

```
1 using Microsoft.Win32;
2 using System;
3 using System.Collections.Generic;
4 using System.IO;
5 using System.Linq;
6 using System.Text;
7 using System.Text.RegularExpressions;
8 using System.Threading.Tasks;
9 using System.Windows;
10 using System.Windows.Controls;
11 using System.Windows.Data;
12 using System.Windows.Documents;
13 using System.Windows.Input;
14 using System.Windows.Media;
15 using System.Windows.Media.Imaging;
16 using System.Windows.Shapes;
17
18 namespace TheNoteBook
19 {
20     /// <summary>
21     /// DocumentViewerWindow.xaml 的互動邏輯
22     /// </summary>
23     public partial class DocumentViewerWindow : Window
24     {
25         string FileName = "新文件"; // 儲存當前檔案名稱
26         public DocumentViewerWindow()
27         {
28             InitializeComponent();
29             cmbFontFamily.ItemsSource =
30                 Fonts.SystemFontFamilies.OrderBy(f => f.Source); // 設定文
31                 字風格的Items
32             cmbFontSize.ItemsSource = new List<double>() { 8, 9, 10,
33                 11, 12, 14, 16, 18, 20, 22, 24, 26, 28, 36, 48, 72 }; //
34                 設定文字大小的Items
35
36             // 取得當前游標所在的行與列
37             TextPointer p1 =
38                 rtbEditor.Selection.Start.GetLineStartPosition(0);
39             TextPointer p2 = rtbEditor.Selection.Start;
40             int c = p1.GetOffsetToPosition(p2);
41             int someBigNumber = int.MaxValue;
42             int lineMoved, currentLineNumber;
43             rtbEditor.Selection.Start.GetLineStartPosition(-
```

```
        someBigNumber, out lineMoved);
39         currentLineNumber = -lineMoved;
40
41         lb1.Content = $"第{currentLineNumber + 1}列 · 第{c}行|
        {FileName}"; //更改狀態列
42     }
43     private void Open_Executed(object sender,
        ExecutedRoutedEventArgs e) // 開啟舊的rtf檔
44     {
45         OpenFileDialog dlg = new OpenFileDialog();
46         dlg.Filter = "Rich Text Format (*.rtf)|*.rtf|All files|
        *.*"; // 設定開檔的類型
47         if (dlg.ShowDialog() == true) // 開啟檔案
48         {
49             FileStream fs = new FileStream(dlg.FileName,
                FileMode.Open); // 無法開剛剛做完的檔案 · 可能需要close
50             TextRange range = new TextRange
                (rtbEditor.Document.ContentStart,
                rtbEditor.Document.ContentEnd);
51             range.Load(fs, DataFormats.Rtf);
52         }
53
54         FileName = System.IO.Path.GetFileName(dlg.FileName); // 取
            得當前檔案名稱
55
56         //取得當前游標的下標
57         int rtbEditorLine = Regex.Split
            (rtbEditor.Selection.Text.ToString(), "\n").Length;
58         TextPointer p1 =
            rtbEditor.Selection.Start.GetLineStartPosition(0);
59         TextPointer p2 = rtbEditor.Selection.Start;
60         int c = p1.GetOffsetToPosition(p2);
61         int someBigNumber = int.MaxValue;
62         int lineMoved, currentLineNumber;
63         rtbEditor.Selection.Start.GetLineStartPosition(-
            someBigNumber, out lineMoved);
64         currentLineNumber = -lineMoved;
65
66         lb1.Content = $"第{currentLineNumber + 1}列 · 第{c}行|
            {FileName}"; // 更新狀態列
67     }
68
69     private void NewFile_Executed(object sender,
```

```
    ExecutedRoutedEventArgs e) // 開一個新的檔案
70     {
71         DocumentViewerWindow viewerWindow = new DocumentViewerWindow(); // 設定一個新的視窗
72         viewerWindow.Show(); // 顯示新的視窗
73     }
74     private void Close_Executed(object sender, ExecutedRoutedEventArgs e)
75     {
76         this.Close(); // 關閉當前的視窗
77     }
78
79     private void Save_Executed(object sender, ExecutedRoutedEventArgs e) // 儲存檔案
80     {
81         SaveFileDialog dlg = new SaveFileDialog();
82         dlg.Filter = "Rich Text Format (*.rtf)|*.rtf|All files|*.**"; // 設定存檔類型
83         if (dlg.ShowDialog() == true)
84         {
85             FileStream fs = new FileStream(dlg.FileName, FileMode.CreateNew); // 無法儲存相同檔名
86             TextRange range = new TextRange(rtbEditor.Document.ContentStart, rtbEditor.Document.ContentEnd);
87             range.Save(fs, DataFormats.Rtf);
88         }
89     }
90
