

# C# Mobile

## Les 3

- Views
- Styling



**DE HOGESCHOOL  
MET HET NETWERK**

Elfde-Liniestraat 24, 3500 Hasselt, [www.pxl.be](http://www.pxl.be)



# Doelstellingen

- De junior-collega kan de verschillende Views beschikbaar in .NET MAUI implementeren
- De junior-collega kan Pages, Layouts en Views combineren tot eenvoudige applicaties
- De junior-collega kan via styling UI-elementen aanpassen in een bepaalde stijl
- De junior-collega kan resources instellen om *styles* te voorzien, zowel lokaal als globaal



# Views

# Views in .NET MAUI

- UI elementen die we in de Pages en Layouts gebruiken om informatie aan de user te tonen en/of interactie te bekomen
- In andere omgevingen soms widgets of controls genoemd
- Via een *renderer* wordt een visueel element getoond in de UI
- Kunnen opgedeeld worden in categorieën
  - Initiate commands: button, searchbar, ...
  - Present data: scrollview, labels, images, webview, contentview, ...
  - Set values: slider, datepicker, ...
  - Edit tekst: editor, entry
  - Indicate activity: activityindicator, progressbar
  - Display collections: CarouselView, **CollectionView**, **ListView**, ...

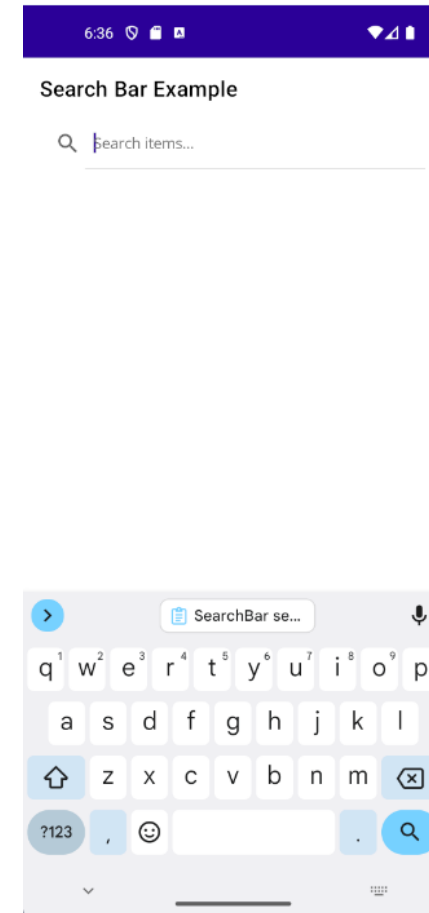
# Views in .NET MAUI

- We hebben al kennis gemaakt met enkele Views:
  - Label
  - Image
  - Slider
  - Button
  - Entry



# Views – Initiate commands: SearchBar

- Een View control dat een zoekbalkje toont
  - Toetsenbord is zo aangepast met vergrootglas
  - Via TextChanged of SearchButtonPressed kan je reageren op event



# Views – Present data

- ContentView:
  - Wordt zelden apart gebruikt  
=> dient als basis voor *user-defined compound views*
  - We komen hier op terug in de les Binding
- Element met *single content*



# Views – ScrollView

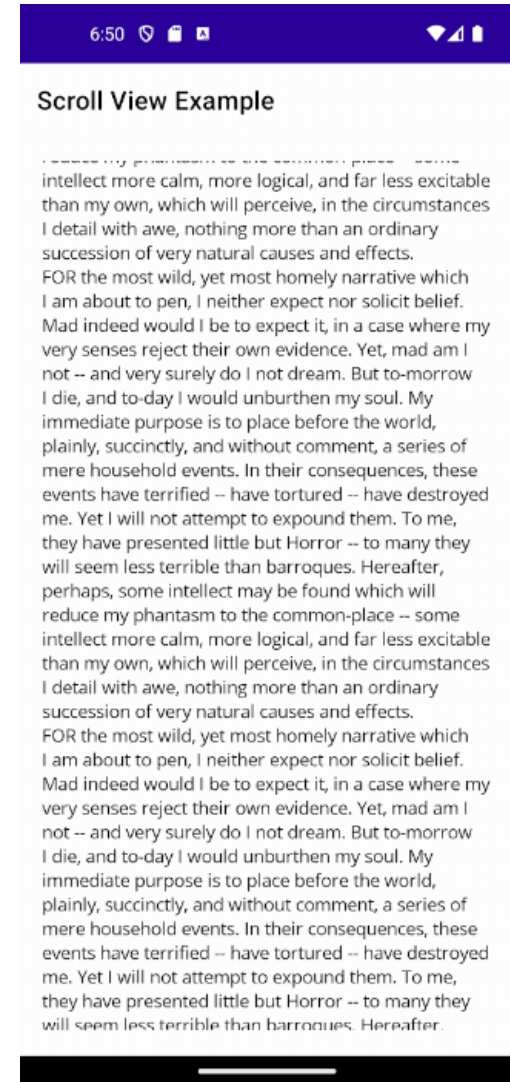
- Zorgt ervoor dat je kan scrollen als zijn content groter is dan de voorziene plaats
- Element met *single content*
- **Opletten:** de documentatie geeft bij verschillende elementen extra waarschuwingen en tips!

