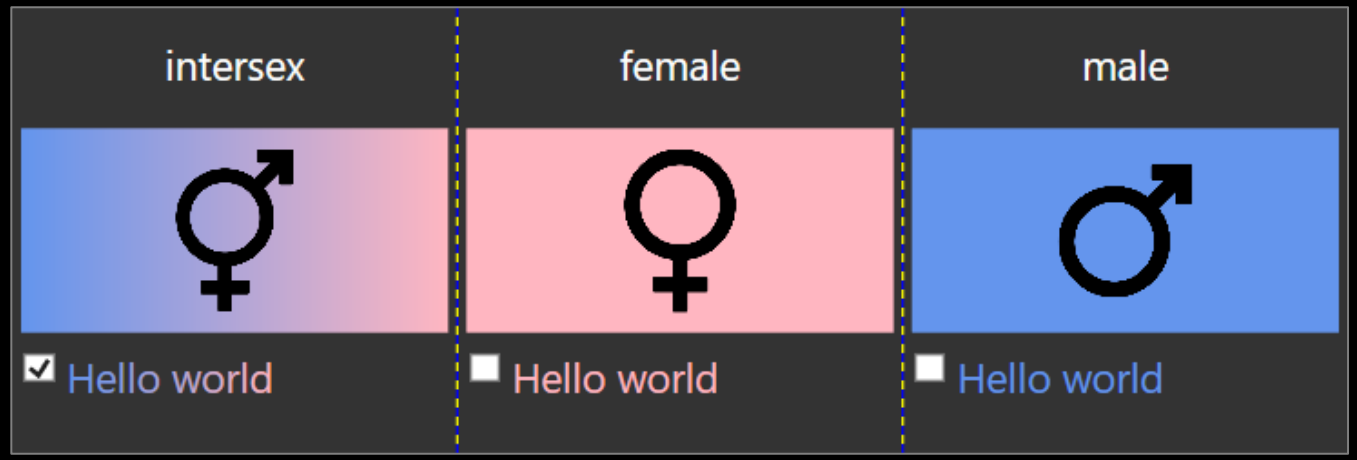
TOOL DEVELOPMENT

MVVM - VALUECONVERTERS

MVVM
VALUECONVERTERS: WHAT & WHY?

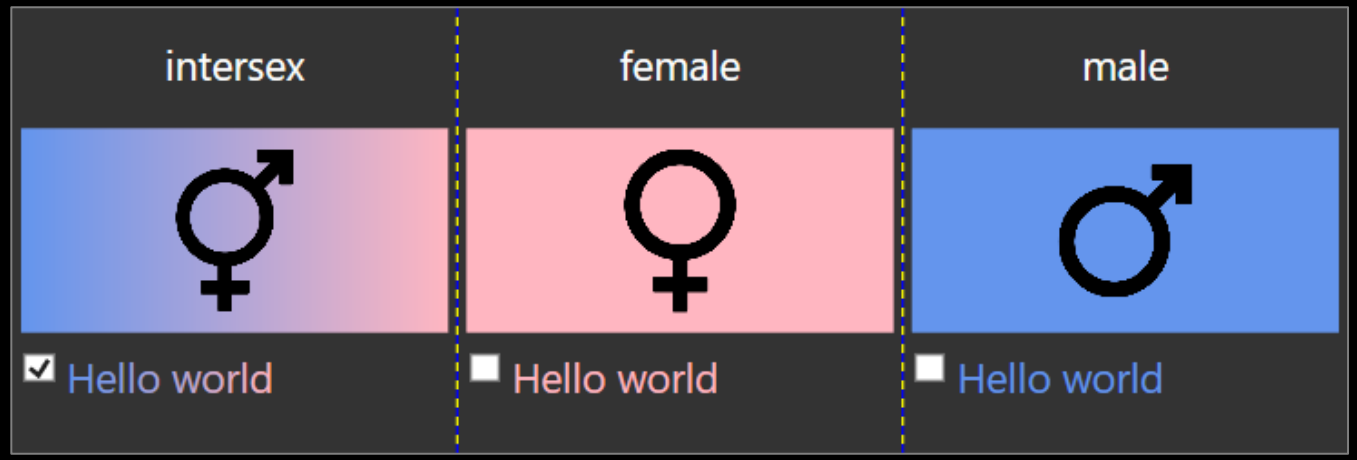



```
public class Person
{
    0 references
    public string Gender { get; set; }
}
```

```
<TextBlock Foreground="White" FontSize="18" HorizontalAlig
           Text="{Binding Gender}"/>
<Rectangle Grid.Row="1" Fill="{Binding Gender, Converter={St
<Image Grid.Row="1" Source="{Binding Gender, Converter={Stat
<CheckBox Grid.Row="2" Content="Hello world"
          Foreground="{Binding Gender, Converter={StaticReso
          FontSize="18" IsChecked="{Binding Gender, Converte
<StackPanel Orientation="Horizontal" VerticalAlignment="Bott
    <Rectangle Fill="{Binding Gender, Converter={StaticResou
    <Rectangle Width="20" Height="20" Margin="2"/>
    <Rectangle Width="20" Height="20" Margin="2"/>
</StackPanel>
```

