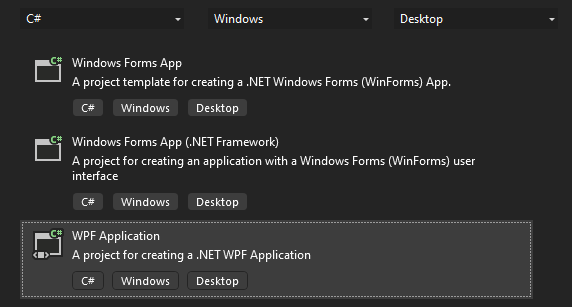
XAML Windows alkalmazás készítése

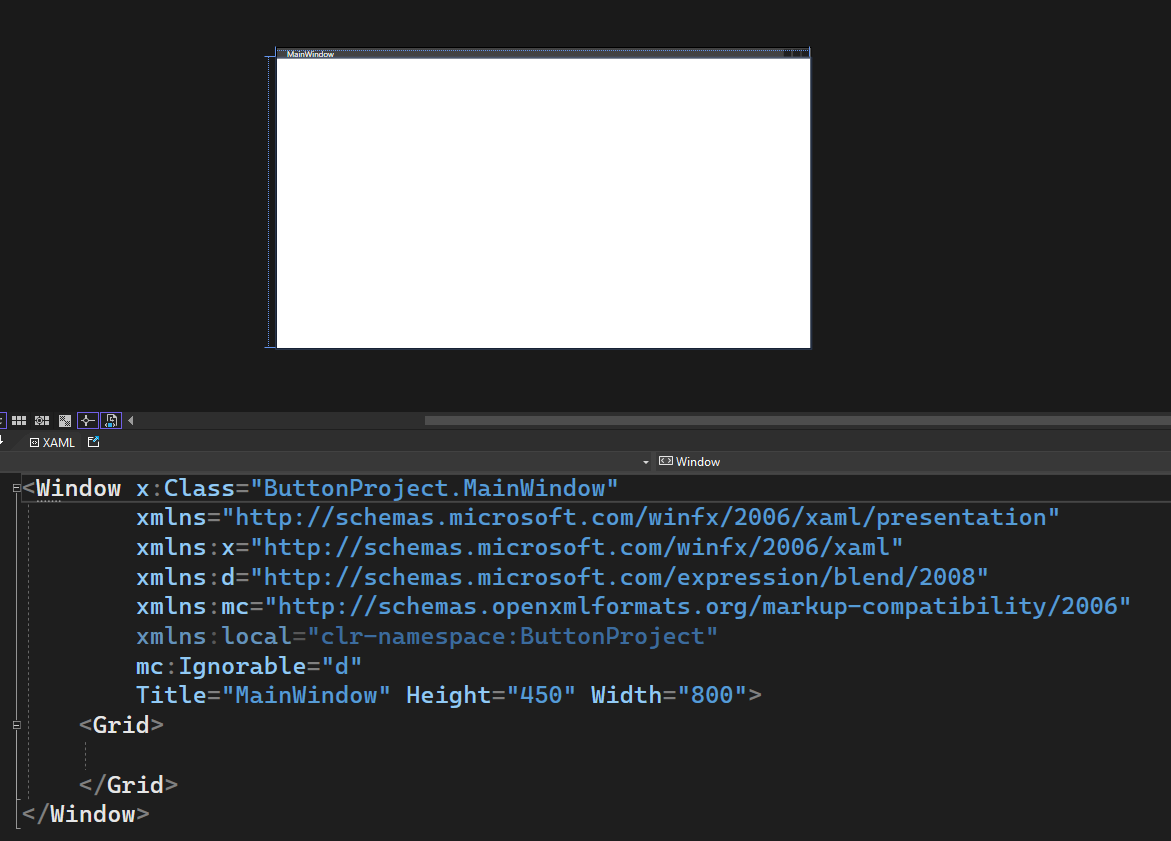
* [XAML – Wikipédia (wikipedia.org)](https://hu.wikipedia.org/wiki/XAML)
* [Eseményvezérelt alkalmazások fejlesztése - XAML](http://sztrabi.web.elte.hu/wp-content/EVA2/lecture06/lecture06#/step-3)
* [WPF - XAML Overview (tutorialspoint.com)](https://www.tutorialspoint.com/wpf/wpf_xaml_overview.htm)

# Gombok

1. Készítsen WPF alkalmazást ButtonProject néven



1. A XAML kódot a MainWindow.xaml állományba írjuk. Közben követhetjük az ablak kinézetét:



[Eseményvezérelt alkalmazások fejlesztése 2 - Window](http://sztrabi.web.elte.hu/wp-content/EVA2/lecture06/lecture06#/step-2)

Ablak címe és ablak mérete a Window tagen belül.

<Window x:Class="\_01\_01\_wpf\_button.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:\_01\_01\_wpf\_button"

mc:Ignorable="d"

**Title="Helló világ program" Height="250" Width="400"**>

## Stack Panel

[Eseményvezérelt alkalmazások fejlesztése 2 - StackPanel](http://sztrabi.web.elte.hu/wp-content/EVA2/lecture07/lecture07.html#/step-5)

A Grid helyett használjunk StackPanel réteget.

A gyerekelemeit egymásután veremszerűen helyezi el. Felsorolásokra nagyon jól használható. Most függőleges lesz az iránya.