## ⚠ Warning

ScrollView objects should not be nested. In addition, ScrollView objects should not be nested with other controls that provide scrolling, such as CollectionView, ListView, and WebView.

## ⊗ Caution

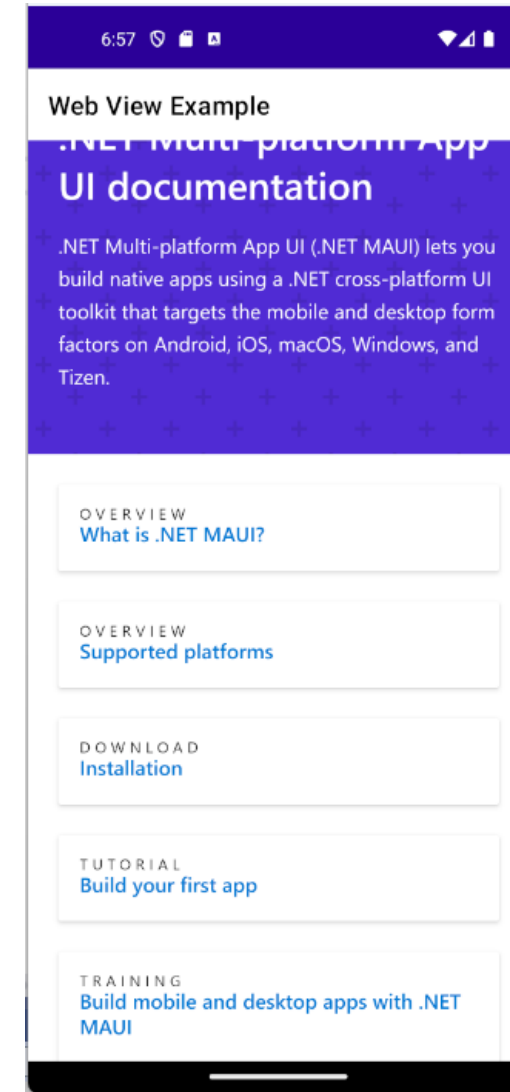
In a vertical ScrollView, avoid setting the `VerticalOptions` property to `Start`, `Center`, or `End`. Doing so tells the ScrollView to be only as tall as it needs to be, which could be zero. While .NET MAUI protects against this eventuality, it's best to avoid code that suggests something you don't want to happen.





