样式表：

例：程序style\_test1

MainWindow.xaml

<Window x:Class="style\_test1.MainWindow"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

Title="MainWindow" Height="350" Width="525">

<Window.Resources>

<Style TargetType="TextBlock">

<Style.Setters>

<Setter Property="FontSize" Value="24" />

<Setter Property="TextDecorations" Value="Underline" />

<Setter Property="FontStyle" Value="Italic" />

</Style.Setters>

</Style>

</Window.Resources>

<StackPanel Margin="5">

<TextBlock Text="Hello WPF!" />

<TextBlock Text="This is a sample for Style!" />

<!--取消样式-->

<TextBlock Text="By Time" Style="{x:Null}" />

</StackPanel>

</Window>

Style的触发器Trigger

基本触发器：

例：程序style\_test2

MainWindow.xaml

<Window x:Class="style\_test2.MainWindow"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

Title="MainWindow" Height="350" Width="525">

<Window.Resources>

<Style TargetType="CheckBox">

<Style.Triggers>

<Trigger Property="IsChecked" Value="true">

<Trigger.Setters>

<Setter Property="FontSize" Value="20" />

<Setter Property="Foreground" Value="Red" />

</Trigger.Setters>

</Trigger>

</Style.Triggers>

</Style>

</Window.Resources>

<StackPanel>

<CheckBox Content="悄悄的我走了" Margin="5"/>

</StackPanel>

</Window>

当CheckBox被选中时，就会触发设置的Style，字体和颜色会发生改变。

多条件触发器：多个条件同时满足才会触发。

例：程序style\_test3

MainWindow.xaml

<Window x:Class="style\_test3.MainWindow"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

Title="MainWindow" Height="350" Width="525">

<Window.Resources>

<Style TargetType="CheckBox">

<Style.Triggers>

<MultiTrigger>

<MultiTrigger.Conditions>

<Condition Property="IsChecked" Value="true" />

<Condition Property="Content" Value="正如我悄悄的来" />

</MultiTrigger.Conditions>

<MultiTrigger.Setters>

<Setter Property="FontSize" Value="20" />

<Setter Property="Foreground" Value="Red" />

</MultiTrigger.Setters>

</MultiTrigger>

</Style.Triggers>

</Style>

</Window.Resources>

<StackPanel>

<CheckBox Content="悄悄的我走了" Margin="5"/>

<CheckBox Content="正如我悄悄的来" Margin="5" />

</StackPanel>

</Window>

数据触发器：DataTrigger

例：程序style\_test4

MainWindow.xaml

<Window x:Class="style\_test4.MainWindow"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:local="clr-namespace:style\_test4"

Title="MainWindow" Height="130" Width="300">

<Window.Resources>

<local:L2BConvert x:Key="cvtr" />

<Style TargetType="TextBox">

<Style.Triggers>

<!--将控件自己作为数据源-->

<DataTrigger Binding="{Binding RelativeSource={x:Static RelativeSource.Self},

Path=Text.Length, Converter={StaticResource cvtr}}" Value="false">

<Setter Property="BorderBrush" Value="Red" />

<Setter Property="BorderThickness" Value="1" />

</DataTrigger>

</Style.Triggers>

</Style>

</Window.Resources>

<StackPanel>

<TextBox Margin="5" />

<TextBox Margin="5,0" />

<TextBox Margin="5" />

</StackPanel>

</Window>

L2BConvert.cs

using System;

using System.Collections.Generic;

using System.Globalization;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Data;

using System.Windows.Media;

using System.Windows.Media.Imaging;

namespace style\_test4

{

public class L2BConvert:IValueConverter

{

public object Convert(object value, Type targetType, object parameter, CultureInfo culture)

{

int textLength = (int)value;

return textLength > 6 ? true : false;

}

public object ConvertBack(object value, Type targetType, object parameter, CultureInfo culture)

{

throw new NotImplementedException();

}

}

}

多数据条件触发器：MultiDataTrigger

例：程序style\_test5

MainWindow.xaml

<Window x:Class="style\_test5.MainWindow"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

Title="MainWindow" Height="146" Width="300">

<Window.Resources>

<Style TargetType="ListBoxItem">

<!--使用Style设置的DataTemplate-->

<Setter Property="ContentTemplate">

<Setter.Value>

<DataTemplate>

<StackPanel Orientation="Horizontal">

<TextBlock Text="{Binding ID}" Width="60" />

<TextBlock Text="{Binding Name}" Width="120" />

<TextBlock Text="{Binding Age}" Width="60" />

</StackPanel>

</DataTemplate>

</Setter.Value>

</Setter>

<!--MultiDataTrigger-->

<Style.Triggers>

<MultiDataTrigger>

<MultiDataTrigger.Conditions>

<Condition Binding="{Binding Path=ID}" Value="2" />

<Condition Binding="{Binding Path=Name}" Value="Tom" />

</MultiDataTrigger.Conditions>

<MultiDataTrigger.Setters>

<Setter Property="Background" Value="Orange" />

</MultiDataTrigger.Setters>

</MultiDataTrigger>

</Style.Triggers>

</Style>

</Window.Resources>

<StackPanel>

<ListBox x:Name="listBoxStudent" Margin="5" />

</StackPanel>

</Window>

MainWindow.xaml.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

namespace style\_test5

{

/// <summary>

/// MainWindow.xaml 的交互逻辑

/// </summary>

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

InitList();

}

void InitList()

{

List<Student> studentList = new List<Student>()

{

new Student {ID = "1", Name = "Tim", Age = "28"},

new Student {ID = "2", Name = "Tom", Age = "27"}

};

listBoxStudent.ItemsSource = studentList;

}

}

}

Student.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace style\_test5

{

public class Student

{

public string ID { get; set; }

public string Name { get; set; }

public string Age { get; set; }

}

}