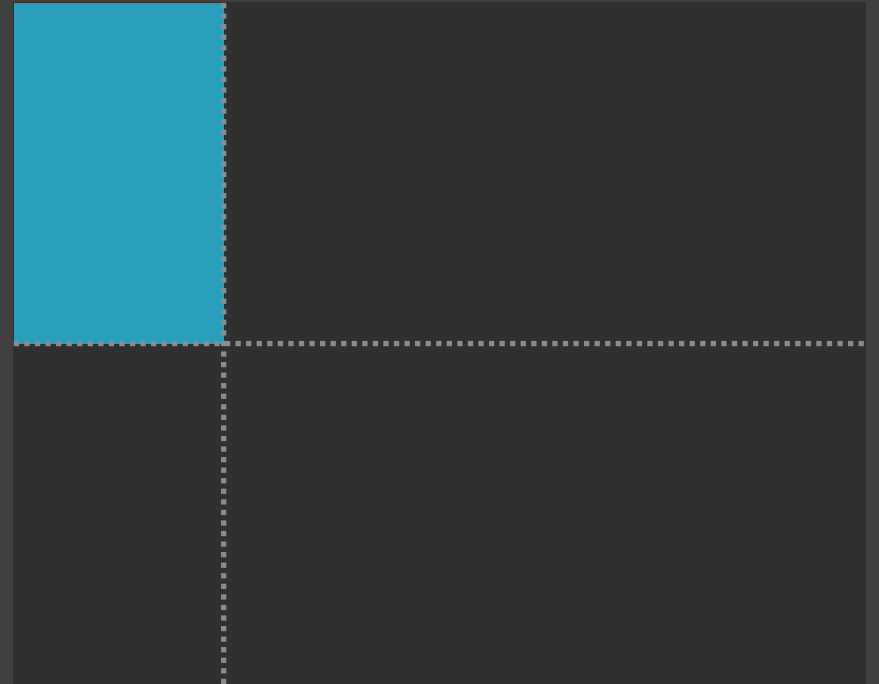StackPanel **Grid** VariableSized WrapGrid Relative Panel Canvas

```
<Grid>
  <Grid.RowDefinitions>
    <RowDefinition/>
    <RowDefinition/>
  </Grid.RowDefinitions>
  <Grid.ColumnDefinitions>
    <ColumnDefinition/>
    <ColumnDefinition/>
  </Grid.ColumnDefinitions>

</Grid>
```

Know the XAML Layout Panels

StackPanel   **Grid**   **VariableSized WrapGrid**   **Relative Panel**   **Canvas**

```xml
<Grid>
  <Grid.RowDefinitions>
    <RowDefinition/>
    <RowDefinition/>
  </Grid.RowDefinitions>
  <Grid.ColumnDefinitions>
    <ColumnDefinition Width="40"/>
    <ColumnDefinition/>
  </Grid.ColumnDefinitions>
  <Rectangle Fill="LightBlue"/>

</Grid>
```