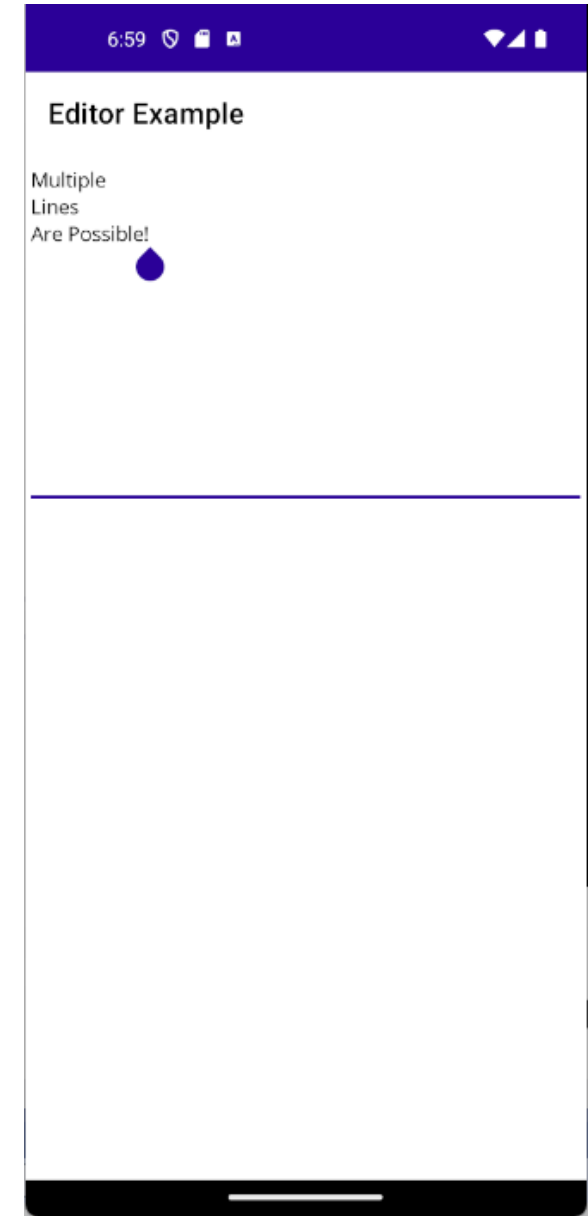
# Views – Webview

- Externe website tonen in de UI
- Eigen stukjes HTML-code tonen



# Views – Edit text: Editor

- Multiline Entry element
  - De gebruiker kan meerdere lijnen input geven
  - Vergelijkbaar met een `<textarea>` uit HTML



# Views – Set values: Switch

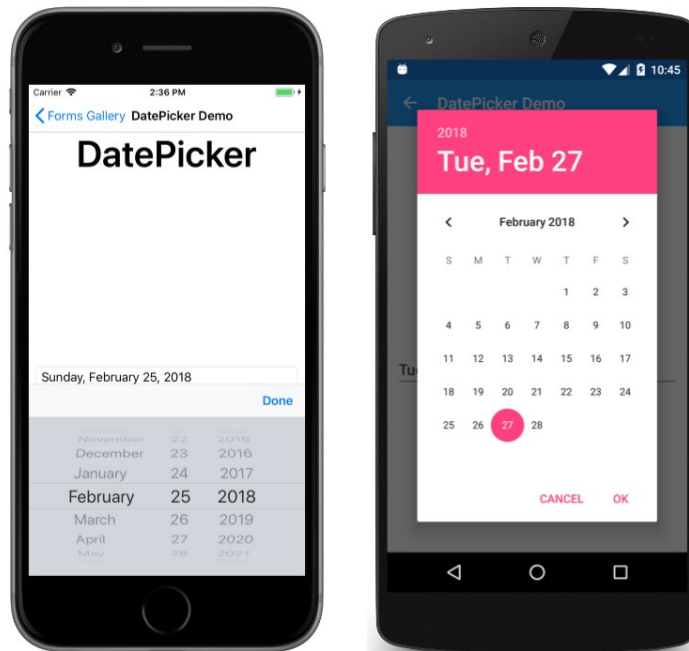
View control om een waarde aan of uit te zetten (Boolean)

