**WPF 的内部世界（控件与布局）**

**目录**

[一、控件与布局](https://juejin.im/post/6844904070050742280)

* WPF作为专门的用户界面技术，布局的功能是它的核心功能之一。友好的用户界面和良好的用户体验离不开设计精良的布局。
* WPF设计师工作量最大的俩部分就是布局和动画，布局是静态的，动画是动态的，用户体验就是用 户在这动静之中与软件功能产生交互时的感受。
* 也就是说，布局就是WPF的衣服！

**二、控件**

一个页面的布局，显示。都是由一个个控件组成的。控件们离不开WPF这个赖以生存的环境，组成了一幅幅美丽生动的画面（布局）。

在开始学习这些布局元素前，我们要知道每个布局元素都有自己的特点，我们要灵活使用。切莫不要无所不用其极，要合理搭配。

（就像生态环境一样，要合理搭配，否则就会出现“生物入侵”这种“偷鸡不成蚀把米”的行为，加重了生态环境的负担。破坏）

**1、 控件的分类**

**粗略而言，日常工作中我们打交道最多的控件无外乎6类，即：**

* 1、布局控件：可以容纳多个控件或嵌套其他布局控件，用于UI上组织和排列控件，如：StackPanel，Grid，Dock，WrapPanel,，Canvas；
* 2、内容控件：只能容纳一个其他控件或布局控件作为它的内容，如：Button，Window；
* 3、带标题的内容控件：相当于一个内容控件，但可以加一个标题，如：Group Box，TabItem；
* 4、条目控件：可以显示一列数据，一般情况下这列数据的类型相同。如ListBox，ComboBox;
* 5、带标题的条目控件：相当于给一个条目控件加上一个标题显示区，如：TreeViewItem，MenuItem，往往用于显示层级数据;
* 6、特殊的内容控件：比如TextBox容纳的是字符串，TeztBlock可以容纳可自由控制格式的文本，Image容纳图片类型数据……这类的控件相对比较独立。

至于，为什么这么分类呢，其实我们只需要细细体会一下，就明白了。实在不行，你从工具栏拖出来，看着它的样子，在看着我的话。“什么？你还不懂”………………………………作者卒

好，我们不对这些控件做太多详细介绍，我们主要介绍布局控件，其他相信属性可以参考下面： WPF 基本控件使用介绍：[blog.csdn.net/niewq/artic…](https://blog.csdn.net/niewq/article/details/50244227" \t "_blank)

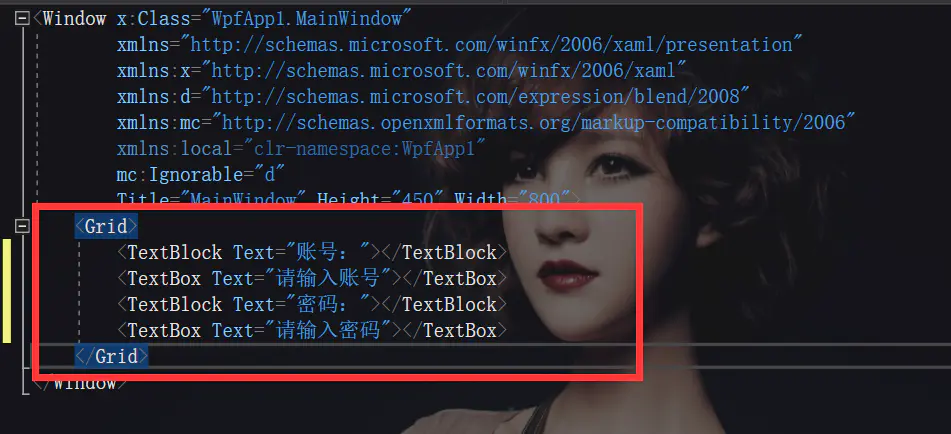
**三、布局控件**

WPF为我们提供了5中布局方式，他们的特点各不相同，可以相互嵌套，让我们来认识一下吧。

* 1、Grid ：列表布局
* 2、StackPanel ：堆栈面板布局
* 3、WrapPanel : 流布局面板（当元素水平对齐，内容超过宽度时，自动换行；当元素垂直对齐，内容超过高度时，自动换列）
* 4、DockPanel：停靠面板
* 5、Canvas：坐标面板