Know the XAML Layout Panels

StackPanel   **Grid**   **VariableSized WrapGrid**   **Relative Panel**   **Canvas**

```xaml
<Grid>
  <Grid.RowDefinitions>
    <RowDefinition/>
    <RowDefinition/>
  </Grid.RowDefinitions>
  <Grid.ColumnDefinitions>
    <ColumnDefinition Width="40"/>
    <ColumnDefinition/>
  </Grid.ColumnDefinitions>
  <Rectangle Fill="LightBlue"
    Grid.Column="1"/>
</Grid>
```
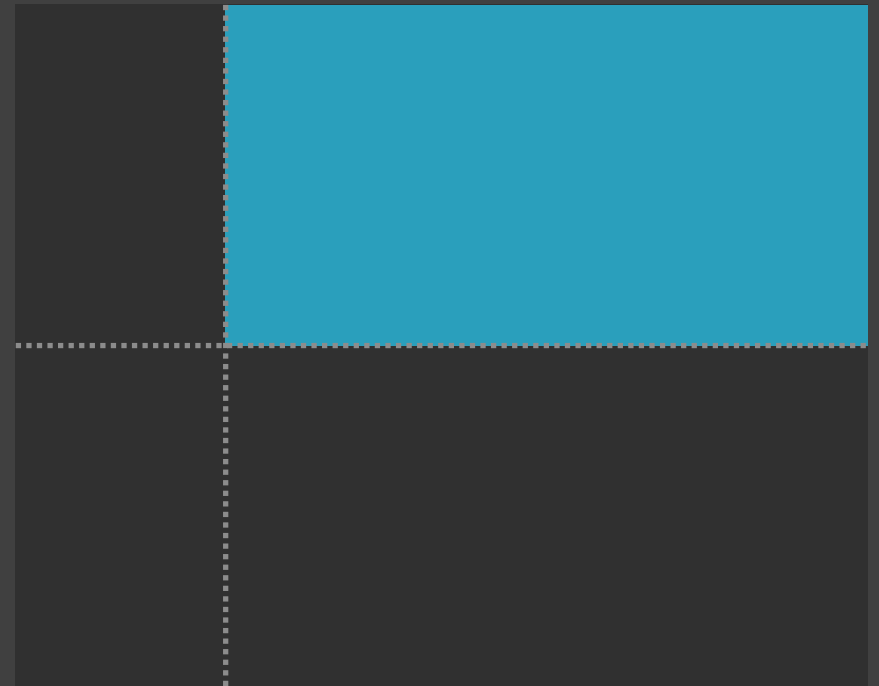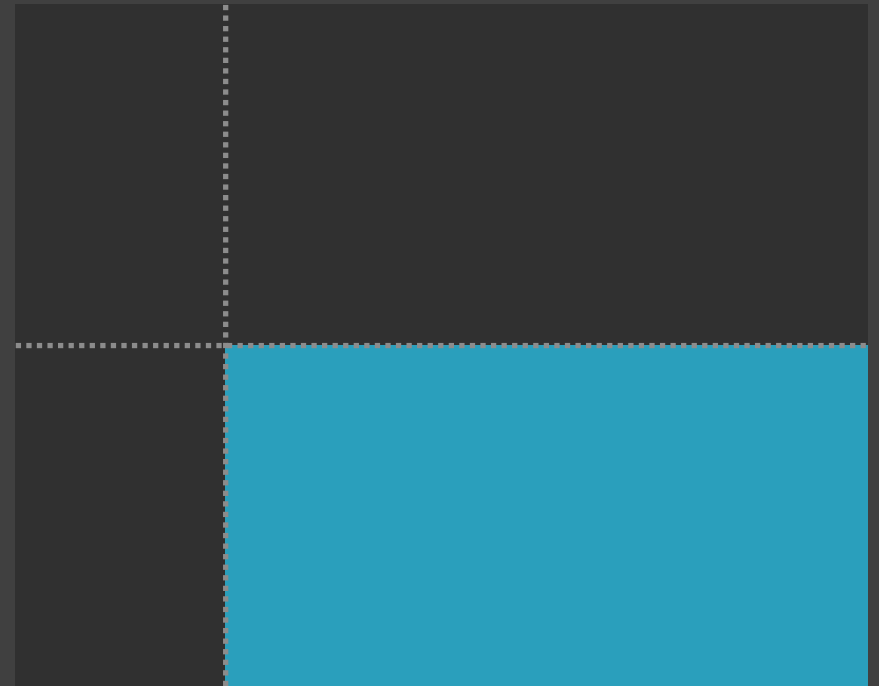
Know the XAML Layout Panels

