## Всплывающие подсказки ToolTip и Popup

### ToolTip

Элемент ToolTip представляет всплывающую подсказку при наведении на какой-нибудь элемент. Для определения всплывающей подсказки у элементов уже есть свойство ToolTip, которому можно задать текст, отображаемый при наведении:

|  |  |
| --- | --- |
| 1 | <Button Content="Tooltip" ToolTip="Всплывающая подсказка для кнопки" Height="30" Width="80" /> |

Также мы можем более точно настроить всплывающую подсказку с помощью свойства Button.ToolTip:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | <Button Content="Tooltip" Height="30" Width="80">      <Button.ToolTip>          <ToolTip>              Всплывающая подсказка для кнопки          </ToolTip>      </Button.ToolTip>  </Button> |

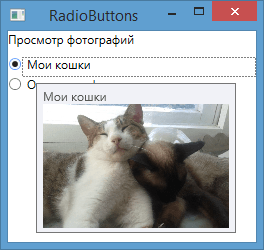
Всплывающие подсказки можно применять не только кнопкам, но и ко всем другим элементам управления, например, к текстовому блоку:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10 | <StackPanel>      <TextBlock Text="TextTooltip 1" ToolTip="Hello Tooltip" />      <TextBlock Text="TextTooltip 2">          <TextBlock.ToolTip>              <ToolTip>                  Hello WPF              </ToolTip>          </TextBlock.ToolTip>      </TextBlock>  </StackPanel> |

Оба определения всплывающей подсказки будут аналогичны.

Поскольку ToolTip является элементом управления содержимого, то в него можно встроить другие элементы для создания более богатой функциональности. Например:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14 | <StackPanel>      <TextBlock Text="Просмотр фотографий" Margin="0 0 0 10" />      <RadioButton GroupName="Photos" Content="Мои кошки" Height="20">          <RadioButton.ToolTip>              <ToolTip Width="200" Height="150">                  <StackPanel>                      <TextBlock Text="Мои кошки" />                      <Image Source="cats.jpg" />                  </StackPanel>              </ToolTip>          </RadioButton.ToolTip>      </RadioButton>      <RadioButton GroupName="Photos" Content="Остальные фото" Height="20" ToolTip="Остальное" />  </StackPanel> |



Здесь у нас два переключателя, и на одном из них определен расширенный элемент ToolTip: а именно в него вложен элемент Image, выводящий изображение, и элемент TextBlock. Таким образом, можно создавать всплывающие подсказки с различным наполнением.

Изображение для элемента Image в данном случае было добавлено в проект.

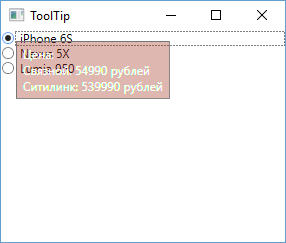
### Свойства ToolTip

Некоторые полезные свойства элемента Tooltip:

* **HasDropShadow**: определяет, будет ли всплывающая подсказка отбрасывать тень.
* **Placement**: определяет, как будет позиционироваться всплывающая подсказка на окне приложения. По умолчанию ее верхний левый угол позиционируется на указатель мыши.
* **HorizontalOffset/VerticalOffset**: определяет смещение относительно начального местоположения.
* **PlacementTarget**: определяет позицию всплывающей подсказки относительно другого элемента управления.

Применим свойства:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17 | <StackPanel>      <RadioButton GroupName="Phones" Content="iPhone 6S">          <RadioButton.ToolTip>              <ToolTip Background="#60AA4030" Foreground="White" HasDropShadow="False"                  Placement="Relative" HorizontalOffset="15" VerticalOffset="10">                  <StackPanel>                      <TextBlock>Цена:</TextBlock>                      <TextBlock>Связной: 54990 рублей</TextBlock>                      <TextBlock>Ситилинк: 539990 рублей</TextBlock>                  </StackPanel>              </ToolTip>          </RadioButton.ToolTip>      </RadioButton>      <RadioButton GroupName="Phones" ToolTipService.Placement="Mouse"          ToolTip="Цена: 29990 рублей" Content="Nexus 5X" />      <RadioButton GroupName="Phones" ToolTip="Цена: 39990 рублей" Content="Lumia 950" />  </StackPanel> |



