**24장. 스따일**

엘리먼트에 적용 되는 프로퍼티의 집합

루프를 사용하지 못하는 XAML에서 동일한 프로퍼티를 가진 엘리먼트를 여러 개 생성할 때 사용하기 적합하다

Style에서 가장 중요한 것은 Setters 프로퍼티이다.

<Setter Property="Button.FontSize" Value="10pt" />

Property 속성은 항상 의존 프로퍼티를 참조하지만 Fontsizeproperty가 아닌 Fontsize로 명시된다.

Value 속성에 null값을 하고 싶을때는 Value="{x:Null}" 로 입력하면 된다.

style 은 6개의 프로퍼티를 정의한다.

Setters, Resources, TargetType, BasedOn, IsSealed, Triggers

예제 1

버튼 안에 스타일 엘리먼트를 만들고 버튼의 fontsize와 foreground를 정의하기 위한 2개의 Setter를 포함하는 버튼

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

HorizontalAlignment="Center" VerticalAlignment="Center"

Foreground="Blue">

<Button.Style>

<Style>

<Setter Property="Button.FontSize" Value="10pt" />

<Setter Property="Control.Foreground" Value="Blue" />

</Style>

</Button.Style>

Button with Local Style

</Button>

예제2

StackPanel.Resource안에 스타일을 정의한 후에 StackPanel안에서 버튼에 스타일 지정하기

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

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

<StackPanel.Resources>

<Style x:Key="normal">

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

<Setter Property="Control.Foreground" Value="Blue" />

<Setter Property="Control.HorizontalAlignment" Value="Center" />

<Setter Property="Control.Margin" Value="24" />

<Setter Property="Control.Padding" Value="20, 10, 20, 10" />

</Style>

</StackPanel.Resources>

<Button Style="{StaticResource normal}">

Button Number 1

</Button>

<Button Style="{StaticResource normal}">

Button Number 2

</Button>

<Button Style="{StaticResource normal}">

Button Number 3

</Button>

</StackPanel>

예제3

다양한 컨트롤 안에 같은 스타일 지정하기

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

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

<StackPanel.Resources>

<Style x:Key="normal">

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

<Setter Property="Control.Foreground" Value="Blue" />

<Setter Property="Control.HorizontalAlignment" Value="Center" />

<Setter Property="Control.Margin" Value="24" />

<Setter Property="Control.Padding" Value="20, 10, 20, 10" />

<Setter Property="Button.IsDefault" Value="true" />

</Style>

</StackPanel.Resources>

<Button Style="{StaticResource normal}">

Button on top of the stack

</Button>

<TextBlock Style="{StaticResource normal}">

TextBlock in the middle of the stack

</TextBlock>

<Button Style="{StaticResource normal}">

Button on the bottom of the stack

</Button>

</StackPanel>

TextBlock에 IsDefault 속성이 없기 때문에 Button.IsDefault 속성을 무시하고 적용된다

예제4

TargetType속성의 등장

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

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

<Grid.Resources>

<Style TargetType="{x:Type Button}">

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

<Setter Property="Control.Foreground" Value="Blue" />

</Style>

</Grid.Resources>

<StackPanel>

<StackPanel.Resources>

<Style TargetType="{x:Type Button}">

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

</Style>

</StackPanel.Resources>

<Button>

Button Number 1

</Button>

<Button>

Button Number 2

</Button>

<Button>

Button Number 3

</Button>

</StackPanel>

</Grid>

예제5

Setter 안에 Gradient를 설정하는 스타일

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

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

<StackPanel.Resources>

<Style x:Key="normal">

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

<Setter Property="Control.HorizontalAlignment" Value="Center" />

<Setter Property="Control.Margin" Value="24" />

<Setter Property="Control.Background">

<Setter.Value>

<LinearGradientBrush StartPoint="1,0" EndPoint="1,1">

<LinearGradientBrush.GradientStops>

<GradientStop Color="LightBlue" Offset="0" />

<GradientStop Color="Aquamarine" Offset="1" />

</LinearGradientBrush.GradientStops>

</LinearGradientBrush>

</Setter.Value>

</Setter>

</Style>

</StackPanel.Resources>

<Button Style="{StaticResource normal}">

Button Number 1

</Button>

<Button Style="{StaticResource normal}">

Button Number 2

</Button>

<Button Style="{StaticResource normal}">

Button Number 3

</Button>

</StackPanel>

예제6

Resource속성안에 Gradient를 정의한뒤에 사용하기

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

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

<StackPanel.Resources>

<Style x:Key="normal">

<Style.Resources>

<LinearGradientBrush x:Key="gradbrush"