StackPanel | **Grid** | VariableSized WrapGrid | Relative Panel | Canvas

```
<Grid>
  <Grid.RowDefinitions>
    <RowDefinition/>
    <RowDefinition/>
  </Grid.RowDefinitions>
  <Grid.ColumnDefinitions>
    <ColumnDefinition Width="40"/>
    <ColumnDefinition/>
  </Grid.ColumnDefinitions>
  <Rectangle Fill="LightBlue"
    Grid.Column="1" Grid.Row="1"/>
</Grid>
```
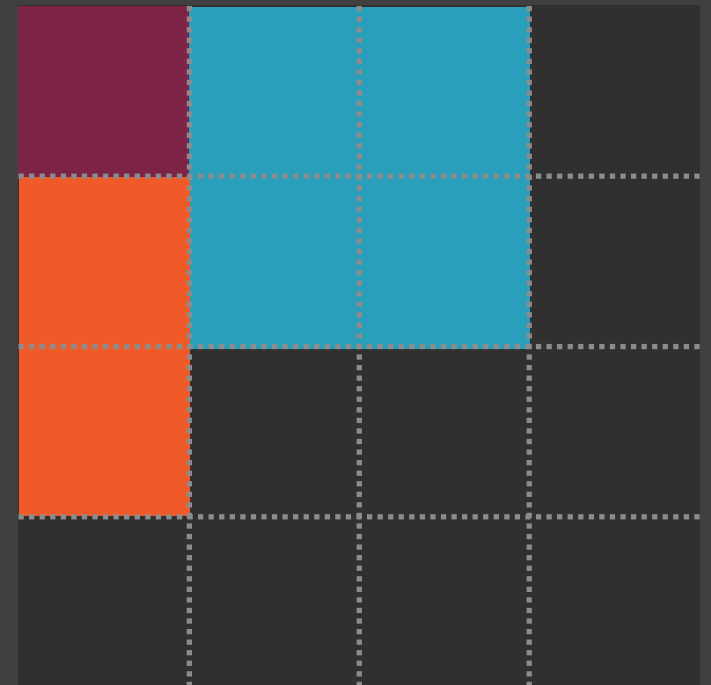
Know the XAML Layout Panels

StackPanel | Grid | **VariableSized WrapGrid** | Relative Panel | Canvas

```xaml
<VariableSizedWrapGrid
    MaximumRowsOrColumns="3"
    ItemHeight="50" ItemWidth="50"
    Orientation="Vertical">
    <Rectangle Fill="Purple"/>
    <Rectangle Fill="Orange"
        VariableSizedWrapGrid.RowSpan="2"/>
    <Rectangle Fill="LightBlue"
        VariableSizedWrapGrid.RowSpan="2"
        VariableSizedWrapGrid.ColumnSpan="2"/>
</VariableSizedWrapGrid>
```

Know the XAML Layout Panels

StackPanel

Grid

VariableSized WrapGrid

Relative Panel

Canvas

```xml
<RelativePanel>
  <TextBlock Text="Firstname" Margin="5"
    x:Name="labelFirstName"/>
  <TextBox Text="Thomas"
    RelativePanel.Below="labelFirstName"/>
</RelativePanel>
```

Firstname

Thomas

Know the XAML Layout Panels

StackPanel

Grid

VariableSized WrapGrid

Relative Panel

Canvas

```
<RelativePanel>
  <TextBlock Text="Firstname" Margin="5"
    x:Name="labelFirstName"/>
  <TextBox Text="Thomas"
    RelativePanel.RightOf="labelFirstName"/>
</RelativePanel>
```

Firstname Thomas

Know the XAML Layout Panels
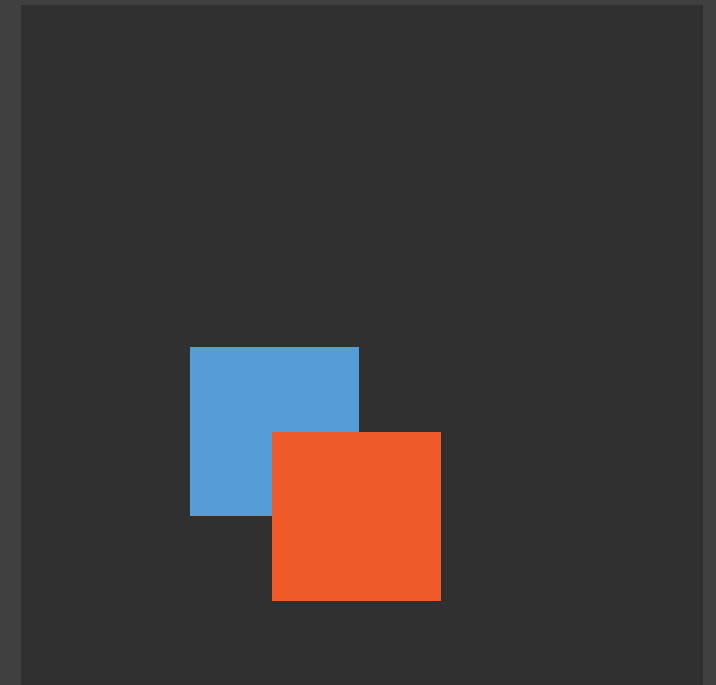
StackPanel

Grid

VariableSized WrapGrid

Relative Panel

Canvas

```xml
<Canvas>
  <Rectangle Fill="LightBlue"
    Height="50" Width="50"
    Canvas.Left="50"
    Canvas.Top="100"/>
  <Rectangle Fill="Orange"
    Height="50" Width="50"
    Canvas.Left="75"
    Canvas.Top="125"/>
</Canvas>
```
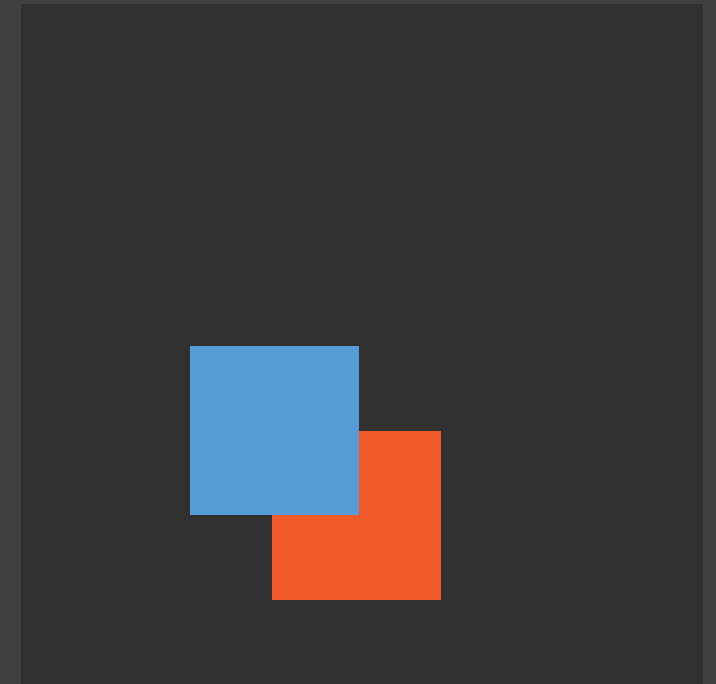
Know the XAML Layout Panels

StackPanel

Grid

VariableSized
WrapGrid

Relative
Panel

Canvas

```xaml
<Canvas>
  <Rectangle Fill="LightBlue"
    Height="50" Width="50"
    Canvas.Left="50" Canvas.ZIndex="1"
    Canvas.Top="100"/>
  <Rectangle Fill="Orange"
    Height="50" Width="50"
    Canvas.Left="75"
    Canvas.Top="125"/>
</Canvas>
```

