

TOPMOTIVE[®]
EXPERIENCING BENCHMARKS

Universal Windows Platform

Attila Aszalos
Szoftverfejlesztő



office@dvse.ro

2016 április 7



Tartalom



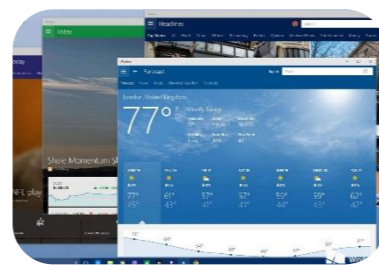
Bevezetés az UWP-be



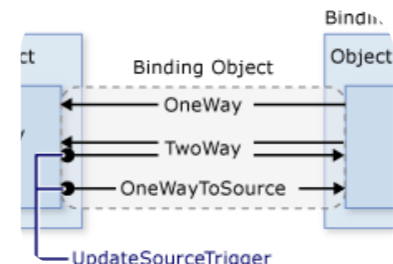
Az első UWP applikáció



Az applikáció élettartama



UI elemek



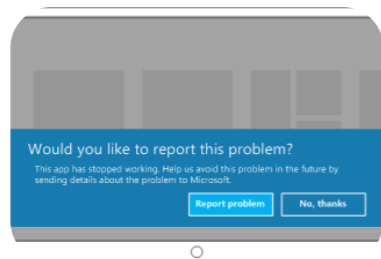
Binding



Navigációs modell



Aszinkron programozás UWP-ben



Dialog-ok



Adaptív elrendezés



Stílusok és téma



Adattárolás



Befejezés



BEVEZETÉS AZ UWP-BE

TOPMOTIVE®
EXPERIENCING BENCHMARKS



Kis történelem (1)

Desktop Mobile IoT Xbox Surface Hub

2015



2013 / 2014



2012



Kis történelem (2)

 Windows 8.1

 Windows 10



Eszközcsalád

Univerzális

Asztali

Mobil

Xbox

IoT

IoT fej
nélküli

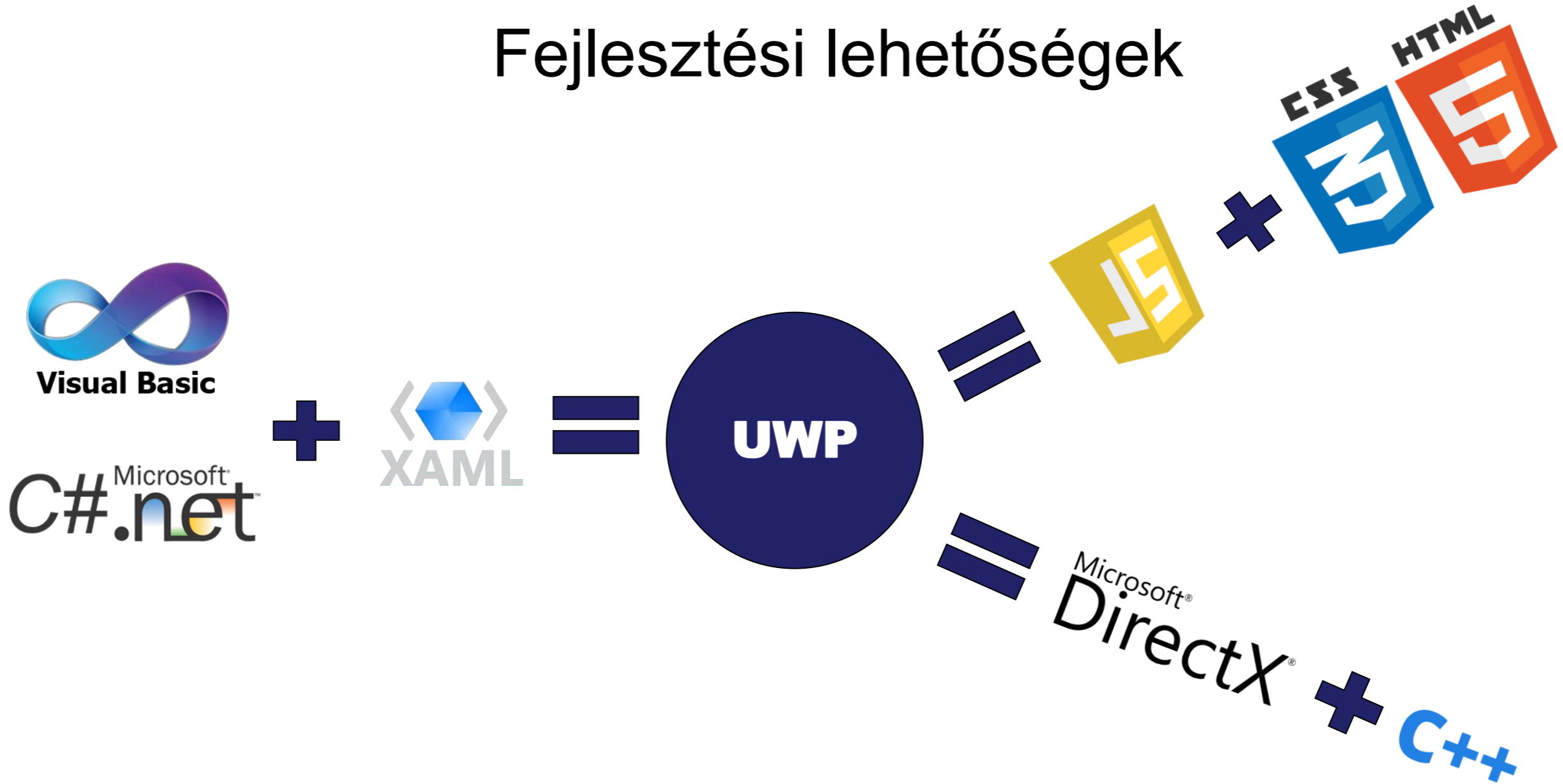
...

Felmerülő „univerzális” problémák

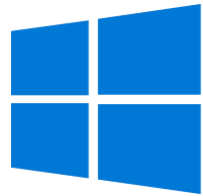
- Képernyőméret
- Fizikai billentyűzet hiánya
- Képernyő elfordítása
- Hardver gombok



Fejlesztési lehetőségek



Szükségletek

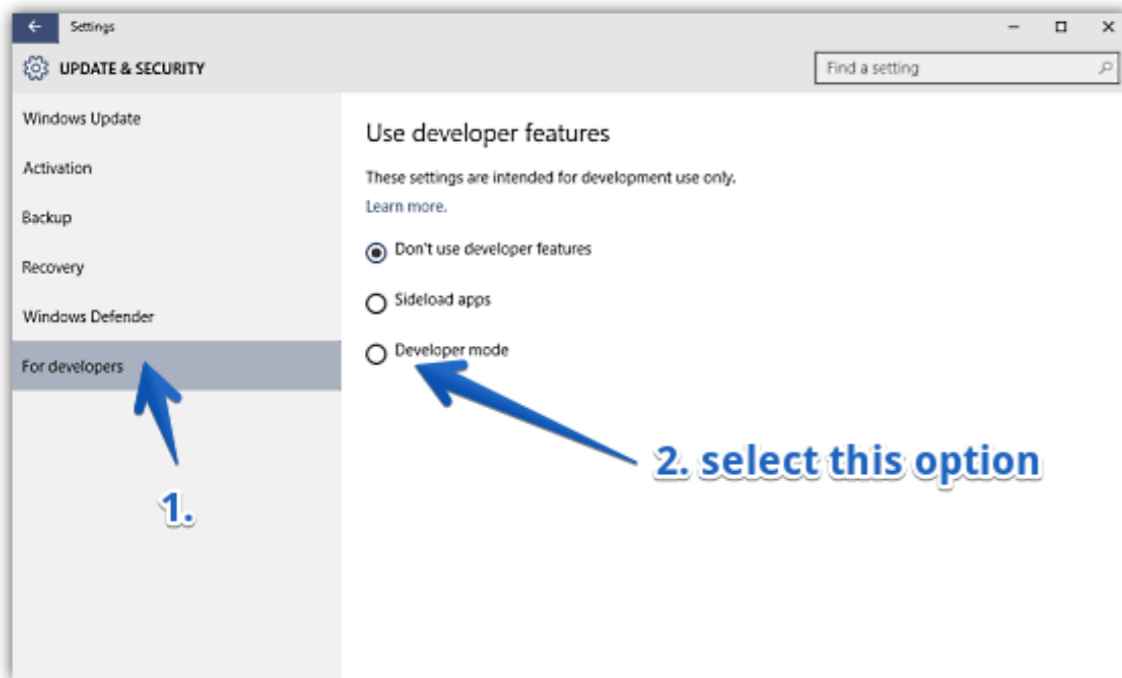


Windows 10



Visual Studio 2015

Microsoft®



- Regisztrálás mint fejlesztő (csak az app áruházba való feltöltéséhez szükséges)



AZ ELSŐ UWP APPLIKÁCIÓM



Új UWP applikáció létrehozása

SapientiaSamples - Microsoft Visual Studio

File Edit View Project Build Debug Team Tools Test Analyze Window Help

New Project... Ctrl+Shift+N

Add New Project

Recent

Installed

Visual C#

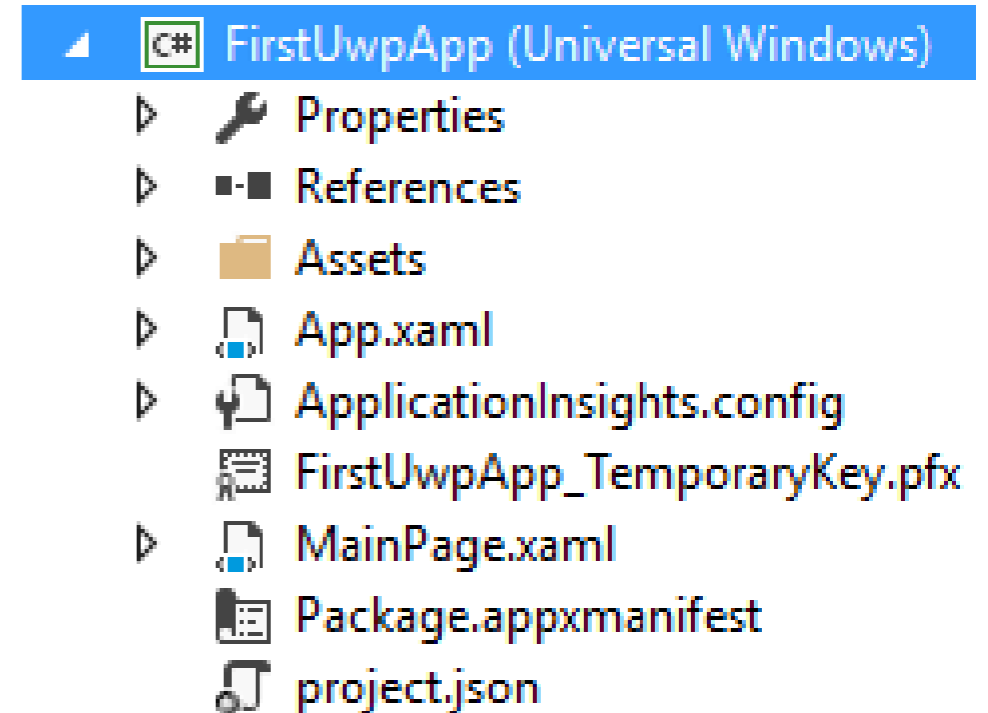
- Windows
 - Universal
 - Windows 8
 - Classic Desktop
- Web

.NET Framework 4.5.2 Sort by: Default

	Blank App (Universal Windows)	Visual C#
	Class Library (Universal Windows)	Visual C#
	Windows Runtime Component (Universal Windows)	Visual C#
	Unit Test App (Universal Windows)	Visual C#

UWP projekt struktúrája

- Assets mappa: az applikáció készleteit (erőforrásait) tartalmazza (képek, hanganyag, stb.)
- App.xaml a kiindulópont (MAIN)
- MainPage.xaml a főoldal
- Package.appxmanifest egyfajta applikáció konfigurációs fájl



App.xaml.cs kiindulási pont

```
protected override void OnLaunched(LaunchActivatedEventArgs e)
{
    Frame rootFrame = Window.Current.Content as Frame;
    if (rootFrame == null)
    {
        rootFrame = new Frame();
        Window.Current.Content = rootFrame;
    }
    if (rootFrame.Content == null)
    {
        rootFrame.Navigate(typeof(MainPage), e.Arguments);
    }
    Window.Current.Activate();
}
```

