

# Chapitre V

## Xamarin.Forms (Notions de base) (II)





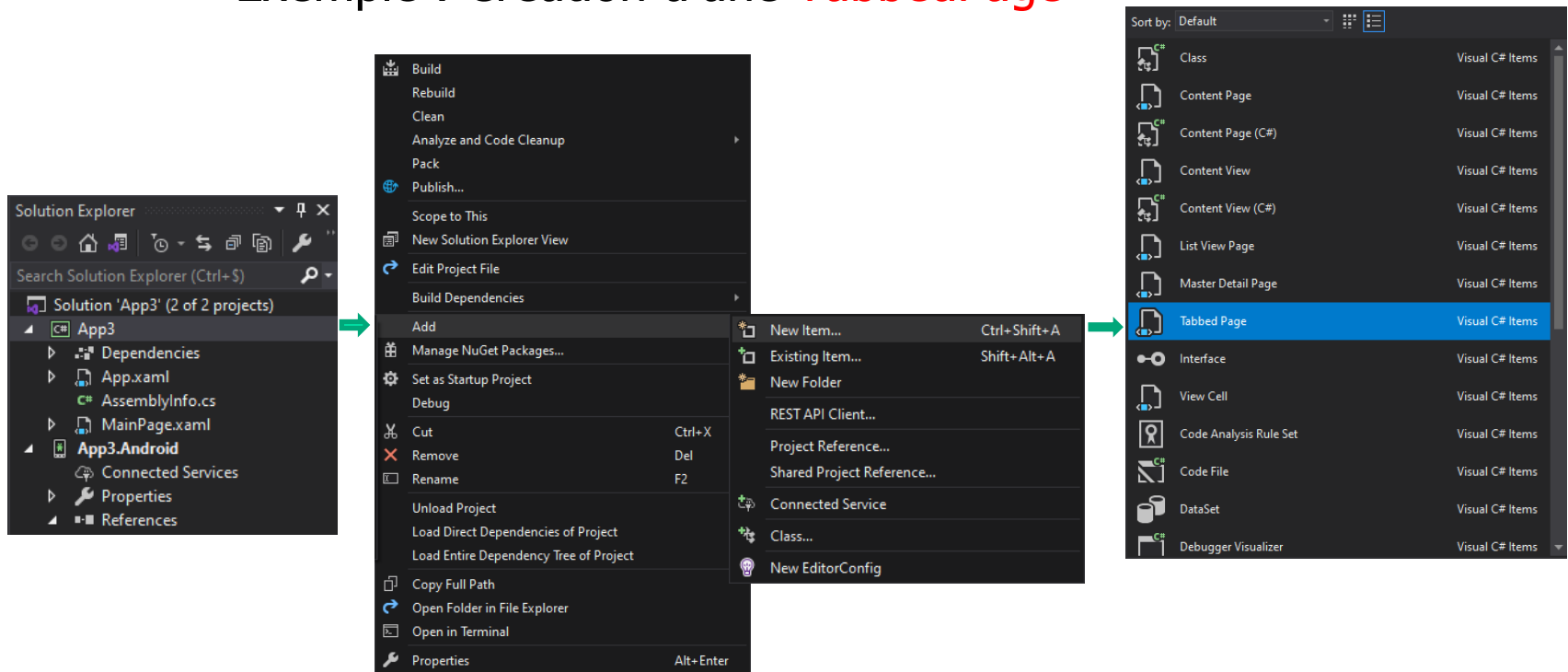
# Plan

- 1 Pages : Création
- 2 Pages : Types
- 3 Navigation



# 1. Pages : Création avec MS Visual Studio 2019 (1/3)

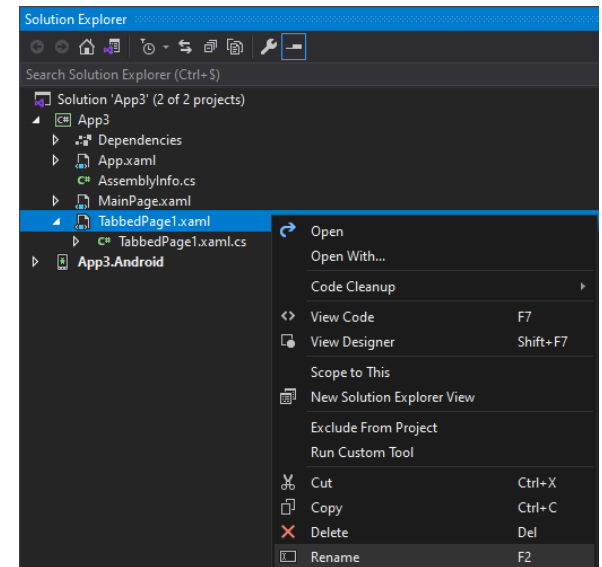
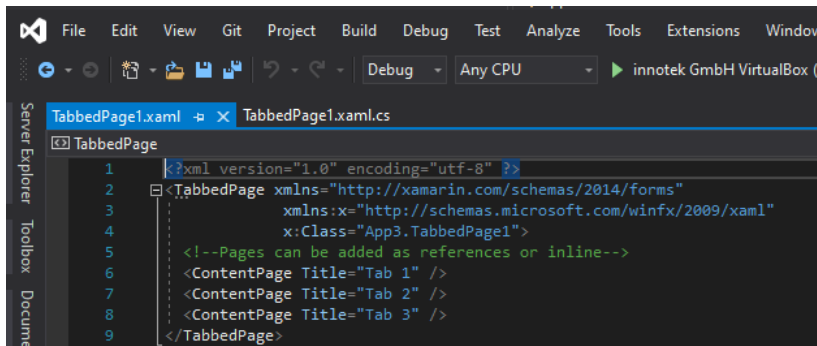
- Création **directe** à partir de MS Visual Studio 2019
  - possibilité de créer un dossier pour ces pages
  - Exemple : Création d'une **TabbedPage**





# 1. Pages : Création avec MS Visual Studio 2019 (2/3)

## ■ Résultat :



■ Il est possible de changer le nom de la page dans

« **Explorateur de solution** » →

- Cliquer avec **bouton droit** sur le **nom de la page**
- Commande « **Renommer** »
- Nouveau nom = « **PagePrincipale** »



