

# 样式系统

张典

An ordinary Avalonia Developer | IRIHI staff

样式是什么

以CSS的语法为基础  
以XAML的形式组织

# 声明样式

# 声明样式

--核心Styles

# 声明样式

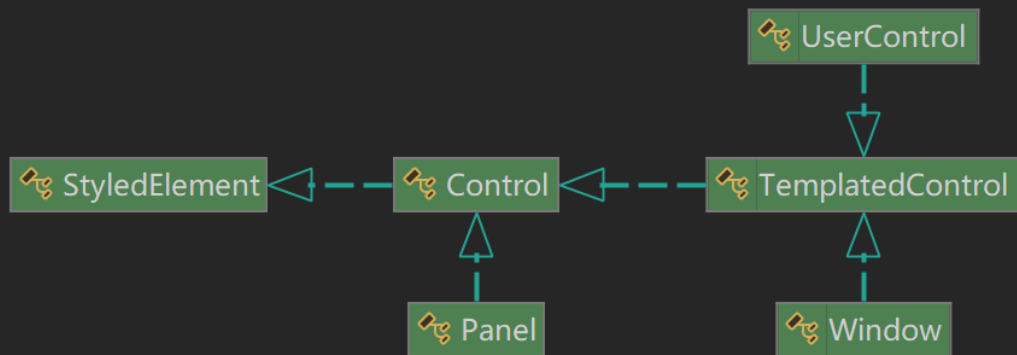
# 声明样式

```
public class StyledElement : Animatable,
```

🔗 259+6 用法    👤 Jeremy Koritzinsky +1

```
public Styles Styles ⇒ _styles ??= new Styles(owner: this);
```

# 声明样式



```
<Application.Styles>
  <Style Selector="Button">
    <Setter Property="FontSize" Value="50" />
    <Setter Property="FontWeight" Value="Bold" />
    <Setter Property="Foreground" Value="Blue" />
    <Setter Property="Background" Value="Red" />
  </Style>
</Application.Styles>
```

Application的Styles属性

```
<UserControl.Styles>
  <Style Selector="Button">
    <Setter Property="FontSize" Value="50" />
    <Setter Property="FontWeight" Value="Bold" />
    <Setter Property="Foreground" Value="Blue" />
    <Setter Property="Background" Value="Red" />
  </Style>
</UserControl.Styles>
```

StyledElement的Styles属性

```
<Styles
  xmlns="https://github.com/avaloniaui"
  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml">
  <Style Selector="Button">
    <Setter Property="FontSize" Value="50" />
    <Setter Property="FontWeight" Value="Bold" />
    <Setter Property="Foreground" Value="Blue" />
    <Setter Property="Background" Value="Red" />
  </Style>
</Styles>
```

