

# Chapitre IV

Xamarin.Forms (Notions de base) (I)









### Plan

- Xamarin.Forms XAML
- Views (contrôles) et Cellules
- Views
- 4 Layouts







#### 1. Xamarin.Forms XAML

- Deux façons pour construire des interfaces graphiques utilisateur (GUI) en Xamarin
  - En utilisant C# dans le fichier source (.cs)
  - En utilisant XAML dans le fichier d'interface (.xaml)
- Il est préférable de définir les GUI moyennant XAML
  - Séparation présentation/traitement
  - Plus facile à comprendre et à maintenir
  - Possibilité d'aperçu (preview) sans exécution complète du code
- eXtensible Application Markup Language
  - Totalement XML



mobiles

Dév. d'applications



### 1. Xamarin.Forms XAML

XAML: interfaces multi-plateformes

```
<ContentPage>
   <StackLayout Spacing="20" Padding="50"</pre>
   VerticalOptions="Center">
        <Entry Placeholder="User name" />
        <Entry Placeholder="Password" IsPassword="True" />
        <Button Text="Login" TextColor="White"</pre>
               BackgroundColor="##FF77D065" />
   </StackLayout>
</ContentPage>
```



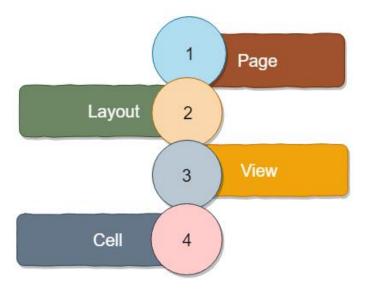






### 1. Xamarin.Forms XAML

- Structure d'une interface XAML
  - Page
    - Layout
      - Views (contrôles)
        - cellules







# 2. Views (contrôles) et Cellules

 Les contrôles sont de type Button, Label, WebView et de cellule (utilisables dans ListView et TableView)

ActivityIndic ator	BoxView	Button	DatePicker	Editor
Entry	Image	Label	ListView	Мар
OpenGLVie w	Picker	ProgressBar	SearchBar	Slider
Stepper	Switch	TableView	TimePicker	WebView
EntryCell	ImageCell	SwitchCell	TextCell	ViewCell



d'applications mobiles



App1.Android

Properties ■·■ References

Connected Services

## 2. Views: Image et Label

Exemple: Image et Label

```
Resources
                                                     🔺 📹 drawable
<Image Source="XamarinLogo"</pre>
                                                        XamarinLogo.png
   Aspect="AspectFit"
                                                     layout
   HeightRequest="{OnPlatform iOS=300, Android=250}"
   WidthRequest="{OnPlatform iOS=300, Android=250}"
   HorizontalOptions="Center" />
<Label Text="Ceci est une image"</pre>
   TextColor="Blue"
   FontAttributes="Italic"
   FontSize="22"
   TextDecorations="Underline"
   HorizontalOptions="Center" />
```





### 2. Views: WebView

Exemple : WebView

<WebView Source="http://cyberawareness.tn"
 MinimumWidthRequest="200"
 MinimumHeightRequest="200"
 HorizontalOptions="FillAndExpand"
 VerticalOptions="FillAndExpand" />



### Welcome to Cyber Awareness Tunisia!



\*\*Our mission is to promote safe and responsible use of digital devices. We seek to ensure Tunisians' cyber welfare for a better online experience.\*\*

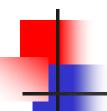
- About us, Cyber Awareness Tunisia







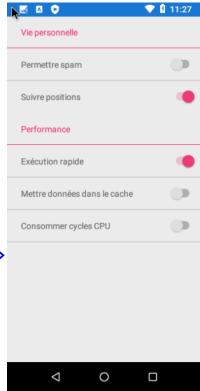




### 2. Views: TableView et SwitchCell

Exemple : TableView et SwitchCell

```
TableView>
   TableView.Root>
    <TableSection Title="Vie personnelle">
d'applications
      <SwitchCell Text="Permettre spam" />
      <SwitchCell Text="Suivre positions" On="True" />
    </TableSection>
    <TableSection Title="Performance">
      <SwitchCell Text="Exécution rapide" On="True" />
      <SwitchCell Text="Mettre données dans le cache" />
      <SwitchCell Text="Consommer cycles CPU" />
    </TableSection>
     FableView.Root>
   ableView>
```







article1

article3

## 2. Views: ListView simple

- Exemple : ListView simple
  - Dans MainPage.xaml

```
<ListView x:Name="lst"></ListView>
```