# 1. Pages : Création avec MS Visual Studio 2019 (3/3)

- Déclarer « **PagePrincipale** » (TabbedPage) nouvellement créée **comme page principale**
- Dans <nom\_projet>.xaml.cs

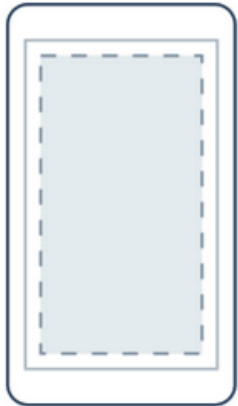
```
using System;
using Xamarin.Forms;
using Xamarin.Forms.Xaml;

namespace <nom_projet>
{
    public partial class App : Application
    {
        public App()
        {
            InitializeComponent();
            MainPage = new PagePrincipale();
        }
    }
}
```



## 2. Pages : Types

- Types de pages dans Xamarin.Forms



ContentPage



MasterDetailPage



NavigationPage



TabbedPage



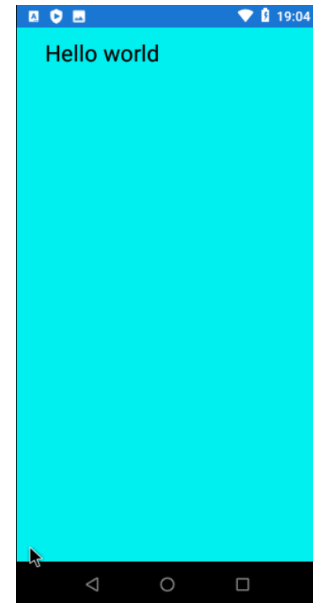
CarouselPage



## 2. Pages : ContentPage

- **ContentPage**
  - Page simple
  - La plus basique des pages
  - Destinée à accueillir principalement des layouts
- **Exemple : MainPage.xaml**

