



Avalonia 專案 安裝 SukiUI 版本號為6.0.3 套件

當App.axaml加上 `<sukiUi:SukiTheme ThemeColor="Blue" Locale="zh-TW" />` 語法
若MainWindow.axaml也將類別Window置換成SukiWindow 那麼此時，我要如何定義該MainWindow
的背景色設定為不具漸層的正綠色

在 SukiUI 6.0.3 版本中，使用 SukiWindow 後，其背景通常受 SukiTheme 控制，可能包含漸層效果。要設定 MainWindow.axaml 的背景為純正綠色（無漸層），直接在 SukiWindow 元素上使用 `Background="Green"` 屬性即可覆寫主題樣式。^[1] ^[2]

XAML 修改範例

將 MainWindow.axaml 修改如下：

```
<sukiUi:SukiWindow
    x:Class="YourNamespace.MainWindow"
    xmlns="https://github.com/avaloniaui"
    xmlns:sukiUi="clr-namespace:SukiUI.Controls;assembly=SukiUI"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    Background="Green"> <!-- 正綠色，無漸層 -->

    <!-- 您的內容 -->

</sukiUi:SukiWindow>
```

此設定會讓整個視窗背景變為固定的純綠色（#FF008000），忽略 SukiTheme 的漸層背景。^[2] ^[1]

注意事項

- 正綠色對應 CSS 顏色 "Green" 或 Brushes.Green，若需自訂色值，可用 "#00FF00" 或 "rgb(0,255,0)"。^[3]
- 若內容區域需獨立背景，可在內部 Grid 或 Panel 上再設 Background，並確保父級 Background 不影響拖曳。^[1]
- 確認 App.axaml 中的 SukiTheme 已正確設定 ThemeColor="Blue" 和 Locale="zh-TW"，以避免透明問題。^[2]



如何在SukiWindow設定單一純色背景

在 SukiUI 中，SukiWindow 繼承自 Avalonia 的 Window，因此直接使用 Background 屬性即可設定單一純色背景，覆寫主題的漸層或預設樣式。[\[18\]](#) [\[19\]](#)

XAML 設定方式

在 MainWindow.xaml 的根元素 SukiWindow 上添加 Background：

```
<sukiUi:SukiWindow
    xmlns:sukiUi="clr-namespace:SukiUI.Controls;assembly=SukiUI"
    Background="Green">

    <!-- 視窗內容 -->

</sukiUi:SukiWindow>
```

此方式產生無漸層的純綠色背景 (#FF008000)，適用於 SukiTheme 已啟用的專案。[\[20\]](#) [\[18\]](#)

顏色指定選項

- **預設顏色名稱**：如 "Green"、"Red" 或 "Transparent"。
- **十六進位**：Background="#00FF00"（純綠）。
- **RGB**：Background="rgb(0,255,0)"。
- **動態資源**：若需主題變體支援，可用 <SolidColorBrush Color="Green"/> 或 DynamicResource。[\[19\]](#) [\[20\]](#)

若背景未生效，確認命名空間正確並無衝突樣式。[\[21\]](#)

✖

如何在 SukiWindow 中設定 BackgroundBrush 資源

在 Avalonia 的 SukiWindow 中，BackgroundBrush 資源可透過 ResourceDictionary 定義，支援靜態或動態引用，允許主題變體（如明亮/暗黑模式）切換純色刷。[\[28\]](#) [\[29\]](#)

資源定義位置

將資源置於 App.xaml、Window.Resources 或外部 .axaml 檔案中（合併至 App）：

```
<Window.Resources>
    <SolidColorBrush x:Key="MyBackgroundBrush" Color="Green" />
</Window.Resources>
```

或支援主題變體：

```
<Window.Resources>
    <ResourceDictionary.ThemeDictionaries>
```

```

    <ResourceDictionary x:Key="Light">
        <SolidColorBrush x:Key="BackgroundBrush" Color="Green" />
    </ResourceDictionary>
    <ResourceDictionary x:Key="Dark">
        <SolidColorBrush x:Key="BackgroundBrush" Color="DarkGreen" />
    </ResourceDictionary>
</ResourceDictionary.ThemeDictionaries>
</Window.Resources>

```

此定義讓刷在不同主題下自動切換單一純色。[\[30\]](#) [\[28\]](#)