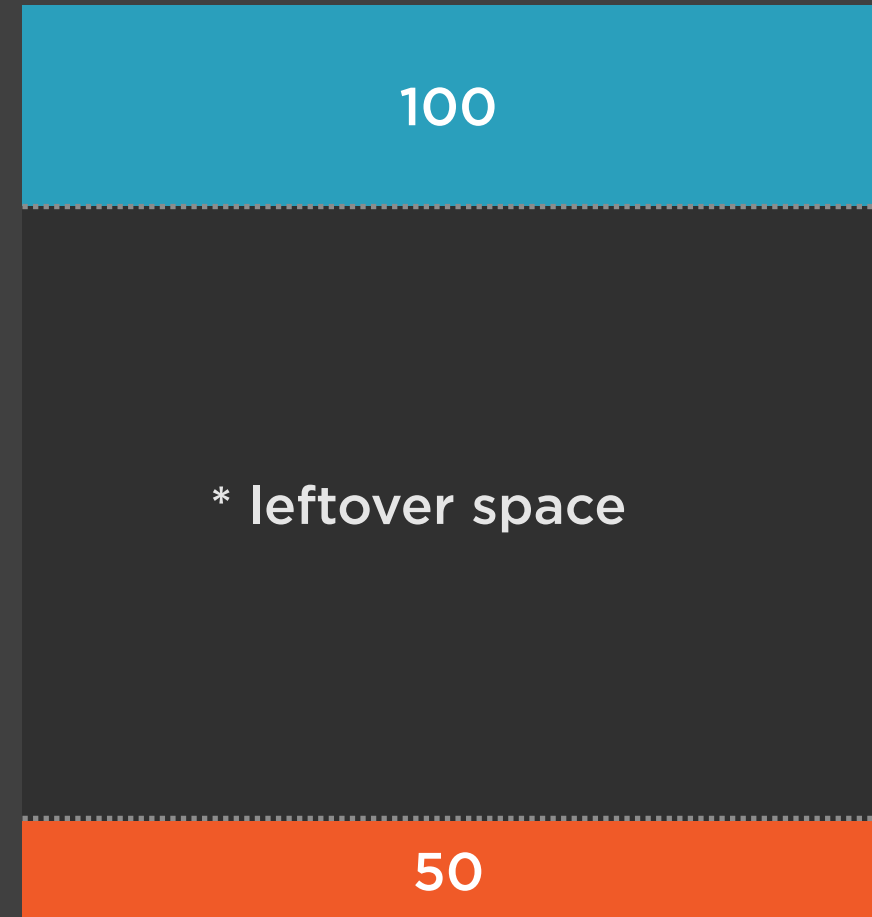# Demo

Build a layout with the Grid

Understand the Size of Rows and Columns

```xml
<Grid>
  <Grid.RowDefinitions>
    <RowDefinition Height="100"/>
    <RowDefinition Height="*"/>
    <RowDefinition Height="Auto"/>
  </Grid.RowDefinitions>

  <Rectangle Fill="LightBlue"/>

  <Rectangle Fill="Orange"
     Grid.Row="2" Height="50"/>