Styles文件



# 样式选择器

# 样式选择器

类型选择器

派生类型选择器

名称选择器

类选择器

伪类选择器

属性选择器

子选择器

任意子选择器

非选择器

或选择器

嵌套选择器

模板选择器

子位置选择器

# 伪类 (PseudoClasses)

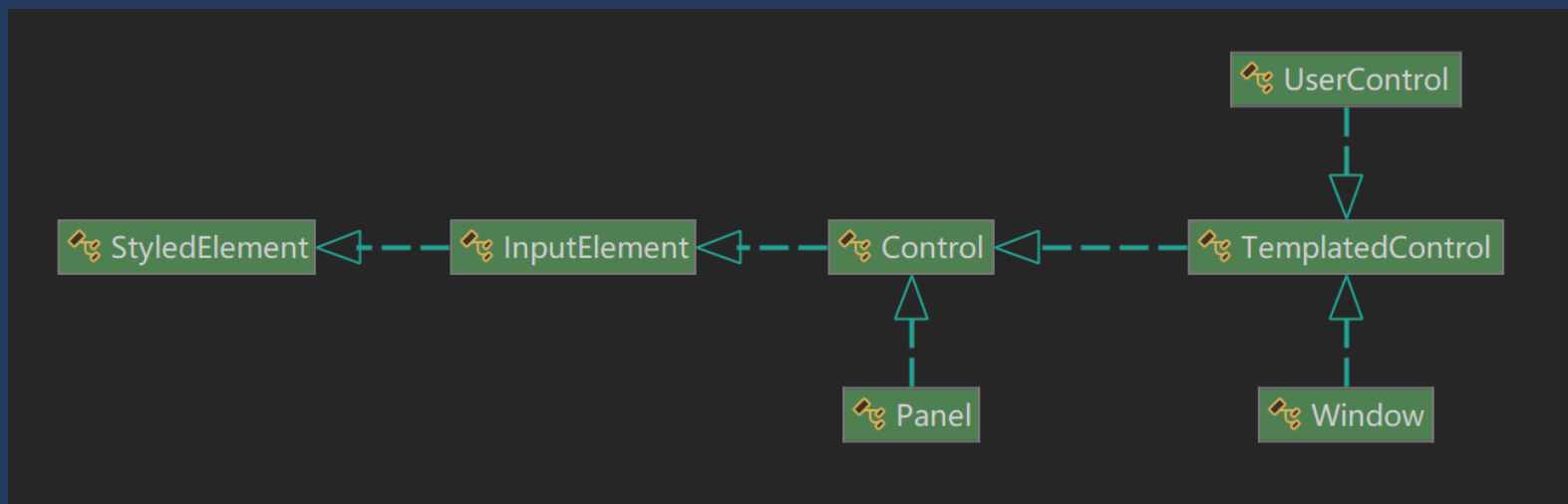
## InputElement

- :disabled 禁用状态
- :pointerover 鼠标悬停状态
- :focus 获得焦点状态
- :focus-visible 可见焦点状态
- :focus-within 包含焦点状态

## Button

- :pressed 按下状态
- :flyout-open Flyout打开状态

# 伪类 (PseudoClasses)



# 伪类 (PseudoClasses)

形似伪类但其实不是伪类的选择器:

派生类型选择器                    :is

非选择器                         :not

子位置选择器                    :nth-child/:nth-last-child

# 样式选择器

```
<Style Selector="^.OnlyIcon">
  <Style Selector="^ /template/ PathIcon#PART_LinkGlyph">
    <Setter Property="Margin" Value="[all:] 0" />
    <Setter Property="IsVisible" Value="True" />
  </Style>
  <Style Selector="^ /template/ ContentPresenter#PART_ContentPresenter">
    <Setter Property="IsVisible" Value="False" />
  </Style>
</Style>
```

HyperlinkButton.axaml

# 样式选择器

```
<Style Selector=".clearButton, ^.ClearButton">
  <Style Selector="^[AcceptsReturn=False][ReadOnly=False]:not(:empty)">
    <Style Selector="^:pointerover,^:focus">
      <Style Selector="^ /template/ Button#PART_ClearButton">
        <Setter Property="IsVisible" Value="True" />
      </Style>
    </Style>
  </Style>
</Style>
```

TextBox.axaml

# 样式的优先级



# 样式的优先级

## ——就近原则

# 样式的优先级

Application

|- Styles [4]

UserControl

|- Styles [3]

|- StackPanel

|- Styles [2]

|- Button

|- Styles [1]

# BindingPriority 枚举

Animation = -1

动画

LocalValue = 0

本地值

StyleTrigger = 1

样式激活状态

Template = 2

模板

Style = 3

样式未激活状态

Inherited = 4

继承

Unset = 2147483647(int.MaxValue)

未设置

# BindingPriority = StyleTrigger

类选择器

Button.Large

伪类选择器

:focus、:pointerover

属性选择器

Button[FontWeight=Normal]

子位置选择器

:nth-child(odd)

:nth-last-child(even)

# BindingPriority = Style

类型选择器

Button、ava|Button

派生类型选择器

:is(Button)

名称选择器

Button#NamedButton

子选择器

StackPanel > Button

任意子选择器

StackPanel Button

非选择器

:not(#NamedButton)

或选择器

TextBlock, Button

嵌套选择器

^ > Button

模板选择器

Button /template/ ContentPresenter

# 小结