MVVM

VALUECONVERTERS: WHAT & WHY?




```
public class Person
{
    0 references
    public string Gender { get; set; }
}
```

```
<TextBlock Foreground="White" FontSize="18" HorizontalAlig
           Text="{Binding Gender}"/>
<Rectangle Grid.Row="1" Fill="{Binding Gender, Converter={St
<Image Grid.Row="1" Source="{Binding Gender, Converter={Stat
<CheckBox Grid.Row="2" Content="Hello world"
          Foreground="{Binding Gender, Converter={StaticReso
          FontSize="18" IsChecked="{Binding Gender, Converte
<StackPanel Orientation="Horizontal" VerticalAlignment="Bott
    <Rectangle Fill="{Binding Gender, Converter={StaticResou
    <Rectangle Width="20" Height="20" Margin="2"/>
    <Rectangle Width="20" Height="20" Margin="2"/>
</StackPanel>
```

VALUECONVERTERS: WHAT?

- A ValueConverter:
  - Gets a value through Binding in xaml
  - Converts this value to another type
  - Returns the result which is then applied in the Binding again

- XAML already automatically uses some built-in converters:

```
<Rectangle Fill="#2494ab" />
```

➢ string to SolidColorBrush

```
rect.Fill = new BrushConverter().ConvertFrom("#2494ab") as SolidColorBrush;
```

```
<Image Source="https://i.imgflip.com/td7ad.jpg" />
```

➢ string to ImageSource / BitmapImage

```
img.Source = new BitmapImage(new Uri("https://i.imgflip.com/td7ad.jpg", UriKind.Absolute));
```

```
<TextBlock Margin="21,13" />
```

➢ string to Thickness, using 21 for left/right and 13 for top/bottom

```
txt.Margin = new Thickness(21, 13, 21, 13);
```

MAKING YOUR OWN VALUECONVERTER

- Converter = class that implements the IValueConverter interface

```
namespace System.Windows.Data
{
    public interface IValueConverter
    {
        object Convert(object value, Type targetType, object parameter, CultureInfo culture);
        object ConvertBack(object value, Type targetType, object parameter, CultureInfo culture);
    }
}
```

- The interface ("contract") forces you to implement:
  - Convert: from source to xaml (such as previous built-in examples)
    - `object value` = input when binding (eg. the gender string)
    - `Type targettype` = what type the output should be, eg. (SolidColor)Brush for Fill
    - `object parameter` = optional, ConverterParameter when Binding in xaml
  - ConvertBack: convert input from user back to classes' property type

EXAMPLE VALUECONVERTER

```
public class GenderToBooleanConverter : IValueConverter
{
    1 reference
    public object Convert(object value, Type targetType, object parameter, CultureInfo culture)
    {

    }

    //this is not always implemented; only when you want to allow user input from this type
    1 reference
    public object ConvertBack(object value, Type targetType, object parameter, CultureInfo culture)
    {

    }
}
```

EXAMPLE VALUECONVERTER

```
public class GenderToBooleanConverter : IValueConverter
{
    1 reference
    public object Convert(object value, Type targetType, object parameter, CultureInfo culture)
    {
        //value through binding; we assume this is a string
        String gender = value.ToString();
    }

    //this is not always implemented; only when you want to allow user input from this type
    1 reference
    public object ConvertBack(object value, Type targetType, object parameter, CultureInfo culture)
    {

    }
}
```

```
IsChecked="{Binding Gender, Conv
```