- Kleuren kunnen aangepast worden via de *Visual States*



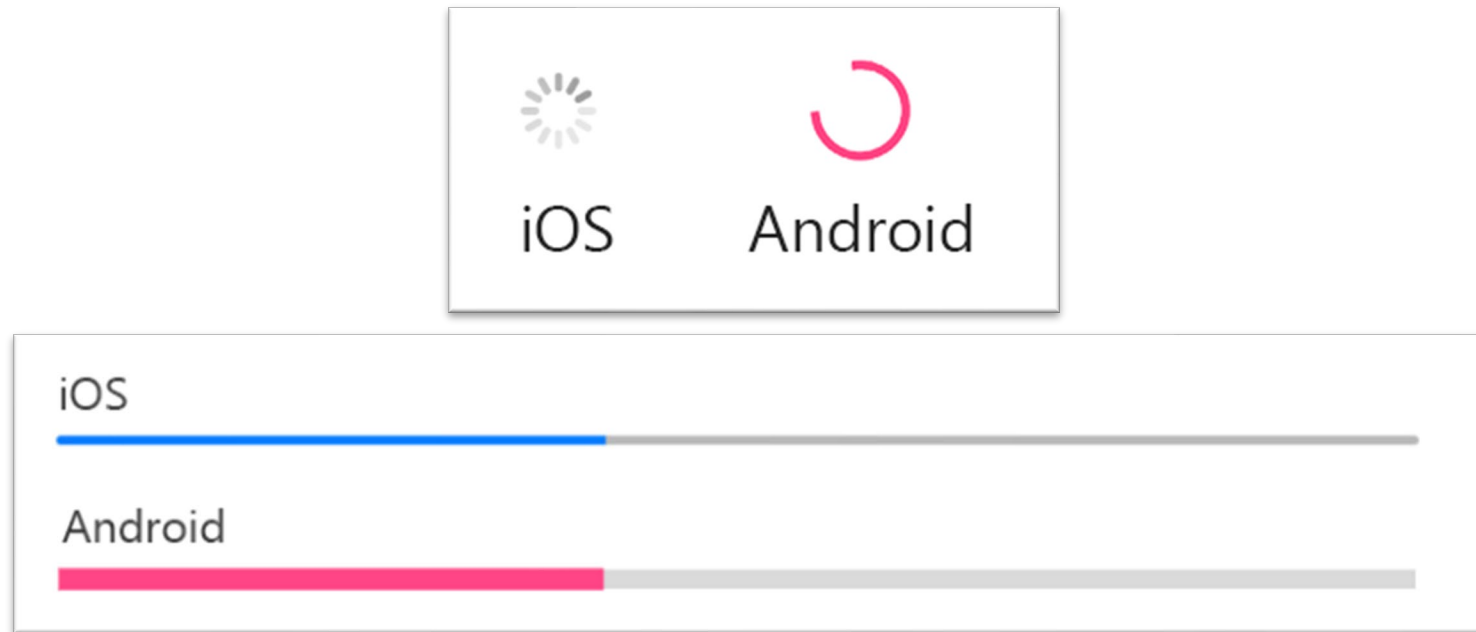
# Views – Set values: DatePicker - TimePicker

- **Element om een datum of tijd te selecteren**
  - In eerste instantie lijkt het op een gewone entry
  - Speciale control verschijnt om een datum of tijd te selecteren (platform-specifiek)



# Views – Indicate activity: ActivityIndicator - Progressbar

- **Element om te tonen dat er 'iets bezig is'**
  - Toont een animatie zodat de gebruiker dit duidelijk ziet
  - Progressbar: de gebruiker ziet de voortgang



# Views – Present data

- Border:
  - Wrapt de child view met een rand (*kleur, schaduw en border-radius*)
  - *Zie Labo*
- Element met *single content*





# Views: Opdracht

- **Maak een applicatie met minstens vier verschillende views uit de vorige slides of de online documentatie**
  - Zoek telkens op hoe je deze moet implementeren!
  - Ga gerust verder op de repo uit het Lab
  - Gebruik een grid layout om ze een plek te geven op de pagina



# .NET MAUI Styling

# Styles

- Tot nu toe: alle stijl apart in elk XAML-element

- Veel copy-paste
- Moeilijk te onderhouden
- Onoverzichtelijk

```
<Label x:Name="ActorName"  
    VerticalOptions="CenterAndExpand"  
    HorizontalOptions="CenterAndExpand"  
    Padding="5,5,5,5"  
    BackgroundColor=■ "Blue"  
    TextColor=□ "White"  
/>
```

**=> We gaan hiervoor Styles gebruiken die we in de  
*Resources* plaatsen**

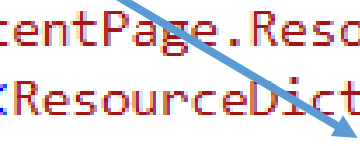
# Intermezzo: Shared Resources

- Elke Layout, Page en de Application kan Resources hebben
- Een *Resource* is een XAML-object dat hergebruikt kan worden
- Bij het opvragen van een Resource zal MAUI zoeken naar de eerste Resource die voldoet
- Resources bijgehouden in een dictionary (key – value)

# Intermezzo: Shared Resources

- Voorbeeld Resource in ContentPage
- Dictionary
  - Key – Value

```
<ContentPage.Resources>  
  <ResourceDictionary>  
    <Color x:Key="ActorAppMainColor">■#0011aa</Color>  
  </ResourceDictionary>  
</ContentPage.Resources>
```

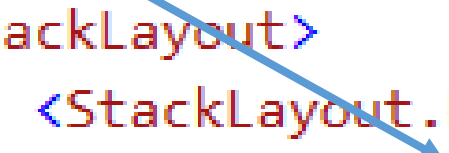


De <ResourceDictionary>-tag is optioneel!

