

Xamarin

Formation

# Présentation du framework

# Présentation du framework

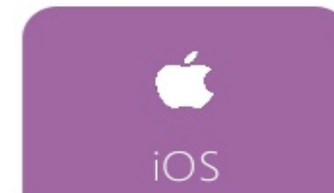
## Techniques de développement d'applications mobiles

- Se conformer au **maximum** de plateformes possible (natif)
- Grande entreprises comme les startups

### Applications mobiles



5 versions d'OS  
x 8 Fabricants  
x 20 Langues  
x 27 Tailles d'écran



3 versions d'OS  
x 8 Devices  
x 20 Langues



2 versions d'OS  
x 17 Devices  
x 20 Langues

# Présentation du framework

## Techniques de développement d'applications mobiles

- Développer des applications **maintenables** et gérant **l'internationalisation**

# Présentation du framework

## Concept Xamarin

- Version libre et open-source du .NET framework
- Développement en C# multi-plateformes

# Présentation du framework

## Plateformes disponibles

- Xamarin.Android : Développement pour la plateforme Google Android
- Xamarin.iOS : Développement pour la plateforme Apple IOS
- Xamarin.UWP : Développement pour Universal Windows Platform (Windows 10 desktop et mobile)

# Présentation du framework

## Xamarin Forms

- Partage de code entre Xamarin.Android, Xamarin.iOS et Xamarin.UWP
- Minimisation de l'utilisation de spécificités de chaque plateforme
- Description des vues en XAML (eXtensible Application Markup Language, dialecte XML)
- Contrôleurs des vues grâce aux fichiers de code-behind (Classes xaml.cs)

# Présentation du framework

## Intégration dans Visual Studio 2017

- Quatre projets pour une solution Xamarin.Forms :
  - **Projet de code partagé** Bibliothèque de classes C# contenant les modèles et les vues Xamarin.Forms
  - **Projet Xamarin.Android** Application spécifique Android
  - **Projet Xamarin.iOS** Application spécifique iOS
  - **Projet Xamarin.UWP** Application spécifique UWP

Chaque projet spécifique dispose d'une référence vers la bibliothèque de classes (code partagé)



# Présentation du framework

## Configuration des émulateurs



- **Android AVD :**
  - > Installation Android Studio
  - > AVD / Création d'un appareil virtuel
- **IOS Iphone :**
  - > Utilisation de Xamarin Studio sur Mac
  - > Connexion SSH depuis Visual Studio sur Windows
  - > Compilation à distance avec Xcode

# Présentation du framework

EXERCICE : Développement d'une première application mobile

## Exercice pratique

- Créer une solution Xamarin.Forms
- Utiliser Bibliothèque de classes comme type de projet
- Réussir à démarrer le projet hello world avec l'émulateur Android

# Langage XAML et C# pour les applications mobiles

# Langage XAML et C# pour mobile

## Introduction au 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"
  xmlns:local="clr-namespace:TestAndroid"
  x:Class="TestAndroid.TestAndroidPage">
  <StackLayout HorizontalOptions="Center">
    <Label Text="Calculs simples" HorizontalOptions="Center" Margin="0, 20, 0, 50" FontSize="20" />
    <Label x:Name="labelStepper" Text="Valeur à ajouter : 1" HorizontalOptions="Center"/>
    <Stepper x:Name="stepper" Minimum="1" Maximum="9" Value="1" ValueChanged="Handle_ValueChanged" HorizontalOptions="Center"/>
    <Button x:Name="bAdd" Text="Ajouter" Clicked="bAdd_Handle_Clicked" Margin="0, 5, 0, 0"/>

