

WPF (Windows Presentation Foundation)

Layoutmanager & Komponenten Handout

Was ist ein Layoutmanager?

Ein Layoutmanager ist ein Container Element in WPF, das dafür zuständig ist, wie andere UI-Elemente angeordnet werden.

Er übernimmt folgende Aufgaben:

- Positionierung von Elementen
- Größenanpassung
- Ausrichtung innerhalb des Fensters

Ein Layoutmanager arbeitet relativ und dynamisch, nicht mit festen Pixelwerten.

Das bedeutet:

- Elemente passen sich automatisch an die Fenstergröße an
- Inhalte können sich je nach Textlänge oder Größe verändern

Welche Layoutmanager gibt es?

Die wichtigsten Layoutmanager in WPF sind:

- Grid
- StackPanel
- WrapPanel
- DockPanel
- Canvas

Grid - Der wichtigste Layoutmanager

Das Grid ist der am häufigsten verwendete Layoutmanager. Er funktioniert ähnlich wie eine Tabelle.

Das Layout wird in Zeilen (Rows) und Spalten (Columns) aufgeteilt.

Aufbau:

- `<Grid.ColumnDefinitions>` definiert Spalten
- `<ColumnDefinition/>` erstellt eine einzelne Spalte
- `<Grid.RowDefinitions>` definiert Reihen
- `<RowDefinition/>` erstellt eine einzelne Reihe

Alle Elemente müssen wie in XML mit schließenden Tags geschrieben werden.

Positionierung von Elementen

UI-Elemente können einer bestimmten Zelle zugeordnet werden.

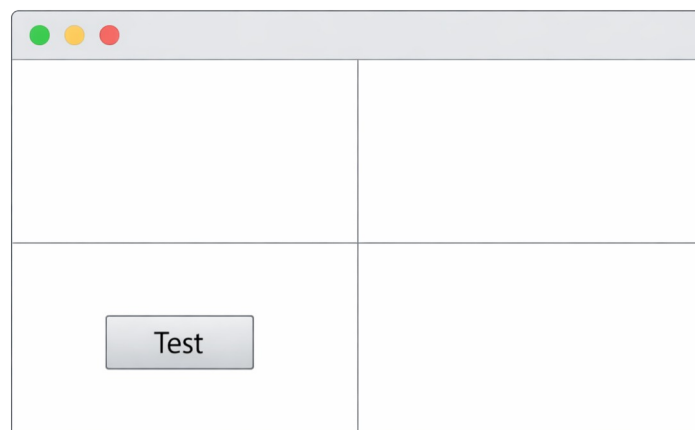
Standardmäßig beginnt die Zählung bei Row 0 und Column 0.

Beispiel:

```
<Grid>
  <Grid.ColumnDefinitions>
    <ColumnDefinition/>
    <ColumnDefinition/>
  </Grid.ColumnDefinitions>

  <Grid.RowDefinitions>
    <RowDefinition/>
    <RowDefinition/>
  </Grid.RowDefinitions>

  <Button Grid.Column="0" Grid.Row="1">Test</Button>
</Grid>
```



StackPanel - Einfach & schnell

Das StackPanel ordnet Elemente automatisch in einer Linie an.

Diese können:

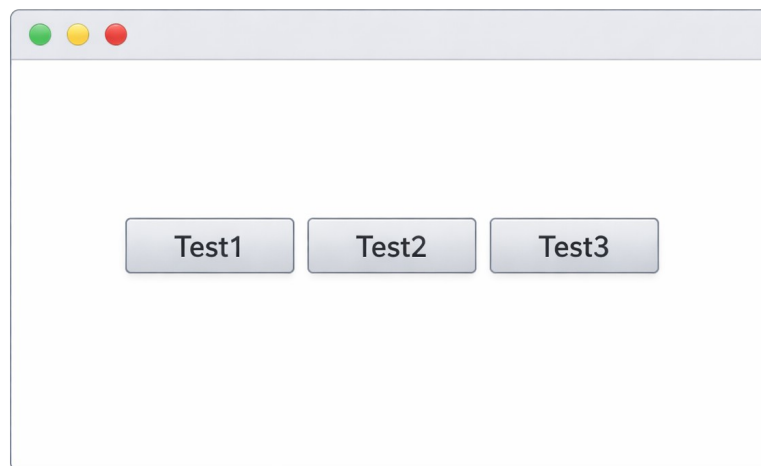
- vertikal untereinander
- horizontal nebeneinander

angeordnet werden.

Die Ausrichtung wird mit der Eigenschaft Orientation festgelegt.

Beispiel:

```
<StackPanel Orientation="Horizontal">  
  <Button>Test1</Button>  
  <Button>Test2</Button>  
  <Button>Test3</Button>  
</StackPanel>
```



WrapPanel - Flexibles Layout

Das WrapPanel funktioniert ähnlich wie das StackPanel.