# Intermezzo: Shared Resources

- Voorbeeld Resource in StackLayout
- Dictionary
  - Key – Value

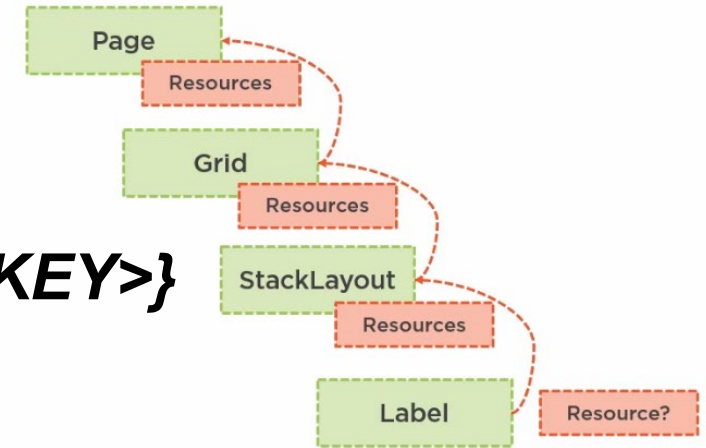
```
<StackLayout>  
  <StackLayout.Resources>  
    <Color x:Key="ActorAppMainColor">■#ff00ff</Color>  
  </StackLayout.Resources>  
</StackLayout>
```






# Intermezzo: Shared Resources

- Zoeken via Markup Extension *{StaticResource <KEY>}*
- “Tree-Walking”



```
<ContentPage.Content>
  <StackLayout>
    <Label x:Name="ActorName"
      VerticalOptions="CenterAndExpand"
      HorizontalOptions="CenterAndExpand"
      Padding="5,5,5,5"
      BackgroundColor=■"Blue"
      TextColor=□"White"
    />
    <Image x:Name="ProfilePicture" WidthRequest="50" HeightRequest="50" A
  />
</StackLayout>
</ContentPage.Content>
```



```
<ContentPage.Content>
  <StackLayout>
    <Label x:Name="ActorName"
      VerticalOptions="CenterAndExpand"
      HorizontalOptions="CenterAndExpand"
      Padding="5,5,5,5"
      BackgroundColor=■{StaticResource ActorAppMainColor}
      TextColor=□"White"
    />
    <Image x:Name="ProfilePicture" WidthRequest="50" HeightRequest="50" Asp
  />
</StackLayout>
</ContentPage.Content>
```

# Styles

- Doel: Het groeperen van Properties van VisualElements
- We plaatsen ze in Shared Resources
- Definiëren in XAML of C#
- Opbouw in XAML:

```
<Style x:Key="StyleName" TargetType="VisualElementToStyle">
```

*Eén of meerdere <Setter>-elementen*

```
</Style>
```

- Dit is een explicit style

# Styles: Voorbeeld

Moet een Key hebben  
want zit in Dictionary!

Moet een TargetType  
hebben:  
Zo weten de Setter's op  
welke Properties ze  
kunnen werken

```
<ContentPage.Resources>
  <ResourceDictionary>
    <Color x:Key="ActorAppMainColor">■#0011aa</Color>
    <Style x:Key="GoBackButton" TargetType="Button">
      <Setter Property="BackgroundColor" Value=■"Red"></Setter>
      <Setter Property="TextColor" Value=□"White"></Setter>
      <Setter Property="HorizontalOptions" Value="Center"></Setter>
      <Setter Property="Padding" Value="20,20,20,20"></Setter>
    </Style>
  </ResourceDictionary>
</ContentPage.Resources>
```

