Data and Properties Binding in WPF

Why We Need Data Binding?

Why We Need Data Binding?

- The purpose of most applications is:
  - Displaying data to users
  - Letting them edit that data
- Developers' job is:
  - Bring the data from a variety of sources
  - Expose the data in object, hierarchical, or relational format
- With WPF's data binding engine, you get more features with less code

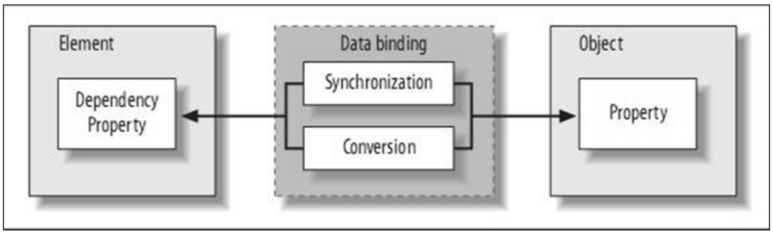
Why We Need Data Binding? (2)

- Data binding is pulling information out of an object into another object or property
  - Data binding means automatically change a property when another property is changed
- Many Windows applications are all about data
- Data binding is a top concern in a user interface technologies like WPF or Silverlight
- WPF and Silverlighto provide very powerful data binding mechanisms

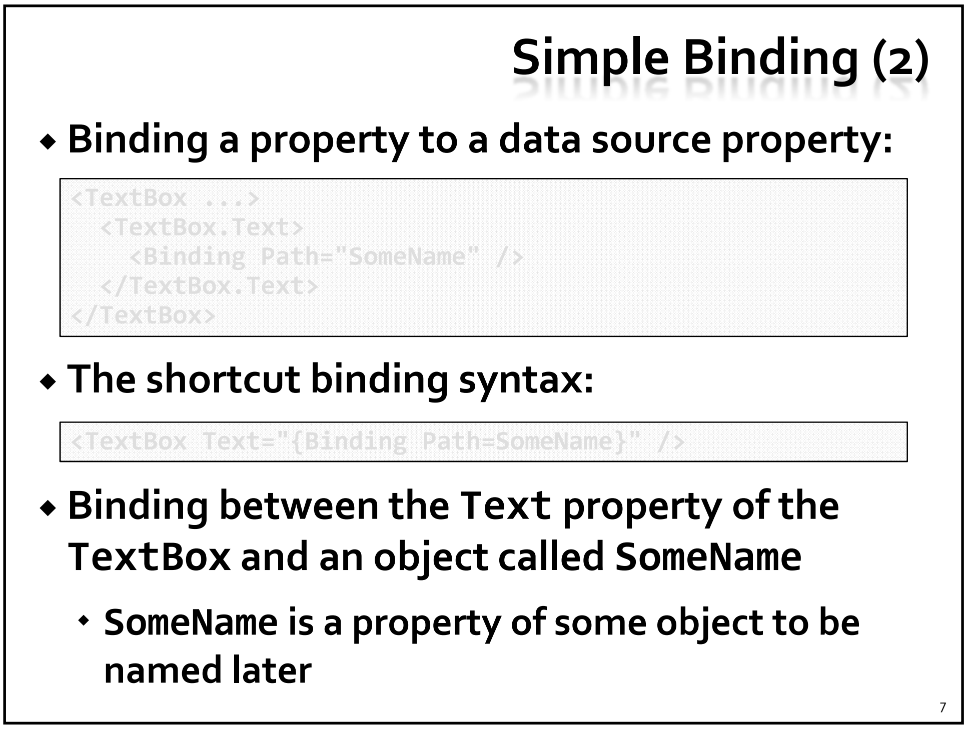
Simple Binding

Simple Binding

- Simple binding in WPF is the act of registering two properties with the data binding engine
  - Letting the engine keep them synchronized
- The synchronization and conversion are duties of the data binding engine in WPF



6



Simple Binding (2)

- Binding a property to a data source property:

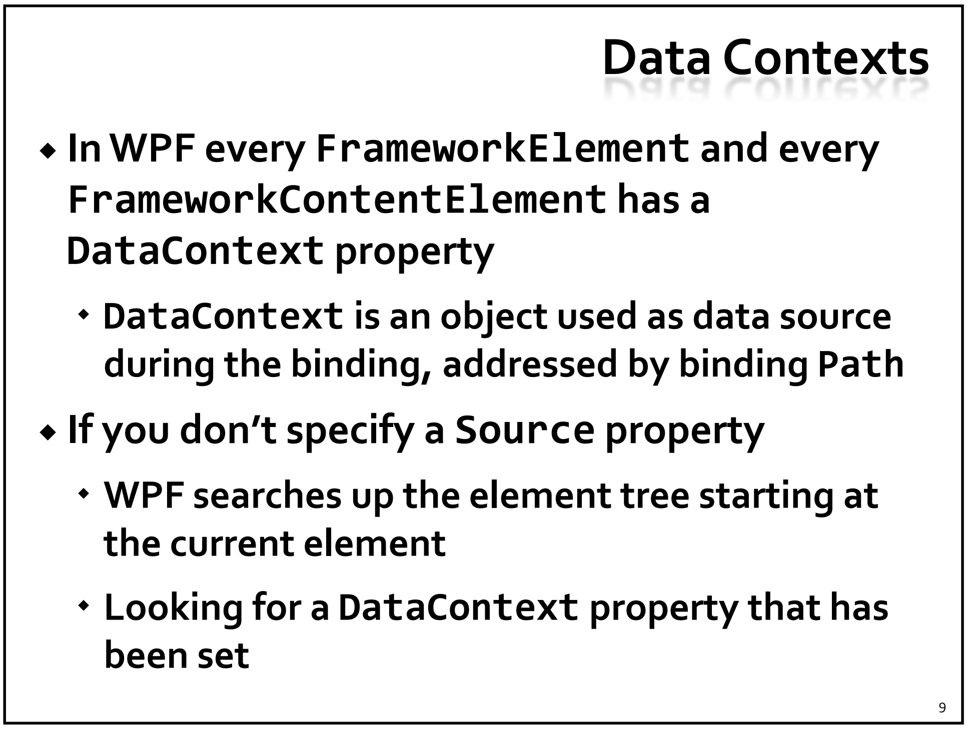
```
<TextBox ...>
  <TextBox.Text>
    <Binding Path="SomeName" />
  </TextBox.Text>
</TextBox>
```

- The shortcut binding syntax:

```
<TextBox Text="{Binding Path=SomeName}" />
```

- Binding between the `Text` property of the `TextBox` and an object called `SomeName`
  - `SomeName` is a property of some object to be named later

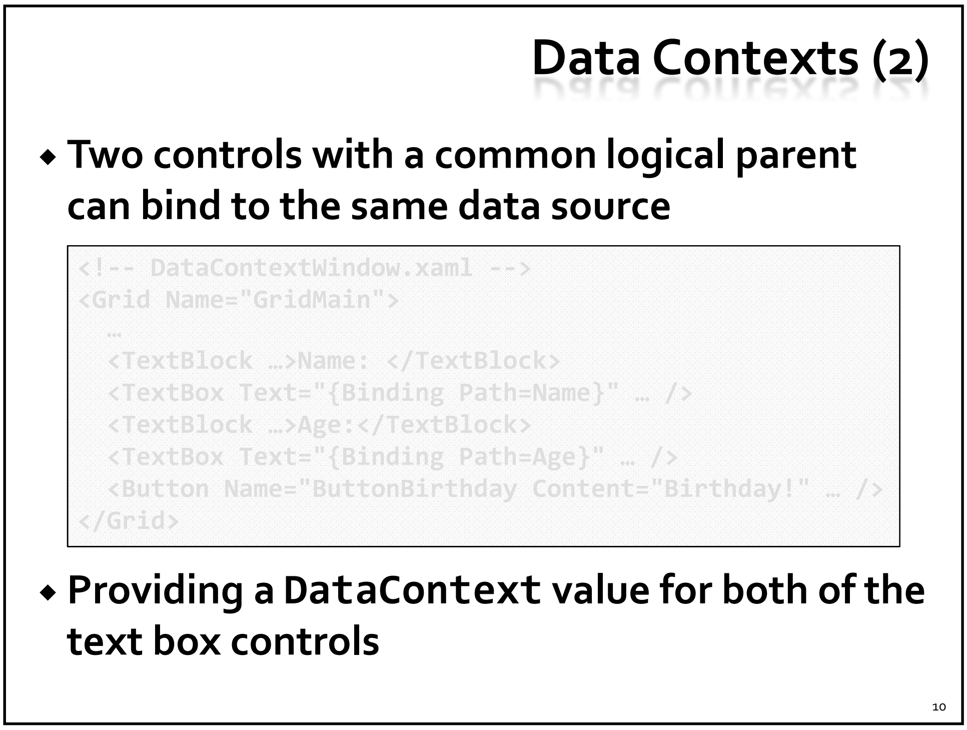
Data Contexts



Data Contexts

- In WPF every `FrameworkElement` and every `FrameworkContentElement` has a `DataContext` property
  - `DataContext` is an object used as data source during the binding, addressed by binding `Path`
- If you don't specify a `Source` property
  - WPF searches up the element tree starting at the current element
  - Looking for a `DataContext` property that has been set

9



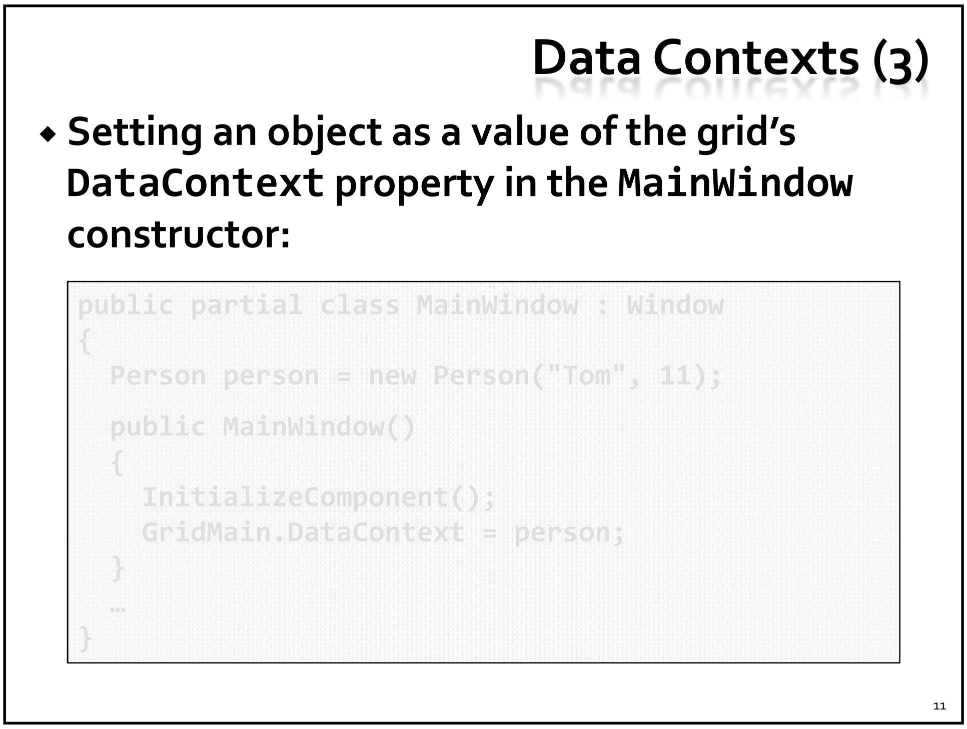
Data Contexts (2)

- Two controls with a common logical parent can bind to the same data source

```
<!-- DataContextWindow.xaml -->
<Grid Name="GridMain">
  …
  <TextBlock …>Name: </TextBlock>
  <TextBox Text="{Binding Path=Name}" … />
  <TextBlock …>Age:</TextBlock>
  <TextBox Text="{Binding Path=Age}" … />
  <Button Name="ButtonBirthday Content="Birthday!" … />
</Grid>
```

- Providing a `DataContext` value for both of the text box controls

10



Data Contexts (3)

- Setting an object as a value of the grid's `DataContext` property in the `MainWindow` constructor:

```
public partial class MainWindow : Window
{
  Person person = new Person("Tom", 11);

  public MainWindow()
  {
    InitializeComponent();
    GridMain.DataContext = person;
  }
  …
}
```

11

Data Contexts

Live Demo

Binding to Other Controls

Binding to Other Controls

- WPF provides binding of one element's property to another element's property

```
<TextBox Name="ageTextBox" Foreground="Red" … />
<Button …
  Foreground="{Binding ElementName=ageTextBox,
  Path=Foreground}" Content="Birthday" />
```

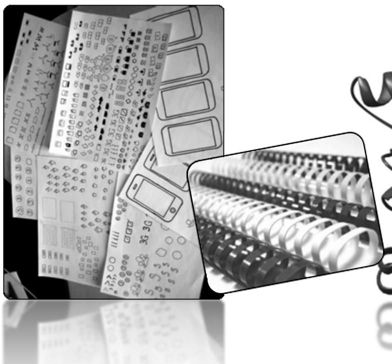
- The button's foreground brush will always follow foreground brush's color of the age `TextBox`

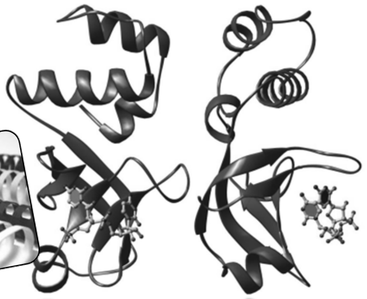
Binding to Other Controls

Live Demo

Name: Dedo Kolio

Age: 200





The Binding Class and Its Properties

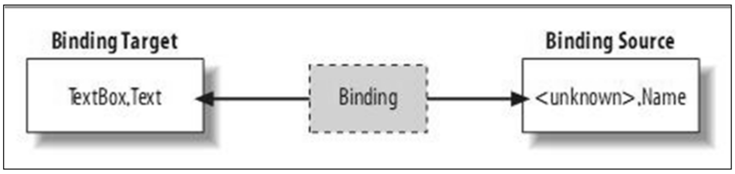
Binding Class (2)

- More Binding class properties
  - ElementName – used when the source of the data is a UI element as well as the target
  - Mode – one of the BindingMode values TwoWay, OneWay, OneTime, OneWayToSource, or Default
  - Path – path to the data in the data source object
  - Source – a reference to the data source

17

Binding Class (3)

- The binding target can be any WPF element
  - Only allowed to bind to the element's dependency properties



- The TextBox control is the binding target
- Object that provides the data is the binding source

18

Simple Data Binding

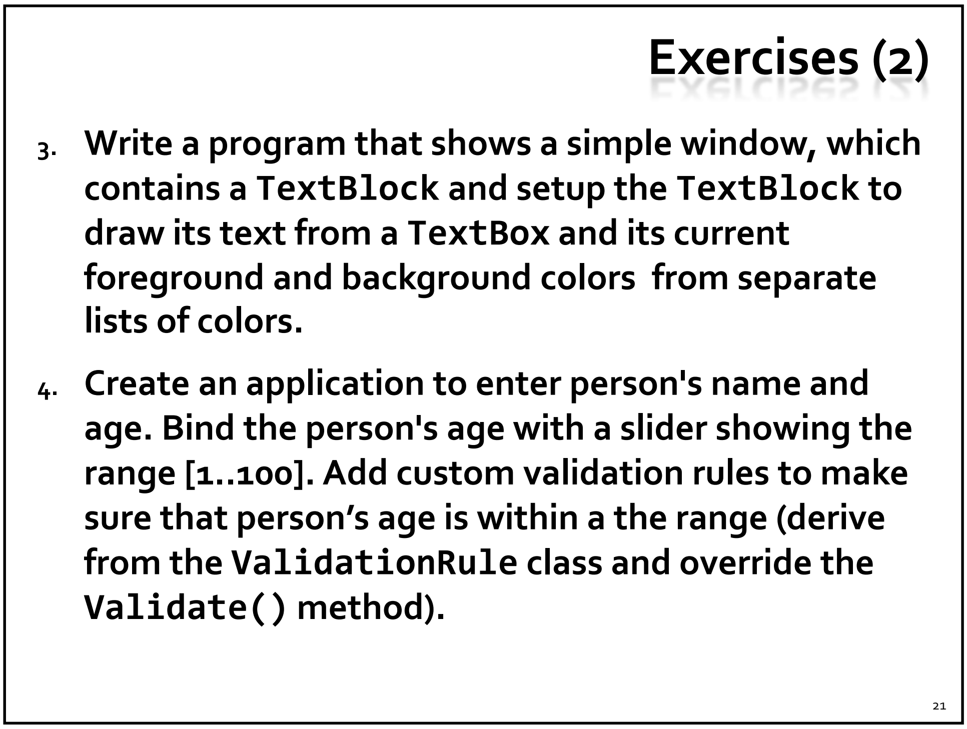
Questions?

http://academy.telerik.com

Exercises

1. Write a program that show a simple window, it contains two controls a `Slider` and a `TextBlock` with a single line of text. If you pull the thumb in the slider to the right, the font size of the text is increased immediately. Add a label that shows the current font size. Use data binding.
2. Add to the previous exercise few buttons each of which applies a preset value to the slider. When you click the "Set to Large" button the code in `Click` event sets the value of the slider, which in turn forces a change to the font size of the text through data binding.

20



Exercises (2)

3. Write a program that shows a simple window, which contains a `TextBlock` and setup the `TextBlock` to draw its text from a `TextBox` and its current foreground and background colors from separate lists of colors.

4. Create an application to enter person's name and age. Bind the person's age with a slider showing the range [1..100]. Add custom validation rules to make sure that person's age is within a the range (derive from the `ValidationRule` class and override the `Validate()` method).