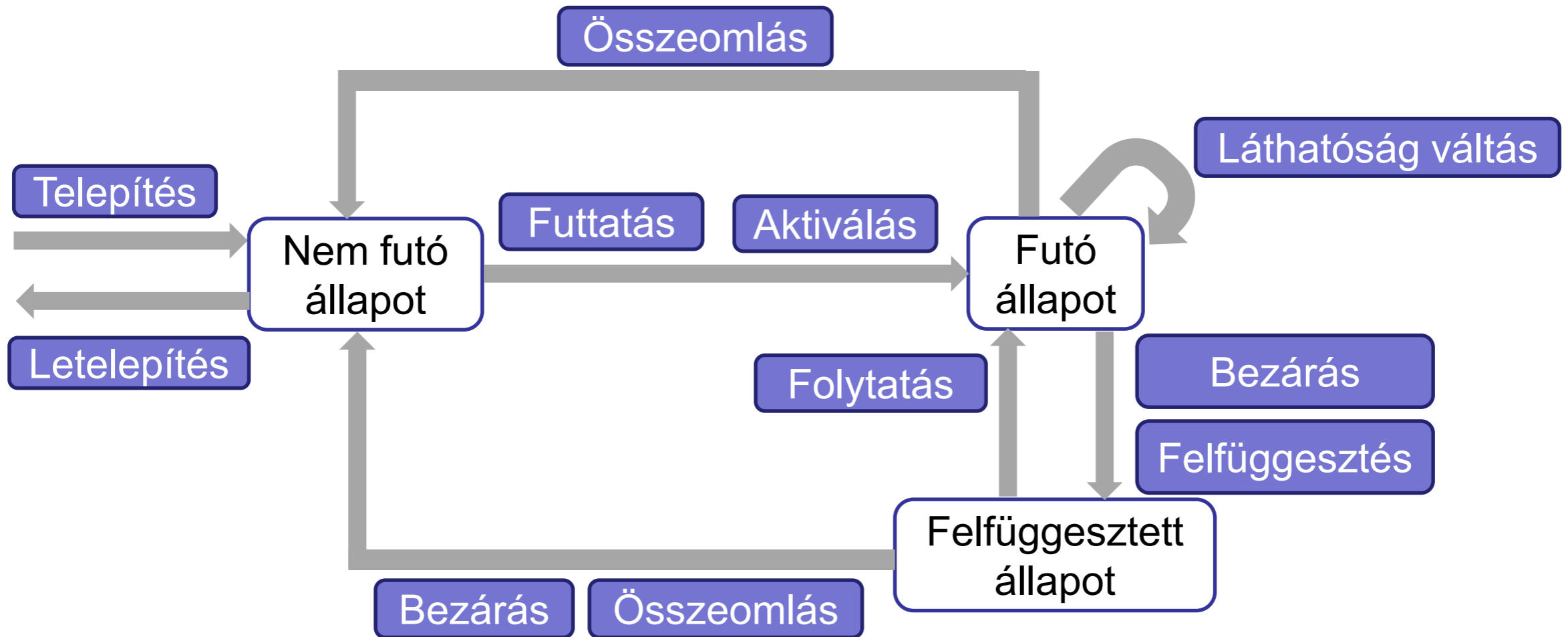




AZ APPLIKÁCIÓ ÉLETTARTAMA

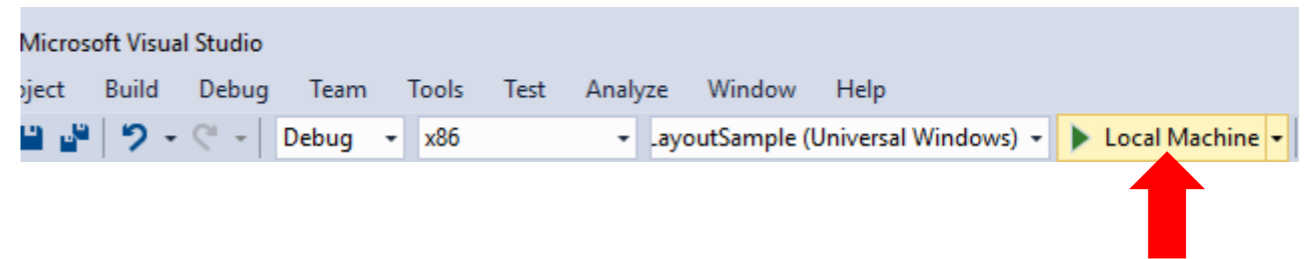
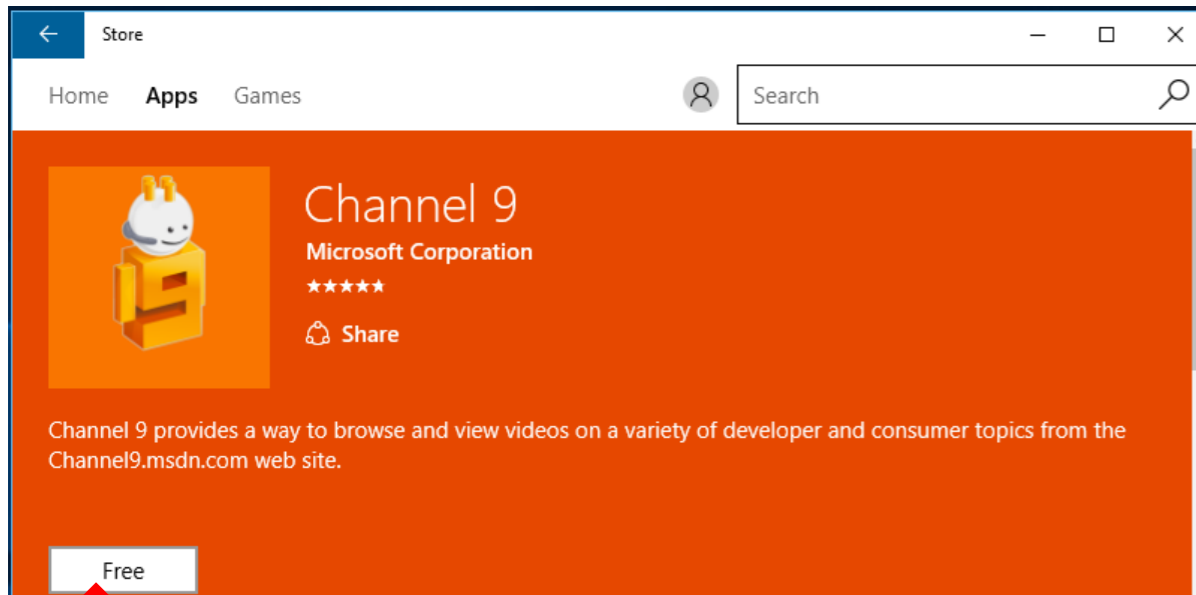


Az applikáció élettartama (Application lifecycle)



Telepítés (Deployment)

- Az első lépés amin átmegy az applikáció.
 - Az áruházból vagy Visual Studio segítségével



Futtatás (App launch)

- Nem volt elindítva
- Futott de összeomlott
- Felfüggesztődött, majd megölte az OpR

Nem futó
állapot

- Opcionális: Splash képernyő

Futó
állapot

```
protected override void OnLaunched(LaunchActivatedEventArgs e)
{
    if (e.PreviousExecutionState == ApplicationExecutionState.ClosedByUser ||
        e.PreviousExecutionState == ApplicationExecutionState.Terminated)
    {
        // a lementett tartalom betöltése
        // a megjelenített tartalom frissítése
    }
    else { // semmi különös }
```

Aktiválás

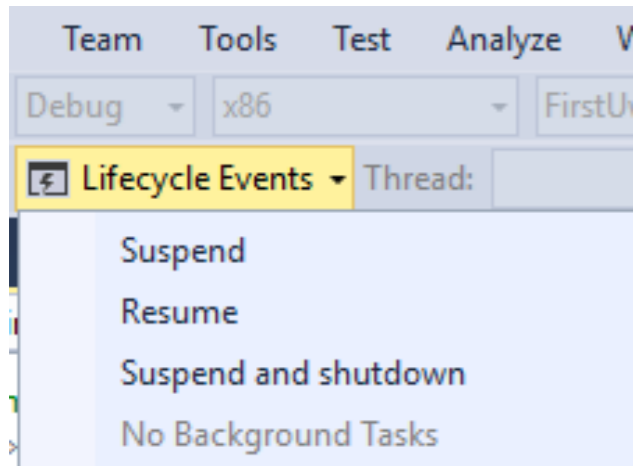
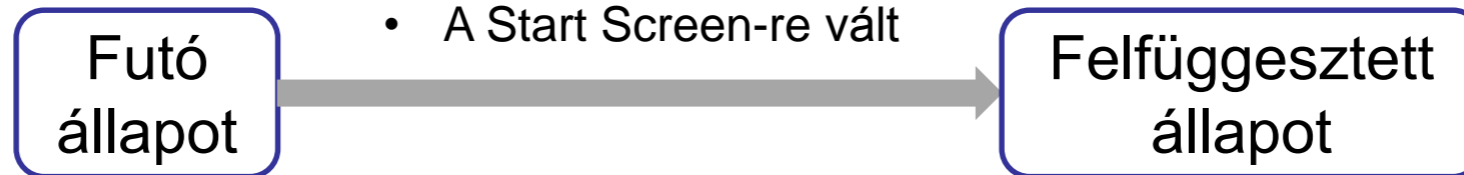
(App activation)

- Több aktiválási mód
 - Vannak speciális aktiválási metódusok
- Most kell visszaállítani az applikáció állapotát
- Amíg tart az aktiválás látható Splash képernyő
- Gyors kell legyen

```
protected override void OnActivated(IActivatedEventArgs args)
{
    switch (args.Kind)
    {
        case ActivationKind.Search:
            break;
        case ActivationKind.ShareTarget:
            break;
        case ActivationKind.FileOpenPicker:
            break;
        case ActivationKind.Contact:
            break;
        case ActivationKind.VoiceCommand:
            break;
        case ActivationKind.WalletAction:
            break;
        ...
    }
}
```

Felfüggesztés (App suspend)

- Egy másik applikációra vált a felhasználó
- A desktop-ra vált
- A Start Screen-re vált



```
private void OnSuspending(object sender, SuspendingEventArgs e)
{
    var deferral = e.SuspendingOperation.GetDeferral();
    // Applikáció állapotának a lementése
    deferral.Complete();
}
```

Láthatóság (App visibility)

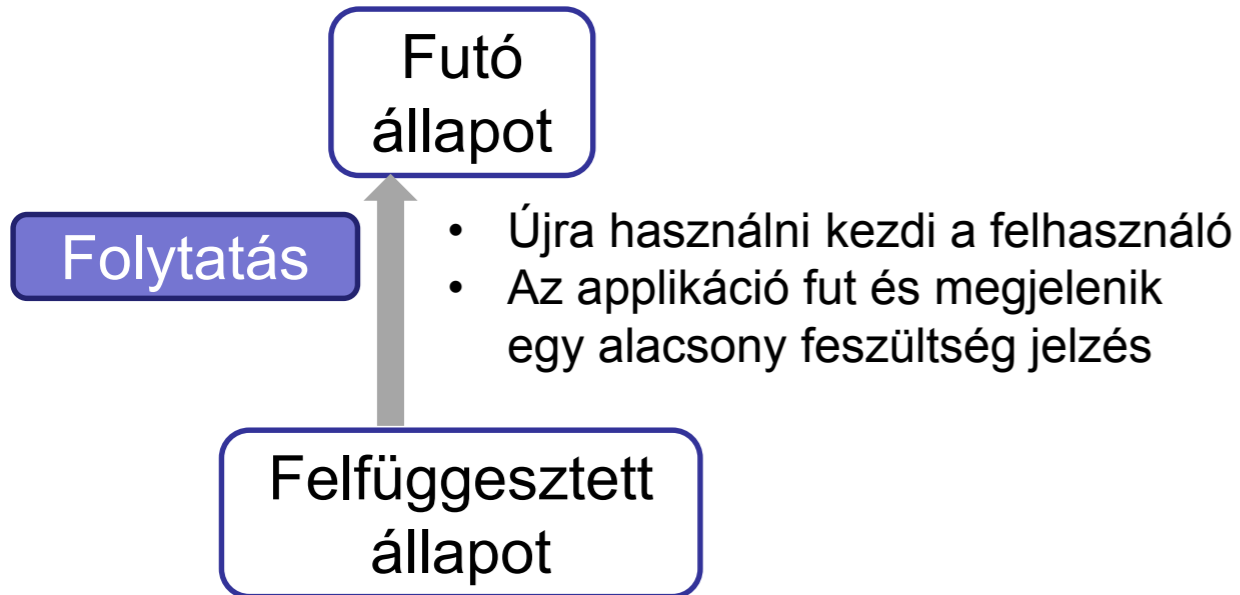
- Amikor egy másik applikációra vált a felhasználó és még nem függesztődik fel



```
protected override void OnLaunched(LaunchActivatedEventArgs e)
{
    Window.Current.VisibilityChanged += OnVisibilityChanged;
    ...
}
```

```
private void OnVisibilityChanged(object sender, VisibilityChangedEventArgs e)
{
    ...
}
```


Applikáció folytatása (App resume)



```
public App()  
{  
    ...  
    this.Resuming += OnResuming;  
}  
  
private void OnResuming(object sender, object e)  
{  
    // Olyan tartalom frissítése ami elavulhat pl.  
    // hely, időjárás, stb.  
}
```

Applikáció bezárása (App close)

```
protected override void OnLaunched(LaunchActivatedEventArgs e)
{
    if (e.PreviousExecutionState == ApplicationExecutionState.ClosedByUser)
    {
        ...
    }
}
```

```
protected override void OnActivated(IActivatedEventArgs args)
{
    if (args.PreviousExecutionState == ApplicationExecutionState.ClosedByUser)
    {
        ...
    }
}
```



Applikáció összeomlás (App crash)

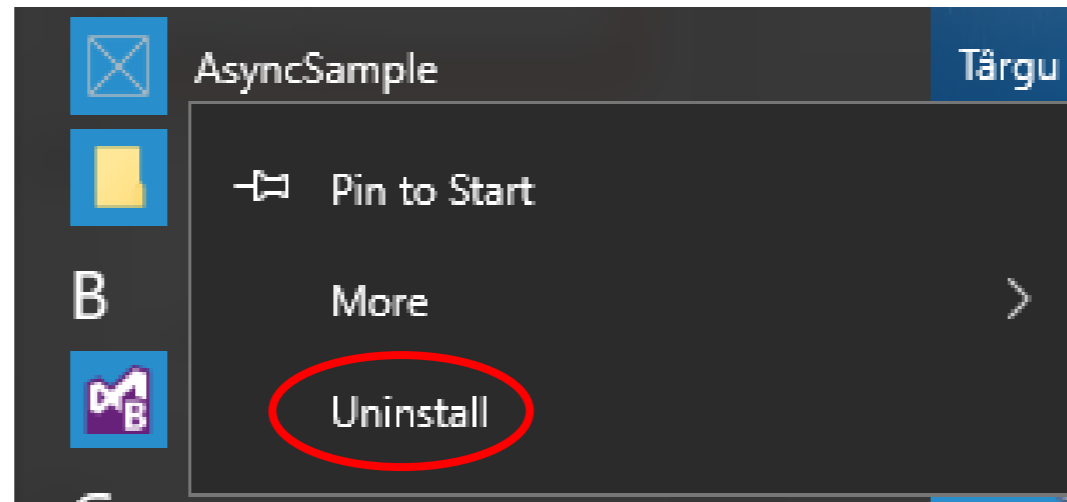
- Ha az applikáció összeomlik, megfagy vagy kivételt dob egy probléma kimutatás lesz elküldve a Microsoft-hoz, a felhasználó beállításaitól függően



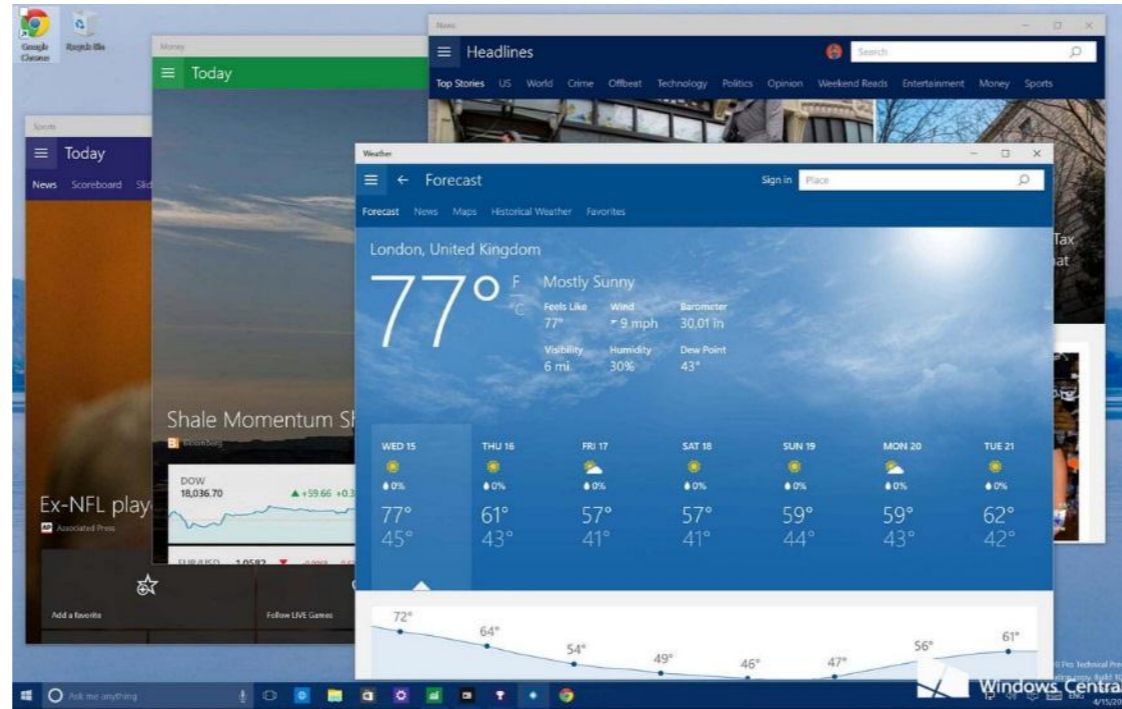
```
if (e.PreviousExecutionState == ApplicationExecutionState.NotRunning)
{
    // Nem ajánlott az elmentett adatokat használni
}
```

Applikáció törlése (App removal)

- Ha a felhasználó törli az applikációt, törlődnek az applikáció lokális adatai is. Azok az adatok, melyek a közös helyekre voltak mentve nem törlődnek pl. Documents, Pictures könyvtárak.