Der Unterschied:

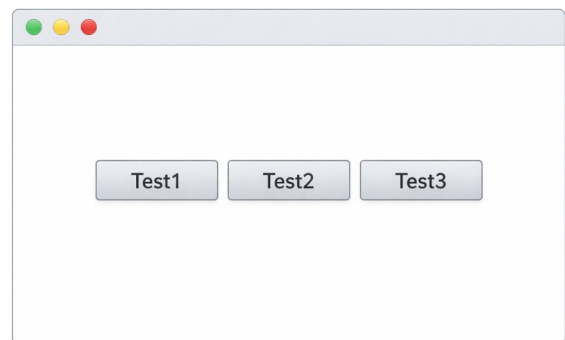
Wenn nicht genug Platz vorhanden ist, werden Elemente automatisch in die nächste Zeile verschoben.

Dadurch eignet sich das WrapPanel besonders für:

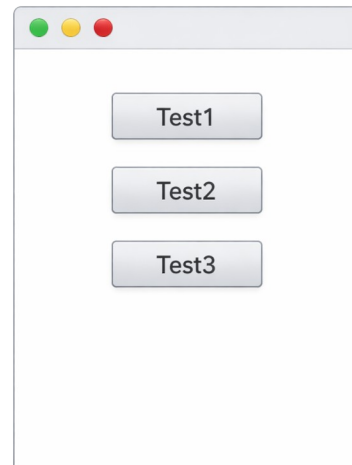
- Bildergalerien
- Kachel-Layouts

Beispiele:

```
<WrapPanel Orientation="Horizontal">  
  <Button>Test1</Button>  
  <Button>Test2</Button>  
  <Button>Test3</Button>  
</WrapPanel>
```



```
<WrapPanel Orientation="Vertical">  
  <Button>Test1</Button>  
  <Button>Test2</Button>  
  <Button>Test3</Button>  
</WrapPanel>
```



DockPanel - Elemente andocken

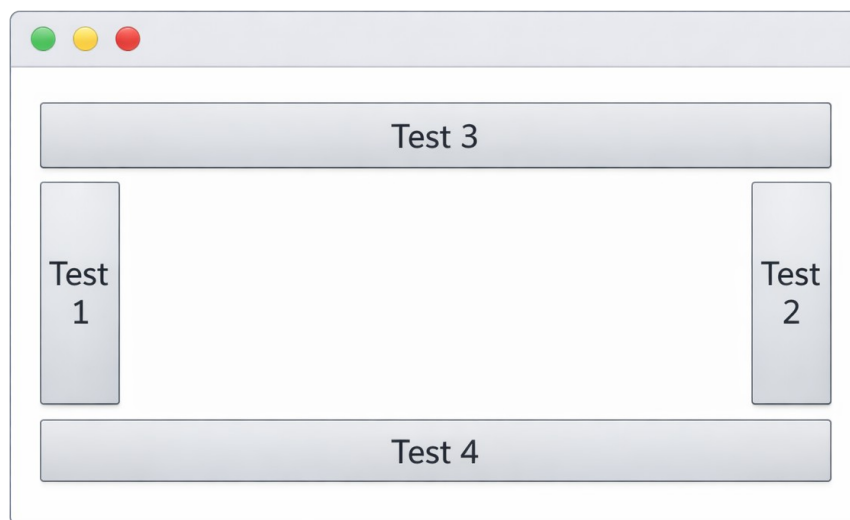
Im DockPanel werden Elemente an den Seiten eines Containers befestigt.

Mögliche Positionen:

- Top
- Bottom
- Left
- Right

Beispiel:

```
<DockPanel>  
  <Button DockPanel.Dock="Left">Test 1</Button>  
  <Button DockPanel.Dock="Right">Test 2</Button>  
  <Button DockPanel.Dock="Top">Test 3</Button>  
  <Button DockPanel.Dock="Bottom">Test 4</Button>  
</DockPanel>
```



Canvas - Absolute Positionierung

Das Canvas arbeitet mit festen Positionswerten.

Elemente werden mit X- und Y-Koordinaten positioniert.

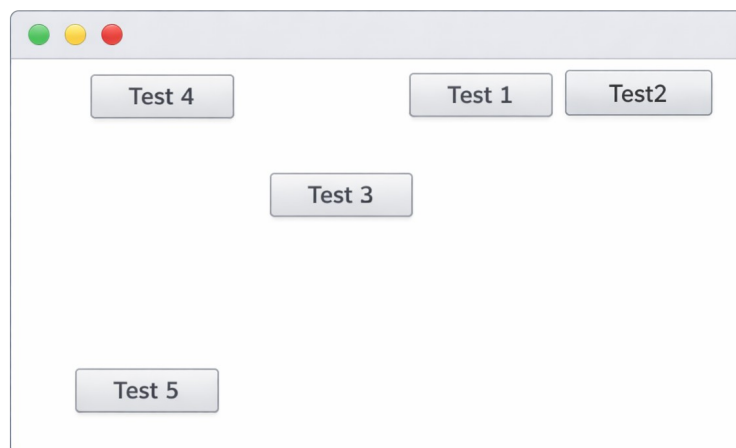
Wichtige Eigenschaften:

- Canvas.Left
- Canvas.Right
- Canvas.Top
- Canvas.Bottom

Im Gegensatz zu anderen Layoutmanagern passt sich das Canvas nicht automatisch an.

