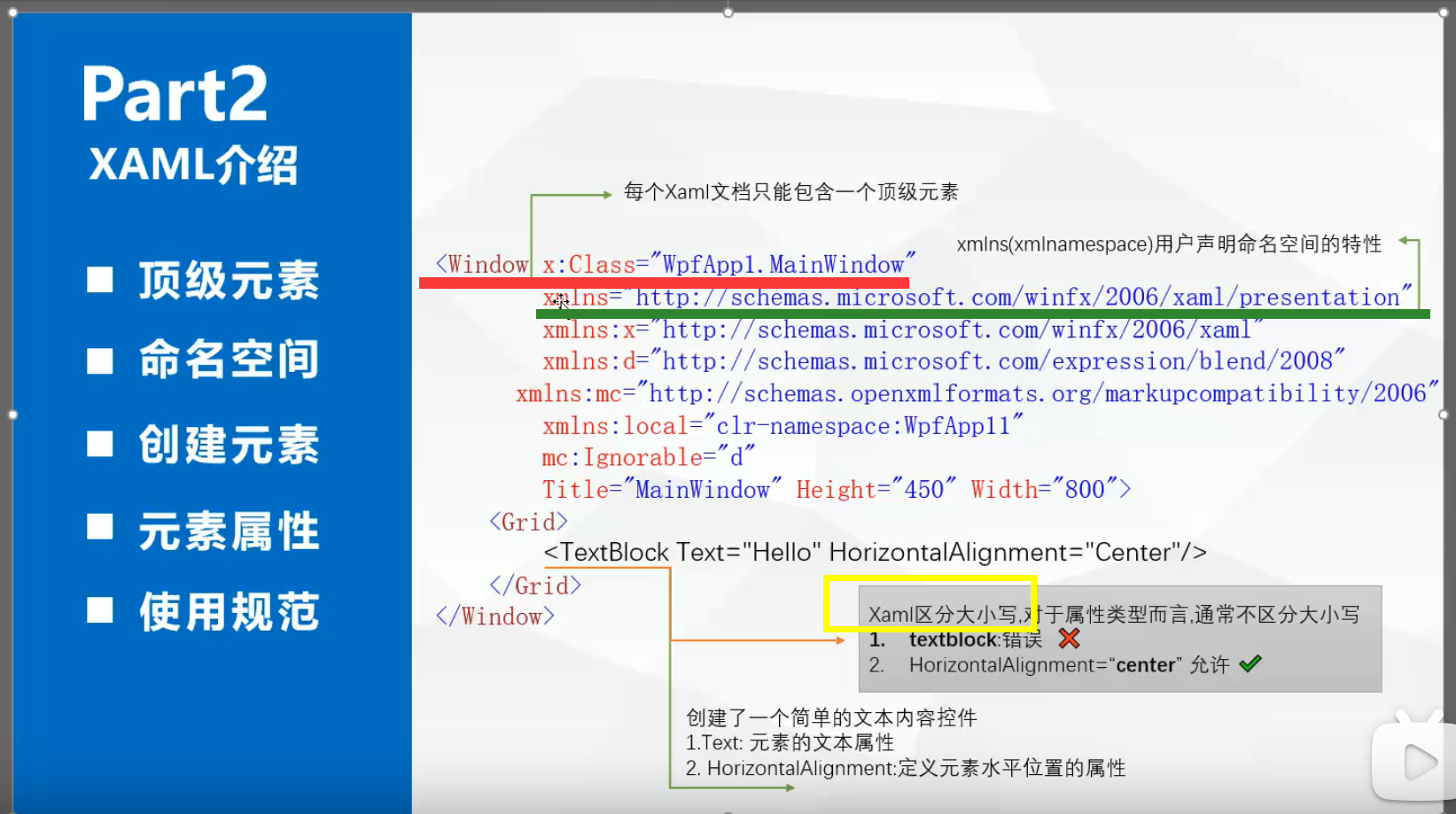
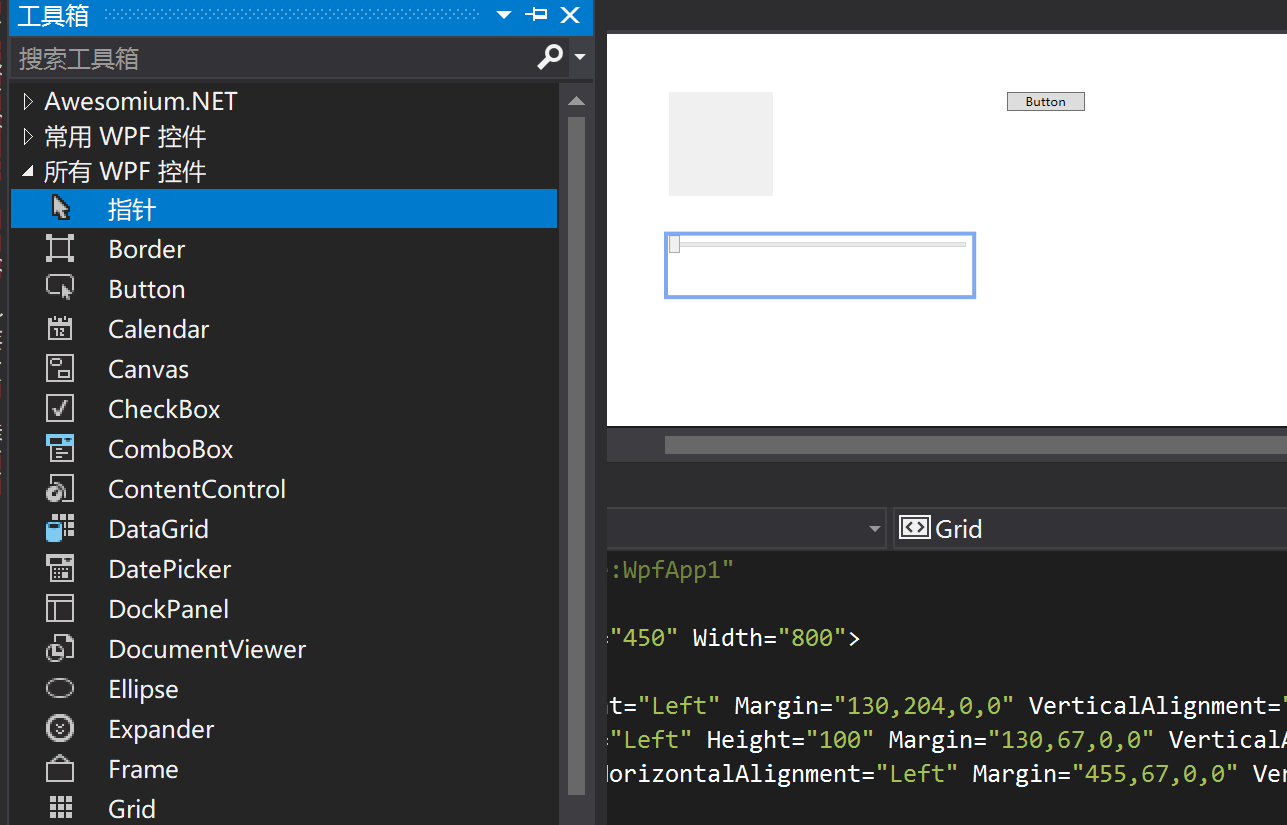
# WPF学习笔记