## Button

Elhelyezünk két gombot és egy szövegmezőt.

Az x:Name attribútum bármilyen objektumon megadható. A mögöttes kódban a megadott x:Name azonosítóval megegyező változóban lesz elérhető a példányosított objektum

[Prognyelvek portál (elte.hu) – x:Name](http://nyelvek.inf.elte.hu/leirasok/XAML/index.php?chapter=4#section_2)

A Content tulajdonság objektum, ezért lehet egy egyszerű szöveg, de akár egy színátmenetes kitöltésű ellipszis is.

Height, width és margin tulajdonságok értelemszerűen használhatók. A margó lehet minden irányban ugyan annyi, vagy irányomként más.

<Button x:Name="btnHello" Content="Kattints rám" Height="30" Width="120" Margin="20 10 5 20" >

Gomb eltüntetése

A gombot eltüntetjük, majd a háttérkódban fogjuk megjeleníteni és eltüntetni.

<Button x:Name="btnDeleteText" Content="Szöveg törlése" **Visibility**="Hidden" Height="40" Width="120" Margin="20 10 5 20">

## TextBlock

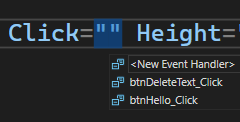
Itt jelenik meg az üzenet:

[A TextBlock vezérlő - Sorok formázása - The complete WPF tutorial (wpf-tutorial.com)](https://wpf-tutorial.com/hu/15/alap-vezerlok/a-textblock-vezerlo-sorok-formazasa/)

<TextBlock x:Name="txtHello" HorizontalAlignment="Center" >Itt jelenik meg az üzenet...</TextBlock>

A gombokhoz események rendelhetők.

Pl. a rákattintás (Click) esemény. Az eseményhez pedig a háttérben futó kód rendelhető.



<Button x:Name="btnHello" Content="Kattints rám" **Click**="btnHello\_Click" Height="30" Width="120" Margin="20 10 5 20" >

A MainWindows.xaml állományban elkészül a btnHello\_Click esemény. Ezt a kódot most a MainWindows.xaml.cs állományba kell írni. Ott folytatjuk…

A Click eseményhez rendelt kód

* A txtHello TextBlock feliratát (Text) megváltoztatjuk.
* A btnDeleteText gombot megjeleníti.

private void btnHello\_Click(object sender, RoutedEventArgs e)

{

txtHello.Text = "Helló világ!";

btnDeleteText.Visibility = Visibility.Visible;

}

Click esemény a másik gombhoz is:

<Button x:Name="btnDeleteText" Content="Szöveg törlése" Click="btnDeleteText\_Click" Visibility="Hidden" Height="40" Width="120" Margin="20 10 5 20">

A másik Click eseményhez rendelt kód:

* A txtHello TextBlock szövegét törli.
* A btnDeleteText gombot eltünteti.

private void btnDeleteText\_Click(object sender, RoutedEventArgs e)

{

txtHello.Text = string.Empty;

btnDeleteText.Visibility = Visibility.Hidden;

}

Gomb stílus

Stílus, amely a gombok háttérszínét megváltoztatja, ha a gombra rávisszük az egeret.

btnHello gomb:

<Button x:Name="btnHello" Content="Kattints rám" Click="" Height="30" Width="120" Margin="20 10 5 20" >

**<Button.Style>**

**<Style TargetType="{x:Type Button}">**

**<Setter Property="Template">**

**<Setter.Value>**

**<ControlTemplate TargetType="{x:Type Button}">**

**<Border Name="border" BorderThickness="1" Padding="4,2" BorderBrush="DarkGray" CornerRadius="3" Background="{TemplateBinding Background}">**

**<ContentPresenter HorizontalAlignment="Center" VerticalAlignment="Center" Name="content"/>**

**</Border>**

**</ControlTemplate>**

**</Setter.Value>**

**</Setter>**

**<Style.Triggers>**

**<Trigger Property="IsMouseOver" Value="True">**

**<Setter Property="Background" Value="Yellow"/>**

**</Trigger>**

**</Style.Triggers>**

**</Style>**

**</Button.Style>**

</Button>

Sablon, amely elipszis alakú gombot készit. Változik a gomb ha rávisszük az egeret vagy megnyomjuk a gombot!

btnDeleteText:

<Button x:Name="btnDeleteText" Content="Szöveg törlése" Click="btnDeleteText\_Click" Visibility="Hidden" Height="40" Width="120" Margin="20 10 5 20">

**<Button.Template>**

**<ControlTemplate TargetType="{x:Type Button}">**

**<Border x:Name="bdr\_main" CornerRadius="20" Margin="4" BorderThickness="1" BorderBrush="Black" Background="LightGray">**

**<ContentPresenter VerticalAlignment="Center" HorizontalAlignment="Center" Margin="8,6,8,6" ContentSource="Content" />**

**</Border>**

**<ControlTemplate.Triggers>**

**<Trigger Property="IsMouseOver" Value="True">**

**<Setter TargetName="bdr\_main" Property="Background" Value="LightGreen"/>**

**</Trigger>**

**<Trigger Property="IsPressed" Value="True">**

**<Setter TargetName="bdr\_main" Property="Background" Value="Red"/>**

**</Trigger>**

**</ControlTemplate.Triggers>**

**</ControlTemplate>**

**</Button.Template>**

</Button>

Futó program:

