

```
1 using Microsoft.Win32;
2 using System;
3 using System.Collections.Generic;
4 using System.IO;
5 using System.Windows;
6 using System.Windows.Documents;
7 using System.Windows.Input;
8 using System.Windows.Media;
9 using System.Xaml;
10
11 namespace _2023_WpfApp4
12 {
13     /// <summary>
14     /// MyDocumentViewer.xaml 的互動邏輯
15     /// </summary>
16     public partial class MyDocumentViewer : Window
17     {
18         Color fontColor = Colors.Black;
19         Color fontBackgroundColor = Colors.Transparent;
20         public MyDocumentViewer()
21         {
22             InitializeComponent();
23             fontColorPicker.SelectedColor = fontColor;
24             fontBackgroundColorPicker.SelectedColor = fontBackgroundColor;
25
26             foreach (FontFamily fontFamily in Fonts.SystemFontFamilies)
27             {
28                 fontFamilyComboBox.Items.Add(fontFamily);
29             }
30             fontFamilyComboBox.SelectedIndex = 8;
31
32             fontSizeComboBox.ItemsSource = new List<Double>() { 8, 9, 10, 12, 20, 24, 32, 40, 50, 60, 80, 90 };
33             fontSizeComboBox.SelectedIndex = 3;
34         }
35
36         private void NewCommand_Executed(object sender, ExecutedRoutedEventArgs e)
37         {
38             // 在這裡實現「新建」的操作，例如打開一個新文件、清空文檔等
39             MyDocumentViewer myDocumentViewer = new MyDocumentViewer();
40             myDocumentViewer.Show();
41         }
42
43         private void OpenCommand_Executed(object sender, ExecutedRoutedEventArgs e)
44         {
45             OpenFileDialog openFileDialog = new OpenFileDialog();
46             openFileDialog.Filter = "Rich Text Format檔案|.rtf|所有檔案|*.*";
```

```
47         if (fileDialog.ShowDialog() == true)
48         {
49             TextRange range = new TextRange
                                     (rtbEditor.Document.ContentStart,
                                     rtbEditor.Document.ContentEnd);
50
51             using (FileStream fileStream = new FileStream
                                     (fileDialog.FileName, FileMode.Open))
52             {
53                 range.Load(fileStream, DataFormats.Rtf);
54             }
55         }
56     }
57
58     private void SaveCommand_Executed(object sender, RoutedEventArgs
59     e)
60     {
61         SaveFileDialog saveFileDialog = new SaveFileDialog();
62         saveFileDialog.Filter = "Rich Text Format (*.rtf)|*.rtf|HTML
63         Files (*.html)|*.html";
64         if (saveFileDialog.ShowDialog() == true)
65         {
66             string selectedFileType = Path.GetExtension
                                     (saveFileDialog.FileName).ToLowerInvariant();
67             if (selectedFileType == ".rtf")
68             {
69                 SaveAsRtf(saveFileDialog.FileName);
70             }
71             else if (selectedFileType == ".html")
72             {
73                 SaveAsHtml(saveFileDialog.FileName);
74             }
75         }
76     }
77
78     private void SaveAsRtf(string fileName)
79     {
80         using (FileStream fs = new FileStream(fileName,
81         FileMode.Create))
82         {
83             TextRange range = new TextRange
                                     (rtbEditor.Document.ContentStart,
                                     rtbEditor.Document.ContentEnd);
84             range.Save(fs, DataFormats.Rtf);
85         }
86     }
87
88     private void SaveAsHtml(string fileName)
89     {
90     }
```

```

...owner\Desktop\writer\WpfApp1\MyDocumentViewer.xaml.cs 3
87      using (FileStream fs = new FileStream(fileName, ➤
           FileMode.Create))
88      {
89          TextRange range = new TextRange ➤
           (rtbEditor.Document.ContentStart, ➤
           rtbEditor.Document.ContentEnd);
90          range.Save(fs, DataFormats.Html);
91      }