**1、Grid列表布局**

Grid一词译为“网格；格子，栅格” 没错，它就像一个网格一样把我们的页面分割成一块又一块。



我们在窗体放置了2个TextBlock和2个TextBox，想实现登录窗口的样式，可是不进人意，他们都重叠在了一起。

**为什么都重叠在了一起呢？**

因为，我们没有对Grid这个容器做相关调整，他现在是一个一行一列的“大格子”。在一个各自当然就重叠了，除非去设置Magin属性，当然我们想要实现的并不是这种效果。

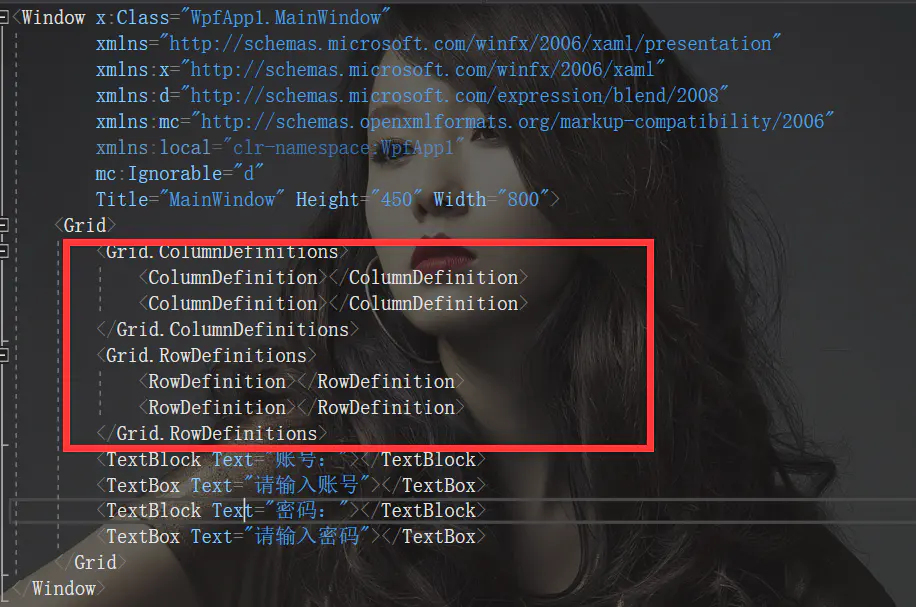
**所以，我们通过设置**

* 列<Grid.ColumnDefinitions></Grid.ColumnDefinitions>
* 行<Grid.RowDefinitions></Grid.RowDefinitions>

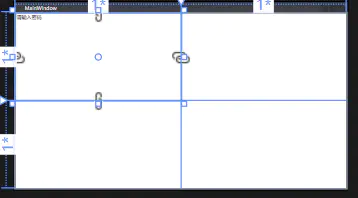
**“分割Grid”**

我们可以通过添加ColumnDefinitions节点和RowDefinitions节点 ，来确定把我们的“Grid分割成几个格子”。

如图，分割成了四个格子：



通过看设计窗口，我们也会发现，Grid被线条分割成了四块：



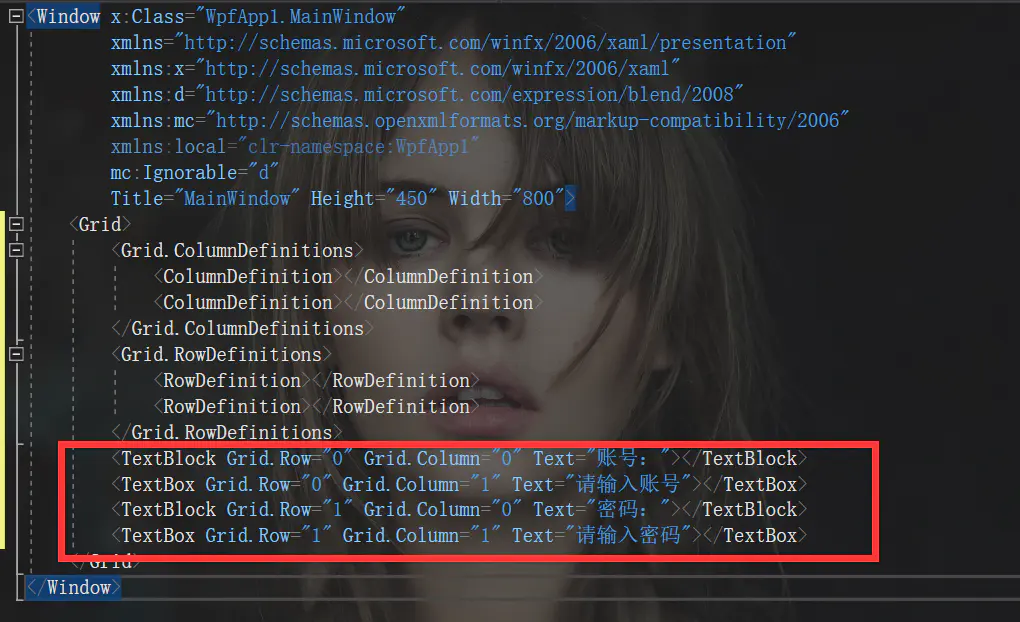
唉？不对啊，为什么我们的控件还重叠的呢？是因为我们没有去设置他们处于哪个格子，接下载，我们设置一下。

