Building PowerShell GUIs in WPF for Free

UNDERSTANDING BASIC WPF UI

Jeff Adkin
PLURALSIGHT AUTHOR

@JeffAdkin www.JAdkin.com

Introduction

We will be going over the following components

- WPF Overview
- WPF vs. Windows Forms
- Understanding XAML
- Creating an XAML Window
- Add Control to WPF Window
- WPF Layouts

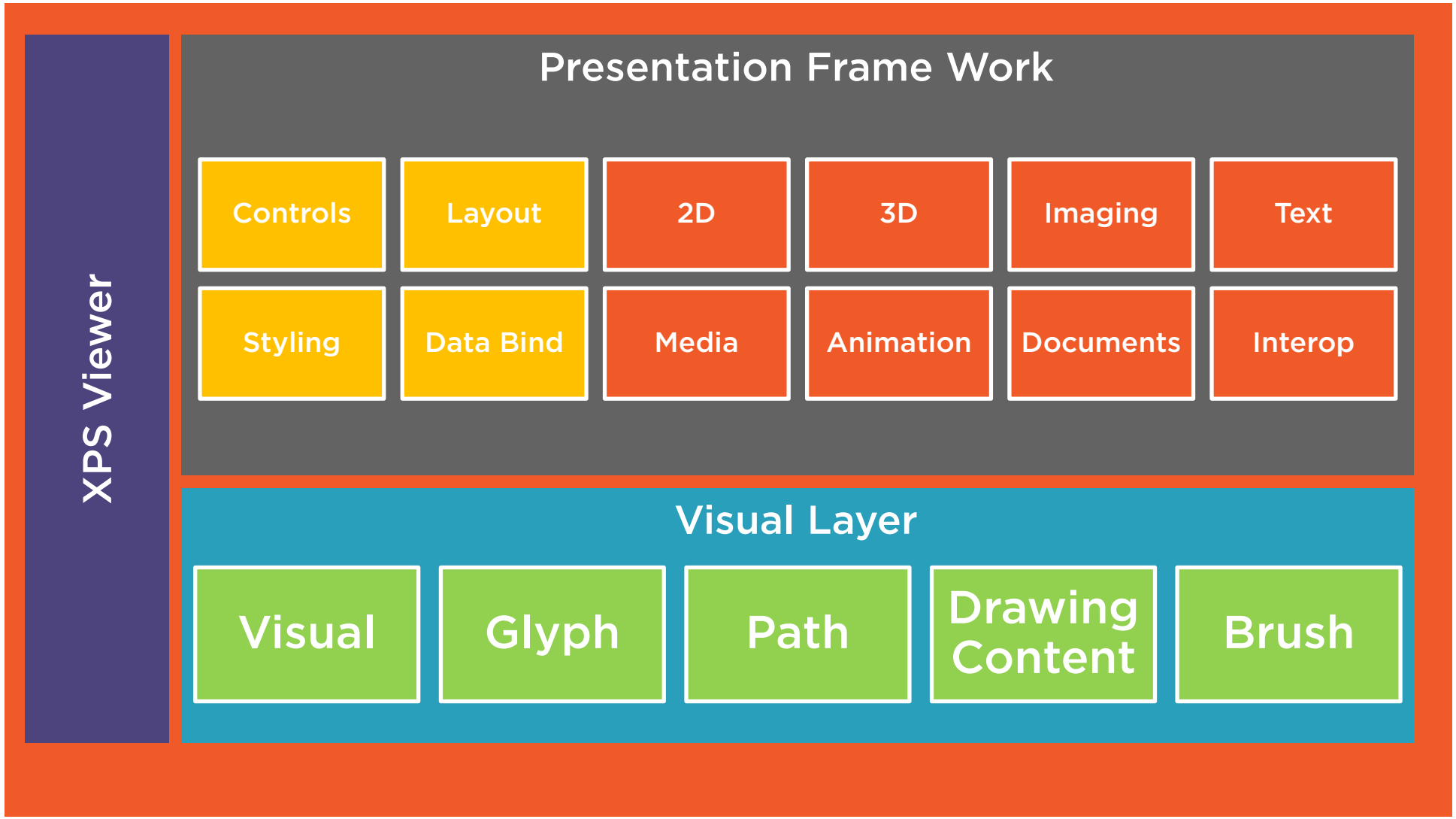
WPF Overview

What Is WPF?