EXAMPLE VALUECONVERTER

```
public class GenderToBooleanConverter : IValueConverter
{
    1 reference
    public object Convert(object value, Type targetType, object parameter, CultureInfo culture)
    {
        //value through binding; we assume this is a string
        String gender = value.ToString();
        //output value in this case should be a boolean
        return gender.ToLower().Contains("male")
            || gender.ToLower().Contains("female") ? false : true;
    }

    //this is not always implemented; only when you want to allow user input from this type
    1 reference
    public object ConvertBack(object value, Type targetType, object parameter, CultureInfo culture)
    {

    }
}
```

```
IsChecked="{Binding Gender, Conv
```

EXAMPLE VALUECONVERTER

```
public class GenderToBooleanConverter : IValueConverter
{
    1 reference
    public object Convert(object value, Type targetType, object parameter, CultureInfo culture)
    {
        //value through binding; we assume this is a string
        String gender = value.ToString();
        //output value in this case should be a boolean
        return gender.ToLower().Contains("male")
            || gender.ToLower().Contains("female") ? false : true;
    }

    //this is not always implemented; only when you want to allow user input from this type
    1 reference
    public object ConvertBack(object value, Type targetType, object parameter, CultureInfo culture)
    {



    }
}
```

EXAMPLE VALUECONVERTER

```
public class GenderToBooleanConverter : IValueConverter
{
    1 reference
    public object Convert(object value, Type targetType, object parameter, CultureInfo culture)
    {
        //value through binding; we assume this is a string
        String gender = value.ToString();
        //output value in this case should be a boolean
        return gender.ToLower().Contains("male")
            || gender.ToLower().Contains("female") ? false : true;
    }

    //this is not always implemented; only when you want to allow user input from this type
    1 reference
    public object ConvertBack(object value, Type targetType, object parameter, CultureInfo culture)
    {
        //reverse => input value is now a boolean
        Boolean isNotMaleOrFemale = (bool)value;



    }
}
```

```
IsChecked="{Binding Gender, Conv
```

```
public string Gender { get; set; }
```

EXAMPLE VALUECONVERTER

```
public class GenderToBooleanConverter : IValueConverter
{
    1 reference
    public object Convert(object value, Type targetType, object parameter, CultureInfo culture)
    {
        //value through binding; we assume this is a string
        String gender = value.ToString();
        //output value in this case should be a boolean
        return gender.ToLower().Contains("male")
            || gender.ToLower().Contains("female") ? false : true;
    }

    //this is not always implemented; only when you want to allow user input from this type
    1 reference
    public object ConvertBack(object value, Type targetType, object parameter, CultureInfo culture)
    {
        //reverse => input value is now a boolean
        Boolean isNotMaleOrFemale = (bool)value;
        //reverse => output value is now string (will be saved in class property)
        return isNotMaleOrFemale ? "intersex" : "(fe)male";
    }
}
```

```
IsChecked="{Binding Gender, Conv
```

```
public string Gender { get; set; }
```

ANOTHER EXAMPLE VALUECONVERTER

```
public class GenderConverter : IValueConverter
{
    1 reference
    public object Convert(object value, Type targetType, object parameter, CultureInfo culture)
    {
        //get binding value, string in this case
        String gender = value.ToString();

        //return a value based on the target type
        switch (gender.ToLower())
        {
            case "male":
            case "man":
                if (targetType == typeof(Brush))
                    return new SolidColorBrush(Colors.CornflowerBlue);
                else
                    return new BitmapImage(new Uri("pack://application:,,,/Resources/male.png", UriKi
            case "female":
            case "woman":
```

OK. BUT HOW DO I THEN USE THESE CONVERTERS?

1. Create the converter class inside a Converters folder in View
2. Add the xmlns namespace where you want to use it, eg.:
   - `xmlns:conv="clr-namespace:[redacted].View.Converters"`
3. Add the converter(s) as a StaticTesource to your xaml + give it a key:
   -
```
<Grid.Resources>
    <conv:GenderToBooleanConverter x:Key="GenderToBool" />
    <conv:GenderConverter x:Key="GenderConv"/>
</Grid.Resources>
```
4. Use the converter static resource(s) when Binding:

```
<CheckBox Grid.Row="2" Content="Hello world" FontSize="18"
          Foreground="{Binding Gender, Converter={StaticResource GenderConv}}"
          IsChecked="{Binding Gender, Converter={StaticResource GenderToBool}}"
```