

C# Essentials

Besturingselementen en layouts

Lector: Tom Quareme

Besturingselementen

- Window
- Label
- TextBlock
- TextBox
- Button
- Image
- CheckBox (selectievakje)
- RadioButton (keuzerondje)
- ListBox (keuzelijst)
- ComboBox
- Slider/ScrollBar
- Menu
- Visibility
- Kleuren
- Brushes

Window

- Open de XAML file van je Window (bijvoorbeeld MainWindow.xaml)
- Hierin kan je de Title property aanpassen.

```
<Window x:Class="BesturingsElementenDemo.MainWindow"
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
        xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
        xmlns:local="clr-namespace:BesturingsElementenDemo"
        mc:Ignorable="d"
        Title="Naam van het venster" Height="450" Width="800">
    <StackPanel>...
```

Window

Eigenschappen (properties)	
Name	naam van window
Title	tekst in titelbalk
Width	breedte van window in pixels
Height	hoogte van window in pixels
BackColor	achtergrondkleur van window
FontFamily	lettertype voor alle objecten in window
WindowState	geeft venstergrootte (Maximized, Minimized of Normal)
Methods	
Close()	sluit het window
Events	
Loaded()	treedt op wanneer het window geladen wordt

Label

- Label om tekst weer te geven.
- Een label kan:
 - een rand maken, kan je disabelen, via content een Image bevatten,
 - je kan er sneltoetsen aan toekennen,...

```
<Label Content="Personeelslid" HorizontalAlignment="Left" VerticalAlignment="Top" Margin="60,51,0,0"  
HorizontalContentAlignment="Right"/>
```

Eigenschappen (properties)	
Name	naam van label
Content	weergave van tekst
Background	achtergrondkleur
Foreground	voorggrondkleur
FontFamily	lettertype en puntgrootte
HorizontalContentAlignment	plaats van de tekst in label (links, rechts,...)
Visibility	zichtbaarheid van label (Hidden/Visible)

Label

```
<!--Klik op ALT en N en de focus gaat naar txtName-->  
<Label Content="_Naam:" Target="{Binding ElementName=TxtNaam}" />  
<TextBox x:Name="TxtNaam" />  
<!--Klik op ALT en M en de focus gaat naar txtMail-->  
<Label Content="_Mail:" Target="{Binding  
ElementName=TxtMail}" />  
<TextBox x:Name="TxtMail" />
```



TextBlock

- Net zoals Label om tekst weer te geven. Maar dat is het dan ook.
⇒ geen afbeeldingen of andere content...
- Is dus minder uitgebreid dan Label, maar wel veel lichter.

TextBox

```
<TextBox x:Name="TxtResultaat" HorizontalAlignment="Left" Height="200" Margin="65,190,0,0"
TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="395" IsEnabled="False"
Background="#FFFAE5E5" FontFamily="Consolas"/>
```

- Een tekstvak waarin je als gebruiker tekst kan ingeven.

Eigenschappen (properties)	
Name	naam van tekstvak
Text	de tekst
Background	achtergrondkleur
Foreground	voorgndkleur
FontFamily/FontWeight	lettertype en puntgrootte, letterdikte
IsEnabled	is de TextBox beschikbaar? (true/false)
IsReadOnly	kan de inhoud enkel gelezen worden? (true/false)
MaxLength	maximaal aantal karakters in TextBox
MaxLines	maximaal aantal lijnen in TextBox
VerticalScrollbarVisibility	schuifbalk (verticaal)
HorizontalScrollbarVisibility	schuifbalk (horizontaal)
TextWrapping	tekstterugloop instellen
Visibility	zichtbaarheid (Hidden/Visible)

TextBox

Methods	
Focus()	tekstvak krijgt de focus
Clear()	inhoud van tekst wissen
Events	
TextChanged()	treedt op bij veranderen van de tekst
GotFocus()	wanneer de TextBox de focus krijgt
LostFocus()	wanneer de TextBox de focus verliest

Bereken

Button

```
<Button x:Name="ButtonBerekenen" Content="_Berekenen" HorizontalAlignment="Left"
Margin="355,35,0,0" VerticalAlignment="Top" Width="105" Height="35" IsDefault="True"
Click="ButtonBerekenen_Click"/>
```

- Een knop waarop je kan klikken.