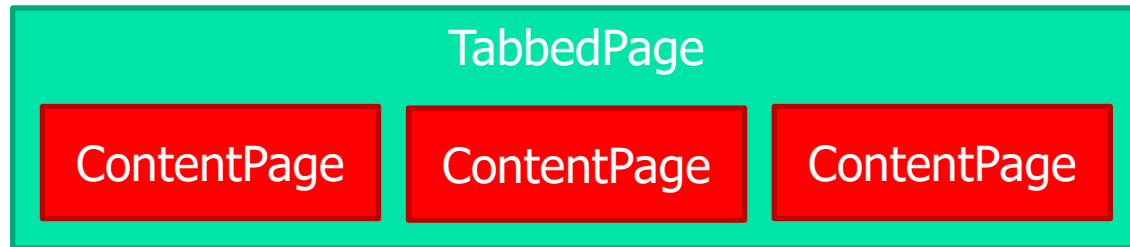
```
<?xml version="1.0" encoding="utf-8" ?>
<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
  xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
  x:Class="App3.MainPage"
  Title="Hello World"
  IconImageSource="Xam2">
  <StackLayout BackgroundColor="Cyan">
    <Label Text="Hello world" FontSize="Title"
      TextColor="Black" Padding="30,10,30,10"/>
  </StackLayout>
</ContentPage>
```



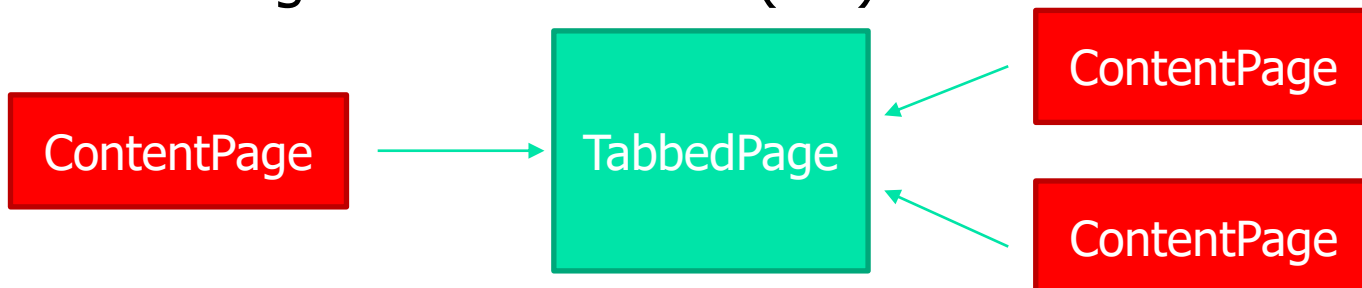


## 2. Pages : TabbedPage (1/3)

- **TabbedPage** : Une page contenant d'autres pages (de type **ContentPage** par exemple)
- Dans TabbedPage, les pages peuvent être
  - Déclarées en ligne (xaml)



- Chargées dans le code (C#)







## 2. Pages : TabbedPage (2/3)

- Exemple
- Déclaration en ligne dans PagePrincipale.xaml

```
<?xml version="1.0" encoding="utf-8" ?>
<TabbedPage xmlns="http://xamarin.com/schemas/2014/forms"
  xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
    x:Class="App3.PagePrincipale">
  <TabbedPage.Children>
    <ContentPage Title="Page1">
      ...
    </ContentPage>
    <ContentPage Title="Page2">
      ...
    </ContentPage>
    <ContentPage Title="Page3">
      ...
    </ContentPage>
  </TabbedPage.Children>
</TabbedPage>
```



## 2. Pages : TabbedPage (3/3)

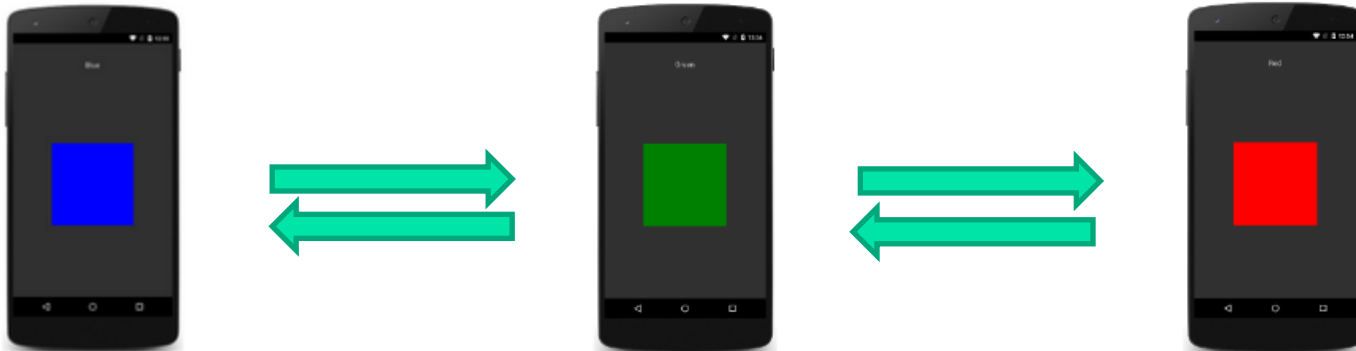
- Exemple
- Chargement de pages 1 à 3 dans PagePrincipale.xaml.cs
- PagePrincipale.xaml (TabbedPage)
  - Page1.xaml (ContentPage)
  - Page2.xaml (ContentPage)
  - Page3.xaml (ContentPage)

```
...
namespace <nom_projet>
{
    [XamlCompilation(XamlCompilationOptions.Compile)]
    public partial class PagePrincipale : TabbedPage
    {
        public PagePrincipale()
        {
            InitializeComponent();
            Children.Add(new Page1());
            Children.Add(new Page2());
            Children.Add(new Page3());
        }
    }
}
```