课件来自 [WPF深入讲解合集-bilibili](https://www.bilibili.com/video/BV1HC4y1b76v)

项目源代码：https://github.com/HenJigg/Microsoft-ToDo-Video

## 开篇入门



* 每个XAML文档只能有一个顶级元素
* 有命名空间的性质
* XAML格式区分大小写，但属性类型可以不区分大小写
* 
* 创建元素可以在工具箱拖控件到设计窗口，也可以手动写

### WPF的编译过程

.xaml和.cs文件经历如下过程：转换成.baml资源（即.xaml的二进制形式，解析速度更佳），嵌入到程序集中，应用程序工作时再从程序集读取这些控件信息

这里课件是用 [dnSpy](https://github.com/dnSpy/) 反编译读.dll文件得出该结论，还反编译了虎牙直播的图标资源作为示例，看起来不错（？）

## 布局和控件

小技巧：VS用`Ctrl+K`和`Ctrl+C`快捷键可以快速注释掉多行代码

### 常见的布局属性

|  |  |
| --- | --- |
| 常用属性 | 功能 |
| HorizontalAlignment | 设置元素水平位置 |
| VerticalAlignment | 设置元素垂直位置 |
| Margin | 指定元素与容器的边距 |
| Height | 指定元素高度 |
| Width | 指定元素宽度 |
| WinHeight / WinWidth | 指定元素最小高度和宽度 |
| MaxHeight / MaxWidth | 指定元素最大高度和宽度 |
| Padding | 指定元素内部边距 |

### 常见的布局容器

|  |  |
| --- | --- |
| 布局容器 |  |
| Grid | 设置框架整体页面布局  可把空间切分成多行多列 |
| StackPanel | 可用Orientation属性设置元素排列方式，默认是垂直布局 |
| WrapPanel | 可用Orientation属性设置元素排列方式，默认是水平布局  具备自动换行的特点 |
| DockPanel | 设置元素的锚定位置 |
| UniformGrid | 设置元素均匀分布地填充空间  可以显式指定行数列数 |

#### 例：Grid分隔2行2列表格：

<Grid ShowGridLines="True">

<Grid.RowDefinitions>

<RowDefinition> </RowDefinition>

<RowDefinition> </RowDefinition>

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition> </ColumnDefinition>

<ColumnDefinition> </ColumnDefinition>

</Grid.ColumnDefinitions>

</Grid>

#### 例：Orientation="Horizontal"

<StackPanel Orientation="Horizontal">

<!-- 在未指定时, WrapPanel默认状态是Orientation="Horizontal", StackPanel默认Orientation="Horizontal"-->

<Button Width="100" Height="30" Background="Red"></Button>

<Button Width="100" Height="30" Background="Yellow"></Button>

<Button Width="100" Height="30" Background="Blue"></Button>

<Button Width="100" Height="30" Background="Green"></Button>

<Button Width="100" Height="30" Background="Pink"></Button>

</StackPanel>

#### 例：DockPanel

<DockPanel>

<Button DockPanel.Dock="Bottom" Width="100" Height="30"></Button>

<Button DockPanel.Dock="Left" Width="100" Height="30"></Button>

<Button DockPanel.Dock="Right" Width="100" Height="30"></Button>

<Button DockPanel.Dock="Top" Width="100" Height="30"></Button>

</DockPanel>

#### 例：UniformGrid

<UniformGrid Columns="3" Rows="4">

<Button Width="100" Height="30"></Button>

<Button Width="100" Height="30"></Button>

<Button Width="100" Height="30"></Button>

<Button Width="100" Height="30"></Button>

<Button Width="100" Height="30"></Button>

</UniformGrid>

### 控件结构