Eigenschappen (properties)	
Name	naam van tekstvak
Content	de tekst (vb/ “_Bereken” (druk op Alt+B om Click event te activeren)
Width	breedte in pixels
Height	hoogte in pixels
Background	achtergrondkleur
Foreground	voorggrondkleur
FontFamily/FontWeight	lettertype en puntgrootte, letterdikte (Thin, Bold, SemiBold,...)
FontStyle	Italic (schuine tekst),...
IsDefault	is dit de standaard Button? (true/false)
IsEnabled	is de Button beschikbaar? (true/false)
Visibility	zichtbaarheid van label (Hidden/Visible)

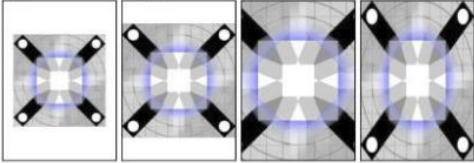
Button

Methods	
Focus()	Button krijgt de focus
Events	
Click()	wanneer je op de Button klikt
GotFocus()	wanneer de Button de focus krijgt
LostFocus()	wanneer de Button de focus verliest
MouseEnter()	wanneer je met de muis over de Button hovert
MouseLeave()	wanneer je de muis weg van de Button hovert

Image

- Afbeeldingsvak

```
<Image x:Name="ImgDuimen" Source="duimen0mhoog.jpg" Stretch="Fill"/>
```

Eigenschappen (properties)	
Name	naam van afbeeldingsvak
Width	breedte in pixels
Height	hoogte in pixels
Source	pad naar het bestand met de afbeelding
IsEnabled	is de Image beschikbaar? (true/false)
Visibility	zichtbaarheid van Image (Hidden/Visible)
Stretch	formaat van de afbeelding (None, Uniform, UniformToFill, Fill) 

Image

- **Afbeelding toekennen aan een image in XAML:**

```
<Image HorizontalAlignment="Left" Height="61" Margin="837,337,0,0"  
VerticalAlignment="Top" Width="54" Source="Telaat.png"/>
```

- **Afbeelding toekennen aan een Button in XAML:**

Als je over Button hoovert, verdwijnt je afbeelding ⇒ content wordt getoond
Voorkom je door in designer Background en Content van Button dezelfde afbeelding te geven!

```
<Button.Background>  
    <ImageBrush ImageSource="Telaat.png" Stretch="UniformToFill"/>  
</Button.Background>
```

```
<Button.Content>  
    <Image Source="Telaat.png"/>  
</Button.Content>
```

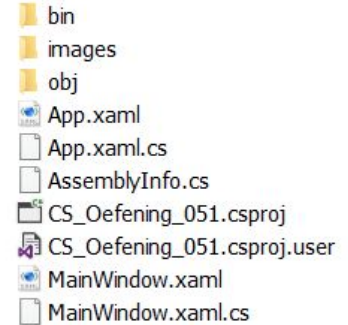
Image

- **Afbeelding aan een image programmeren via C# code:**
 - Maak een mapje “images” aan ter hoogte van bin en obj.
 - Stel je hebt .NET Core project: 3x “..” gebruiken
 - Stel je hebt .NET Framework project: 2x “..” gebruiken
 - @ teken vóór “”: om te voorkomen dat je \\ telkens typt.

```
string bestandsnaam = @"..\..\..\images\Telaat.png";  
ImgAfb.Source = new BitmapImage(  
    new Uri(AppDomain.CurrentDomain.BaseDirectory + bestandsnaam, UriKind.RelativeOrAbsolute));
```

- **Nieuwe afbeelding op een Button programmeren via C# code:**

```
Image afb = new Image();  
string bestandsnaam = @"..\..\..\images\Telaat.png";  
afb.Source = new BitmapImage(new Uri(AppDomain.CurrentDomain.BaseDirectory + bestandsnaam,  
    UriKind.RelativeOrAbsolute));  
afb.Stretch = Stretch.Fill;  
BtnBereken.Content = afb;
```



CheckBox (selectievakje)

- Met de CheckBox kan je meerdere verschillende keuzes aanvinken:

```
<CheckBox x:Name="ChkWerkzoekend" Content="Werkzoekend" HorizontalAlignment="Left" Height="26"
Margin="55,150,0,0" VerticalAlignment="Top" Width="173" Click="ChkWerkzoekend_Click"/>
```

Eigenschappen (properties)	
IsChecked	is de CheckBox aangevinkt? (true/false)
FlowDirection	LeftToRight zet het vakje links, RightToLeft zet het vakje rechts
Events	
Checked()	treedt op wanneer IsChecked true wordt
Click()	wanneer je op de CheckBox klikt
Unchecked()	treedt op wanneer IsChecked false wordt

CheckBox (selectievakje)

- CheckBox heeft een property `IsChecked` om te controleren of de checkbox is aangevinkt.
- `IsChecked` geeft een `bool` terug, dus moet je controleren op `== true`.

```
private void Button_Click(object sender, RoutedEventArgs e)
{
    if (CbzMijnCheckbox.IsChecked == true)
    {
        // Code...
    }
}
```


RadioButton (keuzerondje)

- Alle RadioButtons vormen automatisch 1 groep in het Window.
 - Je kan ze in een apart paneel steken om apart te groeperen.
 - (Panels: Grid, Canvas, StackPanel, DockPanel,...)
- Maar 1 keuze tegelijk wordt geaccepteerd binnen een groep.

```
<RadioButton x:Name="RadJongen" Content="Jongen" IsChecked="True" Checked="RadJongen_Checked"
Width="243" Canvas.Left="10" Canvas.Top="33" HorizontalAlignment="Center" VerticalAlignment="Center"
VerticalContentAlignment="Center"/>
```