**确定位置，与合并单元格**

**当我们的容器处于Grid布局容器里时，会增加附加属性：**

* 行所在位置： Grid.Row="0"
* 列所在位置： Grid.Column="0"

默认不设置，值为0 ，所以才会出现重叠的情况，来我们调整一下位置。

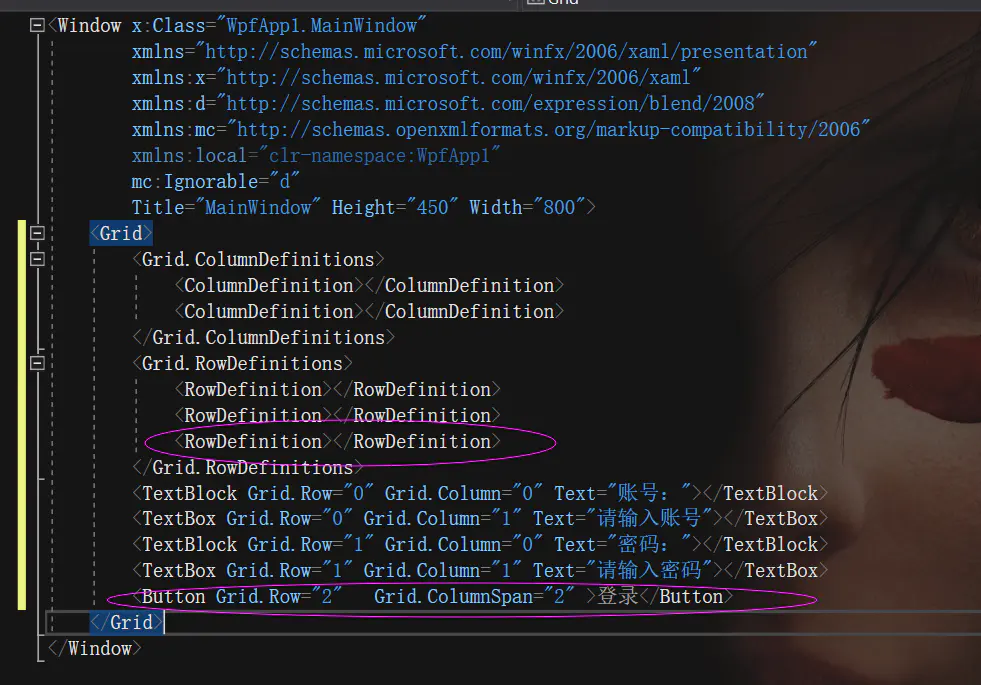


好了，一切恢复正常了，在增加一个按钮，登录怎么能没有按钮呢！

**通过设置**

* 行单元格合并： Grid.RowSpan="1"
* 列单元格合并： Grid.ColumnSpan="1"