    <Button Text="Voir la liste" Clicked="Handle_Clicked"/>

    <Button Text="Combien de 1 dans la liste ?" Clicked="bNumber_Handle_Clicked" x:Name="bNumber"/>
  </StackLayout>
</ContentPage>
```

- Langage utilisant la norme XML (eXtensible Markup Language)
- Une balise a des attributs et peut contenir des enfants
- L'interface est décrite grâce à des Layouts (conteneurs) et des Views (éléments graphiques)

# Langage XAML et C# pour mobile

## Structure d'un fichier XAML

- Une vue décrite en XAML doit contenir la balise `ContentPage` avec ses attributs pour référencer correctement les possibilités de `Xamarin.Forms`
- Chaque balise XAML peut avoir des propriétés et des enfants (ex : une `Grid` a pour enfants des vues)
- Les extensions du langage (Markup Extensions) permettent une interaction forte avec le code C# (`x:Name` pour identifier une balise, `x:Reference` pour récupérer un pointeur sur un objet ou un attribut d'une classe, etc...)

# Langage XAML et C# pour mobile

## Interaction avec le code-behind

- Chaque fichier XAML est connecté à un fichier de code-behind en C# (fichier .xaml.cs)
- Le code-behind permet un contrôle sur les éléments décrits dans le XAML
- Utilisation de l'attribut `x:Name` en XAML pour récupérer les composants dans le code-behind

# Langage XAML et C# pour mobile

## Binding des données

- Possibilité d'associer deux objets entres eux grâce au binding
- Le binding permet d'actualiser automatiquement les propriétés des composants
- MVVM

[https://developer.xamarin.com/guides/xamarin-forms/xaml/xaml-basics/data\\_bindings\\_to\\_mvvm/](https://developer.xamarin.com/guides/xamarin-forms/xaml/xaml-basics/data_bindings_to_mvvm/)

# Langage XAML et C# pour mobile

## EXERCICE : Réaliser une interface graphique en XAML et C#

### Exercice pratique

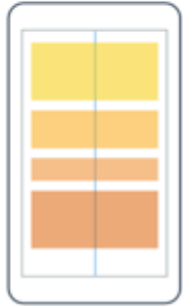
- Créer une application ayant une seule page
- Réaliser une page de login (Bande de titre, champs identifiant et mot de passe, bouton de connexion)
- Les éléments doivent être correctement positionnés (centrage des éléments, bouton en bas de page...)



# Types d'interfaces graphiques

# Types d'interfaces graphiques

## Les différentes organisations d'éléments (Layouts)



StackLayout

- Les éléments sont placés les uns après les autres
- Possibilité d'orienter le sens du StackLayout  
(Positionnement des éléments de façon horizontale ou verticale)

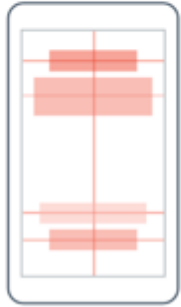


RelativeLayout

- Les éléments sont placés les uns par rapport aux autres
- Chaque élément dispose de contraintes sur les autres éléments  
(enfants ou parents)

# Types d'interfaces graphiques

## Les différentes organisations d'éléments (Layouts)



AbsoluteLayout

- Les éléments sont positionnés de façon exacte
- Type de layout rigide (peu de possibilité pour le responsive)



GridLayout

- Positionnement des éléments dans une grille définie
- Définition de lignes et de colonnes (tailles fixes ou variables)
- Chaque élément peut être positionné dans une case de la grille, ou s'étalant sur plusieurs cases de la grille

