Adatkötés UI elemhez

Ha **UI** vezérlőn adatkötést állítunk be egy másik **UI** vezérlő valamilyen értékéhez, akkor az ***ElementName*** paraméterben meg kell adni annak a vezérlőnek a nevét, amelyhez hozzákötjük, míg a ***Path*** paraméterben meg kell adni, hogy milyen értéket akarunk megkapni. Pár példa:

- amit beírunk a *text1* nevű *TextBoxba*, megjelenik a *TextBlock*-ban

```
<Grid>
    <TextBox x:Name="txt1" HorizontalAlignment="Left" Margin
    <TextBlock Text="{Binding ElementName=txt2, Path=Text}"
```

- itt egy *Slider* értékét (*Value*) jelenítjük meg egy *TextBox*-ban. Az adatkötésben több paramétert is megadunk

```
<TextBox x:Name="txt2" Text="{Binding ElementName=sld1, Path=Value, UpdateSourceTrigger=PropertyChanged, Mode=TwoWay, Delay=4000}"
<Slider x:Name="sld1" IsSnapToTickEnabled="True" Minimum="10" Maximum="250" HorizontalAlignment="Left" Margin="397,25,0,0" Vertica
```

az `UpdateSourceTrigger=PropertyChanged` azt állítja be, hogy a rendszer milyen esemény hatására frissítse a *TexBox*-ban megjelenő értéket. A `PropertyChanged` azt jelenti, hogy a tulajdonság megváltozásakor egyből frissít. Épp ezért van beállítva a `Delay` tulajdonság is, mert ez pedig késleltetést jelent ezredmásodpercben. Tehát itt azt állítottam be, hogy módosuljon a megjelenített érték, ha változik a *Sliderben* a *Value*, de a módosítás megjelenítésével várjon 4 másodpercet! A `Mode=TwoWay` azt jelenti, hogy a módosítás kétirányú. Ha elmozdítom a *Slider csúszkáját*, akkor módosul a *TexBox* tartalma. Ha módosítom a *TexBox* tartalmát, akkor módosul a *Slideren* a csúszka helyzete!

- itt egy téglalap szélességét, magasságát és a vonalvastagságát állítottuk be adatkötéssel

```
<Rectangle Width="{Binding ElementName=sld1, Path=Value}" Height="{Binding ElementName=sld2, Path=Value}" StrokeThickness="{Binding ElementName=sld3, Path=Value}
<Slider x:Name="sld3" IsSnapToTickEnabled="True" Minimum="1" Maximum="25" HorizontalAlignment="Left" Margin="163,73,0,0" VerticalAlignment="Top" Width="120"/>
```

- itt egy StackPanelben lévő RadioButton sorozat csak akkor aktív, ha a CheckBox be van pucukázva

```
<CheckBox x:Name="cbx" Content="CheckBox" HorizontalAlignment="Left" Margin="192,11,0,0" VerticalAlignment="Top"/>
<StackPanel Grid.Column="1" IsEnabled="{Binding ElementName=cbx, Path=IsChecked}">
    <RadioButton Content="1" Margin="5"/>
    <RadioButton Content="2" Margin="5"/>
    <RadioButton Content="3" Margin="5"/>
</StackPanel>
```

- Itt egy ListBox-ot saját magához kötöttünk. Ha rákattintok valamelyik elemére, akkor a keretének a színe olyan lesz, mint ami az adott elem tartalma

```
<ListBox x:Name="eee"  BorderBrush="{Binding ElementName=eee, Path=SelectedItem.Content}" Grid.Column="2" SelectedIndex="1" Grid.Row="1" BorderThickness="3">
    <ListBoxItem Content="Yellow"/>
    <ListBoxItem Content="Black"/>
    <ListBoxItem Content="Blue"/>
    <ListBoxItem Content="Red"/>
</ListBox>
```

Arra figyelni kell, hogy egyes vezérlőknél az érték megváltozása esemény lefuthat inicializáláskor is, amikor még más vezérlők esetleg NULL értékkel léteznek. Ez hibához vezethet, így javasolt az esemény metódusába betenni egy olyan sort, hogy null esetén ne csináljon semmit!

```
private void sldr_ValueChanged(object sender,
RoutedPropertyChangedEventArgs<double> e)
 {
    if (sldg == null || sldb == null || slda == null)
      return;
```