- Dans MainPage.xaml.cs
  - A l'intérieur du constructeur de MainPage

Solution Explorer

∠ C# App1

🔚 Solution 'App1' (3 of 3 projects)

C# MainPage.xaml.cs

Dependencies

App.xaml
C\* AssemblyInfo.cs
MainPage.xaml





## 2. Views: ListView avec valeurs attachées (1/2)

- Exemple (1/2): ListView avec valeurs attachées
  - Dans MainPage.xaml

```
ListView x:Nam
<ListView.Item
<DataTemplate
</DataTemplate
</pre>

/ListView.Item

/ListView.Item

/ListView

Initial
Ist.It
I
                                    <ListView x:Name="lst">
                                            <ListView.ItemTemplate>
                                                     <DataTemplate>
                                                              <TextCell Text="{Binding Nom}" Detail="{Binding Num}" >
                                                     </DataTemplate>
                                            </ListView.ItemTemplate>
                                                                                                                                                                        Dans MainPage.xaml.cs

    A l'intérieur du constructeur de MainPage
```



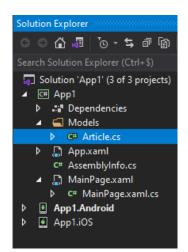


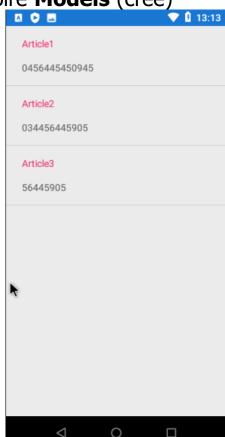
# 2. Views : ListView avec valeurs attachées (2/2)

Exemple (2/2): ListView avec valeurs attachées

On ajoute aussi une classe Article.cs dans un répertoire Models (crée)

```
class Article
    {
        public string Nom { get; set; }
        public string Num { get; set; }
```









## 2. Views: ListView avec valeurs + images attachées (1/2)

- Exemple (1/2): ListView avec valeurs + images attachées
  - Dans MainPage.xaml

```
<ListView x:Name="lst">
   <ListView.ItemTemplate>
    <DataTemplate>
     <ImageCell Text="{Binding Nom}" Detail="{Binding Num}" ImageSource="{Binding</pre>
    </DataTemplate>
   </ListView.ItemTemplate>
```

A l'intérieur du constructeur de MainPage

```
InitializeComponent();
lst.ItemsSource = new List<Article>() {
   new Article() { Nom = "Article1", Num = "0456445450945", Image = "Xam1", },
   new Article() { Nom = "Article2", Num = "034456445905", Image = "Xam2", },
   new Article() { Nom = "Article3", Num = "56445905", Image = "Xam3", },
```





# 2. Views: ListView avec valeurs + images attachées (2/2)

- Exemple (2/2): ListView avec valeurs + images attachées
  - On ajoute aussi une classe Article.cs dans un répertoire Models (crée)

```
class Article
    {
        public string Nom { get; set; }
        public string Num { get; set; }
        public string Image { get; set; }
```

}

Dév. d'applications mobiles

```
Solution 'App1' (3 of 3 projects)
Dependencies
    Models
    C# Article.cs
  App.xaml
     C# AssemblyInfo.cs

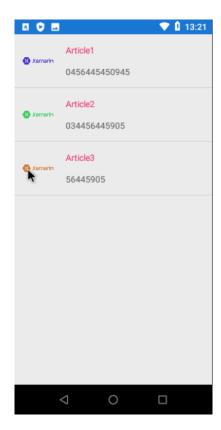
    MainPage.xaml

    C# MainPage.xaml.cs

    App1.Android
     Connected Services
     Properties
    ■·■ References
    Resources
     drawable
          Xam1.png

    Xam2.png

✓ Xam3.png
```