默认值位1，合并几个就写几。 经过稍作修改，我们的界面变成了这样：





额……长得有点丑，我们添加些属性来调整一下，

**设置宽高**

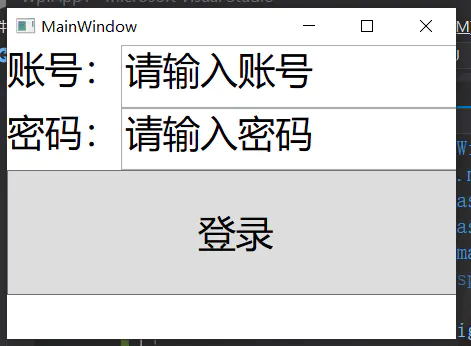
我们通过对ColumnDefinition的Width来设置宽和RowDefinition的Height来设置高

**Width和Height支持像素，比例，以及自适应**

* 像素： 直接用数字表示即可
* 比例： 以\*做单位
* 自适应： 设置值位 auto

好，我们来运用上面的知识，调整一下我们的视图：





**总结**

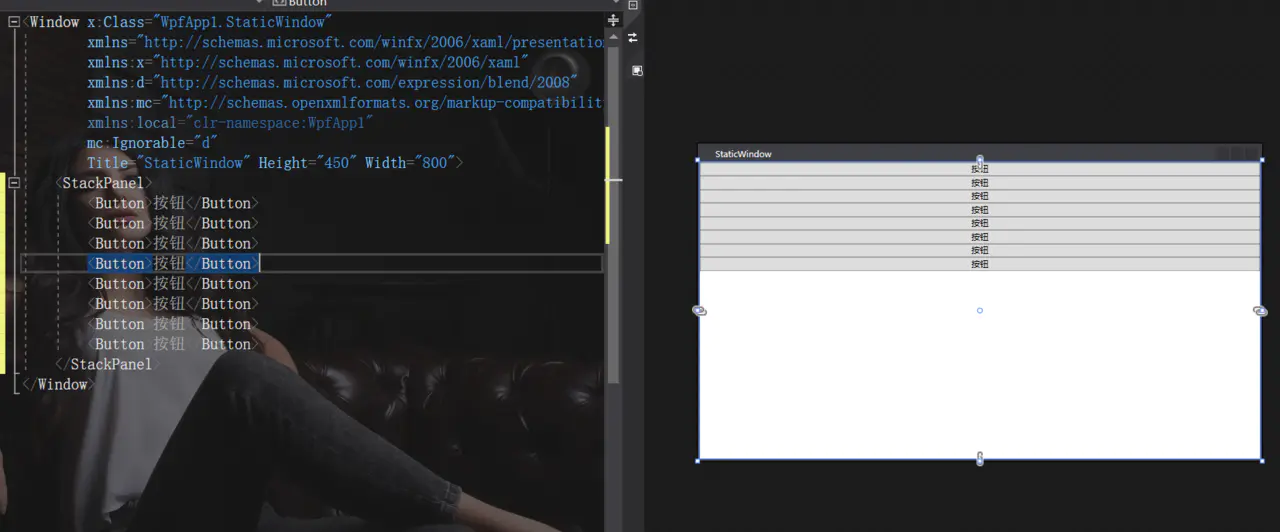
好，我们来总结一下上面的宽高：

* 首先我们对Grid的列经行了比例设置1：3（\*，3\*），当我们拉动窗体大小时，会发现它们的大小是等比例变化的。
* 其次，对Grid的行设定了俩个固定高度，我们可以发现无论窗体怎么变化他们的高度是不变的。
* 最后一行设定了自适应，我刻意把按钮的高度设为100，我们可以看到表格最后一行也为100，可以知道auto是根据内容来自适应的。

**2、StackPanel堆栈面板**

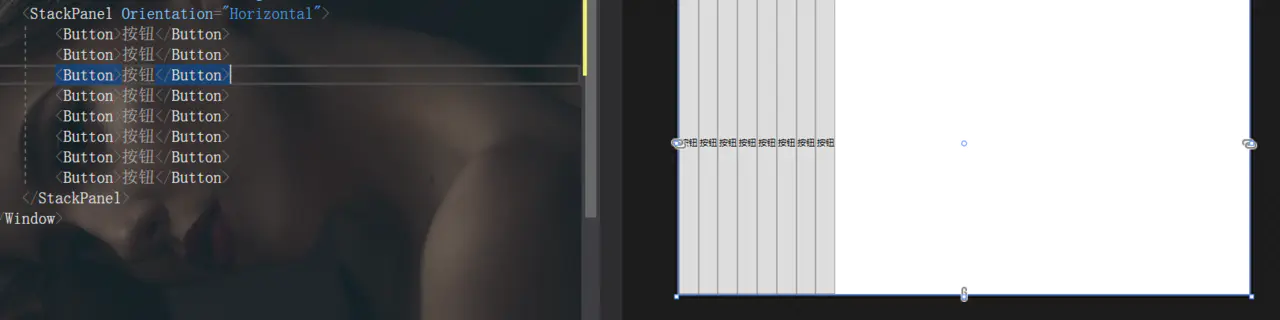
大家都对“堆栈”俩个词不陌生吧，堆栈面板就好像容器在“排队”一样，我们把Window下的Grid 换位StackPanel，来体验一下吧。