UI ELEMEK



TOPMOTIVE®
EXPERIENCING BENCHMARKS



DVSE **MSD**
as

AWDOC **ALDOC**

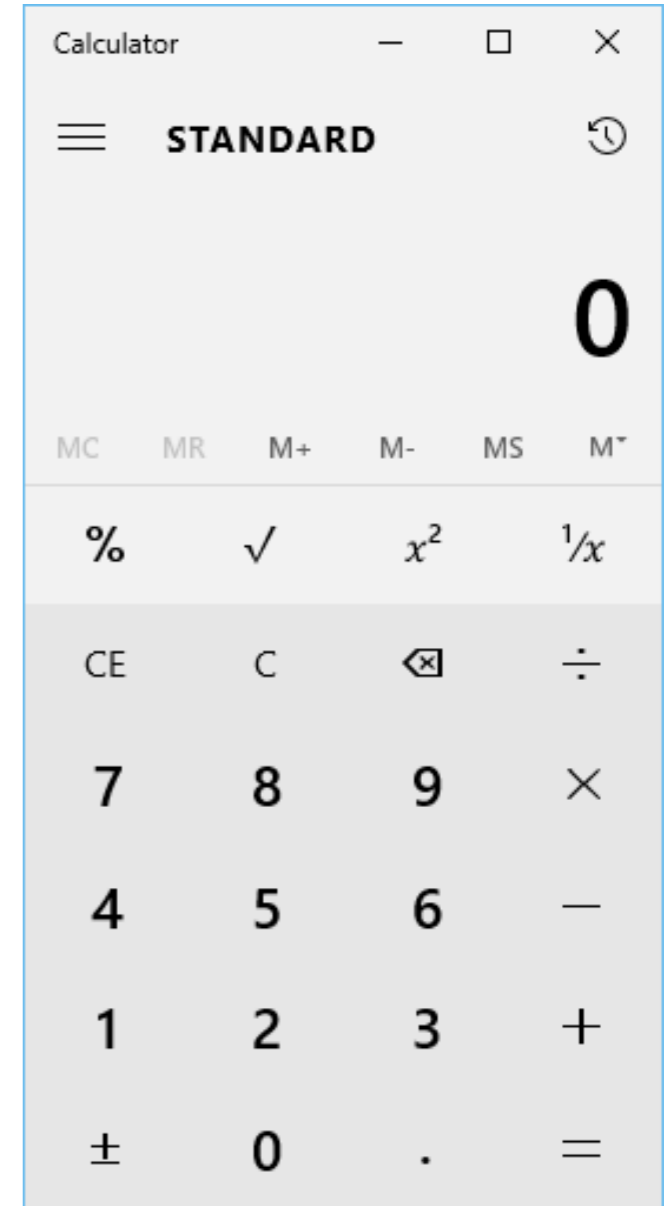
TOPMOTIVE
data

TOPMOTIVE
SYSTEMS

performance

UI elemek bevezető

- Oldalakkból (**Page**) épül fel. Egyszerre csak 1 oldal látható.
- Egy oldal UI elemekből (**UI element**) épül fel.
- A UI elemeket ún. konténer elemekbe (**Container control**) helyezzük, melyek különböző elrendezéseket (**layout**) tesznek lehetővé



Konténer elemek (1)

- Border

```
<Border Background="Red" BorderBrush="Green" BorderThickness="2">  
  <!-- Csak egy gyereke lehet -->  
</Border>
```

Below is a border



- Canvas

```
<Canvas Grid.Row="1" Background="LightBlue">  
  <Border Canvas.Left="12" Canvas.Top="24" Canvas.ZIndex="1"  
    BorderThickness="1" BorderBrush="Red">  
    <TextBlock Text="Hello!" />  
  </Border>  
  <Border Canvas.Left="18" Canvas.Top="18" Canvas.ZIndex="0" Background="Green"  
    Width="24" Height="10"/>  
</Canvas>
```

Below is a canvas



Konténer elemek (2)

- Stack panel

```
<StackPanel Orientation="Vertical">  
  <Border Width="44" Height="44" Background="Orchid"/>  
  <Border Width="88" Height="44" Background="Gray"/>  
  <Border Width="66" Height="22" Background="Cyan"/>  
</StackPanel>
```

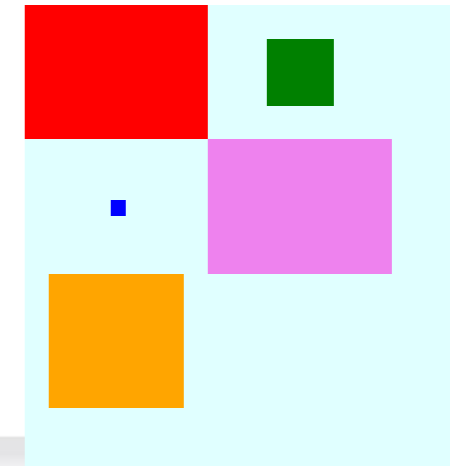
Below is a Stack Panel



- VariableSizedWrapGrid

```
<VariableSizedWrapGrid Width="140" Orientation="Horizontal"  
  Background="LightCyan">  
  <Border Background="Red" Width="60" Height="44"/>  
  <Border Background="Green" Width="22" Height="22"/>  
  <Border Background="Blue" Width="5" Height="5"/>  
  <Border Background="Violet" Width="99" Height="99"/>  
  <Border Background="Orange" Width="44" Height="44"/>  
</VariableSizedWrapGrid>
```

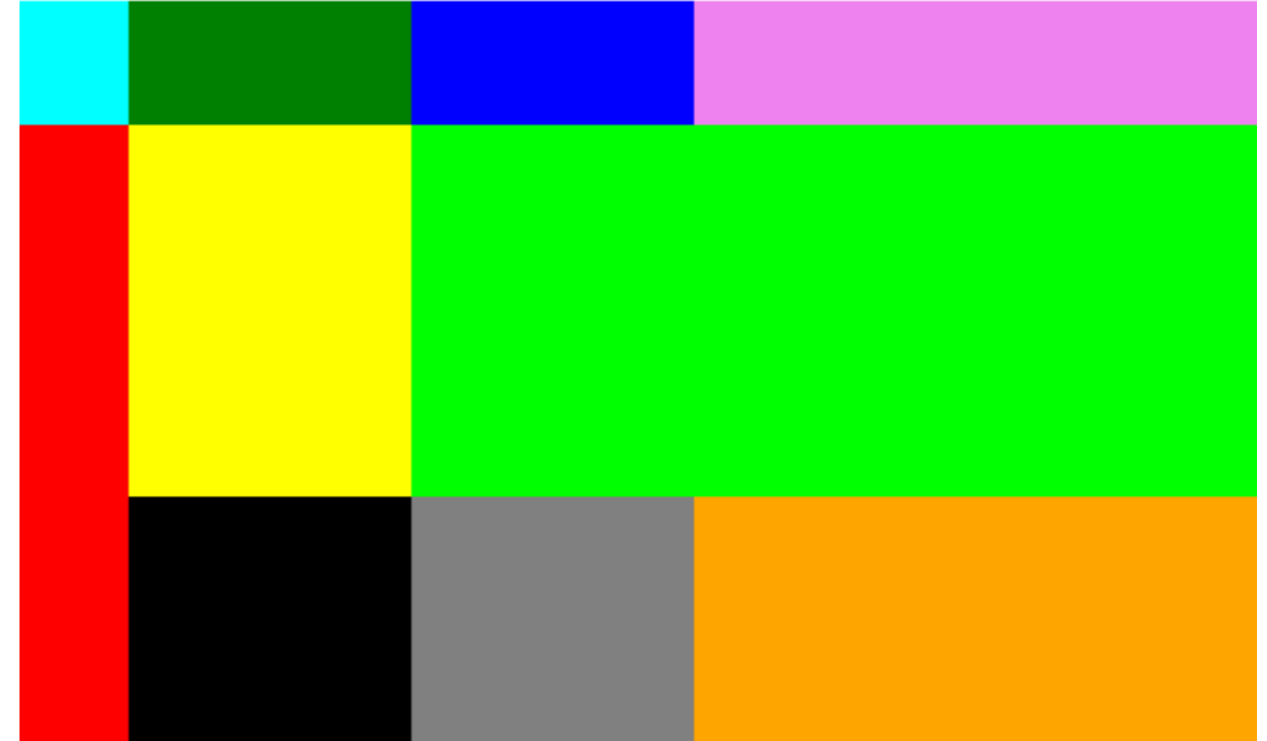
Below is a VariableSizedWrapGrid



Konténer elemek – Grid

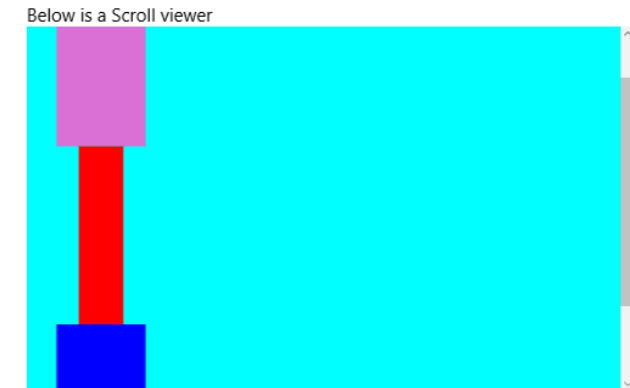
```
<Grid>
  <Grid.RowDefinitions>
    <RowDefinition Height="1*" />
    <RowDefinition Height="3*" />
    <RowDefinition Height="2*" />
  </Grid.RowDefinitions>
  <Grid.ColumnDefinitions>
    <ColumnDefinition Width="44" />
    <ColumnDefinition Width="*" />
    <ColumnDefinition Width="*" />
    <ColumnDefinition Width="2*" />
  </Grid.ColumnDefinitions>
  <Border Grid.Row="0" Grid.Column="0" Background="Cyan" />
  <Border Grid.Row="0" Grid.Column="1" Background="Green" />
  <Border Grid.Row="0" Grid.Column="2" Background="Blue" />
  <Border Grid.Row="0" Grid.Column="3" Background="Violet" />
  <Border Grid.Row="1" Grid.Column="0" Grid.RowSpan="2" Background="Red" />
  <Border Grid.Row="1" Grid.Column="1" Background="Yellow" />
  <Border Grid.Row="1" Grid.Column="2" Grid.ColumnSpan="2" Background="Lime" />
  <Border Grid.Row="2" Grid.Column="1" Background="Black" />
  <Border Grid.Row="2" Grid.Column="2" Background="Gray" />
  <Border Grid.Row="2" Grid.Column="3" Background="Orange" />
</Grid>
```

Below is a 3 by 4 grid



Konténer elemek – ScrollView

```
<ScrollView VerticalScrollMode="Enabled"
  ZoomMode="Enabled" MinZoomFactor="0.1" MaxZoomFactor="2.0"
  Background="Aqua">
  <Grid>
    <Grid.RowDefinitions>
      <RowDefinition/>
      <RowDefinition/>
      <RowDefinition/>
    </Grid.RowDefinitions>
    <Border Grid.Row="0" Width="300" Height="600" Background="Orchid"/>
    <Border Grid.Row="1" Width="150" Height="600" Background="Red"/>
    <Border Grid.Row="2" Width="300" Height="600" Background="Blue"/>
  </Grid>
</ScrollView>
```



Konténer elemek – ViewBox

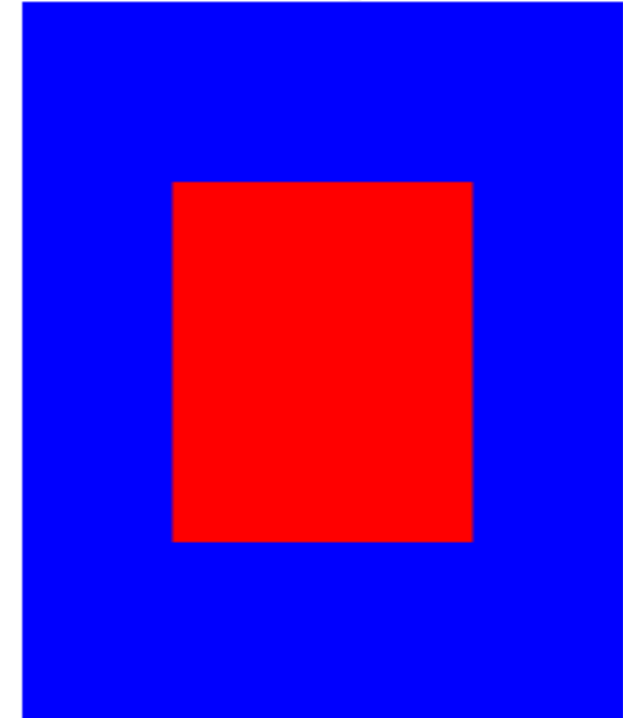
- ViewBox

```
<Grid>
  <Grid.ColumnDefinitions>
    <ColumnDefinition/>
    <ColumnDefinition/>
  </Grid.ColumnDefinitions>
  <Grid Grid.Column="0">
    <Border Background="Blue" Width="44" Height="44"/>
    <Border Background="Red" Width="22" Height="22"/>
  </Grid>

  <Viewbox Grid.Column="1" Stretch="Fill">
    <Grid>
      <Border Background="Blue" Width="44" Height="44"/>
      <Border Background="Red" Width="22" Height="22"/>
    </Grid>
  </Viewbox>
</Grid>
```

