

در این مقاله نحوه ی استفاده از کنترل TreeView در تکنولوژی WPF را می آموزیم. همچنین با به کارگیری انواع کنترل ها در درون TreeView آشنا می شویم .

نحوه ی کار با TreeView در WPF



برنامه نویسی را از برنامه نویسان حرفه ای بیاموزید



مقدمه

یک [TreeView](#) داده ها را به صورت سلسله مراتبی به نمایش در می آورد. در [TreeView](#) یک رابطه ی والد-فرزندی وجود دارد که گره والد می تواند گسترش داده شود یا جمع شود. سمت چپ بخش [Windows Explorer](#) نمونه ای از [TreeView](#) است.

تگ [TreeView](#) یک کنترل [TreeView](#) را در قالب [XAML](#) به نمایش می گذارد.

```
<TreeView></TreeView>
```

مشخصات مربوط به عرض و ارتفاع ، عرض و ارتفاع [TreeView](#) را تعیین می کنند. مشخصه [Name](#)، نام کنترل را تعیین می کند که یک مشخصه ی واحد برای کنترل محسوب می شود. مشخصه ی [Margin](#) مکان [TreeView](#) را در صفحه تعیین می کند. مشخصه های [HorizontalAlignment](#) و [VerticalAlignment](#) برای خط کشی های عمودی و افقی را تعیین می کنند. تکه کد زیر مشخصات مربوط به [name](#)، [width](#) و [height](#) کنترل [TreeView](#) را تنظیم می کند. این کد همچنین تنظیمات صفحه بندی عمودی و افقی را نیز تعیین می کند.

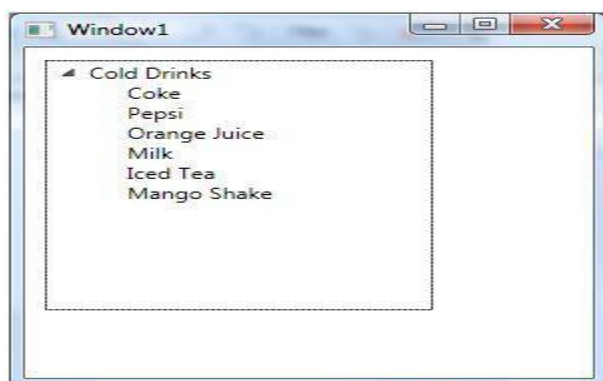
```
<TreeView Margin="10,10,0,13" Name="TreeView1" HorizontalAlignment="Left"
  VerticalAlignment="Top" Width="194" Height="200" />
```

افزودن آیتم های TreeView

کنترل TreeView حاوی یک مجموعه از TreeViewItem ها است . مشخصه ی Header متنی است که بر روی view نمایش داده می شود. تکه کد زیر یک گره والد و شش گره فرزند به TreeView اضافه می کند.

```
<TreeView Margin="10,10,0,13" Name="TreeView1" HorizontalAlignment="Left"
  VerticalAlignment="Top" Width="194" Height="200">
  <TreeViewItem Header="Cold Drinks">
    <TreeViewItem Header="Coke"></TreeViewItem>
    <TreeViewItem Header="Pepsi"></TreeViewItem>
    <TreeViewItem Header="Orange Juice"></TreeViewItem>
    <TreeViewItem Header="Milk"></TreeViewItem>
    <TreeViewItem Header="Iced Tea"></TreeViewItem>
    <TreeViewItem Header="Mango Shake"></TreeViewItem>
  </TreeViewItem>
</TreeView>
```

به صورت پیش فرض ، گره والد به صورت فشرده و جمع شده است ولی زمانی که بر روی آن کلیک می کنیم، نمایش گسترده ی آن مطابق شکل زیر است:



افزودن آیتم ها به صورت داینامیک به TreeView

در بخش قبلی ، یاد گرفتیم که چگونه آیتم ها را در زمان طراحی به TreeView اضافه کنیم. ما می توانیم آیتم ها را از کد به TreeView اضافه کنیم.

حالا بیایید UI را با اضافه کردن یک TextBox و button تکمیل کنیم. کد XAML برای کنترل های Button و TextBox به صورت زیر است:

```
<TextBox Height="23" HorizontalAlignment="Left" Margin="8,14,0,0"
        Name="textBox1" VerticalAlignment="Top" Width="127" />
<Button Height="23" Margin="140,14,0,0" Name="button1" VerticalAlignment="Top"
        HorizontalAlignment="Left" Width="76" Click="button1_Click">
    Add Item
</Button>
```