Beispiel:

```
<Canvas>  
  <Button Canvas.Right="200">Test 1</Button>  
  <Button Canvas.Right="150">Test 2</Button>  
  <Button Canvas.Top="100" Canvas.Right="350">Test 3</Button>  
  <Button Canvas.Left="150">Test 4</Button>  
  <Button Canvas.Bottom="50" Canvas.Left="150">Test 5</Button>  
</Canvas>
```



Weitere WPF-Komponenten

Neben Layoutmanagern gibt es auch UI-Steuerelemente.

Wichtige Beispiele:

- TextBox
- PasswordBox
- TabControl
- GroupBox

TextBox

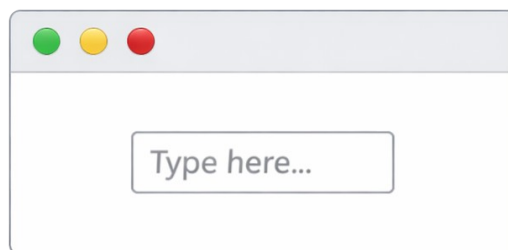
Ein Eingabefeld für normalen Text.

Eigenschaften:

- ermöglicht Benutzereingaben
- wird häufig in Formularen verwendet

Beispiel:

```
<TextBox Width="100" Height="40"/>
```



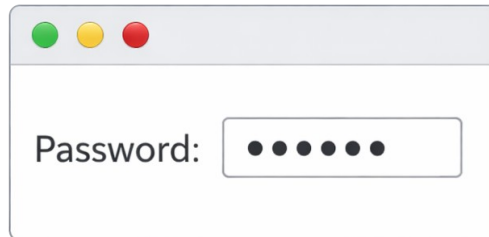
PasswordBox

Ein Eingabefeld für Passwörter.

Die eingegebenen Zeichen werden als Punkte angezeigt.

Beispiel:

```
<PasswordBox Width="100"/>
```



TabControl

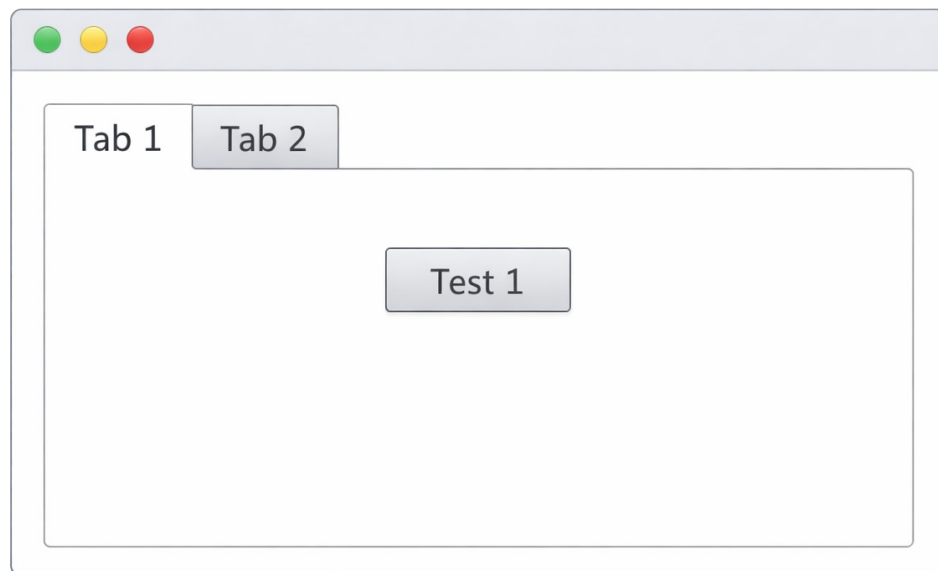
Ein Container für mehrere Seiten innerhalb eines Fensters.

Eigenschaften:

- Darstellung mit Tabs
- ermöglicht Navigation zwischen Bereichen

Beispiel:

```
<TabControl>  
  <TabItem Header="Tab 1">  
    <Button>Test</Button>  
  </TabItem>  
  <TabItem Header="Tab 2">  
    <Button>Test</Button>  
  </TabItem>  
</TabControl>
```



GroupBox

Eine GroupBox ist ein Rahmen mit einer Überschrift.

Sie dient dazu, zusammengehörige Elemente zu gruppieren.

Beispiel:

```
<GroupBox Width="300" Height="100">  
  <StackPanel Orientation="Horizontal">  
    <Button>Test1</Button>  
    <Button>Test2</Button>  
  </StackPanel>  
</GroupBox>
```