## 2. Pages : CarouselPage/CarouselView

- Création similaire à **TabbedPage**
- **Mais!**
  - La navigation se fait avec des gestes de type swipe right/left
  - Pas de « raccourcis » pour aller directement à une page donnée





## 3. Navigation

- Afin de se déplacer dans l'application
- => **Un système de navigation?**
  - Passage d'une page à une autre
  - Une forme d'hierarchie



# 3. Navigation avec NavigationPage (1/3)

- **NavigationPage** : permet d'établir une **pile de navigation**
  - Dans la pile, il y a un nombre de pages (de type **ContentPage** par exemple)
    - Pour **avancer** dans les pages : `PushAsync(new <page_suivante>())`
    - Pour **reculer** dans les pages : `PopAsync()`
    - Pour **revenir à la page racine** : `PopToRootAsync()`
  - Cette navigation est asynchrone! (càd ne bloque pas l'exécution du reste de l'application)
- **Exemple**
  - Application **App** (App.xaml + App.xaml.cs) avec deux pages
    - **Page1** (Page1.xaml + Page1.xaml.cs) => **page racine**
    - **Page2** (Page1.xaml + Page1.xaml.cs)
  - On souhaite naviguer entre **Page1** et **Page2** moyennant une **NavigationPage**
  - **Programmation en C# (dans les fichiers .cs)**
- **Implémentation? => Diapo suivant!**



## 3. Navigation avec NavigationPage (2/3)

- Dans App
  - App.xaml.cs
    - Dans le constructeur

```
public App()  
{  
    MainPage = new NavigationPage(new Page1()); //Page1 devient la racine  
}
```

- Dans Page1

- Page1.xaml

```
<Button x:Name="button" Text="Next Page" Clicked="OnNextPageButtonClicked"  
VerticalOptions="CenterAndExpand" />
```

- Page1.xaml.cs

```
async void OnNextPageButtonClicked(object sender, EventArgs e)  
{  
    await Navigation.PushAsync(new Page2());  
}
```



## 3. Navigation avec NavigationPage (3/3)

- Dans Page2

- Page2.xaml

```
<Button x:Name="button" Text="Root page" Clicked="OnRootPageButtonClicked"
VerticalOptions="CenterAndExpand" />
```

- Page2.xaml.cs

```
async void OnRootPageButtonClicked(object sender, EventArgs e)
{
    await Navigation.PopToRootAsync(new Page2());
}
```



## 3. Navigation « programmatique »

- D'une manière générale, le passage d'une page à une autre peut être programmé avec C#
- Exemple (principe)
  - Passage de Page1 vers Page2
    - Dans Page1.xaml, on ajoute un événement associée à un contrôle. L'événement pointe sur un call-back C# dans Page1.xaml.cs.
      - Contrôle = Button
      - Evènement = Clicked
      - Call-back c# = ChargerPageSuivante()
    - Dans Page1.xaml.cs, on ajoute le call-back C#.
      - Call-back c# = ChargerPageSuivante()
- Exemple (programmation)
  - Page1.xaml

```
<Button x:Name="button" Text="Vers page 2"
Clicked="ChargerPageSuivante" VerticalOptions="CenterAndExpand"
/>
```
  - Page2.xaml.cs

```
async void ChargerPageSuivante(object sender, EventArgs e)
{
    await Navigation.PopToRootAsync(new Page2());
}
```