</Grid>
```

100

* leftover space

50

Demo

**Adjust the Grid in
the Coffee Customers application**

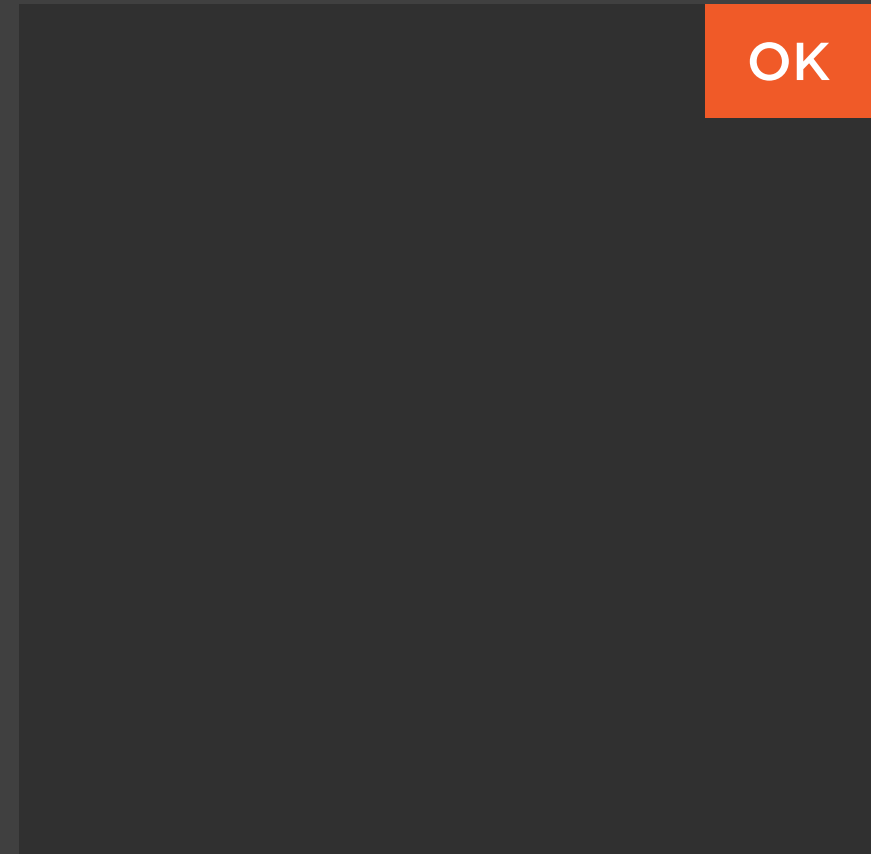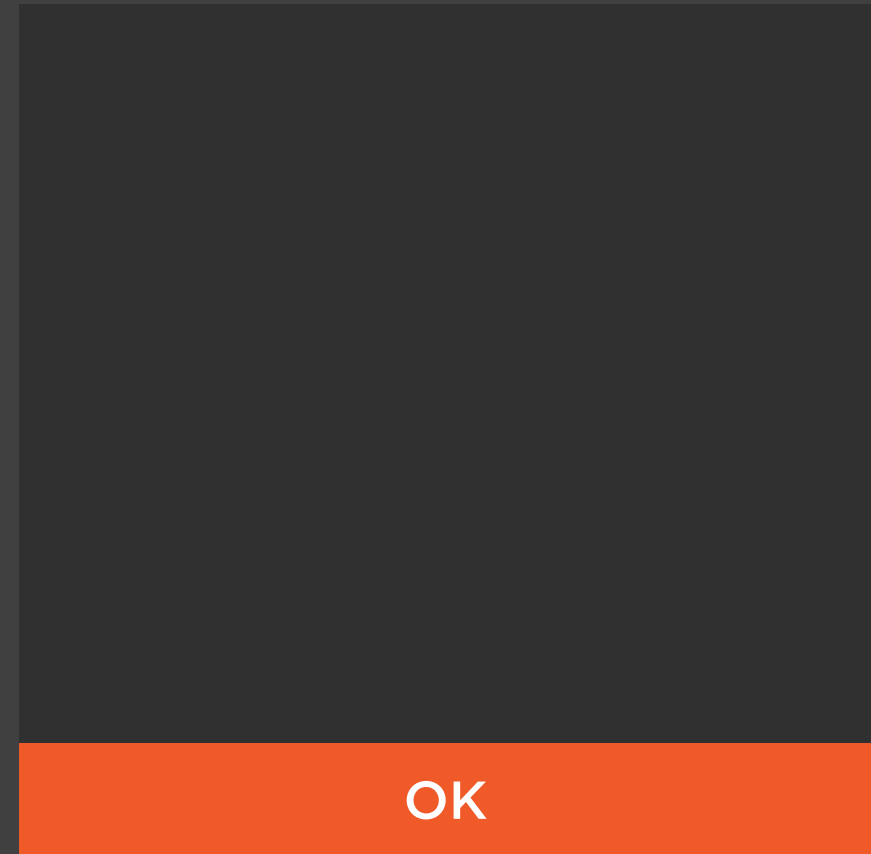# Use Layout Properties to Position Elements

```xml
<Grid Width="200" Height="200">

    <Button Content="OK"
        HorizontalAlignment="Left"
        VerticalAlignment="Top"/>


</Grid>
```

