

```

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
10 namespace wordCopy
11 {
12     /// <summary>
13     /// MyDocumentViewer.xaml 的互動邏輯
14     /// </summary>
15     public partial class MyDocumentViewer : Window
16     {
17         Color fontColor = Colors.Red;
18
19         public MyDocumentViewer()
20         {
21             InitializeComponent();
22             fontColorPicker.SelectedColor = fontColor;
23             // 修改這一行
24             foreach (FontFamily fontFamily in Fonts.SystemFontFamilies)
25             {
26                 fontFamilyComboBox.Items.Add(fontFamily.Source);
27             }
28             fontFamilyComboBox.SelectedIndex = 2;
29
30             fontSizeComboBox.ItemsSource = new List<Double>() { 8, 9,
31                 10, 12, 20, 24, 32, 40, 50, 60, 80, 90 };
32             fontSizeComboBox.SelectedIndex = 3;
33         }
34
35         private void NewCommand_Executed(object sender,
36             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,
44             ExecutedRoutedEventArgs e)
45         {
46             OpenFileDialog fileDialog = new OpenFileDialog();
47             fileDialog.Filter = "Rich Text Format檔案|*.rtf|Hyper Text
48                 Markup Language|.html|所有檔案|*.*";
49             if (fileDialog.ShowDialog() == true)
50             {
51                 TextRange range = new TextRange
52                     (rtbEditor.Document.ContentStart,
53                     rtbEditor.Document.ContentEnd);
54             }
55         }
56     }
57 }

```

```
48         if(range.Text == "" )
49         {
50             MessageBox.Show("檔案無內容");
51         }
52         using (FileStream fileStream = new FileStream
53             (fileDialog.FileName, FileMode.Open))
54         {
55             range.Load(fileStream, DataFormats.Rtf);
56         }
57     }
58
59     private void SaveCommand_Executed(object sender,
60         ExecutedRoutedEventArgs e)
61     {
62         SaveFileDialog fileDialog = new SaveFileDialog();
63         fileDialog.Filter = "Rich Text Format檔案|*.rtf|Hyper Text
64             Markup Language|.html|所有檔案|*.*";
65         if(fileDialog.ShowDialog() == true)
66         {
67             TextRange range = new TextRange
68                 (rtbEditor.Document.ContentStart,
69                 rtbEditor.Document.ContentEnd);
70             using (FileStream fileStream = new FileStream
71                 (fileDialog.FileName, FileMode.Create))
72             {
73                 range.Save(fileStream, DataFormats.Rtf);
74             }
75         }
76     }
77
78     private void rtbEditor_SelectionChanged(object sender,
79         RoutedEventArgs e)
80     {
81         //判斷選中的文字是否為粗體，並同步更新boldButton的狀態
82         object property = rtbEditor.Selection.GetProperty(
83             TextElement.FontWeightProperty);
84         boldButton.IsChecked = (property is FontWeight &&
85             (FontWeight)property == FontWeights.Bold);
86
87         //判斷選中的文字是否為斜體，並同步更新italicButton的狀態
88         property = rtbEditor.Selection.GetProperty(
89             TextElement.FontStyleProperty);
90         italicButton.IsChecked = (property is FontStyle &&
91             (FontStyle)property == FontStyles.Italic);
92
93         //判斷選中的文字是否有底線，並同步更新underlineButton的狀態
94         property = rtbEditor.Selection.GetProperty(
95             Inline.TextDecorationsProperty);
96         underlineButton.IsChecked = (property !=
97             DependencyProperty.UnsetValue && property ==
98             TextDecorations.Underline);
99     }
```

```
87 // 判斷選中的文字的字體，同步更新font
88 property = rtbEditor.Selection.GetPropertyValue
    (TextElement.FontFamilyProperty);
89 fontFamilyComboBox.SelectedItem = property;
90 // 判斷選中的文字的字體大小
91 property = rtbEditor.Selection.GetPropertyValue
    (TextElement.ForegroundProperty) as SolidColorBrush;
92 fontSizeComboBox.SelectedItem = property;
93
94
95
96 SolidColorBrush? foregroundProperty =
    rtbEditor.Selection.GetPropertyValue
    (TextElement.ForegroundProperty) as SolidColorBrush;
97
98 if(foregroundProperty != null)
99 {
100     fontColorPicker.SelectedColor =
        foregroundProperty.Color;
101 }
102
103 }
104
105 private void fontColorPicker_SelectedColorChanged(object
    sender, RoutedPropertyChangedEventArgs<Color?> e)
106 {
107     fontColor = (Color)e.NewValue;
108     SolidColorBrush fontBrush = new SolidColorBrush(fontColor);
109     rtbEditor.Selection.ApplyPropertyValue
        (TextElement.ForegroundProperty, fontBrush);
110 }
111 private void fontFamilyComboBox_SelectionChanged(object sender,
    System.Windows.Controls.SelectionChangedEventArgs e)
112 {
113     if(fontFamilyComboBox.SelectedItem != null)
114     {
115         rtbEditor.Selection.ApplyPropertyValue
            (TextElement.FontFamilyProperty,
            fontFamilyComboBox.SelectedItem);
116     }
117 }
118
119 private void fontSizeComboBox_SelectionChanged(object sender,
    System.Windows.Controls.SelectionChangedEventArgs e)
120 {
121     if (fontSizeComboBox.SelectedItem != null)
122     {
123         rtbEditor.Selection.ApplyPropertyValue
            (TextElement.FontSizeProperty,
            fontSizeComboBox.SelectedItem);
124     }
125 }
126
```

```
127     private void clearButton_Click(object sender, RoutedEventArgs ↗
128         e)
129     {
130         rtbEditor.Document.Blocks.Clear();
131     }
132     private void backgroundPicker_SelectedColorChanged(object ↗
133         sender, RoutedPropertyChangedEventArgs<Color?> e)
134     {
135         if (e.NewValue.HasValue)
136         {
137             Color backgroundColor = e.NewValue.Value;
138             SolidColorBrush backgroundBrush = new SolidColorBrush ↗
139                 (backgroundColor);
140             rtbEditor.Background = backgroundBrush;
141         }
142     }
143 }
```