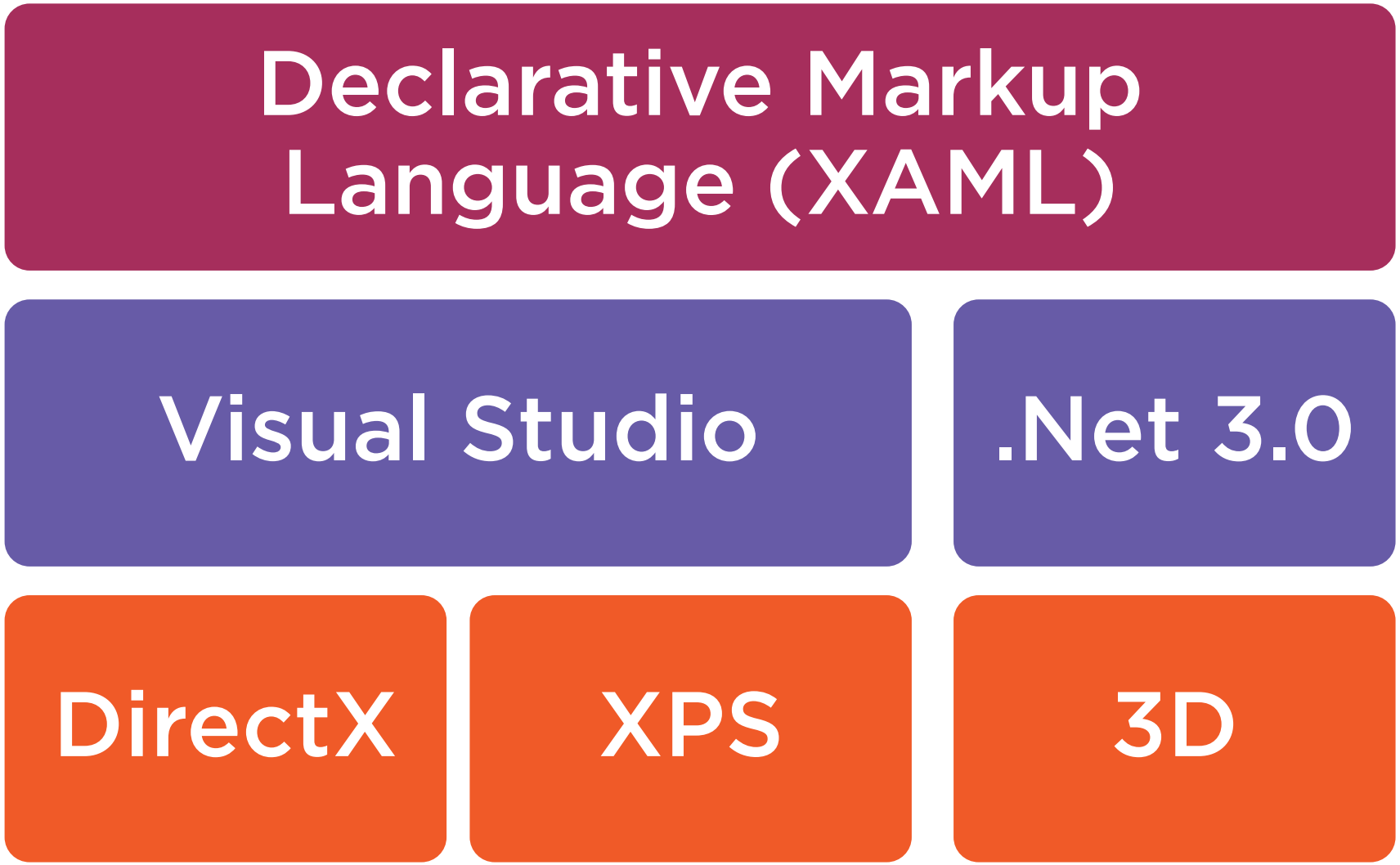
‘Windows Presentation Foundation (or **WPF**) is a graphical subsystem for rendering user interfaces in Windows-based applications. WPF, previously known as "Avalon", was initially released as part of .NET Framework 3.0. WPF employs XAML, an XML-based language, to define and link various interface elements.’

https://en.wikipedia.org/wiki/Windows_Presentation_Foundation

WPF Architecture



WPF Key Points



WPF vs. Windows Forms

WPF vs. Windows Forms

WPF	Forms
• New	• Old
• DirectX	• User32/GDI
• 2D and 3D	• 2D
• GPU/CPU	• CPU
• Not many 3rd Party Controls	• Lots of 3rd Party Controls

Main Differences

	Windows Forms	Windows Forms/GDI+	Direct3D	WPF
Graphical interface, e.g., forms and controls	X			X
On-screen documents	X			X
Fixed-format documents				X
Images		X		X
Video and audio				X
Two-dimensional graphics		X		X
Three-dimensional graphics			X	X

Understanding XAML

What Is XAML?