StartPoint="1,0" EndPoint="1,1">

<LinearGradientBrush.GradientStops>

<GradientStop Color="LightBlue" Offset="0" />

<GradientStop Color="Aquamarine" Offset="1" />

</LinearGradientBrush.GradientStops>

</LinearGradientBrush>

</Style.Resources>

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

<Setter Property="Control.HorizontalAlignment" Value="Center" />

<Setter Property="Control.Margin" Value="24" />

<Setter Property="Control.Background"

Value="{StaticResource gradbrush}" />

</Style>

</StackPanel.Resources>

<Button Style="{StaticResource normal}">

Button Number 1

</Button>

<Button Style="{StaticResource normal}">

Button Number 2

</Button>

<Button Style="{StaticResource normal}">

Button Number 3

</Button>

</StackPanel>

예제7

TargetType속성에 대한 사용법

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

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

<StackPanel.Resources>

<Style TargetType="{x:Type Button}">

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

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

</Style>

<Style TargetType="{x:Type TextBlock}">

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

</Style>

</StackPanel.Resources>

<Button>

Button with Text Content

</Button>

<TextBlock>

TextBlock Text

</TextBlock>

<Button>

<TextBlock>

Button with TextBlock Content

</TextBlock>

</Button>

</StackPanel>

예제 8

BasedOn속성의 사용법

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

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

<StackPanel.Resources>

<Style x:Key="normal">

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

<Setter Property="Control.Foreground" Value="Blue" />

<Setter Property="Control.HorizontalAlignment" Value="Center" />

<Setter Property="Control.Margin" Value="24" />

<Setter Property="Control.Padding" Value="20, 10, 20, 10" />

</Style>

<Style x:Key="hotbtn" BasedOn="{StaticResource normal}">

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

</Style>

</StackPanel.Resources>

<Button Style="{StaticResource normal}">

Button Number 1

</Button>

<Button Style="{StaticResource hotbtn}">

Button Number 2

</Button>

<Button Style="{StaticResource normal}">

Button Number 3

</Button>

</StackPanel>

예제9

BasedOn속성과 TargetType속성 같이 사용하기

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

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

<StackPanel.Resources>

<Style TargetType="{x:Type Button}">

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

<Setter Property="Control.Foreground" Value="Blue" />

<Setter Property="Control.HorizontalAlignment" Value="Center" />

<Setter Property="Control.Margin" Value="24" />

</Style>

<Style x:Key="hotbtn"

TargetType="{x:Type Button}"

BasedOn="{StaticResource {x:Type Button}}">

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

</Style>

</StackPanel.Resources>

<Button>

Button Number 1

</Button>

<Button Style="{StaticResource hotbtn}">

Button Number 2

</Button>

<Button>

Button Number 3

</Button>

</StackPanel>

예제10

TargetType의 여러가지 타입에 적용

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

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

<StackPanel.Resources>

<Style TargetType="{x:Type Control}">

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

<Setter Property="Control.Foreground" Value="Blue" />

<Setter Property="Control.HorizontalAlignment" Value="Center" />

<Setter Property="Control.Margin" Value="24" />

</Style>

<Style TargetType="{x:Type Button}"

BasedOn="{StaticResource {x:Type Control}}">

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

</Style>

<Style TargetType="{x:Type Label}"

BasedOn="{StaticResource {x:Type Control}}">

<Setter Property="Control.Foreground" Value="Green" />

</Style>

<Style TargetType="{x:Type TextBox}"

BasedOn="{StaticResource {x:Type Control}}">

</Style>

</StackPanel.Resources>

<Button>

Button Control

</Button>

<Label>

Label Control

</Label>

<TextBox>

TextBox Control

</TextBox>

</StackPanel>

예제11

Setter에 바인딩 하기

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

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

<StackPanel.Resources>

<Style TargetType="{x:Type Button}">

<Setter Property="FontSize"

Value="{Binding ElementName=scroll, Path=Value}" />

<Setter Property="HorizontalAlignment" Value="Center" />

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

</Style>

</StackPanel.Resources>

<ScrollBar Name="scroll" Orientation="Horizontal" Margin="24"

Minimum="11" Maximum="100" Value="24" />

<Button>

Button Number 1

</Button>

<Button>

Button Number 2

</Button>

<Button>

Button Number 3

</Button>

</StackPanel>

예제12

Canvas에서의 스타일 정의하기

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

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

<Canvas.Resources>

<Style TargetType="{x:Type Ellipse}">

<Setter Property="Stroke" Value="Black" />

<Setter Property="StrokeThickness" Value="3" />

<Setter Property="Width" Value="96" />

<Setter Property="Height" Value="96" />

</Style>

</Canvas.Resources>

<Ellipse Canvas.Left="100" Canvas.Top="50" Fill="Blue" />

<Ellipse Canvas.Left="150" Canvas.Top="100" Fill="Red" />

<Ellipse Canvas.Left="200" Canvas.Top="150" Fill="Green" />

<Ellipse Canvas.Left="250" Canvas.Top="100" Fill="Cyan" />

<Ellipse Canvas.Left="300" Canvas.Top="50" Fill="Magenta" />

</Canvas>

예제13

반복적인 작업을 대체할 수 있는 스타일의 예

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

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

<Canvas.Resources>

<Style x:Key="base" TargetType="Line">

<Setter Property="Stroke" Value="Blue" />

</Style>

<Style x:Key="horz" TargetType="Line"

BasedOn="{StaticResource base}">

<Setter Property="X1" Value="100" />

<Setter Property="X2" Value="300" />

<Setter Property="Y2"

Value="{Binding RelativeSource={RelativeSource self},

Path=Y1}" />

</Style>

<Style x:Key="vert" TargetType="Line"

BasedOn="{StaticResource base}">

<Setter Property="Y1" Value="100" />

<Setter Property="Y2" Value="300" />

<Setter Property="X2"

Value="{Binding RelativeSource={RelativeSource self},

Path=X1}" />

</Style>

</Canvas.Resources>

<Line Style="{StaticResource horz}" Y1="100" />

<Line Style="{StaticResource horz}" Y1="125" />

<Line Style="{StaticResource horz}" Y1="150" />

<Line Style="{StaticResource horz}" Y1="175" />

<Line Style="{StaticResource horz}" Y1="200" />

<Line Style="{StaticResource horz}" Y1="225" />

<Line Style="{StaticResource horz}" Y1="250" />

<Line Style="{StaticResource horz}" Y1="275" />

<Line Style="{StaticResource horz}" Y1="300" />

<Line Style="{StaticResource vert}" X1="100" />

<Line Style="{StaticResource vert}" X1="125" />

<Line Style="{StaticResource vert}" X1="150" />

<Line Style="{StaticResource vert}" X1="175" />

<Line Style="{StaticResource vert}" X1="200" />

<Line Style="{StaticResource vert}" X1="225" />

<Line Style="{StaticResource vert}" X1="250" />

<Line Style="{StaticResource vert}" X1="275" />

<Line Style="{StaticResource vert}" X1="300" />

</Canvas>

예제14

Page에서의 스타일 설정하기

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

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

Title="I. Down the Rabbit-Hole">

<Page.Resources>

<Style TargetType="{x:Type Paragraph}" x:Key="Normal">

<Setter Property="TextIndent" Value="0.25in" />

</Style>

<Style TargetType="{x:Type Paragraph}" x:Key="ChapterHead">

<Setter Property="TextAlignment" Value="Center" />

<Setter Property="FontSize" Value="16pt" />

</Style>

</Page.Resources>

<FlowDocumentReader>

<FlowDocument>

<Paragraph Style="{StaticResource ChapterHead}">

Chapter I

</Paragraph>

<Paragraph Style="{StaticResource ChapterHead}">

Down the Rabbit-Hole

</Paragraph>

<Paragraph Style="{StaticResource Normal}">

Alice was beginning to get very tired of sitting by

her sister on the bank, and of having nothing to do:

once or twice she had peeped into the book her sister

was reading, but it had no pictures or conversations

in it, &#x201C;and what is the use of a book,&#x201D;

thought Alice, &#x201C;without pictures or

conversations?&#x201D;

</Paragraph>

<Paragraph Style="{StaticResource Normal}">

...

</Paragraph>

</FlowDocument>

</FlowDocumentReader>

</Page>

예제15

EventSetter의 예제

XAML 파일

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

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

x:Class="Petzold.EventSetterDemo.EventSetterDemo"

Title="EventSetter Demo">

<Window.Resources>

<Style TargetType="{x:Type Button}">

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

<Setter Property="HorizontalAlignment" Value="Center" />

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

<EventSetter Event="Click" Handler="ButtonOnClick" />

</Style>

</Window.Resources>

<StackPanel>

<Button>

Button Number 1

</Button>

<Button>

Button Number 2

</Button>

<Button>

Button Number 3

</Button>

</StackPanel>

</Window>

CS파일

namespace Petzold.EventSetterDemo

{

public partial class EventSetterDemo : Window

{

[STAThread]

public static void Main()

{

Application app = new Application();

app.Run(new EventSetterDemo());

}

public EventSetterDemo()

{

InitializeComponent();

}

void ButtonOnClick(object sender, RoutedEventArgs args)

{

Button btn = args.Source as Button;

MessageBox.Show("The button labeled " + btn.Content +

" has been clicked", Title);

}

}

}

예제16

Trigger 속성 사용하기

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

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

<StackPanel.Resources>

<Style x:Key="normal">

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

<Setter Property="Control.HorizontalAlignment" Value="Center" />

<Setter Property="Control.Margin" Value="24" />

<Style.Triggers>

<Trigger Property="Control.IsMouseOver" Value="true">

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

<Setter Property="Control.Foreground" Value="Blue" />

</Trigger>

<Trigger Property="Button.IsPressed" Value="true">

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

</Trigger>

</Style.Triggers>

</Style>

</StackPanel.Resources>

<Button Style="{StaticResource normal}">

Button Number 1

</Button>

<Button Style="{StaticResource normal}">

Button Number 2

</Button>

<Button Style="{StaticResource normal}">

Button Number 3

</Button>

</StackPanel>

예제17

MultiTrigger 사용하기

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

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

<StackPanel.Resources>

<Style x:Key="normal">

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

<Setter Property="Control.HorizontalAlignment" Value="Center" />

<Setter Property="Control.Margin" Value="24" />

<Style.Triggers>

<Trigger Property="Button.IsPressed" Value="True">

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

</Trigger>

<MultiTrigger>

<MultiTrigger.Conditions>

<Condition Property="Control.IsMouseOver" Value="True" />

<Condition Property="Button.IsPressed" Value="False" />

</MultiTrigger.Conditions>

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

<Setter Property="Control.Foreground" Value="Blue" />

</MultiTrigger>

</Style.Triggers>

</Style>

</StackPanel.Resources>

<Button Style="{StaticResource normal}">

Button Number 1

</Button>

<Button Style="{StaticResource normal}">

Button Number 2

</Button>

<Button Style="{StaticResource normal}">

Button Number 3

</Button>

</StackPanel>

예제18

DataTrigger 사용하기

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

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

<StackPanel.Resources>

<Style TargetType="{x:Type Button}">

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

<Setter Property="HorizontalAlignment" Value="Center" />

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

<Style.Triggers>

<DataTrigger Binding="{Binding ElementName=txtbox,

Path=Text.Length}"

Value="0">

<Setter Property="IsEnabled" Value="False" />

</DataTrigger>

</Style.Triggers>

</Style>

</StackPanel.Resources>

<TextBox Name="txtbox" HorizontalAlignment="Center"

Width="2in" Margin="24" />

<Button>

Button Number 1

</Button>

<Button>

Button Number 2

</Button>

<Button>

Button Number 3

</Button>

</StackPanel>

예제19

MultiDataTrigger 사용하기

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

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

<StackPanel.Resources>

<Style TargetType="{x:Type CheckBox}">

<Setter Property="HorizontalAlignment" Value="Center" />

<Setter Property="Margin" Value="12" />

</Style>

<Style TargetType="{x:Type Button}">

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

<Setter Property="HorizontalAlignment" Value="Center" />

<Setter Property="Margin" Value="12" />

<Setter Property="IsEnabled" Value="False" />

<Style.Triggers>

<MultiDataTrigger>

<MultiDataTrigger.Conditions>

<Condition Binding="{Binding ElementName=chkbox1,

Path=IsChecked}"

Value="True" />

<Condition Binding="{Binding ElementName=chkbox2,

Path=IsChecked}"

Value="True" />

</MultiDataTrigger.Conditions>

<Setter Property="IsEnabled" Value="True" />

</MultiDataTrigger>

</Style.Triggers>

</Style>

</StackPanel.Resources>

<CheckBox Name="chkbox1">

Check 1

</CheckBox>

<CheckBox Name="chkbox2">

Check 2

</CheckBox>

<Button>

Button Number 1

</Button>

<Button>

Button Number 2

</Button>

<Button>

Button Number 3

</Button>

</StackPanel>