Eigenschappen (properties)	
IsChecked	is de RadioButton geselecteerd? (true/false)
FlowDirection	LeftToRight zet het rondje links, RightToLeft zet het rondje rechts
Events	
Checked()	treedt op wanneer IsChecked verandert
Click()	wanneer je op de RadioButton klikt

ListBox (keuzelijst)

- Een lijst van gegevens waaruit je er 1 (of meerdere) kan selecteren).

```
<ListBox x:Name="LbxNamen" HorizontalAlignment="Left" Height="256" Margin="51,36,0,0"
VerticalAlignment="Top" Width="165" FontSize="16">
    <ListBoxItem Content="Inge" />
    <ListBoxItem Content="Kirsten"/>
    <ListBoxItem Content="Paul" />
    <ListBoxItem Content="Patricia"/>
</ListBox>
```

ListBox (keuzelijst)

Eigenschappen (properties)	
Items	items van de ListBox lezen of aanpassen
SelectedItems	geselecteerde items van de ListBox lezen of aanpassen
SelectionMode	<ul style="list-style-type: none">• Single: 1 item selecteren• Multi: meerdere items per 1 selecteren (+Ctrl)• Extended: meerdere items per groep selecteren (+Shift)
Events	
SelectionChanged()	treedt op wanneer je een ander item selecteert
Methods	
Items.Add()	nieuw ListBoxItem achteraan de ListBox toevoegen
Items.Clear()	alle items uit ListBox verwijderen
Items.Insert()	nieuw ListBoxItem toevoegen op een bepaalde positie
Items.Remove()	verwijder een zeker item uit de ListBox
Items.RemoveAt()	verwijder item op gegeven index (gooit een Exception als index niet bestaat)
Items.SortDescriptions.Add()	ListBox sorteren

ListBox (keuzelijst)

ListBox alfabetisch sorteren (Ascending of Descending voor aflopend)

```
LbxNamen.Items.SortDescriptions.Add(  
    new System.ComponentModel.SortDescription(  
        "Content",  
        System.ComponentModel.ListSortDirection.Ascending));
```

Alle items uit ListBox verwijderen

```
LbxNamen.Items.Clear();
```

Nieuw ListBoxItem achteraan de ListBox toevoegen

```
ListBoxItem item = new ListBoxItem();  
item.Content = "Chuck Norris";  
LbxNamen.Items.Add(item);
```

Nieuw ListBoxItem toevoegen aan de ListBox op een bepaalde positie

```
ListBoxItem item = new ListBoxItem();  
item.Content = "Chuck Norris";  
LbxNamen.Items.Insert(0, item);
```

ListBox (keuzelijst)

Verwijder geselecteerd item van ListBox

```
// index == -1 : niets geselecteerd
if (LbxNamen.SelectedIndex == -1)
    return;
LbxNamen.Items.RemoveAt(LbxNamen.SelectedIndex); // gooit Exception als niets is geselecteerd
```

Verwijder geselecteerd item van ListBox (alternatief)

```
LbxNamen.Items.Remove(LbxNamen.SelectedItem); // gooit geen Exception
```

Vervang geselecteerd item van ListBox

```
// index == -1 : niets geselecteerd
if (LbxNamen.SelectedIndex == -1)
    return;
((ListBoxItem)LbxNamen.SelectedItem).Content = "Chuck Norris";
```

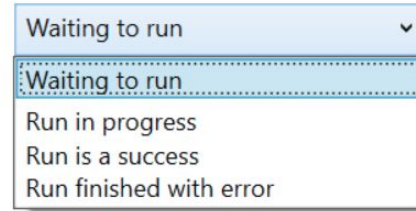
ListBox (keuzelijst)

Zoek item in de ListBox (en selecteer dit nadien automatisch)

```
int index = -1; // zet op -1 omdat we in het begin nog niks gevonden hebben
int i = 0; // huidige positie in ListBox. Start vanaf 0
foreach (ListBoxItem item in LbxNamen.Items)
{
    if (item.Content.Equals("Chuck Norris"))
    {
        // Gevonden! Dus stoppen met zoeken => break om uit de lus te ontsnappen
        index = i;
        break;
    }
    i++;
}

// Gevonden item in ListBox selecteren.
if (index != -1)
    LbxNamen.SelectedIndex = index;
```

ComboBox



- Vervolgkeuzelijst.

```
<ComboBox x:Name="CboOpleiding" SelectionChanged="CboOpleiding_SelectionChanged">
    <ComboBoxItem Content="Programmeren"/>
    <ComboBoxItem Content="Netwerkbeheer"/>
    <ComboBoxItem Content="Internet of Things"/>
    <ComboBoxItem Content="Digitale vormgever"/>
    <ComboBoxItem Content="Drone opleiding"/>
</ComboBox>
```

Eigenschappen (properties)	
Items	items van de lijst lezen of aanpassen
MaxDropDownHeight	hoogte instellen waarbij een aantal items zichtbaar worden
IsEditable	invoer is mogelijk en je kan de lijst uitschuiven (klik of Alt + pijl omlaag)
IsReadOnly	je kan de lijst niet aanpassen
IsTextSearchEnabled	tekst automatisch aanvullen (IsReadOnly moet false zijn!)