‘XAML is a declarative markup language. As applied to the .NET Framework programming model, XAML simplifies creating a UI for a .NET Framework application. You can create visible UI elements in the declarative XAML markup, and then separate the UI definition from the run-time logic by using code-behind files, joined to the markup through partial class definitions.’

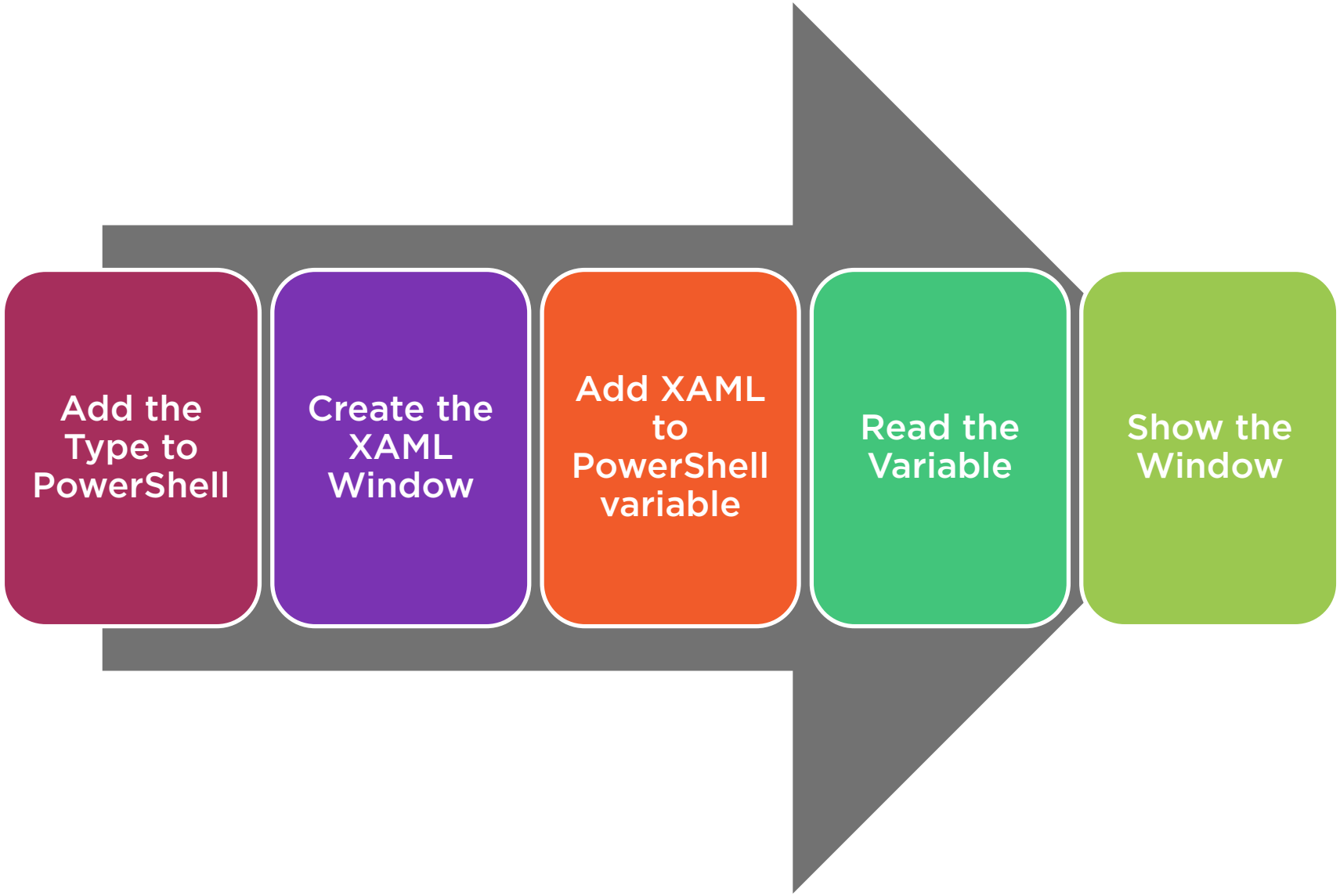
https://msdn.microsoft.com/en-us/library/ms752059(v=vs.110).aspx