Below is a small grid

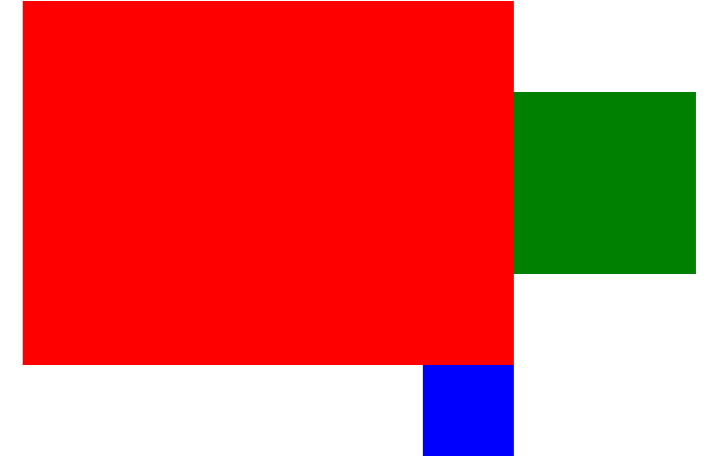
Below is a small grid in a ViewBox



Konténer elemek - RelativePanel

- Relative panel – Windows 10 újdonság

Below is a RelativePanel



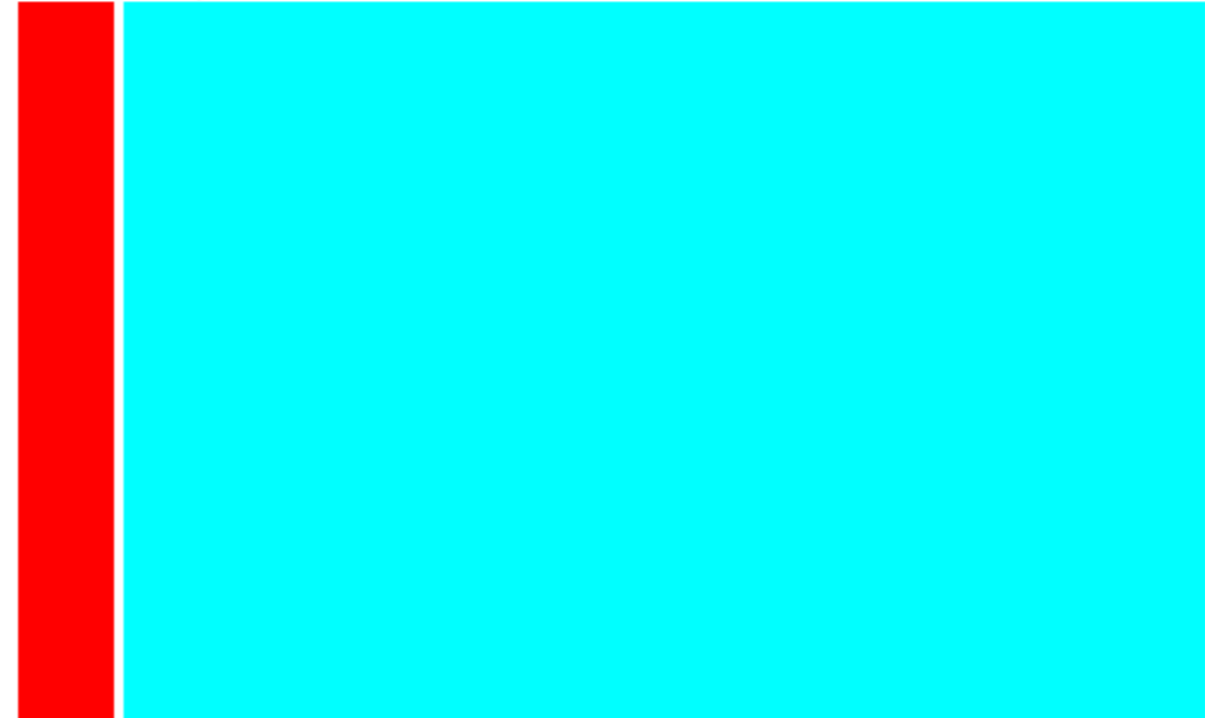
```
<RelativePanel>
  <Border x:Name="RedBorder" Background="Red" Width="120" Height="88"
    RelativePanel.AlignTopWithPanel="True" RelativePanel.AlignLeftWithPanel="True"/>
  <Border x:Name="GreenBorder" Background="Green" Width="44" Height="44"
    RelativePanel.AlignVerticalCenterWith="RedBorder" RelativePanel.RightOf="RedBorder"/>
  <Border x:Name="BlueBorder" Background="Blue" Width="22" Height="22"
    RelativePanel.Below="RedBorder" RelativePanel.AlignRightWith="RedBorder"/>
</RelativePanel>
```


Konténer elemek – SplitView

- Split view – Windows 10 újdonság

```
<SplitView IsPaneOpen="True" OpenPaneLength="44" DisplayMode="CompactOverlay"
  PaneBackground="Yellow">
  <SplitView.Pane>
    <Border Background="Red"/>
  </SplitView.Pane>
  <SplitView.Content>
    <Border Background="Cyan"/>
  </SplitView.Content>
</SplitView>
```

Below is a SplitView



A UI elemek egyéb csoportjai

Szöveges
elemek

Gomb
elemek

Flyout
elemek

Dátum és idő
elemek

App bar és
parancs
elemek

Kollekció és
adat elemek

Kiválasztási
elemek

Média
elemek

Folyamatjelző
elemek

Szöveges elemek

```
<TextBlock Text="Nem editálható címke szöveg" />
```

Nem editálható címke szöveg

```
<TextBox Text="" PlaceholderText="Súgó szöveg" />
```

Súgó szöveg

```
<PasswordBox Password="titkos" />
```

••••••••

```
<AutoSuggestBox QueryIcon="Find"  
PlaceholderText="Keresési kulcsszó..."  
ItemsSource="..."  
QuerySubmitted="AutoSuggestBox_QuerySubmitted"  
TextChanged="AutoSuggestBox_TextChanged"  
SuggestionChosen="AutoSuggestBox_SuggestionChosen"/>
```

Keresési kulcsszó... 🔍

Gomb elemek

```
<Button>  
  <StackPanel Orientation="Horizontal">  
    <TextBlock Text="&#xE722;" FontFamily="Segoe MDL2 Assets"  
      FontSize="24" Margin="4"/>  
    <TextBlock Text="Kamera" VerticalAlignment="Center"/>  
  </StackPanel>  
</Button>
```



```
<HyperlinkButton Content="Felhasználási feltételek" /> Felhasználási feltételek
```

```
<RepeatButton>  
  <StackPanel Orientation="Horizontal">  
    <TextBlock Text="&#xE74A;" ... />  
    <TextBlock Text="Hangerő" VerticalAlignment="Center"/>  
  </StackPanel>  
</RepeatButton>
```



Flyout elemek

Törlés

Biztos törölni szeretné a kiválasztott elemeket?

Igen

```
<Button Content="Törlés">
  <Button.Flyout>
    <Flyout>
      <StackPanel Padding="4">
        <TextBlock Text="Biztos törölni ..."/>
        <Button Content="Igen" Margin="0,4,0,0"/>
      </StackPanel>
    </Flyout>
  </Button.Flyout>
</Button>
```

Rendezés

Kikapcsolva

✓ Növekvő

Csökkenő

```
<MenuFlyout>
  <ToggleMenuFlyoutItem Text="Kikapcsolva"/>
  <MenuFlyoutSeparator/>
  <ToggleMenuFlyoutItem Text="Növekvő" />
  <ToggleMenuFlyoutItem Text="Csökkenő" />
</MenuFlyout>
```

Termék kosárba helyezése

+

```
<Button Content="+"
  TooltipService.ToolTip="Termék kosárba helyezése" />
```

Dátum és idő elemek

Kérem válasszon dátumot

April 2016						
Su	Mo	Tu	We	Th	Fr	Sa
27	28	29	30	31	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
1	2	3	4	5	6	7

```
<CalendarDatePicker PlaceholderText="Kérem válasszon dátumot"
    DateFormat="{0}{day.integer}/{1}{month.integer}/{2}{year.full}" />
```

```
<CalendarView />
```

April	10	2012
May	17	2013
June	18	2014
July	19	2015
August	20	2016
September	21	2017
October	22	2018
November	23	2019
December	24	2020

April 02 2016

```
<TimePicker ClockIdentifier="24HourClock" />
```

12 20

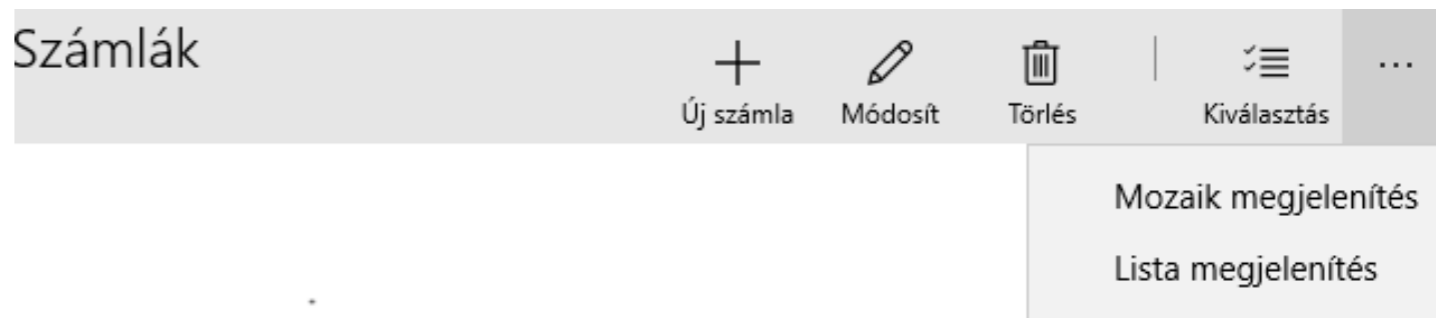
9	17
10	18
11	19
12	20
13	21
14	22
15	23
16	24

```
<DatePicker DayFormat="{0}{day.integer(2)}"
    MonthFormat="{1}{month.full}"
    YearFormat="{2}{year.full(4)}" />
```



AppBar és parancs elemek

```
<CommandBar>  
  <CommandBar.Content>  
    <TextBlock Text="Számlák" Style="{ThemeResource TitleTextBlockStyle}"/>  
  </CommandBar.Content>  
  <CommandBar.PrimaryCommands>  
    <AppBarButton Icon="Add" Label="Új számla"/>  
    <AppBarButton Icon="Edit" Label="Módosít"/>  
    <AppBarButton Icon="Delete" Label="Törlés"/>  
    <AppBarSeparator/>  
    <AppBarButton Label="Kiválasztás">  
      <AppBarButton.Icon>  
        <FontIcon Glyph="&#xE133;"/>  
      </AppBarButton.Icon>  
    </AppBarButton>  
  </CommandBar.PrimaryCommands>  
  <CommandBar.SecondaryCommands>  
    <AppBarToggleButton Label="Mozaik megjelenítés"/>  
    <AppBarToggleButton Label="Lista megjelenítés"/>  
  </CommandBar.SecondaryCommands>  
</CommandBar>
```



Kollekció és adatelemek (1)

```
<FlipView SelectionChanged="FlipView_SelectionChanged" >  
  <Image Source="interior.jpg" />  
  <Image Source="exterior.jpg" />  
  <Image Source="exterior2.jpg" />  
</FlipView>
```



```
<GridView ItemsSource="{Binding UsNewsItems}">  
  <GridView.ItemTemplate>  
    <DataTemplate x:DataType="local:NewsItem">  
      <StackPanel>  
        <Image Source="{x:Bind IconPath}"/>  
        <TextBlock Text="{x:Bind Headline}"/>  
      </StackPanel>  
    </DataTemplate>  
  </GridView.ItemTemplate>  
</GridView>
```

U.S.



Kollekció és adatelemek (2)

Első elem
Második elem
Harmadik elem

```
<ItemsControl>  
  <ItemsControl.Items>  
    <TextBlock Text="Első elem"/>  
    <TextBlock Text="Második elem"/>  
    <TextBlock Text="Harmadik elem"/>  
  </ItemsControl.Items>  
</ItemsControl>
```

Első elem

Második elem

Harmadik elem

```
<ListView>  
  <ListView.Items>  
    <TextBlock Text="Első elem"/>  
    <TextBlock Text="Második elem"/>  
    <TextBlock Text="Harmadik elem"/>  
  </ListView.Items>  
</ListView>
```

Kiválasztási elemek (1)

Egyetértek

```
<CheckBox IsChecked="True" IsThreeState="False" Content="Egyetértek" />
```

Terhes

```
<RadioButton GroupName="pregnant" IsChecked="False" Content="Terhes"/>
```

Nem terhes

```
<RadioButton GroupName="pregnant" IsChecked="True" Content="Nem terhes"/>
```



```
<Slider Minimum="0" Maximum="50" Value="25"
        StepFrequency="5" TickPlacement="Outside" TickFrequency="10"/>
```



```
<ComboBox SelectionChanged="ComboBox_SelectionChanged" Margin="44">
    <TextBlock Text="Maros"/>
    <TextBlock Text="Kovászna"/>
    <TextBlock Text="Hargita"/>
</ComboBox>
```

Kiválasztási elemek (2)

Elsődleges elemek

Elsődleges elemek

Elsődleges elemek



```
<ToggleButton Content="Elsődleges elemek" IsChecked="True" />
```

Elsődleges elemek



```
<ToggleSwitch IsOn="True" Header="Elsődleges elemek"  
OnContent="Be" OffContent="Ki" />
```