# Types d'interfaces graphiques

## Les différentes organisations d'éléments (Layouts)

Autres types d'organisation des éléments :



ContentView



ScrollView



Frame

# Types d'interfaces graphiques

## Les différentes vues (composants graphiques)

Les composants graphiques disponibles dans Xamarin.Forms :

<https://developer.xamarin.com/guides/xamarin-forms/user-interface/controls/views/>

# Types d'interfaces graphiques

## Les différents éléments de cellules

Les types de cellules insérables dans les listes et les grilles :

<https://developer.xamarin.com/guides/xamarin-forms/user-interface/controls/cells/>

# Types d'interfaces graphiques

Créer des applications responsives

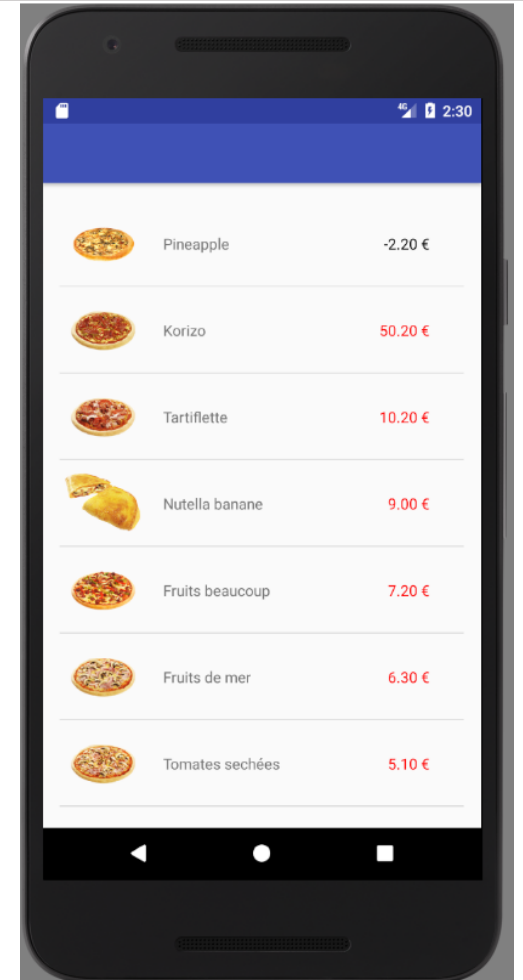
- Utiliser des agencements cohérents (GridLayout, StackLayout, RelativeLayout)
- Ne pas mettre de margin pour placer les composants
- Privilégier les options de positionnement HorizontalOptions et VerticalOptions, et leurs différentes options (Start, Fill, Center, CenterAndExpand, End ...)

# Types d'interfaces graphiques

## EXERCICE : Réaliser une interface responsive

### Exercice pratique

- Réaliser une application d'affichage de pizzas
- Les pizzas seront affichées dans une liste ou un tableau, avec pour chaque pizza : une image, un nom et un prix





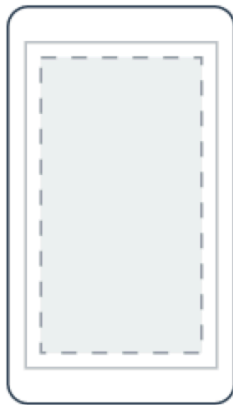
# Navigation dans les applications mobiles

# Navigation dans les applications mobiles

## Les infrastructures de navigation

Les différentes possibilités pour naviguer entre différentes pages :

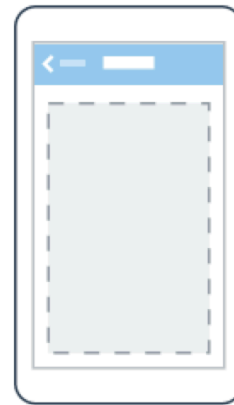
<https://developer.xamarin.com/guides/xamarin-forms/application-fundamentals/navigation/>



ContentPage



MasterDetailPage



NavigationPage



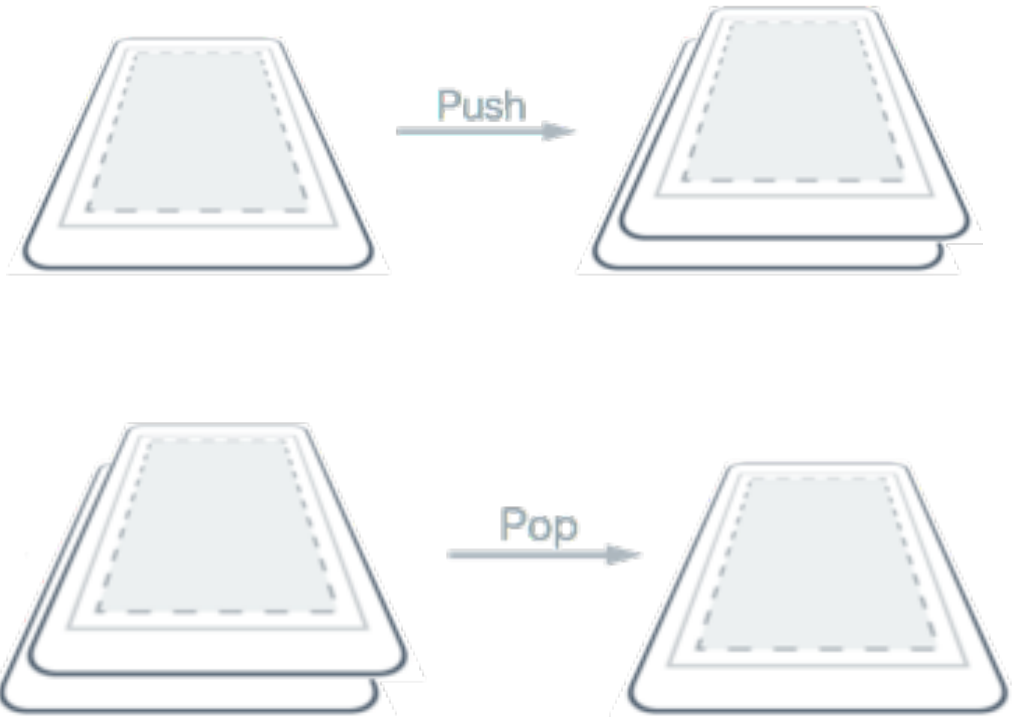
TabbedPage



CarouselPage

# Navigation dans les applications mobiles

Le passage d'une page à une autre



- Utilisation d'une pile pour changer de page
- La page au dessus de la pile est celle qui est actuellement affichée
- Affichage d'une nouvelle page :  
`Navigation.pushAsync(new MaPage());`
- Retour à la page précédente :  
`Navigation.popAsync();`

# Navigation dans les applications mobiles

Cycle de vie d'une application Xamarin.Forms

Plusieurs évènements sont déclenchés avec l'application

- OnStart : Appelé lorsque l'application a démarrée
- OnSleep : Appelé à chaque mise en arrière plan de l'application
- OnResume : Appelé lorsque l'application revient au premier plan

Modèle observer/observable

<https://developer.xamarin.com/guides/xamarin-forms/application-fundamentals/messaging-center/>

# Navigation dans les applications mobiles

## Utiliser les gestes utilisateurs

Différents types de gestes utilisables (Tap, Pinch, Pan) :

<https://developer.xamarin.com/guides/xamarin-forms/application-fundamentals/gestures/tap/>

<https://developer.xamarin.com/guides/xamarin-forms/application-fundamentals/gestures/pinch/>

<https://developer.xamarin.com/guides/xamarin-forms/application-fundamentals/gestures/pan/>