ComboBox

- **Opvragen van geselecteerde items**

```
((ComboBoxItem) CboOpleiding.SelectedItem).Content.ToString(); // Programmeren  
int index = CboOpleiding.SelectedIndex; // 0
```

- **Niets selecteren**

```
CboOpleiding.SelectedIndex = -1;
```

- **Items toevoegen**

```
ComboBoxItem item = new ComboBoxItem();  
item.Content = "Game designer";  
CboOpleiding.Items.Add(item);
```

- **Items verwijderen**

```
CboOpleiding.Items.RemoveAt(CboOpleiding.SelectedIndex);
```


Slider/ScrollBar

```
<Slider x:Name="Schuifregelaar" HorizontalAlignment="Left" Height="35" Margin="85,115,0,0"
VerticalAlignment="Top" Width="395" TickPlacement="BottomRight" AutoToolTipPlacement="TopLeft"
Maximum="100000" TickFrequency="10000" ValueChanged="Slider_ValueChanged" IsSnapToTickEnabled="True"
SmallChange="100" LargeChange="5000"/>
```

- Schuifregelaar/schuifbalk

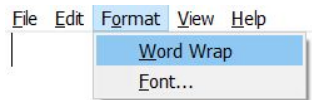
Eigenschappen	
LargeChange	grote sprongen door klikken op Slider/ScrollBar
SmallChange	kleine sprongen door klikken op Slider/ScrollBar
Value	waarde tussen Minimum en Maximum
Minimum	kleinste waarde
Maximum	grootste waarde
Orientation	Horizontal / Vertical
TickFrequency	per eenheid aantal streepjes op balk
TickPlacement	plaats van streepjes
AutoToolTipPlacement	plaats van automatische tooltip (Value tonen)
IsSnapToTickEnabled	

Events	
ValueChanged()	treedt op bij schuiven

Menu

- **Menu:**

- Komt bovenaan je programma onder de titelbalk.
- Bestaat uit menu items die zelf ook weer bestaan uit menu items.



```
<Menu x:Name="Hoofdmenu" HorizontalAlignment="Left" VerticalAlignment="Top">
  <MenuItem Header="Bestand">
    <MenuItem x:Name="MnuAfsluiten" Header="Afsluiten" Click="MnuAfsluiten_Click"/>
  </MenuItem>

  <MenuItem x:Name="MnuAard" Header="Aard oefening">
    <MenuItem x:Name="MnuOptellen" Header="Optellen" Click="MnuOptellen_Click"/>
    <MenuItem x:Name="MnuAftrekken" Header="Aftrekken" Click="MnuAftrekken_Click"/>
    <MenuItem x:Name="MnuVermenigvuldiging" Header="Vermenigvuldiging"
      Click="MnuVermenigvuldiging_Click"/>
    <MenuItem x:Name="MnuDeling" Header="Deling" Click="MnuDeling_Click"/>
  </MenuItem>
</Menu>
```

Menu

Eigenschappen	
IsCheckable	vinkje blijft staan todat je er weer op klikt
ToolTip	tooltip bij menu-item
IsEnabled	actief of inactief (grijs) zetten (true/false)
VerticalAlignment	verticale positie van menu in windows/grid
HorizontalAlignment	horizontale positie van menu in windows/grid

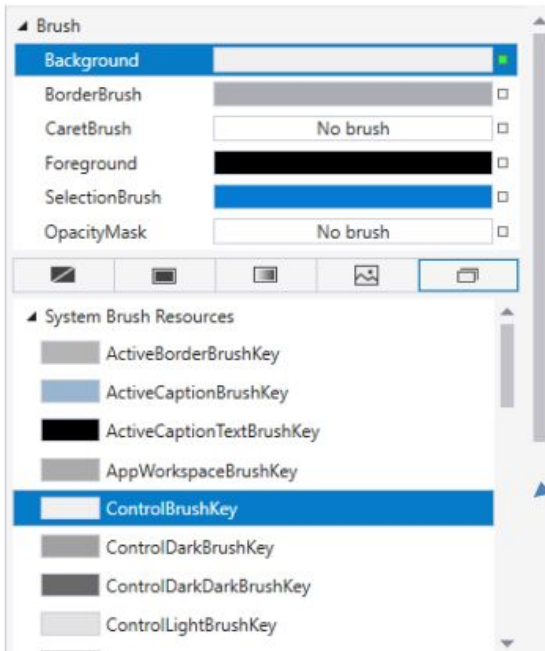
Visibility

- De zichtbaarheid aanpassen via:
 - Visibility.Visible
 - Visibility.Hidden
 - Visibility.Collapsed

```
private void Button_Click(object sender, RoutedEventArgs e)
{
    if (CbzMijnCheckbox.IsChecked == true)
    {
        BtnMijnButton.Visibility = Visibility.Visible;
    }
}
```