XAML Layout

```
<Window>
    <Layout>
        <Control Property="Value"/>
        <Control Property="Value"/>
        <Control Property="Value"/>
    </Layout>
</Window>
```

Creating an XAML Window

Steps to Create a WPF Window



```
<Window xmlns=http://schemas.microsoft.com/winfx/2006/xaml/presentation
Title="My First Form" Height="480"  Width="640">
</Window>
```

Create the XAML Window

An XAML Window allows us to choose the size and title very easily. There are a multitude of Properties on a Window.

For more details - https://msdn.microsoft.com/en-us/library/system.windows.window(v=vs.110).aspx

```
Add-Type -AssemblyName PresentationFramework
```

Add the Type in PowerShell

Add the .Net Framework WPF to PowerShell.

```
[xml]$Form = @"
<Window xmlns=http://schemas.microsoft.com/winfx/2006/xaml/presentation
Title="My First Form" Height="480"  Width="640">
</Window>
"@
```

Add XAML to PowerShell Variable

This allows us to place all of the XAML in a single XML variable to be able to load from.

```
$NR=(New-Object System.Xml.XmlNodeReader $Form)

$Win=[Windows.Markup.XamlReader]::Load( $NR )
```

Read the Variable

Load the XML Node Reader to read the XAML $Form variable then load the variable $NR.

```
$Win.ShowDialog()
```

Show the Window

Everything is read and loaded now we simply show the window.

Demo

Add Control to WPF Window

```
[xml]$Form  = @"
  <Window xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
  Title="My First Form" Height="480"  Width="640">
  <Button Name="MyButton" Width="120" Height="85" Content = 'Hello' />
  </Window>
"@
```

XAML Button

In XAML we can easily add a Control to the Window, but can we add 2 controls?