SukiWindow 使用方式

在 SukiWindow 上引用資源：

```

<sukiUi:SukiWindow Background="{DynamicResource BackgroundBrush}">
    <!-- 內容 -->
</sukiUi:SukiWindow>

```

使用 DynamicResource 以支援運行時變更；若僅靜態，改用 StaticResource。[\[29\]](#) [\[31\]](#)

✖

示範在 App.axaml 定義 BackgroundBrush 的完整範例

在 Avalonia 的 App.axaml 中定義全域 BackgroundBrush 資源，讓 SukiWindow 可透過 DynamicResource 引用，實現純色背景設定。

App.axaml 完整範例

```

<Application xmlns="https://github.com/avaloniaui"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:sukiUi="clr-namespace:SukiUI.Controls;assembly=SukiUI"
    x:Class="YourNamespace.App">

    <Application.Styles>
        <!-- SukiUI 主題 -->
        <sukiUi:SukiTheme ThemeColor="Blue" Locale="zh-TW" />
    </Application.Styles>

    <!-- 全域資源定義 -->
    <Application.Resources>
        <!-- 單一純綠色背景刷 -->
        <SolidColorBrush x:Key="AppBackgroundBrush" Color="Green" />

        <!-- 或支援主題變體的背景刷 -->
        <ResourceDictionary.ThemeDictionaries>
            <ResourceDictionary x:Key="Light">
                <SolidColorBrush x:Key="AppBackgroundBrush" Color="#00FF00" />
            </ResourceDictionary>

```

```

        <ResourceDictionary x:Key="Dark">
            <SolidColorBrush x:Key="AppBackgroundBrush" Color="DarkGreen" />
        </ResourceDictionary>
    </ResourceDictionary.ThemeDictionaries>
</Application.Resources>

</Application>

```

MainWindow.axaml 使用範例

```

<sukiUi:SukiWindow xmlns="https://github.com/avaloniaui"
    xmlns:sukiUi="clr-namespace:SukiUI.Controls;assembly=SukiUI"
    x:Class="YourNamespace.MainWindow"
    Background="{DynamicResource AppBackgroundBrush}">

    <!-- 視窗內容 -->

</sukiUi:SukiWindow>

```

此設定讓所有 SukiWindow 自動使用 App.axaml 定義的純色背景，並支援運行時主題切換。^[38]

✱

提供 App.axaml 中 BackgroundBrush 的完整 XAML 範例

在 Avalonia 的 App.axaml 中定義全域 BackgroundBrush 資源，讓所有 SukiWindow 可統一引用純色背景。

App.axaml 完整範例

```

<Application xmlns="https://github.com/avaloniaui"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:sukiUi="clr-namespace:SukiUI.Controls;assembly=SukiUI"
    x:Class="YourNamespace.App">

    <Application.Styles>
        <!-- SukiUI 主題設定 -->
        <sukiUi:SukiTheme ThemeColor="Blue" Locale="zh-TW" />
    </Application.Styles>

    <!-- 全域資源字典 -->
    <Application.Resources>
        <!-- 單一純綠色背景刷 -->
        <SolidColorBrush x:Key="BackgroundBrush" Color="Green" />

        <!-- 或者支援明暗主題變體 -->
        <!--
        <ResourceDictionary.ThemeDictionaries>
            <ResourceDictionary x:Key="Light">
                <SolidColorBrush x:Key="BackgroundBrush" Color="#00FF00" />
            </ResourceDictionary>
        </ResourceDictionary.ThemeDictionaries>
        -->
    </Application.Resources>

```

```

        </ResourceDictionary>
        <ResourceDictionary x:Key="Dark">
            <SolidColorBrush x:Key="BackgroundBrush" Color="#008000" />
        </ResourceDictionary>
    </ResourceDictionary.ThemeDictionaries>
    -->
</Application.Resources>

</Application>

```

MainWindow.axaml 引用方式

```

<sukiUi:SukiWindow xmlns="https://github.com/avaloniaui"
    xmlns:sukiUi="clr-namespace:SukiUI.Controls;assembly=SukiUI"
    x:Class="YourNamespace.MainWindow"
    Background="{DynamicResource BackgroundBrush}">

    <Grid>
        <!-- 視窗內容 -->
    </Grid>

</sukiUi:SukiWindow>

```

此範例中，BackgroundBrush 資源定義於 App.axaml，全域可用於任何 SukiWindow，並覆寫 SukiTheme 的預設漸層背景。[\[49\]](#)

✱

如何在 App.axaml 使用動態資源切換背景顏色

在 Avalonia 的 App.axaml 中使用動態資源切換背景顏色，透過 ThemeDictionaries 定義明暗主題變體，並搭配 DynamicResource 實現運行時自動切換。

App.axaml 完整範例

```

<Application xmlns="https://github.com/avaloniaui"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:sukiUi="clr-namespace:SukiUI.Controls;assembly=SukiUI"
    x:Class="YourNamespace.App">

    <Application.Styles>
        <sukiUi:SukiTheme ThemeColor="Blue" Locale="zh-TW" />
    </Application.Styles>

    <Application.Resources>
        <!-- 動態背景刷：根據系統主題自動切換 -->
        <ResourceDictionary.ThemeDictionaries>
            <!-- 淺色主題背景 -->
            <ResourceDictionary x:Key="Light">
                <SolidColorBrush x:Key="BackgroundBrush" Color="#00FF00" /> <!-- 純綠 -->
            </ResourceDictionary>

```

```

        <!-- 深色主題背景 -->
        <ResourceDictionary x:Key="Dark">
            <SolidColorBrush x:Key="BackgroundBrush" Color="#008000" /> <!-- 暗綠 -->
        </ResourceDictionary>
    </ResourceDictionary.ThemeDictionaries>
</Application.Resources>

</Application>

```

MainWindow.axaml 使用方式

```

<sukiUi:SukiWindow xmlns="https://github.com/avaloniaui"
    xmlns:sukiUi="clr-namespace:SukiUI.Controls;assembly=SukiUI"
    x:Class="YourNamespace.MainWindow"
    Background="{DynamicResource BackgroundBrush}">

    <Grid>
        <!-- 內容將自動適應主題背景色 -->
    </Grid>

</sukiUi:SukiWindow>

```

程式碼切換主題（可選）

在 App.axaml.cs 中可程式化切換：

```

public partial class App : Application
{
    public override void Initialize()
    {
        AvaloniaXamlLoader.Load(this);
    }

    public void SwitchToDarkTheme()
    {
        RequestedThemeVariant = ThemeVariant.Dark;
    }
}

```

系統主題變更時，DynamicResource 會自動更新所有引用 BackgroundBrush 的控制項。[\[60\]](#) [\[61\]](#)

✱

示範在 App.axaml 用 AppThemeBinding 切換背景色

Avalonia 不支援 .NET MAUI 的 AppThemeBinding 語法，但可透過 ThemeDictionaries 結合 DynamicResource 實現類似動態主題切換背景色效果。

App.axaml 完整範例 (模擬 AppThemeBinding)

```
<Application xmlns="https://github.com/avaloniaui"
              xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
              xmlns:sukiUi="clr-namespace:SukiUI.Controls;assembly=SukiUI"
              x:Class="YourNamespace.App">

    <Application.Styles>
        <sukiUi:SukiTheme ThemeColor="Blue" Locale="zh-TW" />
    </Application.Styles>

    <Application.Resources>
        <!-- 主題變體背景資源 (類 AppThemeBinding 效果) -->
        <ResourceDictionary.ThemeDictionaries>
            <!-- Light 主題：純綠色 -->
            <ResourceDictionary x:Key="Light">
                <SolidColorBrush x:Key="BackgroundBrush" Color="#00FF00" />
            </ResourceDictionary>

            <!-- Dark 主題：暗綠色 -->
            <ResourceDictionary x:Key="Dark">
                <SolidColorBrush x:Key="BackgroundBrush" Color="#008000" />
            </ResourceDictionary>
        </ResourceDictionary.ThemeDictionaries>
    </Application.Resources>

</Application>
```

MainWindow.axaml 使用動態資源

```
<sukiUi:SukiWindow xmlns="https://github.com/avaloniaui"
                   xmlns:sukiUi="clr-namespace:SukiUI.Controls;assembly=SukiUI"
                   x:Class="YourNamespace.MainWindow"
                   Background="{DynamicResource BackgroundBrush}">

    <Grid>
        <!-- 內容自動適應主題背景 -->
        <TextBlock Text="主題切換測試" HorizontalAlignment="Center"
                   VerticalAlignment="Center" />
    </Grid>

</sukiUi:SukiWindow>
```

程式碼觸發主題切換

在 App.axaml.cs 中：

```
public partial class App : Application
{
    public override void OnFrameworkInitializationCompleted()
    {
        if (ApplicationLifetime is IClassicDesktopStyleApplicationLifetime desktop)
```

```

        {
            desktop.MainWindow = new MainWindow();
        }
        base.OnFrameworkInitializationCompleted();
    }

    // 切換到深色主題
    public void SwitchToDarkTheme()
    {
        RequestedThemeVariant = ThemeVariant.Dark;
    }

    // 切換到淺色主題
    public void SwitchToLightTheme()
    {
        RequestedThemeVariant = ThemeVariant.Light;
    }
}

```

此方案透過 Avalonia 原生 ThemeDictionaries + DynamicResource 實現與 AppThemeBinding 相同的自動主題切換效果，背景色會隨系統主題變更而動態更新。[71] [72]

✱✱

範例：App.xaml 同時支援淺色與深色背景的完整 XAML

Avalonia App.axaml 可使用 ResourceDictionary.ThemeDictionaries 同時定義淺色與深色背景資源，搭配 DynamicResource 實現自動切換。

App.axaml 完整範例

```

<Application xmlns="https://github.com/avaloniaui"
              xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
              xmlns:sukiUi="clr-namespace:SukiUI.Controls;assembly=SukiUI"
              x:Class="YourNamespace.App">

    <Application.Styles>
        <!-- SukiUI 主題 -->
        <sukiUi:SukiTheme ThemeColor="Blue" Locale="zh-TW" />
    </Application.Styles>

    <!-- 雙主題背景資源 -->
    <Application.Resources>
        <ResourceDictionary.ThemeDictionaries>
            <!-- 淺色主題背景：鮮綠色 -->
            <ResourceDictionary x:Key="Light">
                <SolidColorBrush x:Key="BackgroundBrush" Color="#00FF00" />
                <SolidColorBrush x:Key="PrimaryTextBrush" Color="Black" />
            </ResourceDictionary>

            <!-- 深色主題背景：暗綠色 -->

```



```

        <ResourceDictionary x:Key="Dark">
            <SolidColorBrush x:Key="BackgroundBrush" Color="#008000" />
            <SolidColorBrush x:Key="PrimaryTextBrush" Color="White" />
        </ResourceDictionary>
    </ResourceDictionary.ThemeDictionaries>
</Application.Resources>

</Application>

```

MainWindow.axaml 引用範例

```

<sukiUi:SukiWindow xmlns="https://github.com/avaloniaui"
    xmlns:sukiUi="clr-namespace:SukiUI.Controls;assembly=SukiUI"
    x:Class="YourNamespace.MainWindow"
    Background="{DynamicResource BackgroundBrush}">

    <Grid>
        <TextBlock Text="雙主題背景測試"
            Foreground="{DynamicResource PrimaryTextBrush}"
            HorizontalAlignment="Center"
            VerticalAlignment="Center"
            FontSize="24" />
    </Grid>

</sukiUi:SukiWindow>

```

此範例中，視窗背景會根據系統主題（Light/Dark）自動切換綠色系純色，文字顏色也同步調整確保可讀性，完美覆寫 SukiTheme 的漸層背景。 [82] [83]

✱

1. <https://github.com/AvaloniaUI/Avalonia/issues/13164>
2. <https://kikipoulet.github.io/SukiUI/zh/documentation/getting-started/launch.html>
3. <https://docs.avaloniaui.net/docs/basics/user-interface/styling/control-themes>
4. https://blog.csdn.net/gitblog_00540/article/details/151490281
5. <https://docs.avaloniaui.net/docs/guides/styles-and-resources/how-to-use-theme-variants>
6. <https://bugfine.com/avalonia/avalonia-使用-sukiui-的步驟/>
7. <https://stackoverflow.com/questions/78640457/how-to-customize-the-options-of-avaloniaui-app-on-the-finder-in-macos>
8. <https://github.com/kikipoulet/SukiUI>
9. <https://github.com/kikipoulet/SukiUI/issues/571>
10. https://blog.csdn.net/gitblog_07179/article/details/142225598
11. <https://github.com/kikipoulet/SukiUI/releases>
12. https://www.reddit.com/r/AvaloniaUI/comments/1ixtc2c/override_default_theme_colors/
13. https://www.reddit.com/r/PySimpleGUI/comments/cvj8te/can_i_change_a_window_colour_without_destroying/
14. <https://blog.csdn.net/fengershishe/article/details/145866442>

15. <https://stackoverflow.com/questions/65333019/easier-way-in-net-avalonia-to-change-the-background-color-of-the-windows-syste>
16. https://www.reddit.com/r/BubbleCard/comments/1cxxqlj/trying_to_change_popup_background_color_t_o/
17. <https://github.com/kikipoulet/SukiUI/issues/322>
18. <https://bugfine.com/avalonia/avalonia-使用-sukiui-的步骤/>
19. <https://docs.avaloniaui.net/docs/guides/styles-and-resources/how-to-use-theme-variants>
20. <https://docs.avaloniaui.net/docs/guides/graphics-and-animation/gradients>
21. https://www.reddit.com/r/AvaloniaUI/comments/1ixtc2c/override_default_theme_colors/
22. <https://github.com/kikipoulet/SukiUI>
23. https://blog.csdn.net/gitblog_07445/article/details/148888460
24. <https://github.com/AvaloniaUI/Avalonia/issues/3700>
25. <https://stackoverflow.com/questions/65333019/easier-way-in-net-avalonia-to-change-the-background-color-of-the-windows-syste>
26. https://blog.csdn.net/gitblog_07849/article/details/148888787
27. <https://github.com/AvaloniaUI/Avalonia/issues/12365>
28. <https://docs.avaloniaui.net/docs/guides/styles-and-resources/resources>
29. https://blog.csdn.net/weixin_42253874/article/details/142336760
30. <https://avaloniachina.github.io/avalonia-docs/docs/guides/styles-and-resources/resources/>
31. <https://github.com/AvaloniaUI/Avalonia/discussions/13517>
32. <https://github.com/AvaloniaUI/Avalonia/issues/11036>
33. <https://stackoverflow.com/questions/17502467/set-control-background-color-using-dynamic-resource-in-wpf>
34. <https://docs.avaloniaui.net/ru/docs/guides/styles-and-resources/resources>
35. <https://learn.microsoft.com/zh-tw/dotnet/maui/user-interface/system-theme-changes?view=net-maui-10.0>
36. <https://www.syncfusion.com/succinctly-free-ebooks/avaloniaui-succinctly/brushes-graphics-and-animations>
37. https://blog.csdn.net/gitblog_00951/article/details/153855760
38. <https://docs.avaloniaui.net/docs/guides/styles-and-resources/resources>
39. <https://www.cnblogs.com/hsiang/p/18385440>
40. <https://cloud.tencent.com/developer/information/XAML性能:设置画笔的Alpha通道与FrameworkElement的不透明度>
41. [https://cloud.tencent.com.cn/developer/information/XAML babysteps。使用子元素设置按钮背景色-article](https://cloud.tencent.com.cn/developer/information/XAML%20babysteps%20。使用子元素设置按钮背景色-article)
42. <https://www.cnblogs.com/jiang08/archive/2012/10/26/2740943.html>
43. <https://learn.microsoft.com/zh-tw/windows/uwp/composition/using-the-visual-layer-with-xaml>
44. <https://learn.microsoft.com/zh-tw/dotnet/maui/user-interface/styles/xaml?view=net-maui-8.0>
45. https://blog.csdn.net/chz_cslg/article/details/49076691
46. <https://mylabtw.blogspot.com/2016/08/xamarin-xaml-style.html>
47. <https://www.soinside.com/tag/xaml/34>