Kleuren

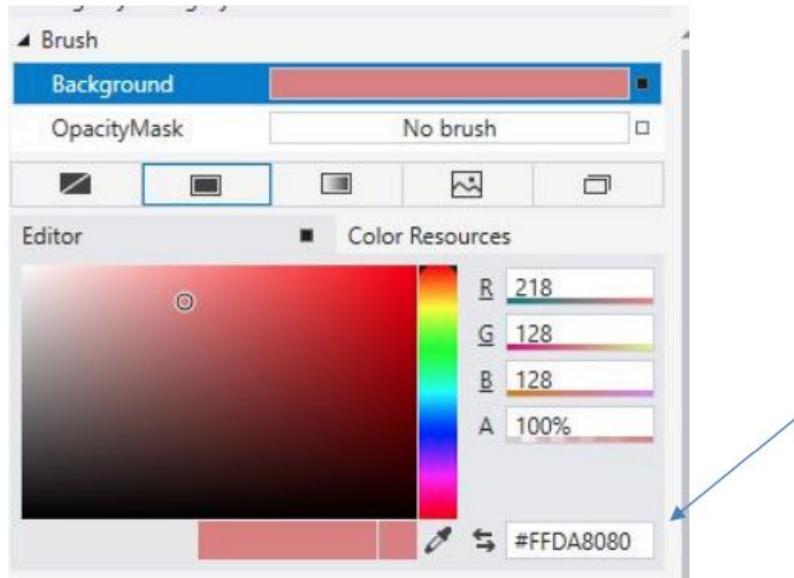
- In de XAML:



```
< ... Background="{DynamicResource {x:Static SystemColors.ControlBrushKey}}" />
```

Kleuren

- In de XAML:



< ... Background="#FFDA8080">

Brushes

- Kleuren in code veranderen via Brushes.
 - Brushes.KleurNaam

```
private void Button_Click(object sender, RoutedEventArgs e)
{
    BtnBereken.Background = Brushes.Red;
    BtnBereken.Content = "gevonden";
}
```

Brushes

- Zelf hexadecimale kleurencodes (ARGB) gebruiken ipv. ingebouwde namen.

```
using System.Windows.Media;  
// Kleurenallet met hexadecimale waarde gebruiken.  
var bc = new BrushConverter();  
TxtSpeler.Background = (Brush)bc.ConvertFrom("#FFE8E6E6");
```


Brushes

 AliceBlue	#FFF0F8FF	 DarkTurquoise	#FF00CED1	 LightSeaGreen	#FF20B2AA	 PapayaWhip	#FFFFEFD5
 AntiqueWhite	#FFFAEBD7	 DarkViolet	#FF9400D3	 LightSkyBlue	#FF87CEFA	 PeachPuff	#FFFFDAB9
 Aqua	#FF00FFFF	 DeepPink	#FFFF1493	 LightSlateGray	#FF778899	 Peru	#FFCD853F
 Aquamarine	#FF7FFFD4	 DeepSkyBlue	#FF00BFFF	 LightSteelBlue	#FFB0C4DE	 Pink	#FFFFC0CB
 Azure	#FFF0FFFF	 DimGray	#FF696969	 LightYellow	#FFFFFFE0	 Plum	#FFDDA0DD
 Beige	#FFF5F5DC	 DodgerBlue	#FF1E90FF	 Lime	#FF00FF00	 PowderBlue	#FFB0E0E6
 Bisque	#FFFFE4C4	 Firebrick	#FFB22222	 LimeGreen	#FF32CD32	 Purple	#FF800080
 Black	#FF000000	 FloralWhite	#FFFFFFAF0	 Linen	#FFFAF0E6	 Red	#FFFF0000
 BlanchedAlmond	#FFFFEBCD	 ForestGreen	#FF228B22	 Magenta	#FFFF00FF	 RosyBrown	#FFBC8F8F
 Blue	#FF0000FF	 Fuchsia	#FFFF00FF	 Maroon	#FF800000	 RoyalBlue	#FF4169E1
 BlueViolet	#FF8A2BE2	 Gainsboro	#FFDCDCDC	 MediumAquamarine	#FF66CDAA	 SaddleBrown	#FF8B4513
 Brown	#FFA52A2A	 GhostWhite	#FFF8F8FF	 MediumBlue	#FF0000CD	 Salmon	#FFFA8072
 BurlyWood	#FFDEB887	 Gold	#FFFFD700	 MediumOrchid	#FFBA55D3	 SandyBrown	#FFFA4A60
 CadetBlue	#FF5F9EA0	 Goldenrod	#FFDAA520	 MediumPurple	#FF9370DB	 SeaGreen	#FF2E8B57
 Chartreuse	#FF7FFF00	 Gray	#FF808080	 MediumSeaGreen	#FF3CB371	 SeaShell	#FFFFFF5EE
 Chocolate	#FFD2691E	 Green	#FF008000	 MediumSlateBlue	#FF7B68EE	 Sienna	#FFA0522D
 Coral	#FFFF7F50	 GreenYellow	#FFADFF2F	 MediumSpringGreen	#FF00FA9A	 Silver	#FFC0C0C0
 CornflowerBlue	#FF6495ED	 Honeydew	#FFF0FFF0	 MediumTurquoise	#FF48D1CC	 SkyBlue	#FF87CEEB

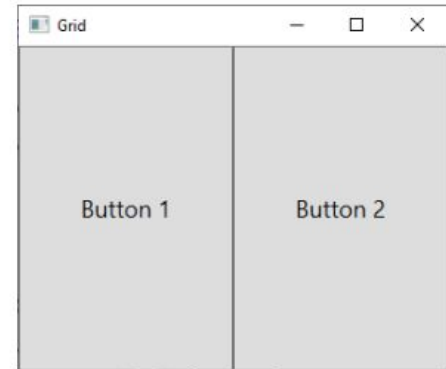
Panelen

- Zijn containers voor WPF besturingselementen.
- Bepalen layout van window.
- Soorten panelen:
 - Grid
 - Canvas
 - StackPanel
 - WrapPanel
 - DockPanel
 - Viewbox

Grid (raster)

- Een raster kan meerdere rijen en kolommen bevatten.
- Voor de eerste component (vb/ Button) moet je niet vertellen welke kolom of rij deze is. Maar het mag wel.
 - Grid.Column="0" Grid.Row="0"

```
<Grid>
  <Grid.ColumnDefinitions>
    <ColumnDefinition Width="*" /> (nemen beiden evenveel
    <ColumnDefinition Width="*" /> kolommen of breedte in)
  </Grid.ColumnDefinitions>
  <Button Content="Button 1" />
  <Button Grid.Column="1" Content="Button 2" />
</Grid>
```



Grid (raster)

<Grid>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="2*" /> (neemt 2x zoveel plaats in qua kolommen)

<ColumnDefinition Width="1*" />

<ColumnDefinition Width="1*" />

</Grid.ColumnDefinitions>

<Grid.RowDefinitions>

<RowDefinition Height="2*" /> (neemt 2x zoveel plaats in rijen)

<RowDefinition Height="1*" />

<RowDefinition Height="1*" />

</Grid.RowDefinitions>

<Button>Button 1</Button>

<Button Grid.Column="1">Button 2</Button>

<Button Grid.Column="2">Button 3</Button>

<Button Grid.Row="1">Button 4</Button>

<Button Grid.Column="1" Grid.Row="1">Button 5</Button>

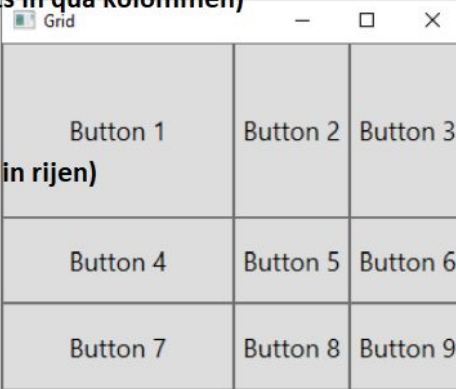
<Button Grid.Column="2" Grid.Row="1">Button 6</Button>

<Button Grid.Row="2">Button 7</Button>

<Button Grid.Column="1" Grid.Row="2">Button 8</Button>

<Button Grid.Column="2" Grid.Row="2">Button 9</Button>

</Grid>



Button 1	Button 2	Button 3
Button 4	Button 5	Button 6
Button 7	Button 8	Button 9

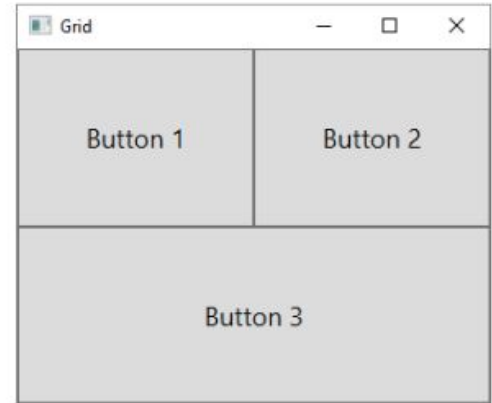
Grid (raster)



```
<Grid>
  <Grid.ColumnDefinitions>
    <ColumnDefinition Width="1*" />
    <ColumnDefinition Width="Auto" /> <!--Breedte past zich aan de content van button aan-->
    <ColumnDefinition Width="100" />
  </Grid.ColumnDefinitions>
  <Button>Button 1</Button>
  <Button Grid.Column="1">PXL Digital in Hasselt</Button>
  <Button Grid.Column="2">Button 3</Button>
</Grid>
```

Grid (raster)

```
<Grid>
  <Grid.ColumnDefinitions>
    <ColumnDefinition Width="1*" />
    <ColumnDefinition Width="1*" />
  </Grid.ColumnDefinitions>
  <Grid.RowDefinitions>
    <RowDefinition Height="*" />
    <RowDefinition Height="*" />
  </Grid.RowDefinitions>
  <Button>Button 1</Button>
  <Button Grid.Column="1">Button 2</Button>
  <Button Grid.Row="1" Grid.ColumnSpan="2">Button 3</Button>
  <!--Spanwijdte over 2 kolommen-->
</Grid>
```



Canvas

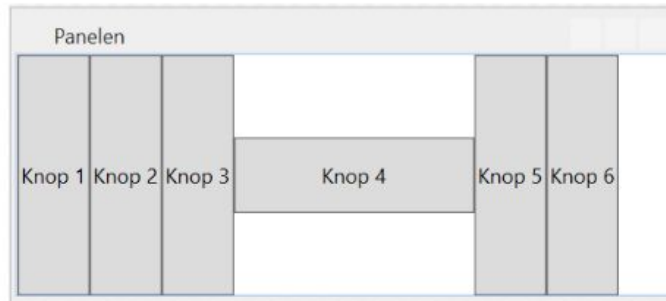
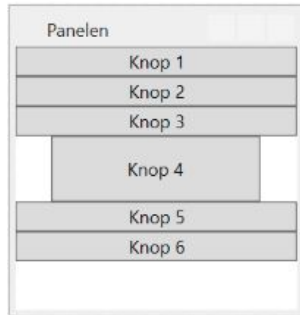
- Eenvoudigste paneel.
- Doet niets, groepeer gewoon componenten die er in staan.
 - Relatieve positie t.o.v. canvas via Canvas.Top, Canvas.Left,...
⇒ als je canvas verplaatst, verplaats je ook elementen die in canvas

~~zitten~~

```
<Canvas x:Name="CanTotaal" HorizontalAlignment="Left" Height="329" Margin="59,105,0,0"
VerticalAlignment="Top" Width="387" Background="{DynamicResource {x:Static
SystemColors.ControlDarkBrushKey}}">
  <Label Content="Totaal 2" HorizontalAlignment="Left" Height="35" VerticalAlignment="Top"
    Width="67" HorizontalContentAlignment="Right" Canvas.Left="12" Canvas.Top="18"/>
  <TextBox HorizontalAlignment="Left" Height="35" TextWrapping="Wrap" VerticalAlignment="Top"
    Width="57" Canvas.Left="95" Canvas.Top="18"/>
</Canvas>
```

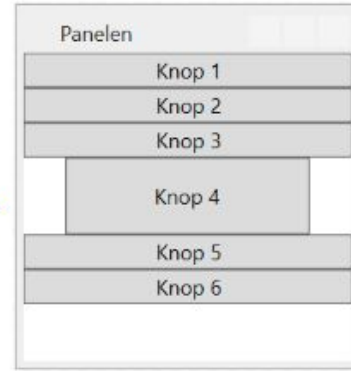

StackPanel

- Stapelt besturingselementen verticaal of horizontaal:
 - Orientation="Vertical" (standard)
 - Elementen krijgen dezelfde breedte (volgens breedste element).
 - Orientation="Horizontal"
 - Elementen krijgen dezelfde hoogte (volgens hoogste element).
- Als je wil dat besturingselementen volledige breedte of hoogte innemen.

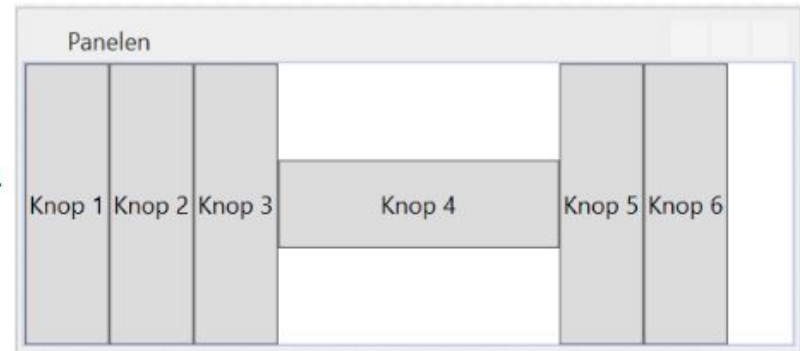


StackPanel

```
<StackPanel>  
  <Button>Knop 1</Button>  
  <Button>Knop 2</Button>  
  <Button>Knop 3</Button>  
  <Button Width="140" Height="44">Knop 4</Button>  
  <Button>Knop 5</Button>  
  <Button>Knop 6</Button>  
</StackPanel>
```



```
<StackPanel Orientation="Horizontal">  
  <Button>Knop 1</Button>  
  <Button>Knop 2</Button>  
  <Button>Knop 3</Button>  
  <Button Width="140" Height="44">Knop 4  
  </Button>  
  <Button>Knop 5</Button>  
  <Button>Knop 6</Button>  
</StackPanel>
```



WrapPanel

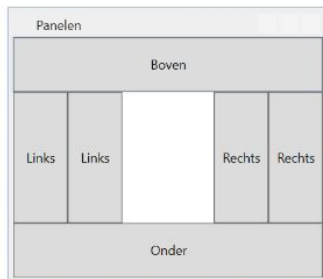
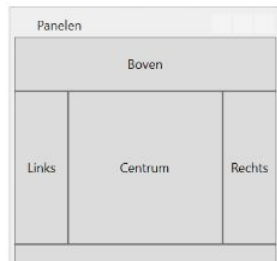
- Plaatst inwendige elementen naast elkaar of onder elkaar.
 - Orientation="Horizontal" (standaard)
 - Wanneer rij vol \Rightarrow nieuwe rij starten
 - Orientation="Vertical"
 - Wanneer kolom vol \Rightarrow nieuwe kolom starten

DockPanel

- Element boven, onder, links en rechts van elkaar plaatsen.
- Het eerste element komt standaard links, tenzij anders aangegeven.
- Het laatste element komt volledig gevuld in het midden.
 - Zet LastChildFill op false als je dit niet wil.

```
<DockPanel>
  <Button DockPanel.Dock="Top" Height="50" Content="Boven"/>
  <Button DockPanel.Dock="Bottom" Height="50" Content="Onder"/>
  <Button DockPanel.Dock="Left" Width="50" Content="Links"/>
  <Button DockPanel.Dock="Right" Width="50" Content="Rechts"/>
  <Button Content="Centrum"/>
</DockPanel>

<DockPanel LastChildFill="false">
  <Button DockPanel.Dock="Top" Height="50" Content="Boven"/>
  <Button DockPanel.Dock="Bottom" Height="50" Content="Onder"/>
  <Button DockPanel.Dock="Left" Width="50" Content="Links"/>
  <Button DockPanel.Dock="Right" Width="50" Content="Rechts"/>
  <Button DockPanel.Dock="Right" Width="50" Content="Rechts"/>
  <Button Width="50" Content="Links"/>
</DockPanel>
```



Viewbox

- Past de inwendige elementen hun grootte aan naargelang wat er beschikbaar is.
- Formaat van inhoud aanpassen en schalen (transformatie).
- Vaak gebruikt voor 2D-afbeeldingen.

Voorbeeld 1: Viewbox heeft inwendig een heel Grid

```
<Viewbox>
<Grid >
  <Grid.ColumnDefinitions>
    <ColumnDefinition Width="Auto" />
    <ColumnDefinition Width="*" />
    <ColumnDefinition Width="20" />
  </Grid.ColumnDefinitions>
  <Grid.RowDefinitions>
    <RowDefinition Height="Auto" />
    <RowDefinition Height="Auto" />
    <RowDefinition Height="*" />
    <RowDefinition Height="20" />
  </Grid.RowDefinitions>
  <Label>Naam:</Label>
  <TextBox Grid.Column="1" Margin="0,0,0,10" MinWidth="300" />
  <Label Grid.Row="1">E-mail:</Label>
  <TextBox Grid.Row="1" Grid.Column="1" Margin="0,0,0,10" />
  <Label Grid.Row="2">Opmerkingen:</Label>
  <TextBox Grid.Row="2" Grid.Column="1" AcceptsReturn="True" MinHeight="250" />
</Grid>
</Viewbox>
```



Viewbox

Voorbeeld 2: Viewbox voor bepaalde besturingselementen binnen het Grid

```
<Grid>
  <Grid.RowDefinitions>
    <RowDefinition Height="98*" />
    <RowDefinition Height="97*" />
    <RowDefinition Height="138*" />
  </Grid.RowDefinitions>
  <Grid.ColumnDefinitions>
    <ColumnDefinition Width="316*" />
    <ColumnDefinition Width="271*" />
  </Grid.ColumnDefinitions>
  <Button Content="Button 1" HorizontalAlignment="Left" Height="45" Margin="69,39,0,0"
  VerticalAlignment="Top" Width="183" />
  <Button Content="Button 2" HorizontalAlignment="Left" Height="42" Margin="69,17,0,0"
  VerticalAlignment="Top" Width="181" Grid.Row="1" />
  <Viewbox Margin="16,34,25,27" Grid.Column="1" Grid.RowSpan="2">
    <Image Source="DB.jpg" />
  </Viewbox>
  <Viewbox VerticalAlignment="Center" Grid.Row="2" Grid.RowSpan="2">
    <TextBlock Text="Tekst in TextBlock is schaalbaar" />
  </Viewbox>
</Grid>
```