# Styles: Gebruiken

- Toepassen via {StaticResource ...}
- Dit is een **explicit** style

```
<Button Text="Go back"  
        BackgroundColor="Red"  
        TextColor="White"  
        HorizontalOptions="Center"  
        Padding="20,20,20,20"></Button>
```



```
<Button Text="Go back"  
        Style="{StaticResource GoBackButton}"></Button>
```

# Styles: Implicit Styles

- **Implicit styles**
  - GEEN key
  - WEL TargetType
- **Alle elementen van dit type krijgen deze stijl**
  - Zelfde gedrag als een algemene CSS regel in Web Development

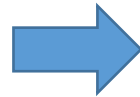
```
<Style TargetType="ActivityIndicator">  
  <Setter Property="Color" Value="White"></Setter>  
</Style>
```

# Styles: Inheritance

- **Stijl laten baseren op andere stijl**
  - Via **BasedOn**-property

```
<Style x:Key="BaseButtonStyle" TargetType="Button">
    <Setter Property="BackgroundColor" Value="Red"></Setter>
    <Setter Property="TextColor" Value="Purple"></Setter>
    <Setter Property="BorderColor" Value="Red"></Setter>
    <Setter Property="BorderWidth" Value="5"></Setter>
    <Setter Property="VerticalOptions" Value="CenterAndExpand"></Setter>
</Style>

<Style x:Key="MainButtonStyle" TargetType="Button">
    <Setter Property="BackgroundColor" Value="{StaticResource ApplicationMainColor}"></Setter>
    <Setter Property="TextColor" Value="Purple"></Setter>
    <Setter Property="BorderColor" Value="Red"></Setter>
    <Setter Property="BorderWidth" Value="5"></Setter>
    <Setter Property="VerticalOptions" Value="CenterAndExpand"></Setter>
</Style>
```



```
<Style x:Key="BaseButtonStyle" TargetType="Button">
    <Setter Property="BackgroundColor" Value="Red"></Setter>
    <Setter Property="TextColor" Value="Purple"></Setter>
    <Setter Property="BorderColor" Value="Red"></Setter>
    <Setter Property="BorderWidth" Value="5"></Setter>
    <Setter Property="VerticalOptions" Value="CenterAndExpand"></Setter>
</Style>

<Style x:Key="MainButtonStyle" TargetType="Button" BasedOn="{StaticResource BaseButtonStyle}">
    <Setter Property="BackgroundColor" Value="{StaticResource ApplicationMainColor}"></Setter>
</Style>
```



# Styles: Dynamic Resources

- Zoals de naam zegt: dynamisch
- Kan dus veranderen *at run-time*
- Via DynamicResource

```
<ContentPage.Content>  
  <StackLayout>  
    <Switch PropertyChanged="SwitchLayout"></Switch>  
  
    <Button  
      Style="{DynamicResource aButtonStyle}"  
      Text="Settings"  
      Clicked="GoToSettings"></Button>
```

```
<Application.Resources>  
  <ResourceDictionary>  
    <Style x:Key="styleA" TargetType="Button">  
      <Setter Property="BackgroundColor" Value="■"Black"></Setter>  
    </Style>  
    <Style x:Key="styleB" TargetType="Button">  
      <Setter Property="BackgroundColor" Value="■"Green"></Setter>  
    </Style>
```

?

<https://docs.microsoft.com/en-us/dotnet/maui/user-interface/styles/xaml#dynamic-style-inheritance>

# Styles: Dynamic Resources

- Zoals de naam zegt: dynamisch
- Kan dus veranderen in *run-time*
- Via DynamicResource

```
<ContentPage.Content>  
  <StackLayout>  
    <Switch PropertyChanged="SwitchLayout"></Switch>  
  
    <Button  
      Style="{DynamicResource aButtonStyle}"  
      Text="Settings"  
      Clicked="GoToSettings"></Button>
```

In C# code!

