## Текстовые элементы управления

### TextBlock

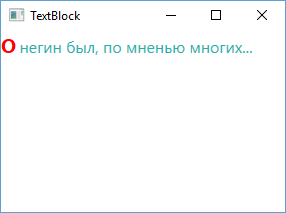
Элемент предназначен для вывода текстовой информации, для создания простых надписей:

|  |  |
| --- | --- |
| 1  2  3  4 | <StackPanel>      <TextBlock>Текст1</TextBlock>      <TextBlock Text="Текст2" />  </StackPanel> |

Ключевым свойством здесь является свойство Text, которое задает текстовое содержимое. Причем в случае <TextBlock>Текст1</TextBlock> данное свойство задается неявно.

С помощью таких свойств, как FontFamily, TextDecorations и др., мы можем настроить отображение текста. Однако мы можем задать и более сложное форматирование, например:

|  |  |
| --- | --- |
| 1  2  3  4 | <TextBlock TextWrapping="Wrap">      <Run FontSize="20" Foreground="Red" FontWeight="Bold">О</Run>      <Run FontSize="16" Foreground="LightSeaGreen">негин был, по мненью многих...</Run>  </TextBlock> |



Элементы Run представляют куски обычного текста, для которых можно задать отдельное форматирование.

Для изменения параметров отображаемого текста данный элемент имеет такие свойства, как **LineHeight**, **TextWrapping** и **TextAlignment**.

Свойство **LineHeight** позволяет указывать высоту строк.

Свойство **TextWrapping** позволяет переносить текст при установке этого свойства TextWrapping="Wrap". По умолчанию это свойство имеет значение NoWrap, поэтому текст не переносится.

Свойство **TextAlignment** выравнивает текст по центру (значение Center), правому (Right) или левому краю (Left): <TextBlock TextAlignment="Right">

Для декорации текста используется свойство **TextDecorations**, например, если TextDecorations="Underline", то текст будет подчеркнут.

Если нам вдруг потребуется перенести текст на другую строку, то тогда мы можем использовать элемент LineBreak:

|  |  |
| --- | --- |
| 1  2  3  4  5 | <TextBlock>      Однажды в студеную зимнюю пору      <LineBreak />      Я из лесу вышел  </TextBlock> |

### TextBox

Если TextBlock просто выводит статический текст, то этот элемент представляет поле для ввода текстовой информации.

Он также, как и TextBlock, имеет свойства **TextWrapping**, **TextAlignment** и **TextDecorations**.

С помощью свойства **MaxLength** можно задать предельное количество вводимых символов.

|  |  |
| --- | --- |
| 1 | <TextBox MaxLength="250" TextChanged="TextBox\_TextChanged">Начальный текст</TextBox> |

В коде C# мы можем обработать событие изменения текста:

|  |  |
| --- | --- |
| 1  2  3  4  5 | private void TextBox\_TextChanged(object sender, TextChangedEventArgs e)  {      TextBox textBox = (TextBox)sender;      MessageBox.Show(textBox.Text);  } |

По умолчанию, если вводимый текст превышает установленные границы поля, то текстовое поле растет, чтобы вместить весь текст. Но визуально это не очень хорошо выглядит. Поэтому, как и в случае с TextBlock, мы можем перенести непомещающийся текст на новую строку, установив свойство TextWrapping="Wrap".

Чобы переводить по нажатию на клавишу Enter курсор на следующую строку, нам надо установить свойство AcceptsReturn="True".

Также мы можем добавить полю возможность создавать табуляцию с помощью клавиши Tab, установив свойство AcceptsTab="True"

Для отображения полос прокрутки TextBox поддерживает свойства **VerticalScrollBarVisibility** и **НоrizontalScrollBarVisibility**:

|  |  |
| --- | --- |
| 1  2 | <TextBox AcceptsReturn="True" Height="100" VerticalScrollBarVisibility="Auto"           HorizontalScrollBarVisibility="Auto">Начальный текст</TextBox> |

Возможно, при создании приложения нам потребуется сделать текстовое поле недоступным для ввода (на время в зависимости от условий или вообще), тогда для этого нам надо установить свойство IsReadOnly="True".

