

Handout – Layoutmanager

1. Canvas:

```
<Canvas> [Kindelemente] </Canvas>
```

Eigenschaften:

- Canvas.Left, bzw. Canvas.Right für die horizontale Ausrichtung
- Canvas.Top, bzw. Canvas.Bottom für die vertikale Ausrichtung

Beispiel:

```
<Canvas>  
    <Image Source="_" Canvas.Left ="5" Canvas.Top="10" />  
    <Button Canvas.Right ="5" Canvas.Bottom="10"> Hallo Welt </Button>  
</Canvas>
```

2. StackPanel:

```
<StackPanel> [Kindelemente] </StackPanel>
```

Eigenschaften:

- Orientation="Vertical" (Standardwert)
-> Alle Elemente sind übereinander gestapelt
- Orientation="Horizontal"
-> Alle Elemente sind nebeneinander gestapelt
- Je nach Orientierung wird entweder die Höhe oder Breite der Elemente durch das Panel vorgegeben. Der Platz wird "gefüllt".

Beispiel:

```
<StackPanel Orientation="Vertical" Margin="10">  
    <TextBlock Text="Willkommen" FontSize="16" />  
    <Button Content="Button 1" Margin="5" />  
    <Button Content="Button 2" Margin="5" />  
    <Button Content="Button 3" Margin="5" />  
    <TextBox Margin="10" Text="Geben Sie hier etwas ein..." />  
</StackPanel>
```

3. DockPanel:

```
<DockPanel> [Kindelemente] </DockPanel>
```

Eigenschaften:

- LastChildFill="True" (Standardwert)
 - Gibt an, ob das letzte Kindelement den verbleibenden Platz im DockPanel füllen soll
- DockPanel.Dock
 - Bestimmt, wo ein Kindelement im DockPanel angedockt wird
Mögliche Attribute: Top, Bottom, Left, Right

Beispiel:

```
<DockPanel>  
  <Button DockPanel.Dock="Top">Oben</Button>  
  <Button DockPanel.Dock="Bottom">Unten</Button>  
  <Button DockPanel.Dock="Left">Links</Button>  
  <Button DockPanel.Dock="Right">Rechts</Button>  
  <TextBlock>Inhalt in der Mitte</TextBlock>  
</DockPanel>
```

4. Grid:

```
<Grid> [Kindelemente] </Grid>
```

Eigenschaften:

- RowDefinitions und ColumnDefinitions werden zur Definition von Zeilen und Spalten genutzt
- Mit Grid.Row und Grid.Column (Standard: 0) können Kindelemente in die verschiedenen Grid-Bereiche eingeteilt werden
- Mit Grid.ColumnSpan und Grid.RowSpan (Standard: 1) können Kindelemente über mehrere Zeilen oder Spalten gestreckt werden

Beispiel:

```
<Grid>
  <Grid.RowDefinitions>
    <RowDefinition Height="Auto" />
    <RowDefinition Height="*" />
  </Grid.RowDefinitions>
  <Grid.ColumnDefinitions>
    <ColumnDefinition Width="*" />
    <ColumnDefinition Width="2*" />
  </Grid.ColumnDefinitions>

  <TextBlock Grid.Row="0" Grid.Column="0">Zeile 0, Spalte 0</TextBlock>
  <TextBlock Grid.Column="1">Zeile 0, Spalte 1</TextBlock>
  <TextBlock Grid.Row="1"
    Grid.ColumnSpan="2">Zeile 1, Spalte 0 + 1</TextBlock>
</Grid>
```

Allgemein kann jedes Panel ineinander verwendet werden und auch in sich selbst verschachtelt sein. Kopieren Sie sich gerne mal das Beispiel hier in ein WPF-Projekt:

```

<DockPanel>
  <DockPanel DockPanel.Dock="Left" LastChildFill="False">
    <GroupBox Header="Auswahl" DockPanel.Dock="Top" Margin="5, 0, 0, 5">
      <StackPanel>
        <Button Margin="5">Hauptmenü</Button>
        <Button Margin="5">Level1</Button>
        <Button Margin="5">Level2</Button>
        <Button Margin="5">Level3</Button>
        <Button Margin="5">Level4</Button>
        <Button Margin="5">Level5</Button>
      </StackPanel>
    </GroupBox>
    <GroupBox Header="Optionen" DockPanel.Dock="Bottom">
      <StackPanel>
        <Button Margin="5">Speichern</Button>
        <Button Margin="5">Laden</Button>
        <Button Margin="5">Beenden</Button>
      </StackPanel>
    </GroupBox>
  </DockPanel>
</Grid>
<Grid>
  <Grid.ColumnDefinitions>
    <ColumnDefinition />
    <ColumnDefinition />
  </Grid.ColumnDefinitions>
  <Grid.RowDefinitions>
    <RowDefinition />
    <RowDefinition />
  </Grid.RowDefinitions>

  <DockPanel Margin="5">
    <Button DockPanel.Dock="Top">Oben</Button>
    <Button DockPanel.Dock="Bottom">Unten</Button>
    <Button DockPanel.Dock="Left">Links</Button>
    <Button DockPanel.Dock="Right">Rechts</Button>
    <TextBlock HorizontalAlignment="Center"
      VerticalAlignment="Center">Inhalt in der Mitte</TextBlock>
  </DockPanel>

  <Canvas Grid.Column="1">
    <Button Canvas.Left="10" Canvas.Top="10">Hallo</Button>
    <Button Canvas.Right="10" Canvas.Bottom="10">Welt</Button>
  </Canvas>

  <StackPanel Grid.Row="1" Grid.ColumnSpan="2" Orientation="Horizontal"
HorizontalAlignment="Center">
    <Button Height="100">100</Button>
    <Button Height="90">90</Button>
    <Button Height="80">80</Button>
    <Button Height="70">70</Button>
    <Button Height="60">60</Button>
    <Button Height="50">50</Button>
    <Button Height="40">40</Button>
    <Button Height="30">30</Button>
    <Button Height="20">20</Button>
    <Button Height="10">10</Button>
    <Button Height="10">10</Button>
    <Button Height="20">20</Button>
    <Button Height="30">30</Button>
    <Button Height="40">40</Button>
    <Button Height="50">50</Button>
    <Button Height="60">60</Button>
    <Button Height="70">70</Button>
    <Button Height="80">80</Button>
    <Button Height="90">90</Button>
    <Button Height="100">100</Button>
  </StackPanel>
</Grid>
</DockPanel>

```