# Styles: Dynamic Resources

- **We definiëren Resource in C#**
  - In dit voorbeeld halen we een style op uit App.xaml
    - ➡ `App.Current.Resources["KEY"]`
  - We kunnen deze ook *at run-time* aanpassen

```
public partial class HomePage : ContentPage
{
    private bool _styleAisActive = false;
    1 reference
    public HomePage()
    {
        InitializeComponent();

        Resources["aButtonStyle"] = App.Current.Resources["styleA"];
    }
}

0 references
private void SwitchLayout(object sender, System.ComponentModel.PropertyChangedEventArgs e)
{
    if (_styleAisActive)
    {
        Resources["aButtonStyle"] = App.Current.Resources["styleA"];
    }
    else
    {
        Resources["aButtonStyle"] = App.Current.Resources["styleB"];
    }
    _styleAisActive = !_styleAisActive;
}
```

# Styles: Globale stijl

- We kunnen de Styles ook in allerhoogste niveau plaatsen  
=> App.xaml

```
<Application xmlns="http://xamarin.com/schemas/2014/forms"
             xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
             x:Class="Voorbeeldlijst2.App">
  <Application.Resources>
    <ResourceDictionary>
      <Style x:Key="GoBackButton" TargetType="Button">
        <Setter Property="BackgroundColor" Value="■"Red"></Setter>
        <Setter Property="TextColor" Value="□"White"></Setter>
        <Setter Property="HorizontalOptions" Value="Center"></Setter>
        <Setter Property="Padding" Value="20,20,20,20"></Setter>
      </Style>
    </ResourceDictionary>
  </Application.Resources>
</Application>
```

# Styles: Globale stijl

- Nadeel: App.xaml wordt heel groot
- Oplossing: Merged dictionaries
  - Aparte ResourceDictionary-files maken (.xaml)
  - Deze samenvoegen in App.xaml in MergedDictionaries
  - Check de default app!

```
<Application.Resources>
  <ResourceDictionary>
    <ResourceDictionary.MergedDictionaries>
      <ResourceDictionary Source="Resources/Styles/Colors.xaml" />
      <ResourceDictionary Source="Resources/Styles/Styles.xaml" />
    </ResourceDictionary.MergedDictionaries>
  </ResourceDictionary>
</Application.Resources>
```

```
<?xml version="1.0" encoding="UTF-8" ?>
<?xaml-comp compile="true" ?>
<ResourceDictionary
  xmlns="http://schemas.microsoft.com/dotnet/2021/maui"
  xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml">

  <Color x:Key="Primary">#512BD4</Color>
  <Color x:Key="Secondary">#DFD8F7</Color>
  <Color x:Key="Tertiary">#2B0B98</Color>
  <Color x:Key="White">White</Color>
  <Color x:Key="Black">Black</Color>
  <Color x:Key="Gray100">#E1E1E1</Color>
  <Color x:Key="Gray200">#C8C8C8</Color>
  <Color x:Key="Gray300">#ACACAC</Color>
  <Color x:Key="Gray400">#919191</Color>
  <Color x:Key="Gray500">#6E6E6E</Color>
  <Color x:Key="Gray600">#404040</Color>
```

```
<?xml version="1.0" encoding="UTF-8" ?>
<?xaml-comp compile="true" ?>
<ResourceDictionary
  xmlns="http://schemas.microsoft.com/dotnet/2021/maui"
  xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml">

  <Style TargetType="ActivityIndicator">
    <Setter Property="Color" Value="{AppThemeBinding Light={StaticResource Primary}, Dark={StaticResource White}}" />
  </Style>

  <Style TargetType="IndicatorView">
    <Setter Property="IndicatorColor" Value="{AppThemeBinding Light={StaticResource Gray200}, Dark={StaticResource Gray500}}" />
    <Setter Property="SelectedIndicatorColor" Value="{AppThemeBinding Light={StaticResource Gray950}, Dark={StaticResource Gray100}}" />
  </Style>
```

# Styles: Light and Dark mode

Reageren op het systeem (light en dark):

<https://docs.microsoft.com/en-us/dotnet/maui/user-interface/system-theme-changes>

```
BackgroundColor="{AppThemeBinding Light=Orange, Dark=Pink}"
```



# Extra:

## Styling with CSS + Theming

- Het is ook mogelijk om CSS te gebruiken om de elementen te stylen
- Zie: <https://docs.microsoft.com/en-us/dotnet/maui/user-interface/styles/css>
- Eigen themes maken:  
<https://docs.microsoft.com/en-us/dotnet/maui/user-interface/theming>

# Bronnen

- <https://docs.microsoft.com/en-us/dotnet/maui/>
- <https://www.youtube.com/watch?v=x-kyAeAGFVg>