Для выделения текста есть свойства **SelectionStart**, **SelectionLength** и **SelectionText**. Например, выделим программно текст по нажатию кнопки:

|  |  |
| --- | --- |
| 1  2  3  4 | <StackPanel>      <TextBox x:Name="textBox1" Height="100" SelectionBrush="Blue" />      <Button Content="Выделить текст" Height="30" Width="100" Click="Button\_Click" Margin="10" />  </StackPanel> |

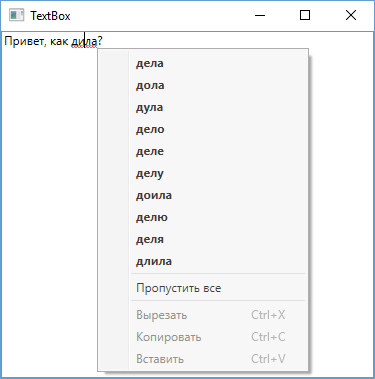
Обработчик нажатия кнопки:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | private void Button\_Click(object sender, RoutedEventArgs e)  {      textBox1.SelectionStart = 5;      textBox1.SelectionLength = 10;      textBox1.Focus();      // данное выражение эквивалентно      //textBox1.Select(5, 10);  } |

#### Проверка орфографии

TextBox обладает встроенной поддержкой орфографии. Чтобы ее задействовать, надо установить свойство SpellCheck.IsEnabled="True". Кроме того, по умолчанию проверка орфографии распространяется только на английский язык, поэтому, если приложение заточено под другой язык, нам надо его явным образом указать через свойство **Language**:

|  |  |
| --- | --- |
| 1  2  3 | <DockPanel>      <TextBox SpellCheck.IsEnabled="True" Language="ru-ru">Привет, как дила?</TextBox>  </DockPanel> |



### Метка (Label)

Главной особенностью меток является поддержка мнемонических команд-клавиш быстрого доступа, которые передают фокус связанному элементу. Например,

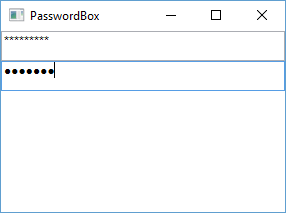
|  |  |
| --- | --- |
| 1  2 | <Label Target="{Binding ElementName=TextBox1}">\_привет</Label>  <TextBox Name="TextBox1" Margin="0 30 0 0" Height="30" Width="100"></TextBox> |

Теперь, нажав на клавишу "п", мы переведем фокус на связанное текстовое поле. При вызове приложения подчеркивание не отображается, чтобы отображать подчеркивание, надо нажать на клавишу Alt. Тогда чтобы перевести фокус на связанное текстовое поле необходимо будет нажать сочетание Alt + "п". Если не предполагается использование клавиш быстрого доступа, то для вывода обычной текста вместо меток лучше использовать элемент TextBlock.

### PasswordBox

Элемент предназначен для ввода парольной информации. По сути это тоже текстовое поле, только для ввода символов используется маска. Свойство **PasswordChar** устанавливает символ маски, отображаемый при вводе пароля. Если это свойство не задано, то по умолчанию для маски символа используется черная точка. Свойство **Password** устанавливает парольную строку, отображаемую по умолчанию при загрузке окна приложения.

|  |  |
| --- | --- |
| 1  2  3  4 | <StackPanel>      <PasswordBox PasswordChar="\*" MinHeight="30" />      <PasswordBox MinHeight="30" />  </StackPanel> |



### RichTextBox

Для вывода текстового содержимого, насыщенного форматированием, графикой, предназначен RichTextBox. Можно даже сказать, что он выводит не просто текст, а документы с более сложным форматированием, чем обычный TextBox. Более подробно о нем, а также как на его основе создать простой текстовый редактор, мы поговорим в главе, посвященной документам.