Здесь у нас три переключателя. У первого мы задаем свойства через элемент ToolTip. Для второго переключателя мы также можем задать свойства, несмотря на то, что здесь мы всплывающую подсказку задаем просто ToolTip="Цена: 29990 рублей" Content="Nexus 5X". В этом случае мы можем использовать прикрепленные свойства класса **ToolTipService**:

* InitialShowDelay: задает задержку перед отображением всплывающей подсказки
* ShowDuration: устанавливает время отображения всплывающей подсказки
* BetweenShowDelay: устанавливает время, в течение которого пользователь сможет перейти к другому элементу с подсказкой, и для этого элемента не будет работать свойство InitialShowDelay (если оно указано)
* ToolTip: устанавливает содержимое всплывающей подсказки
* HasDropShadow: определяет, будет ли подсказка отбрасывать тень
* ShowOnDisabled: устанавливает поведение всплывающей подсказки для недоступного элемента (со значением IsEnabled="True"). Если это свойство равно true, то подсказка отображается для недоступных элементов. По умолчанию равно false.
* Placement / HorizontalOffset / VerticalOffset / PlacementTarget: те же свойства, что и у элемента ToolTip, которые устанавливают положение всплывающей подсказки

#### Программное создание всплывающей подсказки

Допустим, в коде XAML у нас определена следующая кнопка:

|  |  |
| --- | --- |
| 1 | <Button x:Name="button1" Content="Hello" /> |

Тогда в файле кода C# мы могли бы определить всплывающую подсказку для кнопки так:

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | ToolTip toolTip = new ToolTip();  StackPanel toolTipPanel = new StackPanel();  toolTipPanel.Children.Add(new TextBlock { Text = "Заголовок", FontSize=16 });  toolTipPanel.Children.Add(new TextBlock { Text = "Текст" });  toolTip.Content = toolTipPanel;  button1.ToolTip = toolTip; |

### Popup

Элемент Popup также представляет всплывающее окно, только в данном случае оно имеет другую функциональность. Если Tooltip отображается автоматически при наведении и также автоматически скрывается через некоторое время, то в случае с Popup все эти действия нам надо задавать вручную.

Так, чтобы отразить при наведении мыши на элемент всплывающее окно, нам надо соответственным образом обработать событие MouseEnter.

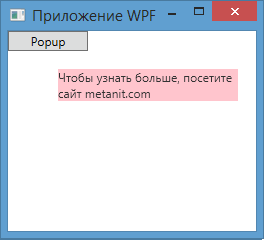
Второй момент, который надо учесть, это установка свойства StaysOpen="False". По умолчанию оно равно True, а это значит, что при отображении окна, оно больше не исчезнет, пока мы не установим явно значение этого свойства в False.

Итак, создадим всплывающее окно:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9 | <StackPanel>      <Button Content="Popup" Width="80" MouseEnter="Button\_MouseEnter\_1" HorizontalAlignment="Left" />      <Popup x:Name="popup1" StaysOpen="False" Placement="Mouse" MaxWidth="180"           AllowsTransparency="True"  >          <TextBlock TextWrapping="Wrap" Width="180" Background="LightPink" Opacity="0.8" >              Чтобы узнать больше, посетите сайт metanit.com          </TextBlock>      </Popup>  </StackPanel> |

И обработчик наведения курсора мыши на кнопку в коде c#:

|  |  |
| --- | --- |
| 1  2  3  4 | private void Button\_MouseEnter\_1(object sender, MouseEventArgs e)  {      popup1.IsOpen = true;  } |



И при наведении указателя мыши на элемент появится всплывающее окно с сообщением.