Demo

WPF Layouts

Types of Layouts

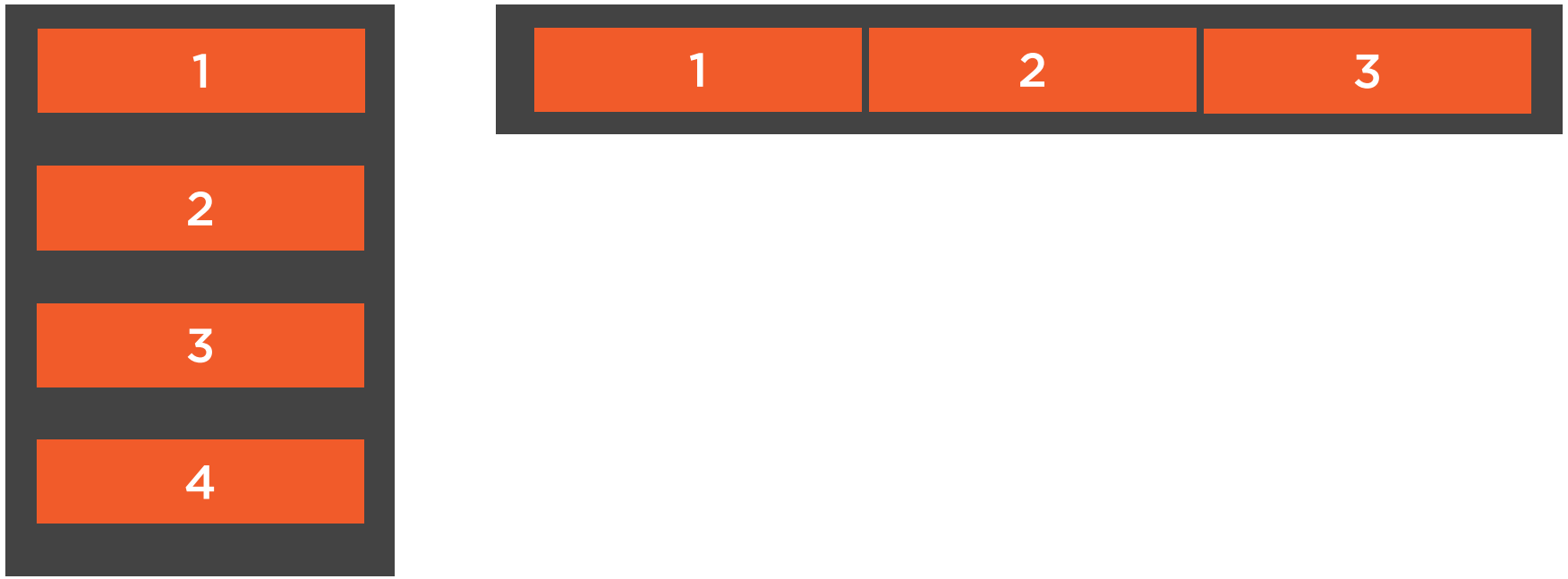
StackPanel

Canvas

Grid

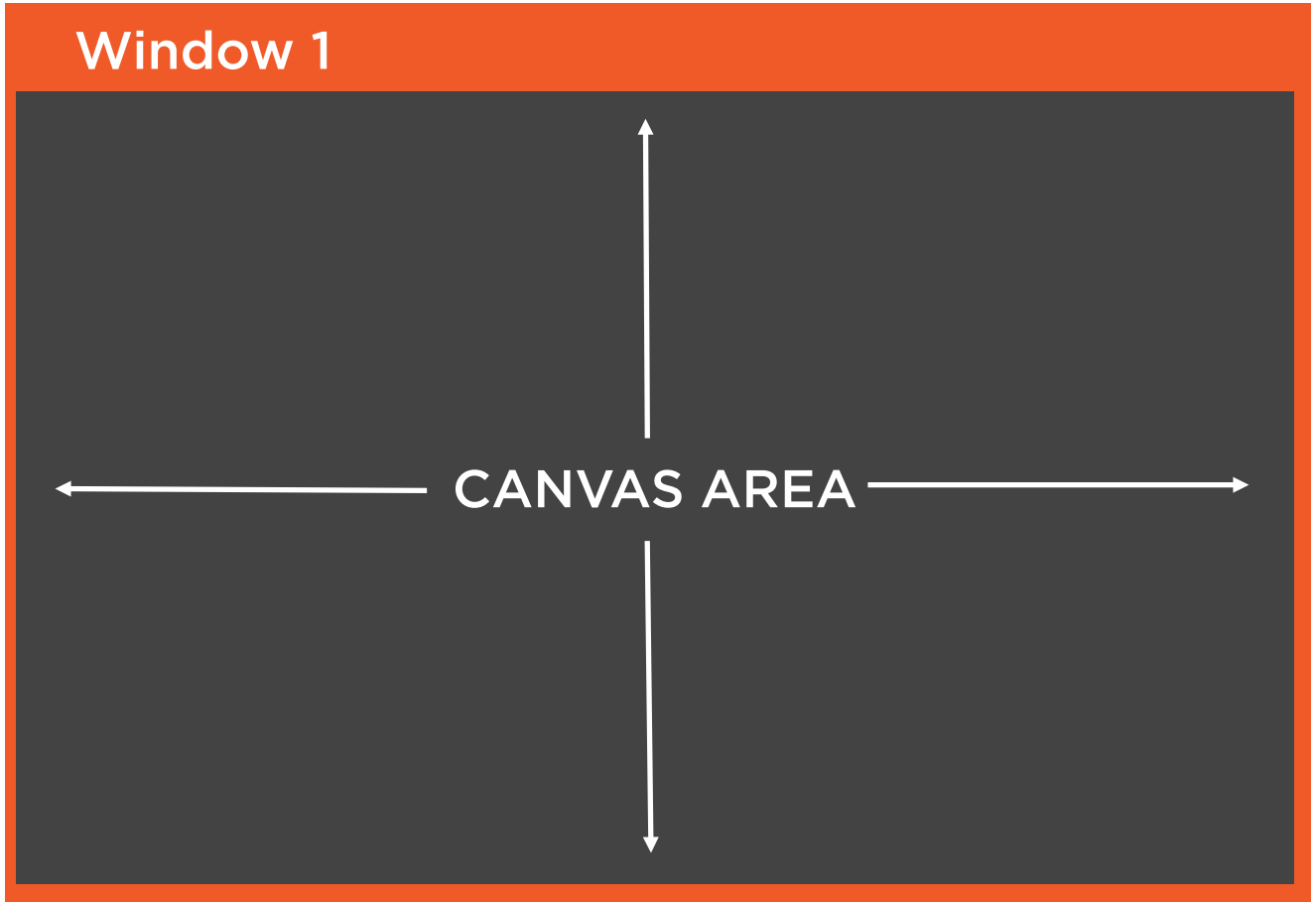
StackPanel

StackPanel creates a layout that stacks child objects either vertically or horizontally