**设置方向**

****

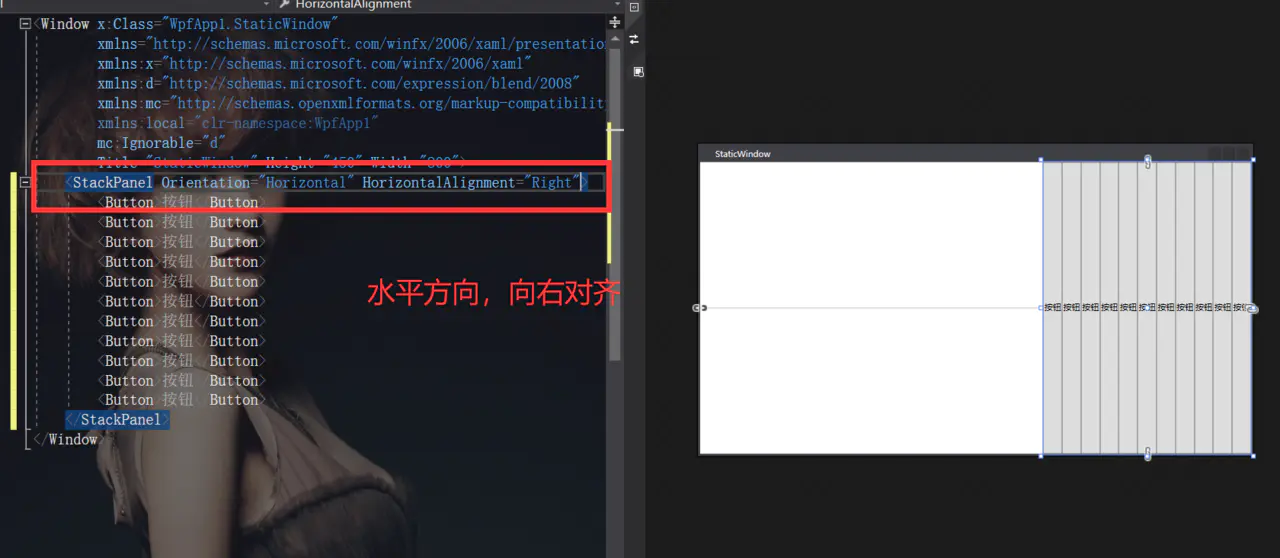
我们在StackPanel里放置了很多按钮，发现他们就像排队一样，一个接一个，水平方向。如果你不喜欢这样，当然是可以改方向的啦！