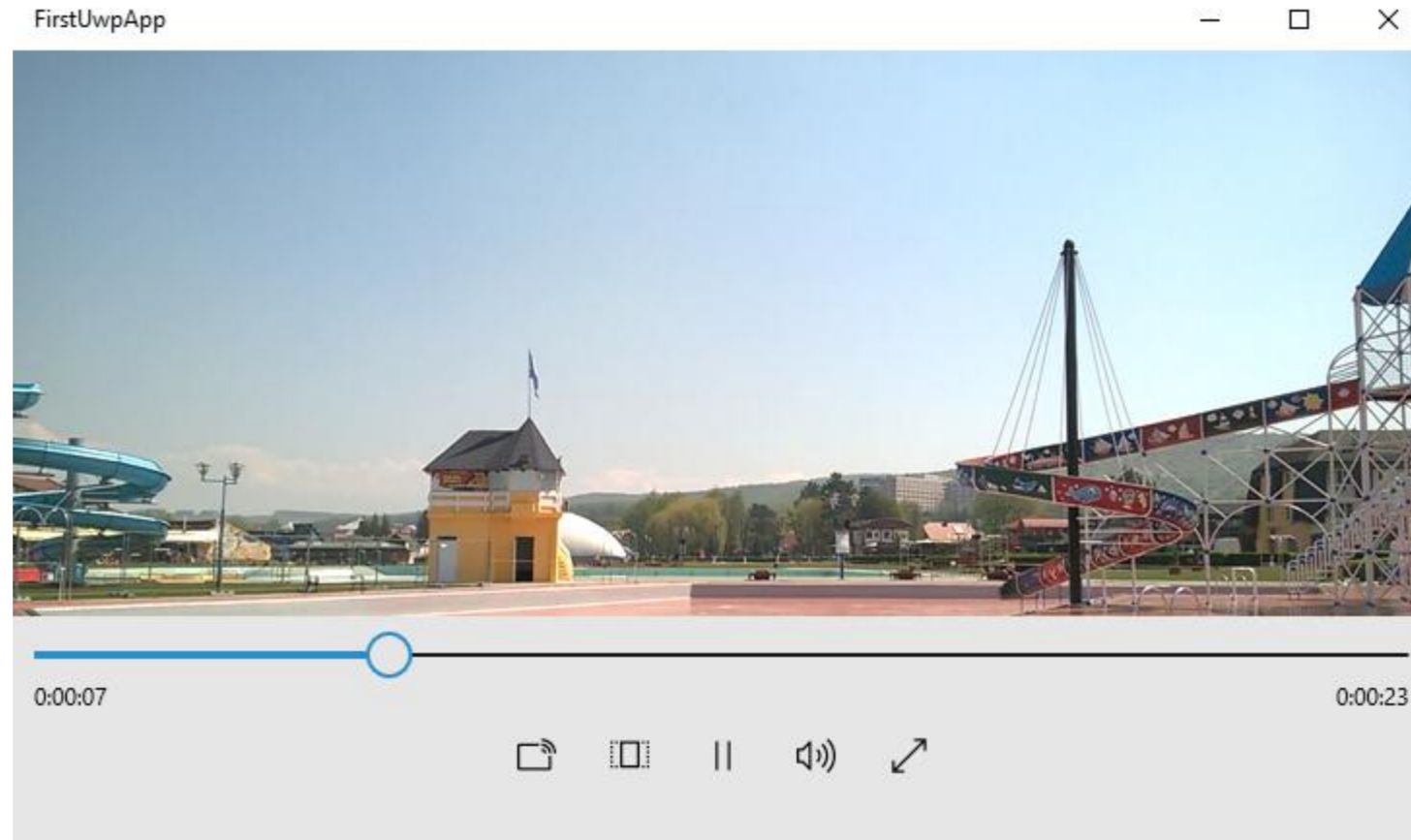
Első fogás

Második fogás

Desszert

```
<ListBox SelectionChanged="ListBox_SelectionChanged">  
  <TextBlock Text="Első fogás" />  
  <TextBlock Text="Második fogás" />  
  <TextBlock Text="Desszert" />  
</ListBox>
```

Média elemek



```
<MediaElement Source="Assets/video.mp4" AutoPlay="False"  
Volume="0.75"  
AreTransportControlsEnabled="True"/>
```

Folyamatjelző elemek



```
<ProgressRing IsActive="True" Width="128" Height="128" Margin="88" />
```

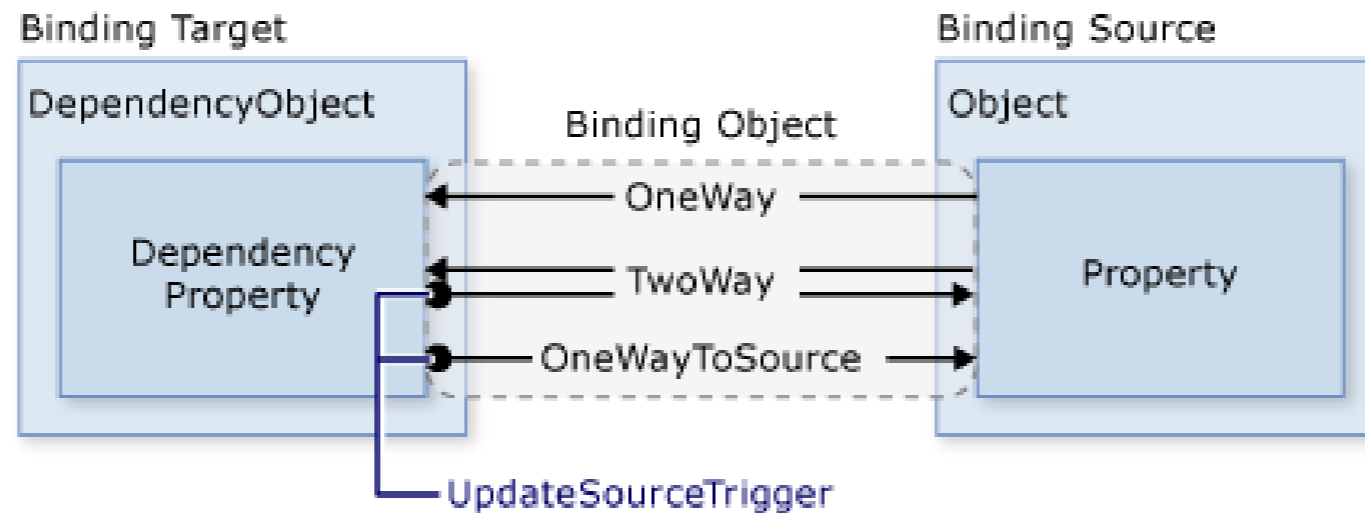


```
<ProgressBar IsIndeterminate="True"/>
```



```
<ProgressBar Minimum="0" Maximum="1" Value="0.74"  
SmallChange="0.1" LargeChange="0.3" />
```





BINDING (ÖSSZEKÖTÉS)



x:Bind

- Újfajta bind-olási mechanizmus, alternatíva a WPF-ből ismert mechanizmusra
- Előnye, hogy ez kompilált, míg a WPF-beni dinamikus (lásd a Binding hibákat az output window-ban, melyek csak futás közben derülnek ki)
- Gyorsabb
- Csak ismert típushoz lehet bind-olni



Fontos tudnivalók

- A WPF-ből ismert **Binding**-hoz képest, az **x:Bind**-nak a **Mode**-jának az alapértelmezett értéke **OneTime**.
- Nem létezik **OneWayToSource** bind-olási mód
- Nem létezik **UpdateSourceTrigger**
- Az átalakítók (**IValueConverter**) ugyanúgy használhatók, mind a hagyományos **Binding** esetében
- A fenti hiányosságok miatt, nem minden esetben használható
- Egy kis lépés a programozónak, de hatalmas ugrás a Microsoft-nak...



Szintaxis

- XAML

```
<UserControl x:Class="Sample.NewsItemUserControl">  
    <Grid>  
        <TextBlock Text="{x:Bind NewsItem.Title}" />  
    </Grid>  
</UserControl>
```

- Code behind

```
public sealed partial class NewsItemUserControl : UserControl  
{  
    private NewsItem NewsItem => (NewsItem)DataContext;  
    ...  
}
```

x:Bind DataTemplate-ben

- XAML

```
<Page x:Class="Sample.MainPage,,  
      xmlns:local="using:Sample">  
  <GridView ItemsSource="{x:Bind NewsItems}">  
    <GridView.ItemTemplate>  
      <DataTemplate x:DataType="local:NewItem">  
        <TextBlock Text="{x:Bind Title}" />  
      </DataTemplate>  
    </GridView.ItemTemplate>  
  </GridView>  
</Page>
```

- Code behind

```
public sealed partial class MainPage : Page  
{  
    private List<NewItem> NewsItems { get; set; }  
    ...  
}
```



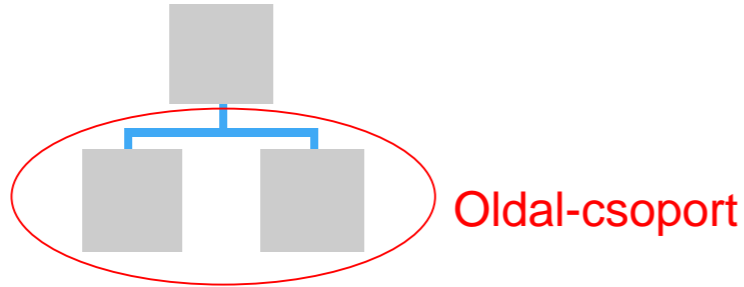


NAVIGÁCIÓS MODELL AZ UWP-BEN



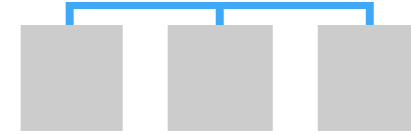
A navigáció alappillérei

Hierarchikus



- Az oldalak fa struktúrát alkotnak: 1 szülőnek több gyereke is lehet, 1 gyereknek csak 1 szüleje van
- A gyerek oldalakra csak a szülőn keresztül vezető úton lehet eljutni
- Ezt használjuk, ha:
 - Az szeretnénk, hogy az oldalak egy bizonyos sorrendben következzenek egymás után
 - Szülő-gyerek kapcsolat van az oldalak között
 - Több mint 7 oldal van egy csoportban

Egyenrangú

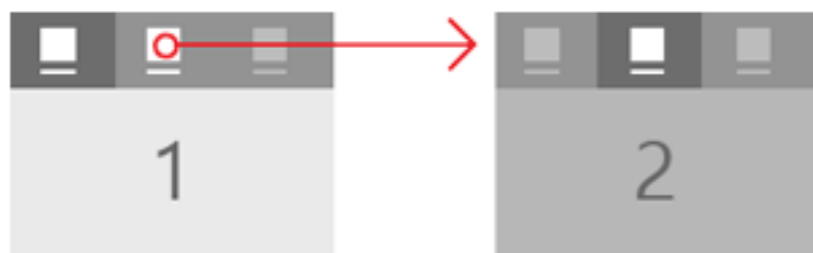


- Az oldalak egymás mellett léteznek
- Bármilyen sorrendben mehetünk egyik oldalról a másikra
- Ezt használjuk, ha:
 - Nem fontos az oldalak sorrendisége
 - Az oldalak teljesen függetlenek egymástól, nem határozható meg egyértelmű szülő-gyerek kapcsolat köztük
 - Kevesebb mint 8 oldal van a csoportban

Egyenrangú navigáció

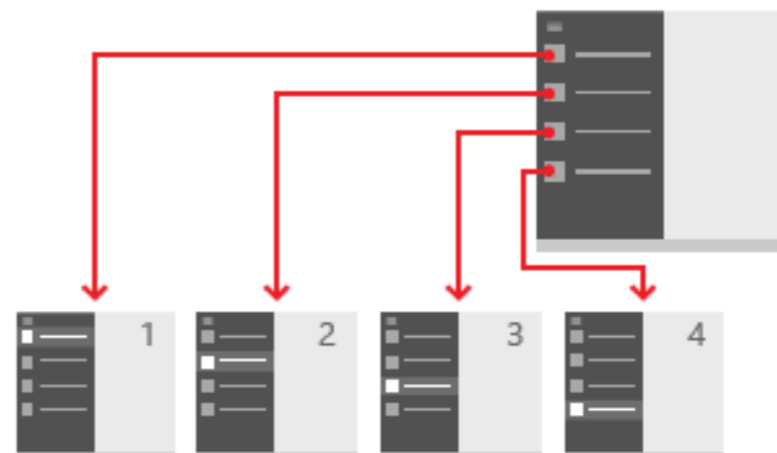
Tabok és pillérek (Tabs and pivot)

- Mindig látható egy hivatkozás lista az azonos szinten levő oldalakra
- Ezt használjuk, ha:
 - Az oldalak száma 2 és 5 között van
 - Arra számítunk, hogy a felhasználó sűrűn fog oldalakat váltani



Navigációs tábla (Navigation pane)

- Látható egy hivatkozás lista a legfelső szinten elhelyezkedő oldalakra
- Ezt használjuk, ha:
 - Nem számítunk arra, hogy a felhasználó gyakran fog oldalt váltani
 - Helyet akarunk spórolni, lelassítva navigációs műveletet
 - Az oldalak a legfelső szinten helyezkednek el



Hierarchikus navigáció

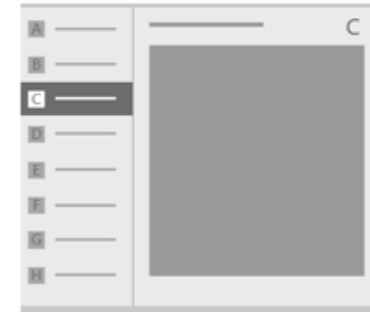
Csomópont (Hub)

- Előnézetet biztosít a gyerek oldalakról
- Az egyenrangú navigációs elemekkel szemben a hivatkozások az oldalon belül vagy az oldalon belüli fejléceken helyezkednek el
- Ezt használjuk, ha:
 - A felhasználó akarják látni a tartalom egy részét anélkül, hogy navigálniuk kellene



Szülő gyerek (Master details)

- Listában elhelyezkedő elemekről biztosít előnézetet
- Egy elem kiválasztása esetén a részletes leírás a gyerek oldalon jelenik meg
- Ezt használjuk, ha:
 - A felhasználók gyakran akarják váltogatni a megjelenített gyerek elemeket
 - A listaelemek törölhetőek, rendezhetőek vagy módosíthatók



Navigáció megvalósítás (1)

- A navigáció a Keret (**Frame**) segítségével valósítható meg
- Elérhető a `this.Frame` –en keresztül, ahol a `this.Page` típusú
- Globális események:

```
frame.Navigating += OnNavigating;  
frame.Navigated += OnNavigated;  
frame.NavigationFailed += OnNavigationFailed;  
frame.NavigationStopped += OnNavigationStopped;
```

```
void OnNavigating(object sender,  
    NavigatingCancelEventArgs e)  
{...}
```

```
void OnNavigated(object sender,  
    NavigationEventArgs e)  
{...}
```

```
void OnNavigationFailed(object sender,  
    NavigationFailedEventArgs e)  
{...}
```

```
void OnNavigationStopped(object sender,  
    NavigationEventArgs e)  
{...}
```



