

Cheatsheet : Environnement de développement de logiciels

Border : Permet d'encadrer un sous-élément dans une bordure.

```
<Border BorderThickness Background BorderBrush>
  <StackPanel>
  </StackPanel>
</Border>
```

BorderBrush définit la couleur extérieure de la bordure.

<Button click="Button_Click" >Nom</Button>

Grid (grille, tableau): <Grid Background>

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

Autres éléments, par exemple :

```
<TextBlock Grid.Row = "2">blabla</TextBlock>
</Grid>
```

Les indices commencent à 0, et si on ne le précise pas (pour le TextBlock par exemple), le 0 est pris par défaut. Le fait d'utiliser 2* dans les tailles des définitions s'appelle le starsizing. On peut également taper Auto.

Par défaut, les éléments ne s'affichent pas sur plusieurs cellules. Pour régler cela, on utilise la propriété Grid.ColumnSpan et Grid.RowSpan.

HyperLinkButton : Il s'agit simplement d'un lien et pas vraiment d'un bouton.

```
<HyperlinkButton Content="texte" Click="méthode" NavigateUri="lien" />
```

Rectangle : <Rectangle Fill Height Width HorizontalAlignment VerticalAlignment Margin/>

RelativePanel : Permet de placer des éléments à l'intérieur de façon relative, c'est à dire les uns par rapport aux autres.

```
<RelativePanel MinHeight>
  <Rectangle Name="RedRectangle" Fill Width ...
  RelativePanel.AlignRightWithPanel="True" />
```

```
<Rectangle ... RelativePanel.LeftOf="RedRectangle"/>
</RelativePanel>
```

Il y a tout un tas de propriétés, comme :

- `RelativePanel.AlignRightWith...`
- `RelativePanel.AlignVerticalCenterWith="RedRectangle"`
- `RelativePanel.AlignHorizontalCenterWithPanel="True"`
- `RelativePanel.AlignBottomWithPanel="True"`
- `RelativePanel.AlignTopWith="RedRectangle"`
- `RelativePanel.Below="RedRectangle"`
- etc, etc

SplitView (menu déroulant sur la gauche) :

```
<SplitView CompactPaneLength OpenPaneLength
    DisplayMode="CompactInline">
```

```
    <SplitView.Pane><StackPanel><TextBlock>...</SplitView.Pane>
    <SplitView.Content><StackPanel><TextBlock>...</SplitView.Content>
</SplitView>
```

StackPanel (comme un div) : <StackPanel *Orientation* *VerticalAlignment*> </StackPanel>

On ne peut avoir qu'un seul StackPanel dans une page, mais il est assez facile de palier à ce problème en mettant plusieurs StackPanel dans ce StackPanel !

CODE :

SplitView :

```
private void Button_StackPanel(object sender, RoutedEventArgs e)
{
    MySplitView.IsPaneOpen = !MySplitView.IsPaneOpen;
}
```

—> MySplitView est l'attribut 'Name' de la SplitView dans le code XAML

Navigation : Il faut créer un bloc <Frame Name="MyFrame"></Frame> dans la MainPage, ainsi que d'autres pages XAML. Ensuite, rien de plus simple !

```
private void Navigation(object sender, RoutedEventArgs e)
{
    MyFrame.Navigate(typeof(Page1));
    MyFrame.GoBack();
    MyFrame.GoForward();
}
```