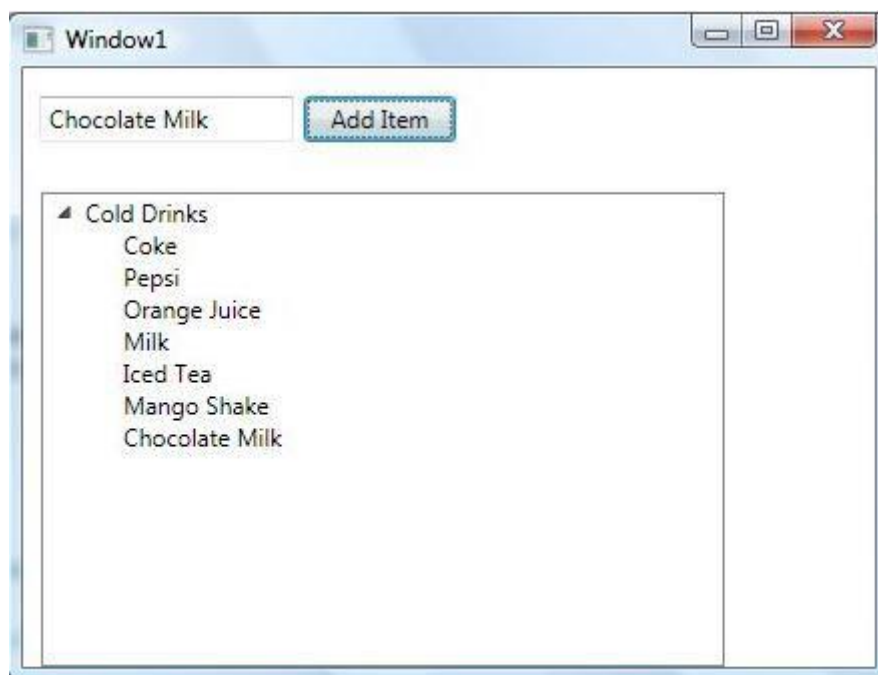
UI نهایی کار به صورت شکل زیر است. در رویداد button click مربوط به دکمه ی Add Item، می خواهیم یک آیتم به گره والد اضافه کنیم.



در رویداد button click، ما محتویات TextBox را با فراخوانی متد `TreeViewItem.Items.Add` به `TreeView` اضافه می کنیم. کد زیر محتویات TextBox را به آیتم های `TreeViewItem` اضافه می کند.

```
private void button1_Click(object sender, RoutedEventArgs e)
{
    TreeViewItem newChild = new TreeViewItem();
    newChild.Header = textBox1.Text;
    Parent.Items.Add(newChild);
}
```

در رویداد button click ما محتویات TextBox را با فراخوانی متد `TreeViewItem.Items.Add` به `TreeView` اضافه می کنیم. حالا اگر شما یک متن در `TextBox` وارد کنید و دکمه ی افزودن را بزنید ، محتویات `TextBox` به `TreeView` اضافه خواهد شد.



حذف آیتم های TreeView

ما می توانیم از متدهای `TreeView.Items.Remove` و یا `TreeView.Items.RemoveAt` برای حذف یک آیتم از مجموعه آیتم های `TreeView` استفاده کنیم . متد `RemoveAt`، با گرفتن `index` مربوط به آیتم ، کار حذف را انجام می دهد.

حالا برنامه را ویرایش می کنیم و یک `button` به نام `Delete Item` به آن اضافه می کنیم . کد `XAML` برای دکمه به صورت زیر است:

```
<Button Height="23" Margin="226,14,124,0" Name="DeleteButton"
  VerticalAlignment="Top" Click="DeleteButton_Click">
  Delete Item</Button>
```

در رویداد `button click` به صورت زیر است، ما `index` مربوط به آیتم انتخاب شده را پیدا می کنیم و متد `TreeView.Items.RemoveAt` را فراخوانی می کنیم.

```
private void DeleteButton_Click(object sender, RoutedEventArgs e)
{
  TreeView1.Items.RemoveAt
  (TreeView1.Items.IndexOf(TreeView1.SelectedItem));
}
```

کد بالا آیتم های ریشه (`root`) را از `TreeView` حذف می کند، نه آیتم های زیر مجموعه. برای حذف آیتم های زیر مجموعه (فرزند) ابتدا نیاز داریم تا آیتم انتخاب شده را پیدا کنیم و سپس متد `TreeViewItem.Items.RemoveAt` را فراخوانی کنیم.

زیباسازی آیتم های `TreeView`

کنترل TreeView در درون یک StackPanel قرار دارد که شامل یک کنترل ScrollViewer است. به این ترتیب زمانی که عرض و ارتفاع panel بیشتر از حد افزایش پیدا کند، scroll viewer فعال می شود و قابلیت scrolling عمودی و افقی را به ما می دهد.

برای زیبا سازی یک TreeView، می توانیم Property های مربوط به تک تک آیتم ها را به صورت جداگانه تنظیم کنیم. و یا می توانیم از System.Resources استفاده کنیم و Style property را تنظیم کنیم. تکه کد زیر مشخصه های foreground، font size و font weight را تنظیم می کند.

```
<Window.Resources>
  <Style TargetType="{x:Type TreeViewItem}">
    <Setter Property="Foreground" Value="Blue"/>
    <Setter Property="FontSize" Value="12"/>
    <Setter Property="FontWeight" Value="Bold" />
  </Style>
</Window.Resources>
```

TreeView جدید به صورت زیر خواهد بود:



نمایش تصویر در TreeView

ما می توانیم هر کنترلی را در داخل `TreeViewItem` قرار بدهیم . برای نمایش یک عکس در کنار یک متن ، به راحتی یک کنترل `TextBlock` و یک کنترل `Image` را در درون یک `StackPanel` قرار می دهیم. مشخصه ی `Image.Source` نام تصویری را که می خواهید در داخل کنترل `Image` نشان داده شود را می گیرد و `TextBlock.Text` رشته ای را می گیرد که می خواهید در داخل `TextBlock` نمایش داده شود.

تکه کد زیر یک تصویر و یک متن به `TreeViewItem` اضافه می کند. نکته اصلی این است که تصویر و متن را به بخش `header` در `TreeViewItem` ها اضافه کنیم.

```
<TreeViewItem Name="Child1">
  <TreeViewItem.Header>
    <StackPanel Orientation="Horizontal">
      <Image Source="coffie.jpg" Height="30"></Image>
      <TextBlock Text="Coffie"></TextBlock>
    </StackPanel>
  </TreeViewItem.Header>
</TreeViewItem>
```

بعد از این که کد را برای همه ی ۵ آیتم در `TreeView` تغییر دادیم، `TreeView` به صورت زیر خواهد بود:



استفاده از CheckBox در TreeView

زمانی که از CheckBox در TreeView استفاده می کنیم، می توانیم تصویر و یا هر متن دلخواه را به عنوان محتویات آن قرار بدهیم.

تکه کد زیر یک CheckBox همراه با تصویر و متن به TreeViewItem اضافه می کند.

```
<TreeViewItem Name="Child1">
  <TreeViewItem.Header>
    <CheckBox Name="CoffieCheckBox">
      <StackPanel Orientation="Horizontal">
        <Image Source="coffie.jpg" Height="30"></Image>
        <TextBlock Text="Coffie"></TextBlock>
      </StackPanel>
    </CheckBox>
  </TreeViewItem.Header>
</TreeViewItem>
```

تکه کد برنامه را تغییر می دهیم تا به همه ی آیتم ها CheckBox اضافه کنیم. نام هر CheckBox در درون Name property تعیین شده است که به این ترتیب می توانیم در صورت لزوم به آن ها دسترسی داشته باشیم.

```
<TreeViewItem Name="Child1">
  <TreeViewItem.Header>
    <CheckBox Name="CoffieCheckBox">
      <StackPanel Orientation="Horizontal">
        <Image Source="coffie.jpg" Height="30"></Image>
        <TextBlock Text="Coffie"></TextBlock>
      </StackPanel>
    </CheckBox>
  </TreeViewItem.Header>
</TreeViewItem>
<TreeViewItem Name="Child2">
  <TreeViewItem.Header>
    <CheckBox Name="IcedTeaCheckBox">
      <StackPanel Orientation="Horizontal">
```



```
<Image Source="IcedTea.jpg" Height="30"></Image>
  <TextBlock Text="Iced Tea"></TextBlock>
</StackPanel>
</CheckBox>
</TreeViewItem.Header>
</TreeViewItem>
<TreeViewItem Name="Child3">
  <TreeViewItem.Header>
    <CheckBox Name="MangoShakeCheckBox">
      <StackPanel Orientation="Horizontal">
        <Image Source="MangoShake.jpg" Height="30"></Image>
        <TextBlock Text="Mango Shake"></TextBlock>
      </StackPanel>
    </CheckBox>
  </TreeViewItem.Header>
</TreeViewItem>
<TreeViewItem Name="Child4">
  <TreeViewItem.Header>
    <CheckBox Name="MilkCheckBox">
      <StackPanel Orientation="Horizontal">
        <Image Source="Milk.jpg" Height="30"></Image>
        <TextBlock Text="Milk"></TextBlock>
      </StackPanel>
    </CheckBox>
  </TreeViewItem.Header>
</TreeViewItem>
<TreeViewItem Name="Child5">
  <TreeViewItem.Header>
    <CheckBox Name="TeaCheckBox">
      <StackPanel Orientation="Horizontal">
        <Image Source="Tea.jpg" Height="30"></Image>
        <TextBlock Text="Tea"></TextBlock>
      </StackPanel>
    </CheckBox>
  </TreeViewItem.Header>
</TreeViewItem>
<TreeViewItem Name="Child6">
  <TreeViewItem.Header>
    <CheckBox Name="OrangeJuiceCheckBox">
```

```
<StackPanel Orientation="Horizontal">  
  <Image Source="OrangeJuice.jpg" Height="30"></Image>  
  <TextBlock Text="Orange Juice"></TextBlock>  
</StackPanel>  
</CheckBox>  
</TreeViewItem.Header>  
</TreeViewItem>  
</TreeViewItem>
```

حالا برنامه را اجرا کنید و خروجی را ببینید.