48. <https://mylabtw.blogspot.com/2016/06/>
49. <https://docs.avaloniaui.net/docs/guides/styles-and-resources/resources>
50. <https://dotblogs.azurewebsites.net/pou/2012/07/07/73267>
51. <https://yuchungchuang.wordpress.com/2019/04/14/wpf-教學-3 : xaml/>
52. <http://gelis-dotnet.blogspot.com/2009/06/xamlwpf.html>
53. https://blog.csdn.net/weixin_30533301/article/details/153091406
54. <https://learn.microsoft.com/zh-tw/windows/apps/develop/platform/xaml/brushes>
55. <https://docs.flutter.tw/get-started/flutter-for/xamarin-forms-devs/>
56. <https://learn.microsoft.com/zh-tw/windows/windows-app-sdk/api/winrt/microsoft.ui.xaml.controls.control.background?view=windows-app-sdk-1.6>
57. <https://ithelp.ithome.com.tw/m/articles/10293772>
58. <https://learn.microsoft.com/zh-tw/windows/windows-app-sdk/api/winrt/microsoft.ui.xaml.media.brush?view=windows-app-sdk-1.6>
59. https://blog.csdn.net/weixin_44538319/article/details/105286629
60. <https://docs.irihi.tech/semi/docs/advanced/resource-customization/>
61. <https://docs.avaloniaui.net/docs/guides/styles-and-resources/resources>
62. <https://ithelp.ithome.com.tw/articles/10293772>
63. <https://www.easemob.com/news/1821>
64. <https://learn.microsoft.com/zh-tw/dotnet/maui/user-interface/theming?view=net-maui-10.0>
65. <https://cloud.tencent.com/developer/ask/sof/108857925/answer/119623970>
66. https://blog.csdn.net/weixin_44668234/article/details/97622664
67. <https://cloud.tencent.cn/developer/information/如何在C%23代码中更改materialDesign:属性? -article>
68. <https://www.cnblogs.com/fangxinliu/p/16891565.html>
69. <https://givemepass.blogspot.com/2011/10/android.html>
70. https://blog.csdn.net/qq_25462179/article/details/146041428
71. <https://docs.avaloniaui.net/zh-Hans/docs/guides/styles-and-resources/how-to-use-theme-variants>
72. <https://docs.avaloniaui.net/docs/guides/styles-and-resources/resources>
73. <https://learn.microsoft.com/zh-tw/dotnet/maui/user-interface/system-theme-changes?view=net-maui-10.0>
74. http://www.xn.1xn1.com/userblog/onenews.aspx?blog_news_nodeuuid=16983211048970000003000
75. <https://learn.microsoft.com/zh-tw/dotnet/maui/user-interface/system-theme-changes?view=net-maui-9.0>
76. https://www.reddit.com/r/csharp/comments/gvh2lk/wpf_how_do_i_bind_a_background_color_in_my_ap_pxaml/
77. <https://learn.microsoft.com/zh-cn/dotnet/maui/user-interface/system-theme-changes?view=net-maui-10.0>
78. <https://github.com/dotnet-presentations/dotnet-maui-workshop/blob/main/Part 6 - AppThemes/README.zh-tw.md>
79. <https://blog.csdn.net/sD7O95O/article/details/135321938>
80. <https://learn.microsoft.com/zh-tw/dotnet/communitytoolkit/maui/essentials/apptHEME-resources>

81. <https://github.com/dotnet-presentations/dotnet-maui-workshop/blob/main/Part 6 - AppThemes/README.zh-cn.md>
82. <https://docs.avaloniaui.net/docs/guides/styles-and-resources/resources>
83. <https://docs.avaloniaui.net/zh-Hans/docs/guides/styles-and-resources/how-to-use-theme-variants>
84. <https://www.ruyut.com/2023/07/csharp-winui3-theme.html>
85. <https://blog.csdn.net/sD7O95O/article/details/146357269>
86. <https://developer.android.com/develop/ui/views/theming/darktheme?hl=zh-tw>
87. <https://learn.microsoft.com/zh-tw/dotnet/maui/user-interface/styles/xaml?view=net-maui-10.0>
88. <https://learn.microsoft.com/zh-tw/windows/apps/develop/platform/xaml/xaml-theme-resources>
89. <https://ithelp.ithome.com.tw/m/articles/10293772>
90. https://blog.csdn.net/qg_43307934/article/details/82908399
91. https://www.reddit.com/r/csharp/comments/dubxh1/wpfs_dark_themed_menus_and_menuitems/
92. <https://learn.microsoft.com/zh-tw/shows/visual-studio-toolbox/new-xaml-features-in-visual-studio>
93. <https://dotblogs.azurewebsites.net/pou/2012/07/07/73267>