91     private void cmbFontFamily_SelectionChanged(object sender, SelectionChangedEventArgs e) // 改變字體
92     {
93         if (cmbFontFamily.SelectedValue != null)
94         {
95             rtbEditor.Selection.ApplyPropertyValue(Inline.FontFamilyProperty, cmbFontFamily.SelectedItem);
96         }
97     }
98
99     private void cmbFontSize_SelectionChanged(object sender, SelectionChangedEventArgs e) // 改變文字大小
100    {
```

```
101         if (cmbFontSize.SelectedValue  $\neq$  null)
102         {
103             rtbEditor.Selection.ApplyPropertyValue
104                 (Inline.FontSizeProperty, Convert.ToDouble
105                 (cmbFontSize.SelectedValue.ToString()));
106         }
107     }
108
109     private void rtbEditor_SelectionChanged(object sender,
110         RoutedEventArgs e) // 游標位置改變時的事件
111     {
112         // 取得列與行的下標
113         int rtbEditorLine = Regex.Split
114             (rtbEditor.Selection.Text.ToString(), "\n").Length;
115         TextPointer p1 =
116             rtbEditor.Selection.Start.GetLineStartPosition(0);
117         TextPointer p2 = rtbEditor.Selection.Start;
118         int c = p1.GetOffsetToPosition(p2);
119         int someBigNumber = int.MaxValue;
120         int lineMoved, currentLineNumber;
121         rtbEditor.Selection.Start.GetLineStartPosition(-
122             someBigNumber, out lineMoved);
123         currentLineNumber = -lineMoved;
124
125         lb1.Content = $"第{currentLineNumber + 1}列 · 第{c}行|
126             {FileName}"; // 更改狀態列
127
128         var temp = rtbEditor.Selection.GetPropertyValue
129             (Inline.FontWeightProperty);
130         btnBold.IsChecked = (temp  $\neq$  DependencyProperty.UnsetValue) &&
131             (temp.Equals(FontWeights.Bold));
132
133         temp = rtbEditor.Selection.GetPropertyValue
134             (Inline.FontStyleProperty);
135         btnItalic.IsChecked = (temp  $\neq$ 
136             DependencyProperty.UnsetValue) && (temp.Equals
137             (FontStyles.Italic));
138
139         temp = rtbEditor.Selection.GetPropertyValue
140             (Inline.TextDecorationsProperty);
141         btnUnderLine.IsChecked = (temp  $\neq$ 
```

```
DependencyProperty.UnsetValue) && (temp.Equals  
(TextDecorations.Underline));  
131  
132     temp = rtbEditor.Selection.GetPropertyValue  
        (Inline.FontFamilyProperty);  
133     cmbFontFamily.SelectedItem = temp;  
134  
135     temp = rtbEditor.Selection.GetPropertyValue  
        (Inline.FontSizeProperty);  
136     cmbFontSize.SelectedItem = temp;  
137  
138 }  
139  
140  
141  
142 private void ColorBtn_Click(object sender, RoutedEventArgs  ↗  
    e) // 更改文字顏色  
143 {  
144     Color brc = GetDialogColor(); // 設定顏色  
145     BrushConverter brushes = new BrushConverter(); // 建立一個  ↗  
        rtbEditor設定顏色需要的類別  
146     rtbEditor.Selection.ApplyPropertyValue  
        (TextElement.ForegroundProperty,  ↗  
        brushes.ConvertFromString(Convert.ToString(brc))); //更改  ↗  
        選中rtbEditor文字顏色  
147     SolidColorBrush Btbrush = new SolidColorBrush(brc);  
148     ColorBtn.Background = Btbrush; // 改變按鈕的背景顏色  
149 }  
150 private Color GetDialogColor() // 叫出調色盤回傳選中顏色  
151 {  
152     System.Windows.Forms.ColorDialog dlg = new  ↗  
        System.Windows.Forms.ColorDialog();  
153     dlg.ShowDialog();  
154     System.Drawing.Color returnColor = dlg.Color;  
155     return Color.FromArgb(returnColor.A, returnColor.R,  ↗  
        returnColor.G, returnColor.B);  
156  
157 }  
158  
159 private void BtnClear_Click(object sender, RoutedEventArgs  ↗  
    e) // 關閉文字文件  
160 {  
161     rtbEditor.Document.Blocks.Clear();
```

```
162     }
163
164     private void BackColorBtn_Click(object sender, RoutedEventArgs e) // 改變背景顏色
165     {
166         Color brc = GetDialogColor(); // 設定顏色
167         SolidColorBrush brush = new SolidColorBrush(brc); // 設定成背景設定顏色的類別
168         BackColorBtn.Background = brush; // 設定背景顏色
169         rtbEditor.Background = brush; // 改變按鈕顏色
170     }
171
172
173     //TextBoxBase.SelectionChanged="cmbFontSize_SelectionChanged"
174 }
175 }
176
```