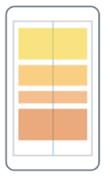




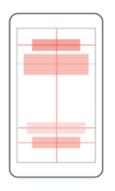


## 3. Layouts

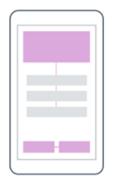
- Layouts?
  - Contrôles pouvant contenir d'autres contrôles!
  - Fournissent :
    - Mise en forme
    - Positionnement



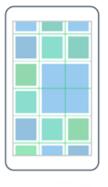
StackLayout



AbsoluteLayout



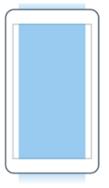
RelativeLayout



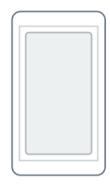
Grid



ContentView



ScrollView



Frame





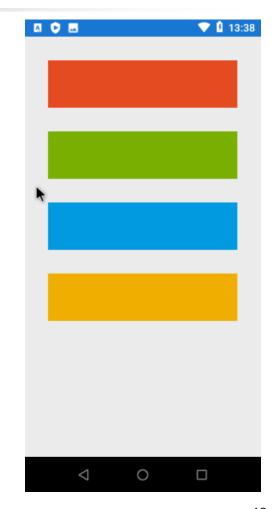
### 3. Layouts: StackLayout

- Organise les contrôles sous forme d'un tas (seule direction)
  - StackOrientation
    - Vertical (défaut) / Horizontal
  - Spacing (double)
    - espace entre contrôles (défaut = 6 device independant units)
- Autre

Dév. d'applications mobiles

- HorizontalOptions
  - Start, Center, End, Fill (+ AndExpand)

```
<StackLayout Padding="32" Spacing="32">
  <BoxView Color="#FFF25022" HeightRequest="64" />
  <BoxView Color="#FF7FBA00" HeightRequest="64" />
  <BoxView Color="#FF01A4EF" HeightRequest="64" />
  <BoxView Color="#FFFFB901" HeightRequest="64" />
  </StackLayout>
```







## 3. Layouts: AbsoluteLayout

- Organise les contrôles avec positionnement direct (position et taille en device-independant units)
  - AbsoluteLayout.LayoutBounds (contrôles)
    - X, Y
    - Ou bien x, y, width, height
    - AbsoluteLayout.AutoSize pour le calcul automatique de largeur/hauteur

```
Stylish Header
```





# 3. Layouts: GridLayout

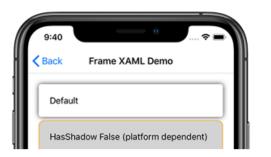
```
Grid Padding="32" RowSpacing="32" ColumnSpacing="32">
d'applications mobiles
           <Grid.RowDefinitions>
               <RowDefinition Height="*" />
               <RowDefinition Height="*" />
           </Grid.RowDefinitions>
           <Grid.ColumnDefinitions>
               <ColumnDefinition Width="*" />
               <ColumnDefinition Width="*" />
           </Grid.ColumnDefinitions>
           <BoxView Grid.Row="0" Grid.Column="0" Color="#FFF25022"</pre>
           <BoxView Grid.Row="0" Grid.Column="1" Color="#FF7FBA00"</pre>
           <BoxView Grid.Row="1" Grid.Column="0" Color="#FF01A4EF"</pre>
           <BoxView Grid.Row="1" Grid.Column="1" Color="#FFFFB901" />
   </Grid>
```





# 3. Layouts additionnels: Frame, ScrollView, et ContentView

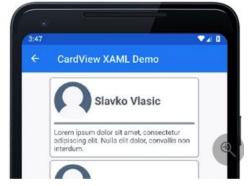
#### Frame





#### **ContentView**





#### **ScrollView**