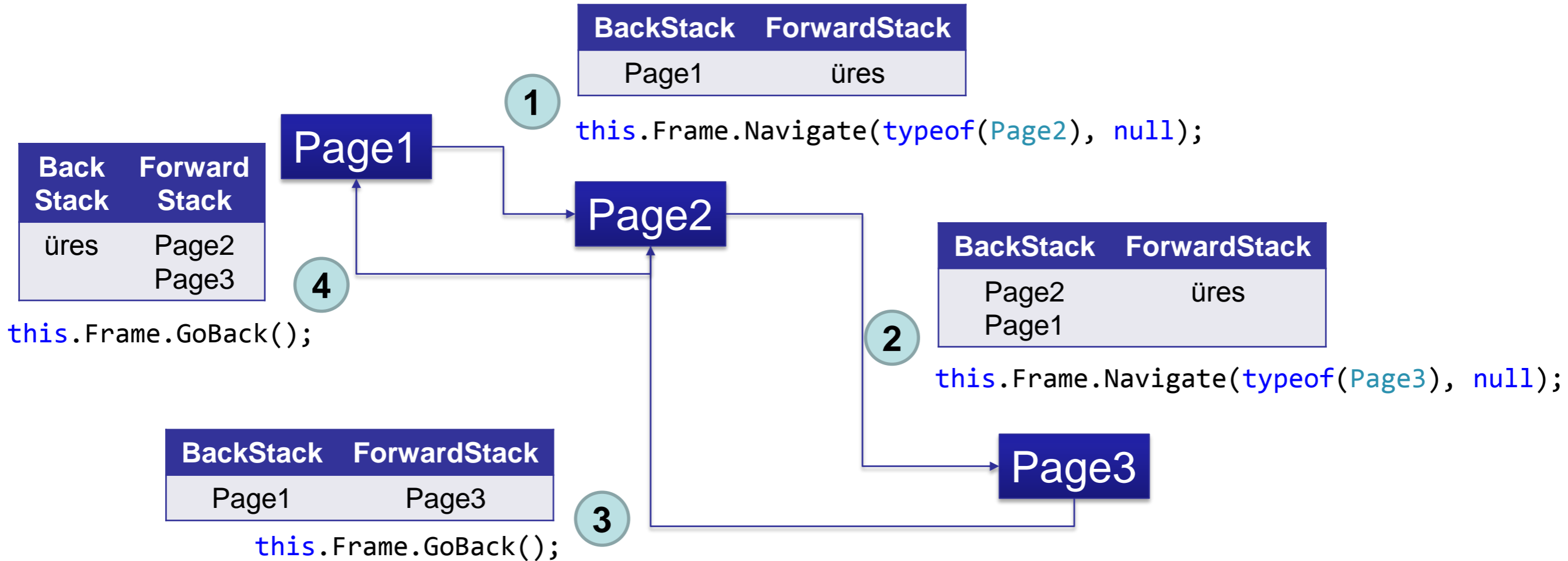
Navigáció megvalósítás (2)

```
public sealed partial class MainPage : Page
{
    private void NavigateButton_Click(object sender, RoutedEventArgs e)
    {
        object parameter = ...;
        this.Frame.Navigate(typeof(PageTwo), parameter);
    }

    // Amikor erre az oldalra érkezett
    protected override void OnNavigatedTo(NavigationEventArgs e)
    {
        ... /* e.Parameter */
        // Mielőtt el fogja hagyni az oldalt
        protected override void OnNavigatingFrom(NavigatingCancelEventArgs e)
        {
            ... /* e.Cancel */
            // Miután elhagyta az oldalt
            protected override void OnNavigatedFrom(NavigationEventArgs e)
            {
                ...
            }
        }
    }
}
```



Frame.BackStack és Frame.ForwardStack



Példa

- SimpleNavigationSample



Megjegyzés

- Alapbeállítás, hogy az oldalak ne őrizzék meg állapotukat

```
public MainPage()  
{  
    ...  
    this.NavigationCacheMode = NavigationCacheMode.Required;  
}
```

```
public enum NavigationCacheMode  
{  
    Disabled = 0,  
    Required = 1,  
    Enabled = 2  
}
```

- Az állapotmegőrzés implementálása meghaladja ennek a bemutatónak a célját
 - Ha implementálunk állapotmegőrzést a navigációs paraméter, **csak alaptípus lehet**: karakterlánc, numerikus vagy GUID



Navigációt elősegítő elemek (1)

- Pivot

- PivotSample

```
<Pivot Title="PIVOT TITLE">  
  <PivotItem>  
    <PivotItem.Header>  
      <StackPanel Orientation="Horizontal">  
        <TextBlock FontFamily="Segoe MDL2 Assets"  
          Text="&#xE77B;" FontSize="48"/>  
        <TextBlock Text="Contacts" VerticalAlignment="Center"  
          Margin="12,0,0,0"/>  
      </StackPanel>  
    </PivotItem.Header>  
    <Grid Background="LightBlue">  
      <TextBlock Text="Content of pivot item 1."/>  
    </Grid>  
  </PivotItem>  
  ...  
</Pivot>
```

PivotSample



PIVOT TITLE



Contacts



Settings

Content of pivot item 1.



LOPMOTIVE
data

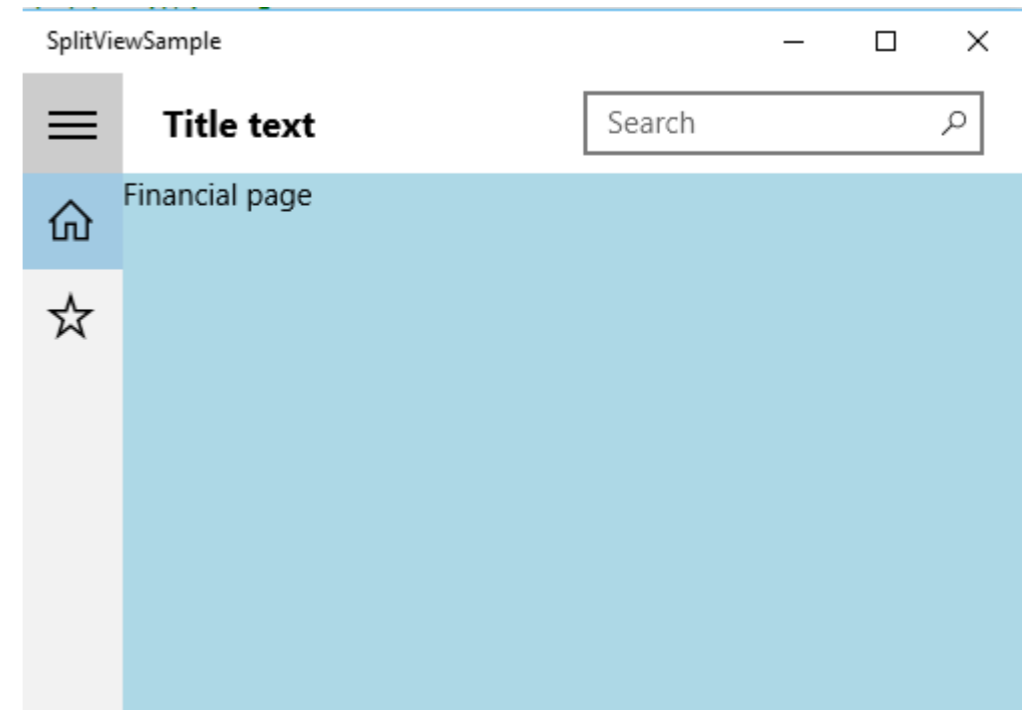
LOPMOTIVE
SYSTEMS



Navigációt elősegítő elemek (2)

- Hamburger navigáció **SplitView** segítségével
 - SplitViewSample

```
<SplitView DisplayMode="CompactOverlay" CompactPaneLength="50" x:Name="MainSplitView">
  <SplitView.Pane>
    <ListBox SelectionMode="Single" x:Name="MenuListBox" SelectionChanged="MenuListBox_SelectionChanged">
      <ListBox.Items>
        <ListBoxItem x:Name="FinancialMenuItem">
          <StackPanel Orientation="Horizontal">
            <TextBlock Text="&#xE80F;" ... />
            <TextBlock Text="Financial" ... />
          </StackPanel>
        </ListBoxItem>
        ...
      </ListBox.Items>
    </ListBox>
  </SplitView.Pane>
  <SplitView.Content>
    <Frame x:Name="MainFrame" x:FieldModifier="public" />
  </SplitView.Content>
</SplitView>
```

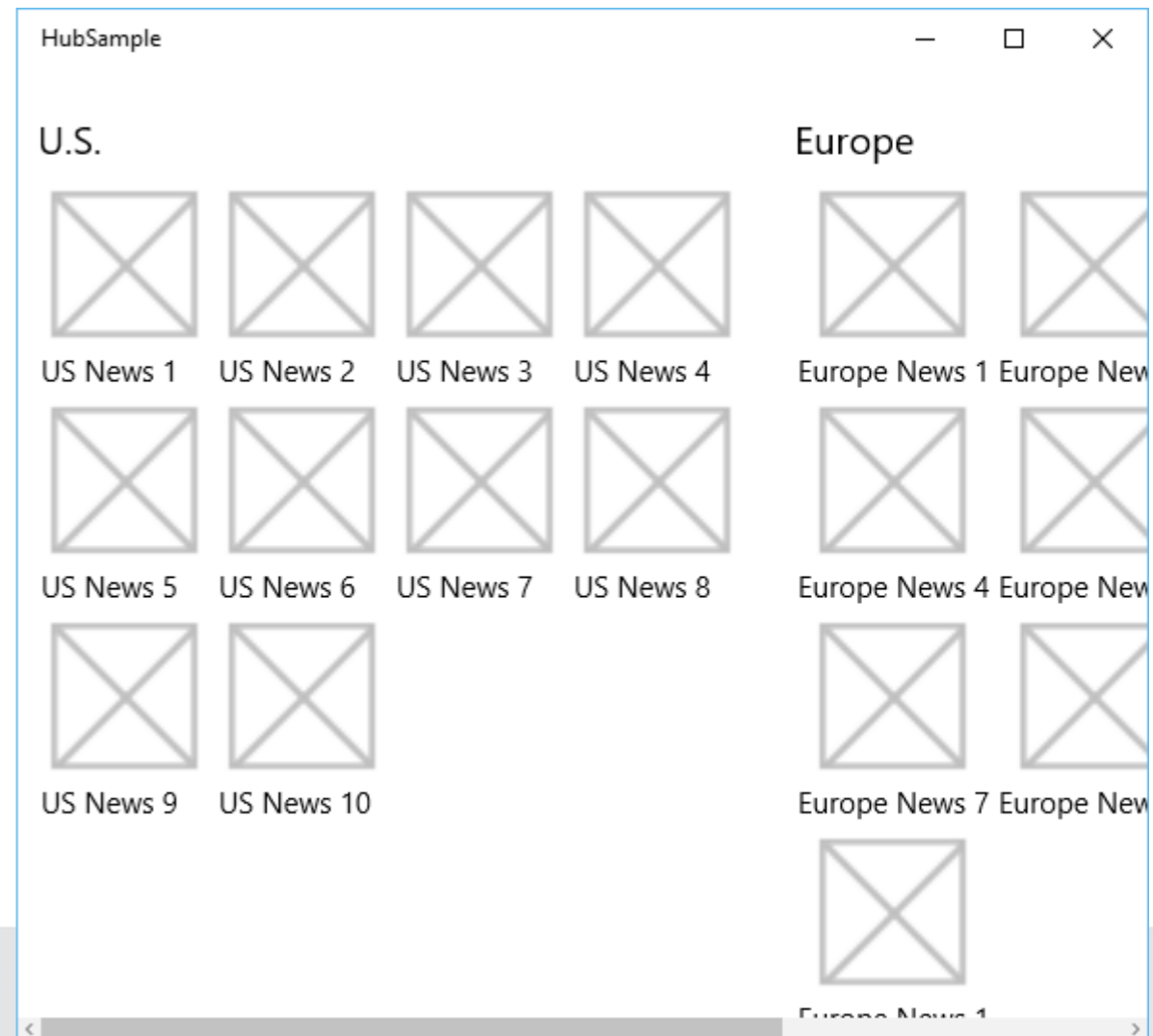


Navigációt elősegítő elemek (3)

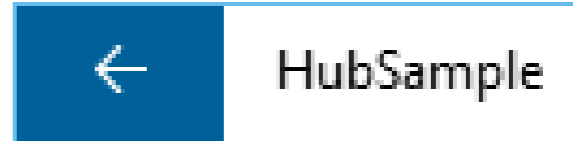
- Hub

- Hub Sample

```
<Hub>
  <HubSection Header="U.S." MaxWidth="400">
    <DataTemplate>
      <GridView ItemsSource="{Binding UsNewsItems}">
        <GridView.ItemTemplate>
          <DataTemplate x:DataType="local:NewsItem">
            <StackPanel>
              <Image Source="{x:Bind IconPath}"/>
              <TextBlock Text="{x:Bind Headline}"/>
            </StackPanel>
          </DataTemplate>
        </GridView.ItemTemplate>
      </GridView>
    </DataTemplate>
  </HubSection>
  <HubSection Header="Europe" MaxWidth="400">
    ...
  </HubSection>
</Hub>
```



Vissza gomb



```
protected override void OnLaunched(LaunchActivatedEventArgs e)
{
    Frame rootFrame = Window.Current.Content as Frame;
    ...
    rootFrame.Navigated += OnNavigated;
    SystemNavigationManager.GetForCurrentView().BackRequested += OnBackRequested;
    ...
}
```



```
private void OnBackRequested(object sender, BackRequestedEventArgs e)
{
    Frame frame = Window.Current.Content as Frame;
    if (frame.CanGoBack) { frame.GoBack(); e.Handled = true; }
}
private void OnNavigated(object sender, NavigationEventArgs e)
{
    Frame frame = Window.Current.Content as Frame;
    SystemNavigationManager.GetForCurrentView().AppViewBackButtonVisibility =
        (frame.CanGoBack ? AppViewBackButtonVisibility.Visible : AppViewBackButtonVisibility.Collapsed)
}
```





ASZINKRON PROGRAMOZÁS UWP-BEN



Példaprogram

- AsyncSample
 - Nem aszinkron
 - Hagyományos aszinkron programozás
 - Async/await aszinkron programozás



Deklarációs szintaxis

Szinkron

```
public interface IDataService
{
    List<string> GetProductNameSuggestions();
}

public class DataService : IDataService
{
    public List<string> GetProductNameSuggestions()...
```

Aszinkron

```
public interface IDataService
{
    Task<List<string>> GetProductNameSuggestionsAsync();
}

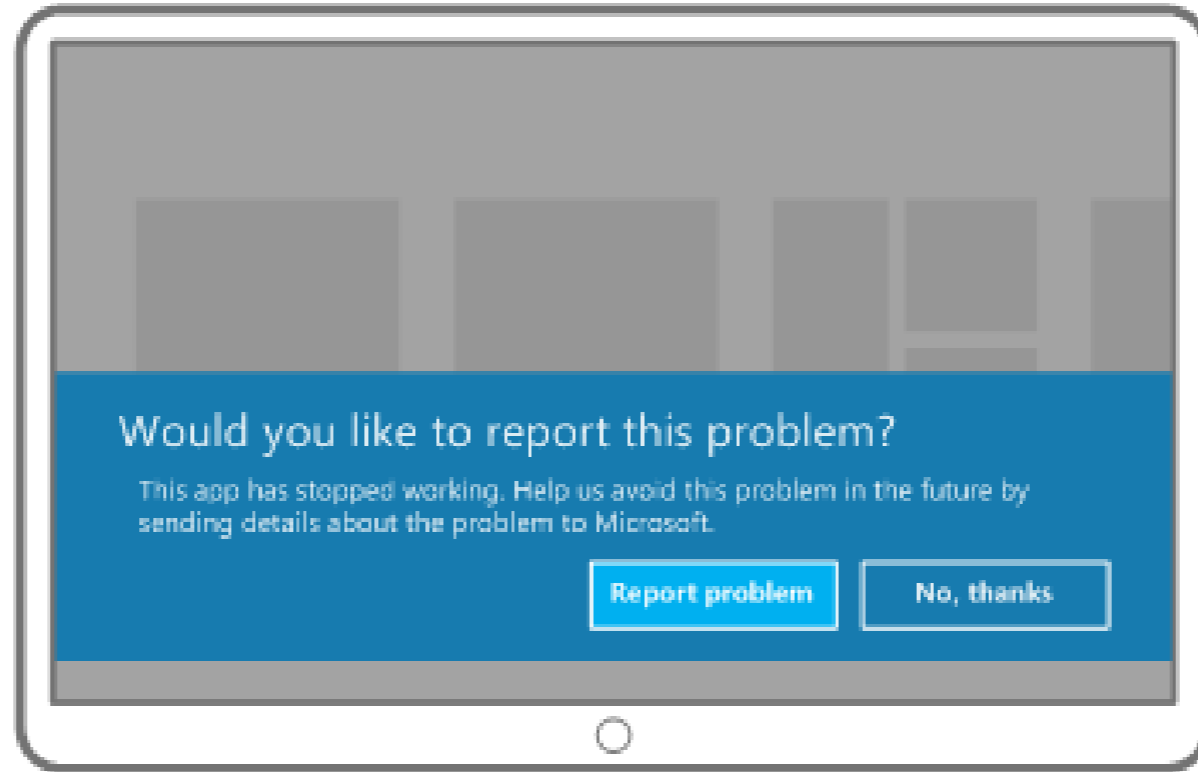
public class DataService : IDataService
{
    public Task<List<string>> GetProductNameSuggestionsAsync()...
```

Meghívási szintaxis

```
public class DataService : IDataService
{
    public Task<List<string>> GetProductNameSuggestionsAsync()...
```

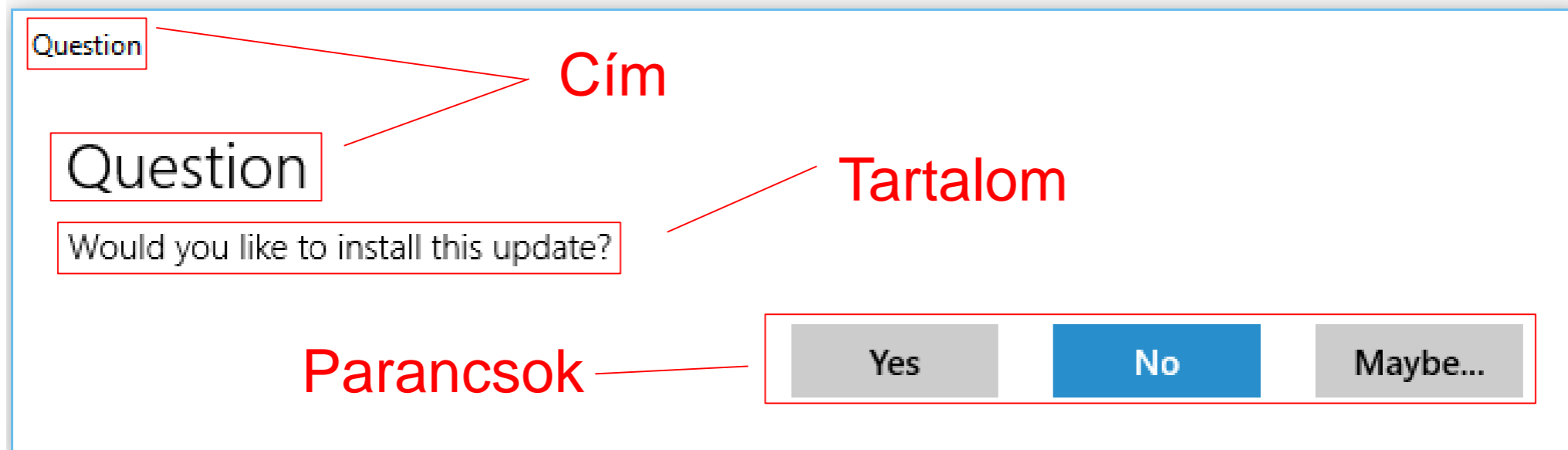
```
private async void GetProductNamesButton_Click(object sender, RoutedEventArgs e)
{
    DataService dataService = new DataService();
    List<string> suggestions = await dataService.GetProductNameSuggestionsAsync();
    ProductNamesComboBox.ItemsSource = suggestions;
}
```

DIALOG-OK



MessageDialog (1)

- Egyszerű üzenetek megjelenítésére használatos



MessageDialog (2)

- Létrehozás: `public MessageDialog(string title, string content)`
- Személyreszabás:

```
public class MessageDialog
{
    public IList<IUICommand> Commands { get; }
    public System.UInt32 DefaultCommandIndex { get; set; }
    public System.UInt32 CancelCommandIndex { get; set; }
}
```

- Megjelenítés: `MessageDialog dialog = new MessageDialog(...);
await dialog.ShowAsync().AsTask();`



MessageDialog (3)

```
public sealed class UICommand : ICommand
{
    public UICommand(String label, ICommandInvoker action);

    ICommandInvoker Invoked { get; set; }
    System.String Label { get; set; }
}

public delegate void ICommandInvoker(ICommand command);
```



ContentDialog

- Különleges megjelenítést igénylő üzenetek megjelenítésére használatos
- XAML-ből tetszés szerint személyre szabható
- Alapból 2 gombot támogat, de kibővíthető



ContentDialog (2)

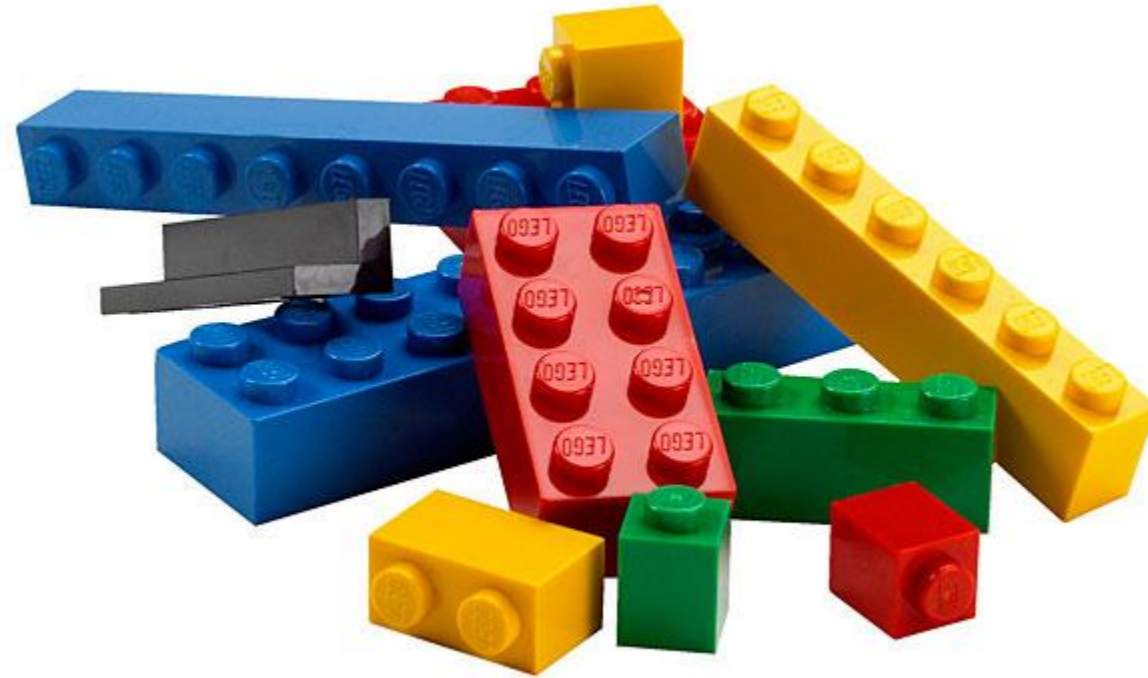
- A projekthez hozzá kell adni egy új „Content Dialog” típusú elemet
- A létrejött osztályt ízlés szerint szerkeszthetjük:
 - Paraméterezhetjük
 - A bezárás típusára használhatunk pl. saját felsorolás típust



Demo

- DialogSamples

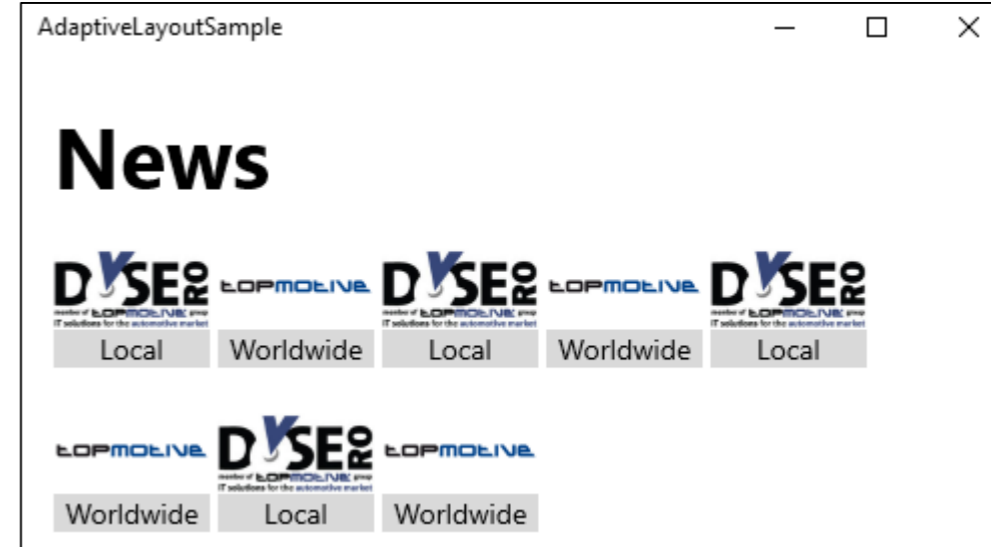




ADAPTÍV ELRENDEZÉS



Miért is adaptív?



VisualStateManager felépítése

- A XAML kódon belül kell elhelyezni

```
<VisualStateManager.VisualStateGroups>  
  <VisualStateGroup>  
    <VisualState x:Name="LargeScreen">  
      ...  
    </VisualState>  
    <VisualState x:Name="SmallScreen">  
      ...  
    </VisualState>  
  </VisualStateGroup>  
</VisualStateManager.VisualStateGroups>
```



VisualState felépítése

```
<VisualState x:Name="LargeScreen">  
  <VisualState.StateTriggers>  
    <AdaptiveTrigger MinWindowWidth="0"/>  
  </VisualState.StateTriggers>  
  <VisualState.Setters>  
    <Setter Target="TitleTextBlock.FontSize" Value="12"/>  
  </VisualState.Setters>  
</VisualState>
```



Kivételes esetek

- Egy lista elemeinek az újraméretezéséhez szükséges lépések
 - A **DataTemplate** tartalma külön **UserControl**-ként definiált elem kell legyen
 - A **VisualStateManager**-t a **UserControl**-on kell definiálni
 - A **UserControl** konstruktorában fel kell iratkozni a **DataContextChanged** eseményre:

```
this.DataContextChanged += (s, e) => this.Bindings.Update();
```



Eszközspecifikus oldalak – 1. lehetőség

- A *ProfilePage* nevű oldalt szeretnénk átrendezni
- A projektben létrehozunk egy új mappát: **DeviceFamily-<Típus>**, ahol a **<Típus>** lehet:
 - Mobile, Desktop, IoT, Xbox, ...
- Ebbe a mappába létrehozunk egy **Blank View** típusú oldalt az eredeti oldal nevével (*ProfilePage.xaml*)
- Ez az oldal az eredeti oldal code behind fájlját használja



Eszközspecifikus oldalak – 2. lehetőség

- A *ProfilePage* nevű oldalt szeretnénk átrendezni
- A projektben létrehozunk egy új Blank View típusú oldalt:
 - **<EredetiOldalNeve>.DeviceFamily-<Típus>.xaml**



Demo

- AdaptiveLayoutSample





TOPMOTIVE[®]
EXPERIENCING BENCHMARKS



STÍLUSOK ÉS TÉMA

DVSE **MSD**
as

AWDOC **ALDOC**

TOPMOTIVE
data

TOPMOTIVE
SYSTEMS

performance

Erőforrás szótár

```
<ResourceDictionary
  xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml">

  <SolidColorBrush x:Key="PrimaryAppColor" Color="Green"/>
  <SolidColorBrush x:Key="SecondaryAppColor" Color="Yellow"/>

  <Style x:Key="TitleTextBlockStyle" TargetType="TextBlock">
    <Setter Property="FontSize" Value="48" />
    <Setter Property="HorizontalAlignment" Value="Center" />
    <Setter Property="FontWeight" Value="Bold"/>
  </Style>
</ResourceDictionary>
```



Összefésült erőforrás szótárak

```
<Application x:Class="FirstUwpApp.App">
  <Application.Resources>
    <ResourceDictionary>
      <ResourceDictionary.MergedDictionaries>
        <ResourceDictionary Source="BrushesDictionary.xaml"/>
        <ResourceDictionary Source="CustomStylesDictionary.xaml"/>
      </ResourceDictionary.MergedDictionaries>
    </ResourceDictionary>
  </Application.Resources>
</Application>
```

- Használat:

```
<Border Background="{StaticResource PrimaryAppColor}">
  <TextBlock Text="Home Page"
    Style="{StaticResource TitleTextBlockStyle}"/>
</Border>
```

Applikáció téma

- Újdonságok:
 - ResourceDictionary.ThemeDictionaries

```
<ResourceDictionary>  
  <ResourceDictionary.ThemeDictionaries>  
    <ResourceDictionary x:Key="Dark" Source="DarkThemeDicitonary.xaml"/>  
    <ResourceDictionary x:Key="Light" Source="LightThemeDicitonary.xaml"/>  
  </ResourceDictionary.ThemeDictionaries>  
</ResourceDictionary>
```

- ThemeResource

```
<Grid Background="{ThemeResource ApplicationPageBackgroundThemeBrush}">
```



Beépített erőforrások – Színek, ecsetek

- Színek világos és sötét témához:
 - SystemAlt...
 - SystemBase...
 - SystemChrome...
 - SystemList...
- Színek magas kontrasztú témához:
 - SystemColor...
- Ecsetek:
 - SystemControl...



Beépített erőforrások – Szövegstílusok

Header

SubHeader

Title

SubTitle

Base

Body

Caption

- HeaderTextBlockStyle
- SubheaderTextBlockStyle
- TitleTextBlockStyle
- SubtitleTextBlockStyle
- BaseTextBlockStyle
- BodyTextBlockStyle
- CaptionTextBlockStyle



Referencia

- Erőforrás szótárak:
 - <https://msdn.microsoft.com/en-us/windows/uwp/controls-and-patterns/resourcedictionary-and-xaml-resource-references>
- Beépített témák:
 - <https://msdn.microsoft.com/en-us/windows/uwp/controls-and-patterns/xaml-theme-resources>





ADATTÁROLÁS



Bevezető

Applikáció adatok (App data)

- Az applikáció tulajdona
- Nem megosztott
- Törlődik az applikáció törlésekor
- 3 típus:
 - Lokális (Local)
 - Szinkronizált (Roaming)
 - Ideiglenes (Temporary)
- 2 tárolási lehetőség:
 - Beállítások
 - Állományok

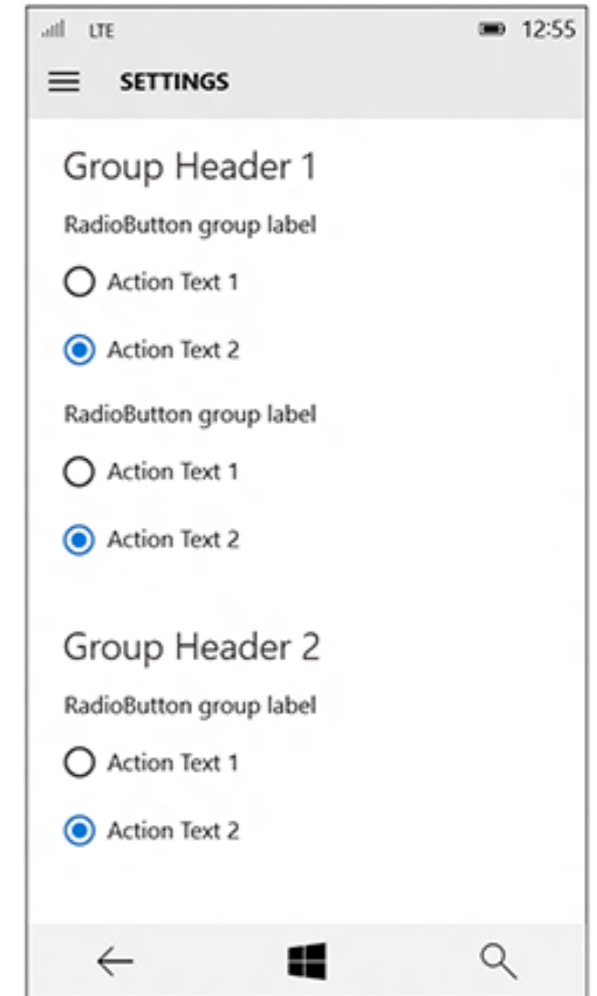
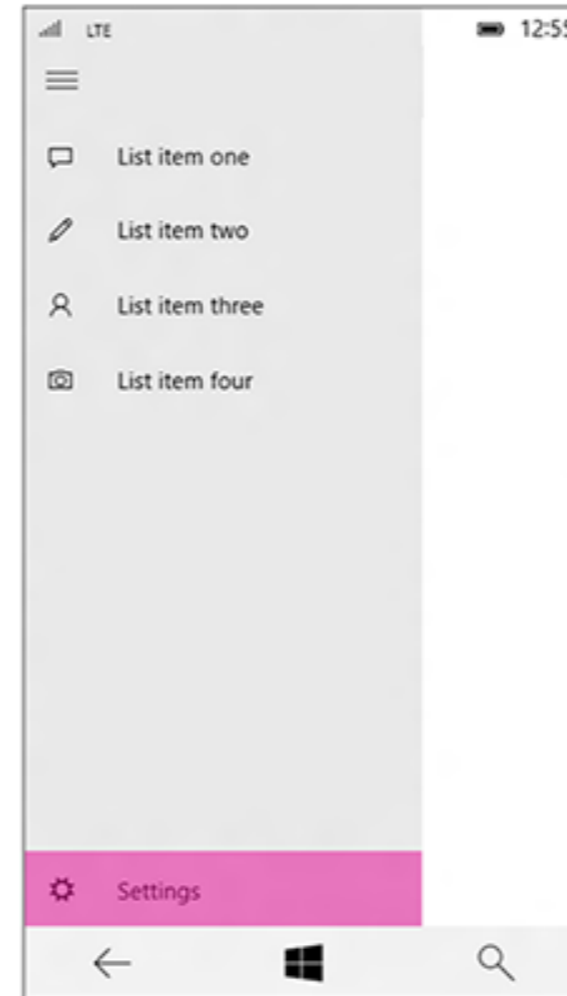
Felhasználó adatok (User data)

- A felhasználó tulajdona
- Megosztott
- Nem törlődik az applikáció törlésekor



Beállítások oldal

- Minél egyszerűbb legyen:
 - On-off kapcsolók
 - Rádió gombok
- A legutolsó menüpont legyen
- „A kevesebb néha több”



Beállítások API (1)

- A beállításokat az `ApplicationData.Current.LocalSettings` segítségével mentjük el és olvassuk ki:

```
ApplicationDataContainer settings = ApplicationData.Current.LocalSettings;
```

```
settings.Values["LanguageCode"] = "en";
```

```
object langCode = settings.Values["LanguageCode"];
```

- Beállítás törlése: `settings.Values.Remove("LanguageCode")`

- Nem tárolható bármi, csak numerikus, bool, dátum, **Guid**, **Point**, **Size**, **Rect**, **ApplicationDataCompositeValue** típusok



Beállítások API (2)

- Beállítások csoportosítása:

```
ApplicationDataContainer settings = ApplicationData.Current.LocalSettings;
```

```
ApplicationDataContainer profileSettings =  
    settings.CreateContainer("ProfileSettings", ApplicationDataCreateDisposition.Always);  
// ugyanúgy használható mint a settings
```

```
settings.DeleteContainer("ProfileSettings");
```

- Beállítások szinkronizálása több eszköz között:

```
ApplicationDataContainer roamingSettings = ApplicationData.Current.RoamingSettings;
```



Állományok API

- Az applikáció adatokat tárolhat az `ApplicationData.Current.LocalFolder` könyvtárban
- Fontosabb fájlrendszer műveletek:

```
StorageFolder localFolder = ApplicationData.Current.LocalFolder;  
// állomány létrehozás/törlés  
StorageFile file = await localFolder.CreateFileAsync("filename");  
await file.DeleteAsync();  
// könyvtár létrehozás/törlés  
StorageFolder other = await localFolder.CreateFolderAsync("otherFolder");  
await other.DeleteAsync();  
// állományok keresése  
IReadOnlyList<StorageFile> files = await localFolder.GetFilesAsync(CommonFileQuery.OrderByTitle);  
files.Where(f => f.ContentType == "audio/mpeg");
```



Serializálás példa

```
public class NewsItem
{
    public string Id { get; set; }
    public string Headline { get; set; }
    public string Content { get; set; }
    public DateTimeOffset LastEdited { get; set; }
}

List<NewsItem> newsItemsList = ...;
StorageFile file = await localFolder.CreateFileAsync("newsitems.json");
DataContractJsonSerializer ser = new DataContractJsonSerializer(typeof(List<NewsItem>));
// szerializálás
using (Stream stream = await file.OpenStreamForWriteAsync())
{
    ser.WriteObject(stream, newsItemsList);
}
// deszerializálás
using (Stream stream = await file.OpenStreamForReadAsync())
{
    newsItemsList = ser.ReadObject(stream) as List<NewsItem>;
}
```



Ideiglenes adatok

- Cache-elésre ajánlott használni
- Nem szinkronizálódnak
- Bármikor törölhető
 - Akár a felhasználó által (Disk Cleanup)
 - Akár a rendszer által

- API:

```
StorageFolder tempFolder = ApplicationData.Current.TemporaryFolder;
```





TOPMOTIVE®
EXPERIENCING BENCHMARKS



DVSE **MSD**
as

AWDOC **ALDOC**

TOPMOTIVE
data

TOPMOTIVE
SYSTEMS

performance

BEFEJEZÉS

Web Service kommunikáció - dióhéjban

- A **System.Net.Http** névteret lehet használni
- Egy fiktív REST API-hoz egy fiktív POST kérés, mely létrehozhat egy új `NewItem`-et

```
HttpClient client = new HttpClient();
```

```
NewItem newItem = new NewItem { Headline = "test" };
```

```
DataContractJsonSerializer ser = new DataContractJsonSerializer(typeof(NewItem));
```

```
string jsonString = null;
```

```
using (MemoryStream stream = new MemoryStream())
```

```
{
```

```
    ser.WriteObject(stream, newItem);
```

```
    jsonString = Encoding.UTF8.GetString(stream.ToArray());
```

```
}
```

```
StringContent content = new StringContent(jsonString, Encoding.UTF8, "application/json");
```

```
await client.PostAsync("http://www.dummynewsrestapi.com/newsitems", content);
```

Eszközcsalád beállítása

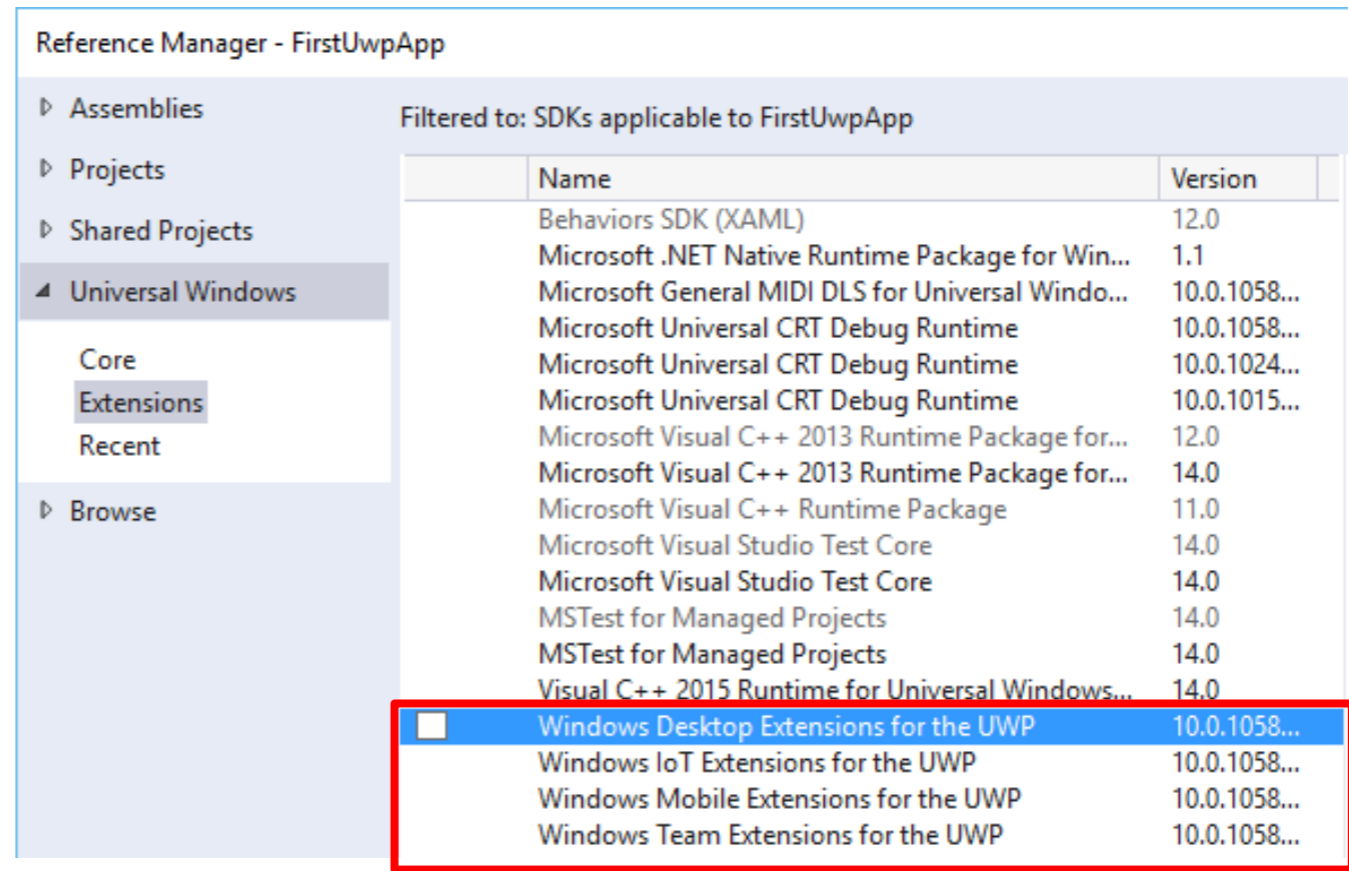
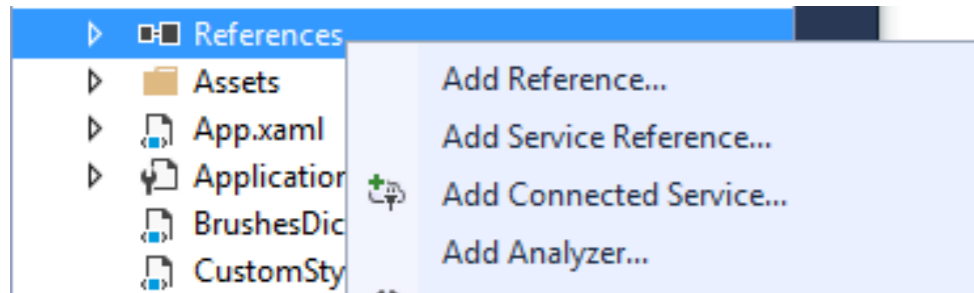
- Package.appxmanifest
 - Kézzel editálható
 - Windows.Mobile
 - Windows.Desktop
 - Windows.Universal

```
<Package>  
  <Dependencies>  
    <TargetDeviceFamily Name="Windows.Mobile"  
      MinVersion="10.0.0.0"  
      MaxVersionTested="10.0.0.0" />  
  </Dependencies>  
</Package>
```



Adaptív kód írása (1)

- Kibővítés SDK (Extensions SDK) hozzáadása



Adaptív kód írása (2)

- MSDN dokumentáció példa:

HardwareButtons.CameraPressed | camerapressed event

Occurs when the user presses the hardware camera button.

Syntax

JavaScript

C#

C++

VB

```
public static event EventHandler<CameraEventArgs> CameraPressed
```

Requirements (Windows 10 device family)

Device family	Mobile, introduced version 10.0.10240.0
API contract	Windows.Phone.PhoneContract, introduced version 1.0



Adaptív kód írása (3)

- Kódpélda:

```
bool isHardwareButtonsAPIPresent =  
    ApiInformation.IsTypePresent("Windows.Phone.UI.Input.HardwareButtons");  
  
if (isHardwareButtonsAPIPresent)  
{  
    Windows.Phone.UI.Input.HardwareButtons.CameraPressed  
        += HardwareButtons_CameraPressed;  
}
```

- Egyéb **ApiInformation** metódusok:

```
ApiInformation.IsApiContractPresent("Windows.Phone.PhoneContract", 1, 0);  
ApiInformation.IsMethodPresent("TypeName", "MethodName");  
ApiInformation.IsPropertyPresent("TypeName", "PropertyName");
```



Hasznos linkek

- <https://channel9.msdn.com/Series/Windows-10-development-for-absolute-beginners>
- <https://channel9.msdn.com/Events/Windows/Developers-Guide-to-Windows-10-RTM>
- <https://dev.windows.com/en-us>
- UWP keretrendszer:
 - <https://github.com/PrismLibrary/Prism>
 - <https://github.com/Windows-XAML/Template10/wiki>





KÉRDÉSEK



Köszönöm a figyelmüket!