Use Layout Properties to Position Elements

```
<Grid Width="200" Height="200">

   <Button Content="OK"
      HorizontalAlignment="Right"
      VerticalAlignment="Top"/>



</Grid>
```

OK

Use Layout Properties to Position Elements

```xml
<Grid Width="200" Height="200">

    <Button Content="OK"
        HorizontalAlignment="Stretch"
        VerticalAlignment="Bottom"/>


</Grid>
```
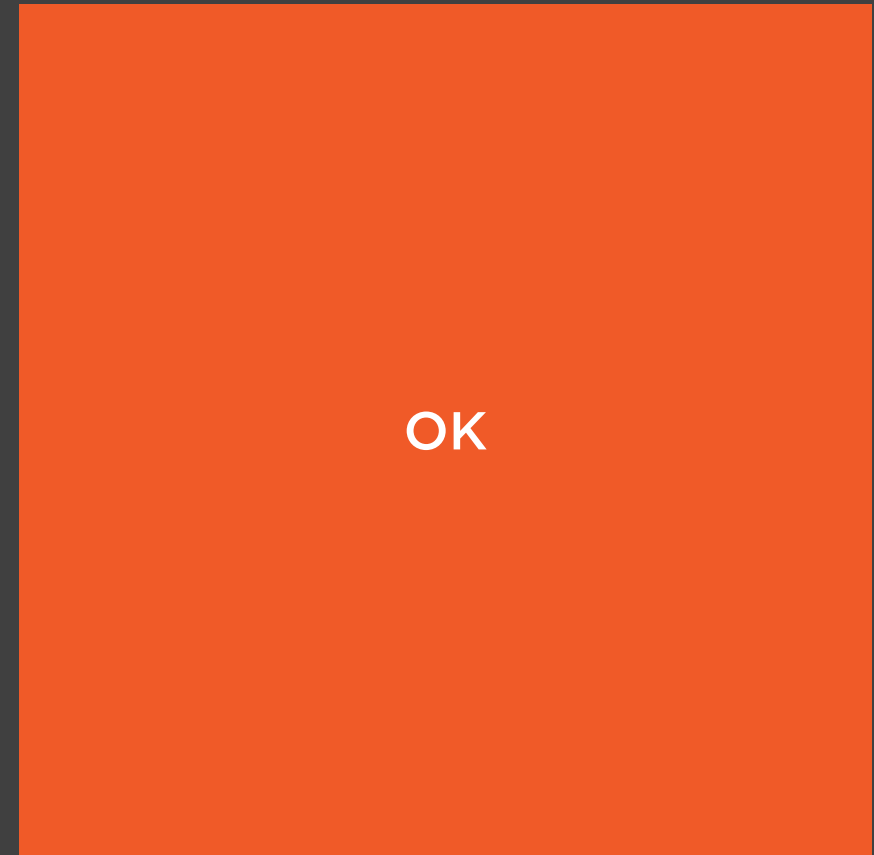
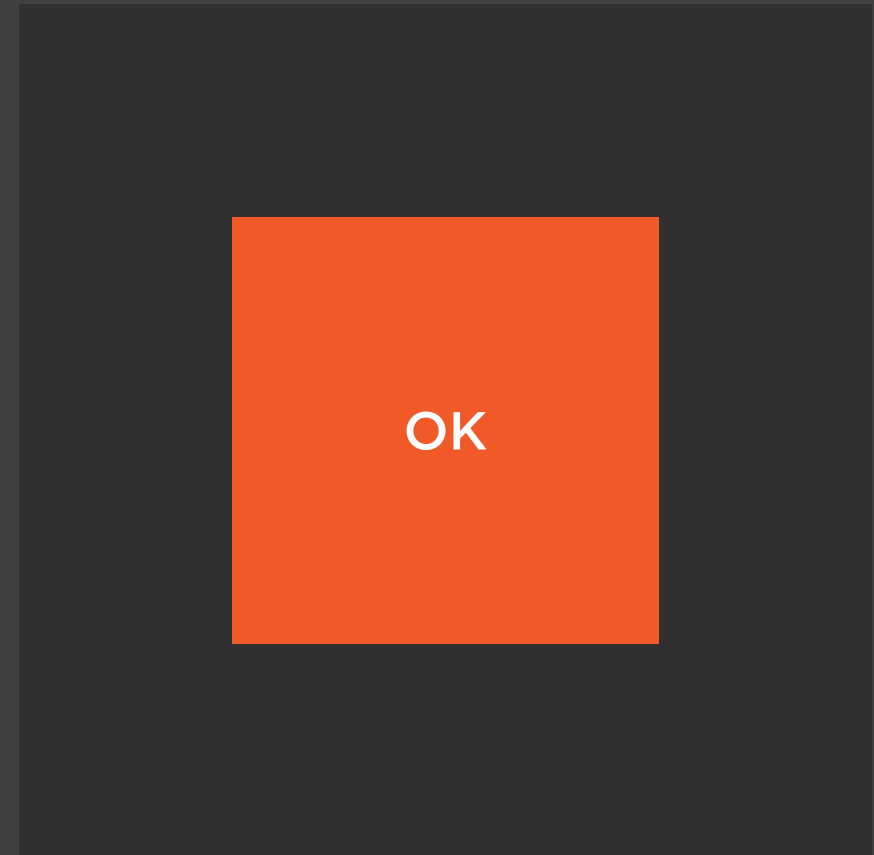# Use Layout Properties to Position Elements

```
<Grid Width="200" Height="200">

    <Button Content="OK"
        HorizontalAlignment="Stretch"
        VerticalAlignment="Stretch"/>


</Grid>
```