92  }
93
94  private void rtbEditor_SelectionChanged(object sender, ➤
           RoutedEventArgs e)
95  {
96      //判斷選中的文字是否為粗體，並同步更新boldButton的狀態
97      object property = rtbEditor.Selection.GetPropertyValue ➤
           (TextElement.FontWeightProperty);
98      boldButton.IsChecked = (property is FontWeight && (FontWeight) ➤
           property == FontWeights.Bold);
99
100     //判斷選中的文字是否為斜體，並同步更新italicButton的狀態
101     property = rtbEditor.Selection.GetPropertyValue ➤
           (TextElement.FontStyleProperty);
102     italicButton.IsChecked = (property is FontStyle && (FontStyle) ➤
           property == FontStyles.Italic);
103
104     //判斷選中的文字是否有底線，並同步更新underlineButton的狀態
105     property = rtbEditor.Selection.GetPropertyValue ➤
           (Inline.TextDecorationsProperty);
106     underlineButton.IsChecked = (property != ➤
           DependencyProperty.UnsetValue && property == ➤
           TextDecorations.Underline);
107
108     //判斷選中的文字的字體，同步更新fontFamilyComboBox的狀態
109     property = rtbEditor.Selection.GetPropertyValue ➤
           (TextElement.FontFamilyProperty);
110     fontFamilyComboBox.SelectedItem = property;
111
112     //判斷所選中的文字的字體大小，同步更新fontSizeComboBox的狀態
113     property = rtbEditor.Selection.GetPropertyValue ➤
           (TextElement.FontSizeProperty);
114     fontSizeComboBox.SelectedItem = property;
115
116     //判斷所選中的文字的字體色彩，同步更新fontColorPicker的狀態
117     SolidColorBrush? foregroundProperty = ➤
           rtbEditor.Selection.GetPropertyValue ➤
           (TextElement.ForegroundProperty) as SolidColorBrush;
118     if (foregroundProperty != null)
119     {
120         fontColorPicker.SelectedColor = foregroundProperty.Color;

```

```
121     }
122     //判斷所選中的文字的字體色彩，同步更新fontBackgroundColorPicker的狀態
123     SolidColorBrush? backgroundProperty =
124         rtbEditor.Selection.GetPropertyValue
125         (TextElement.BackgroundProperty) as SolidColorBrush;
126
127     if (backgroundProperty != null)
128     {
129         fontBackgroundColorPicker.SelectedColor =
130             backgroundProperty.Color;
131     }
132 }
133
134 private void fontColorPicker_SelectedColorChanged(object sender,
135     RoutedPropertyChangedEventArgs<Color?> e)
136 {
137     fontColor = (Color)e.NewValue;
138     SolidColorBrush fontBrush = new SolidColorBrush(fontColor);
139     rtbEditor.Selection.ApplyPropertyValue
140         (TextElement.ForegroundProperty, fontBrush);
141 }
142
143 private void fontBackgroundColorPicker_SelectedColorChanged(object
144     sender, RoutedPropertyChangedEventArgs<Color?> e)
145 {
146     fontBackgroundColor = (Color)e.NewValue;
147     SolidColorBrush backgroundBrush = new SolidColorBrush
148         (fontBackgroundColor);
149     rtbEditor.Selection.ApplyPropertyValue
150         (TextElement.BackgroundProperty, backgroundBrush);
151 }
152
153 private void fontFamilyComboBox_SelectionChanged(object sender,
154     System.Windows.Controls.SelectionChangedEventArgs e)
155 {
156     if (fontFamilyComboBox.SelectedItem != null)
157     {
158         rtbEditor.Selection.ApplyPropertyValue
159             (TextElement.FontFamilyProperty,
160             fontFamilyComboBox.SelectedItem);
161     }
162 }
163
164 private void fontSizeComboBox_SelectionChanged(object sender,
165     System.Windows.Controls.SelectionChangedEventArgs e)
166 {
167     if (fontSizeComboBox.SelectedItem != null)
168     {
169     }
```

```
...owner\Desktop\writer\WpfApp1\MyDocumentViewer.xaml.cs 5
158         rtbEditor.Selection.ApplyPropertyValue          ↗
            (TextElement.FontSizeProperty,               ↗
            fontSizeComboBox.SelectedItem);
159     }
160 }
161
162 private void clearButton_Click(object sender, RoutedEventArgs e)
163 {
164     rtbEditor.Document.Blocks.Clear();
165 }
166 }
167 }
168
```