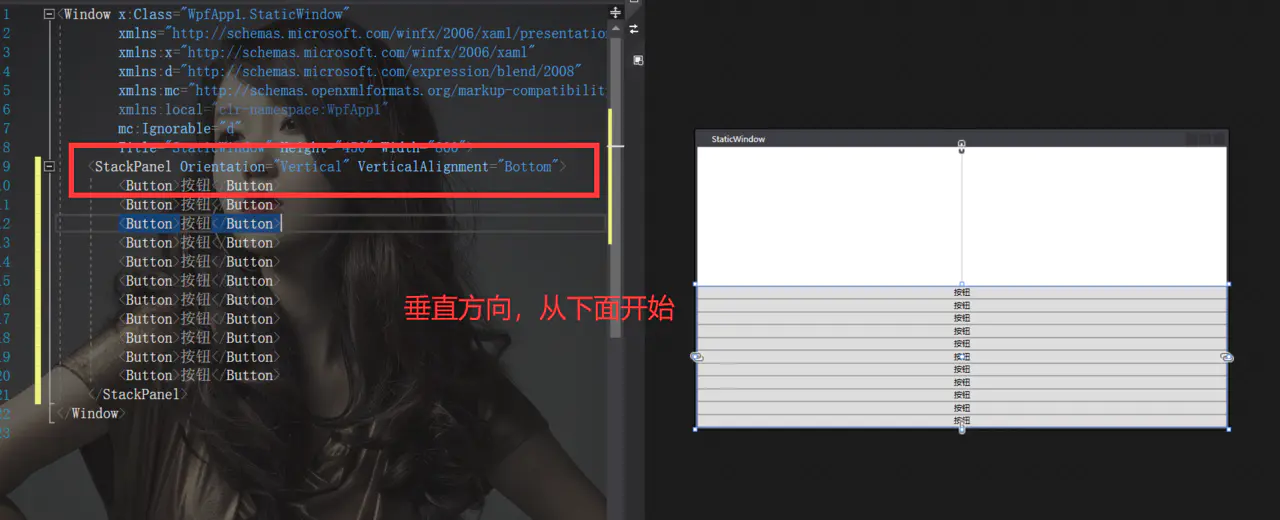
**通过设置Orientation**

* 水平对齐： Orientation="Horizontal"
* 垂直对齐： Orientation="Vertical"（默认）
* 

对于内部元素自身，也可以选择对其对齐方式

* HorizontalAlignment="left" Center,right
* VerticalAlignment="Top" Bottom,Center,Stretch





**3、WrapPanel 流动布局**

当元素水平对齐，内容超过宽度时，自动换行；当元素垂直对齐，内容超过高度时，自动换列）

它可能和上面的长得像，其实是不一样的。当StackPanel的内部子元素数量超出宽度（高度）会溢出窗体，而流动布局会自动换行。常用于动态数据生成。

这就不做详细介绍了，属性都和上面的一样。

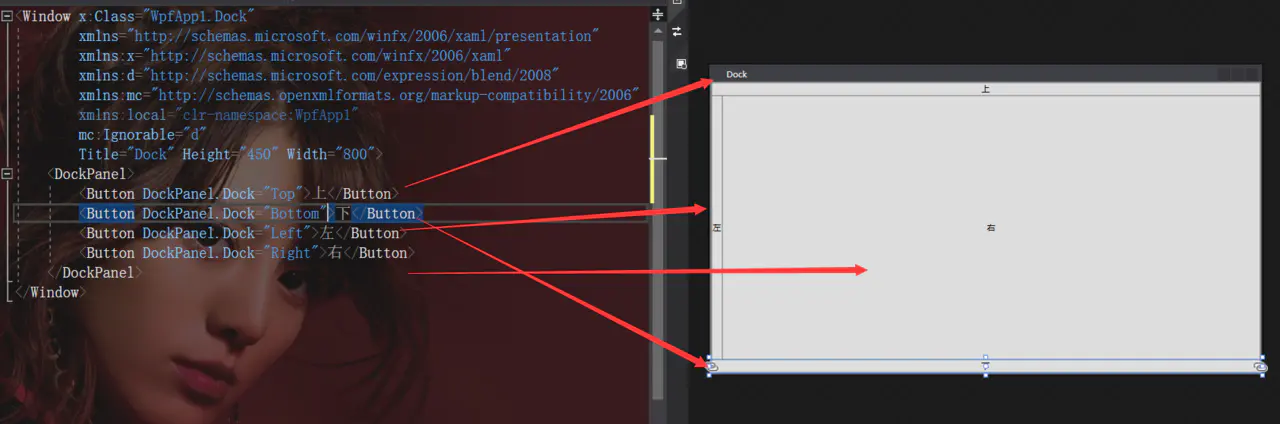
**4、DockPanel 停靠布局**

做过winfrom开发的朋友，都知道一个Docl属性吧，那么我们的Dock布局也是这个道理，我们来实践一下。

**在DockPanel的容器里的控件，会增加一个附加属性DockPanel.Dock**

* DockPanel.Dock="Top"
* DockPanel.Dock="Bottom"
* DockPanel.Dock="Left"
* DockPanel.Dock="Right"

分别是上下左右停靠，要注意的是**控件会随着设定的先后顺序，具有不同的宽高（或大小），默认最后一个停靠控件的大小占剩下界面的全部**



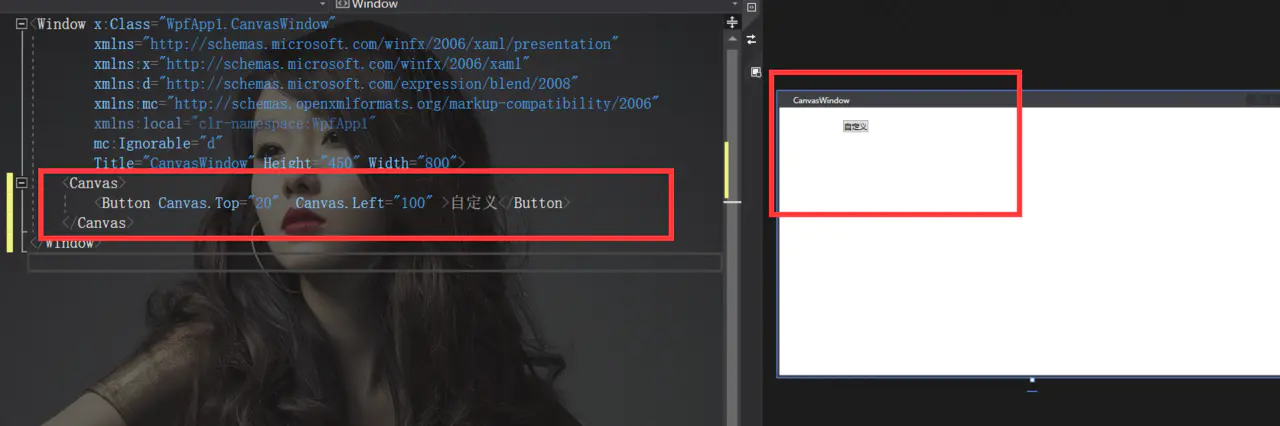
仔细看左 和 右 ，你就会知道我说的注意项，新停靠的控件占剩余页面的所有区域。

**5、Canvas 坐标布局**

好吧，这个真的就相当于winform的布局了， 设置坐标，确定控件的位置。

**在Canvas容器里的控件，会增加附加属性**

* 距离窗口上方： Canvas.Top="20"
* 距离窗口左方：Canvas.Left="100"
* 距离窗口下方：Canvas.Bottom="20"
* 距离窗口右方：Canvas.Right="0"



**四、综合小案例**

如果都看完的话，我们来做一个小案例吧。



<Grid>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="\*"></ColumnDefinition>

<ColumnDefinition Width="4\*"></ColumnDefinition>

</Grid.ColumnDefinitions>

<Grid.RowDefinitions>

<RowDefinition Height="auto"></RowDefinition>

<RowDefinition></RowDefinition>

</Grid.RowDefinitions>

<StackPanel Grid.RowSpan="2" Grid.Column="0">

<Button Height="50">新建StackPanel</Button>

<Button Height="50">保存StackPanel</Button>

<Button Height="50">增加StackPanel</Button>

<Button Height="50">导入StackPanel</Button>

<Button Height="50">导出StackPanel</Button>

<Button Height="50">关闭StackPanel</Button>

</StackPanel>

<DockPanel Grid.Row="0" Grid.Column="1">

<TextBlock DockPanel.Dock="Top" HorizontalAlignment="Center">我是DockPanel</TextBlock>

<WrapPanel>

<Button Width="200">WrapPanel</Button>

<Button Width="200">WrapPanel</Button>

<Button Width="200">WrapPanel</Button>

<Button Width="200">放不下啦WrapPanel</Button>

<Button Width="200">WrapPanel</Button>

</WrapPanel>

</DockPanel>

<Canvas Grid.Row="1" Grid.Column="1">

<TextBlock Canvas.Top="100" Canvas.Left="100">Canvas账号：</TextBlock>

<TextBox Canvas.Top="100" Canvas.Left="200">Canvas请输入账号：</TextBox>

<TextBlock Canvas.Top="130" Canvas.Left="100">Canvas密码：</TextBlock>

<TextBox Canvas.Top="130" Canvas.Left="200">Canvas请输入密码：</TextBox>

</Canvas>

</Grid>