OK

Use Layout Properties to Position Elements

```
<Grid Width="200" Height="200">

  <Button Content="OK"
    HorizontalAlignment="Stretch"
    VerticalAlignment="Stretch"
    Margin="50"/>

</Grid>
```
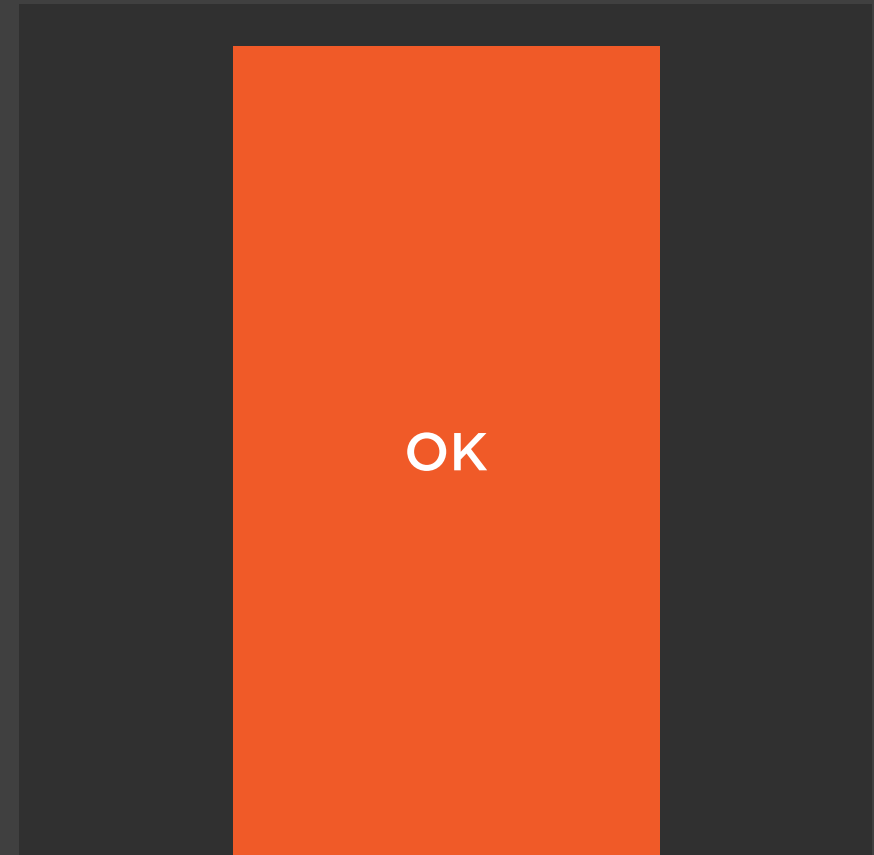
OK

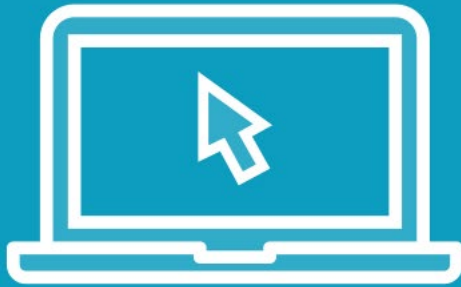# Use Layout Properties to Position Elements

```
<Grid Width="200" Height="200">

   <Button Content="OK"
      HorizontalAlignment="Stretch"
      VerticalAlignment="Stretch"
      Margin="50 10 50 0"/>

</Grid>
```

OK

Demo

Position elements in the Coffee Customers application

Demo

Use a nested Grid

Add additional Buttons
to the navigation

Set Attached Properties in XAML

Attribute Syntax `<Button Content="OK"/>`

Property Element Syntax

```
<Button>
  <Button.Content>
    OK
  </Button.Content>
</Button>
```

Set Attached Properties in XAML

Attribute Syntax   `<Button Grid.Row="1"/>`

Property Element Syntax

```
<Button>
  <Grid.Row>
    1
  </Grid.Row>
</Button>
```

Set Attached Properties in C#

XAML
```xml
<Button Grid.Row="1"/>
```

C#
```csharp
var btn = new Button();
btn.SetValue(Grid.RowProperty, 1);

var column =
    (int)btn.GetValue(Grid.RowProperty);
```

Demo

Set Attached Properties in C#

Move the navigation to the other side

Demo

Access Attached Properties with static methods

Summary

XAML Layout Panels

- Grid
- StackPanel

Set Attached Properties in

- XAML
- C#

**Position elements
with layout properties**