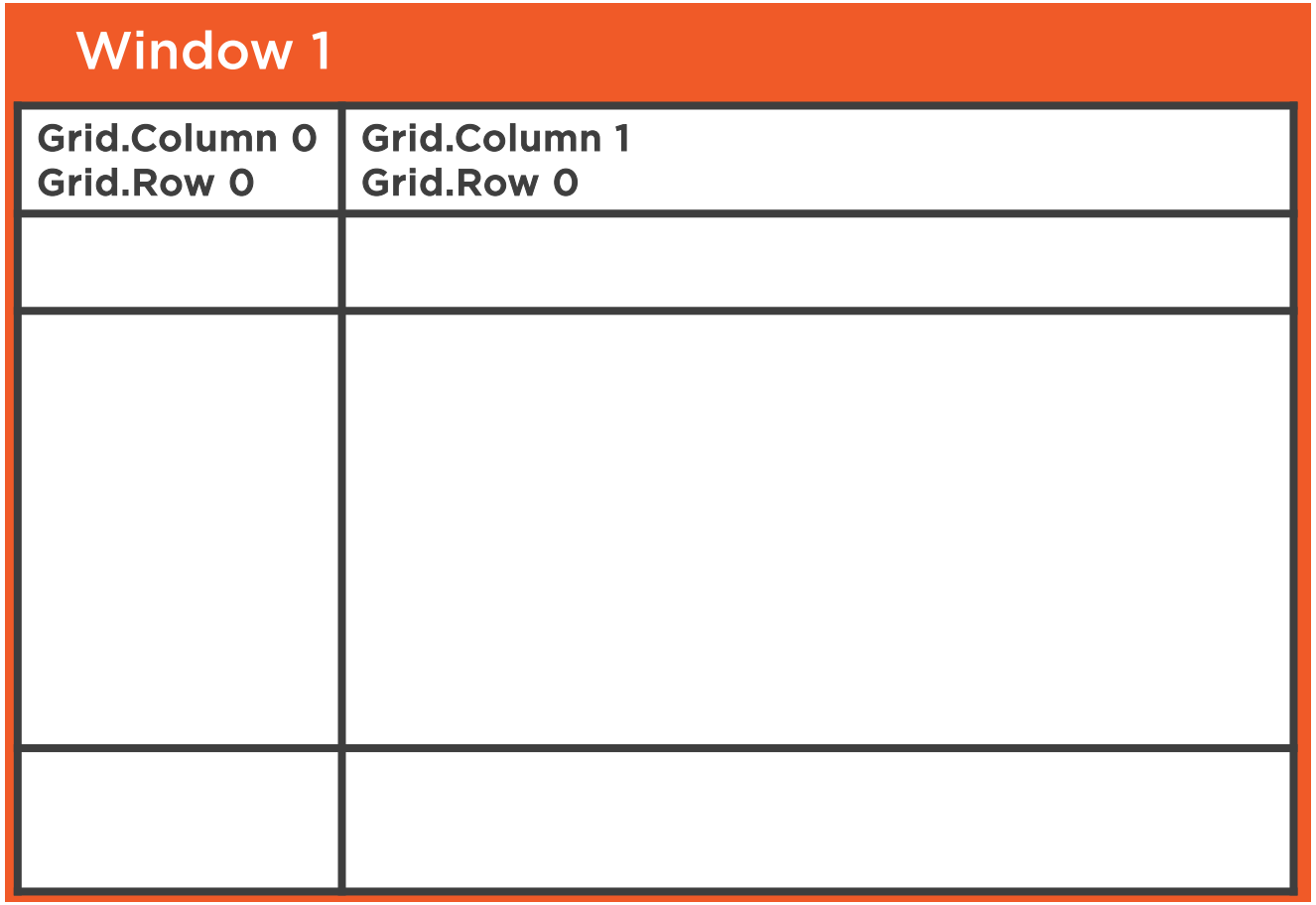
Canvas

Canvas creates a layout the stretches across the whole Window



Grid

The Grid layout uses a grid (Table) to create pre-defined areas for your Controls



Demo