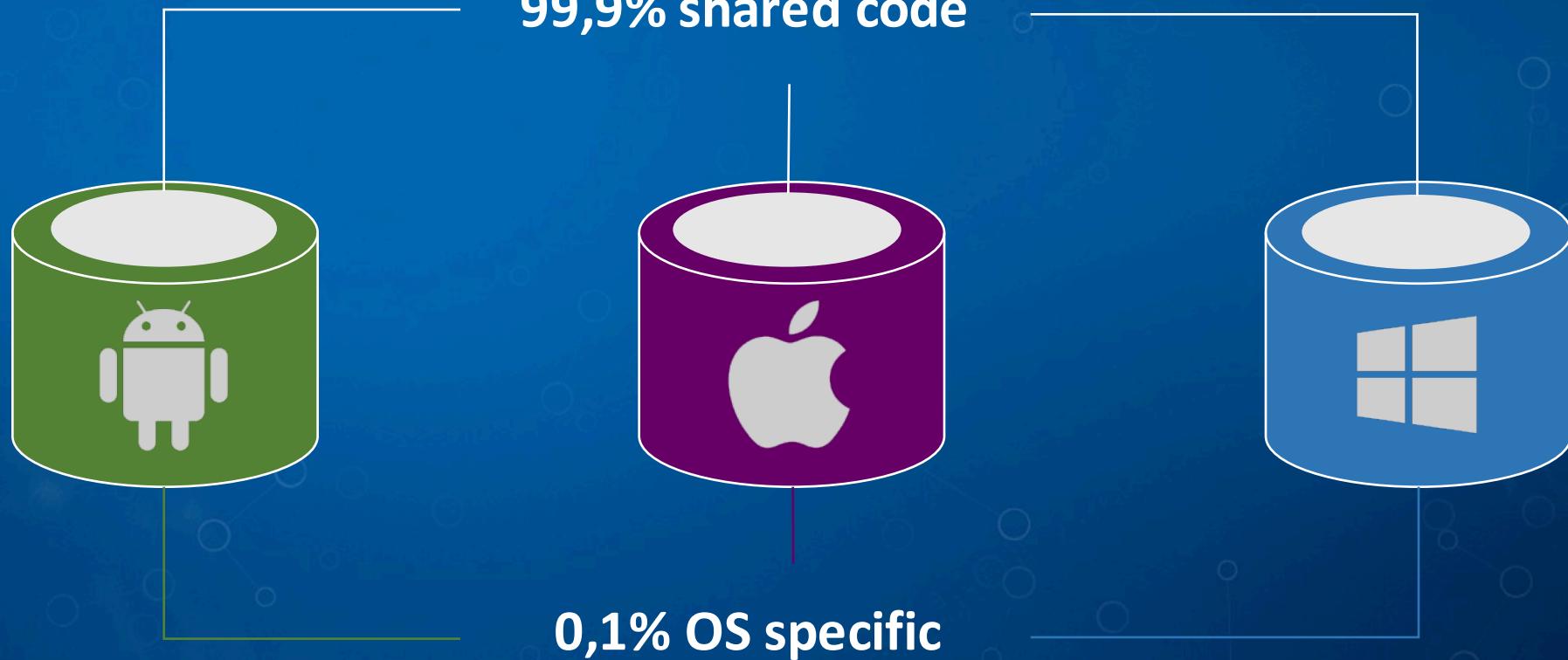


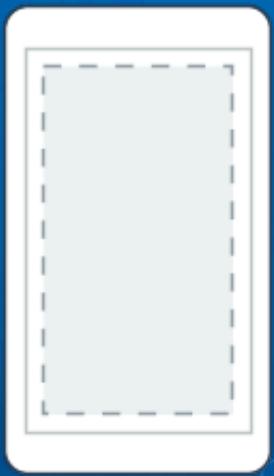
Xamarin.Forms



The promise



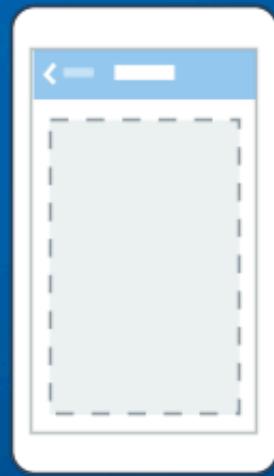
Pages



ContentPage



MasterDetailPage



NavigationPage

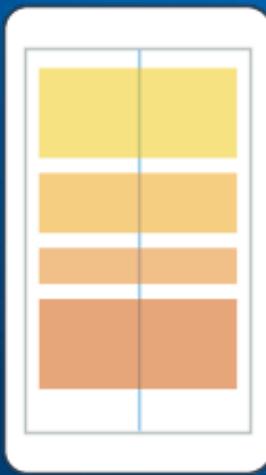


TabbedPage

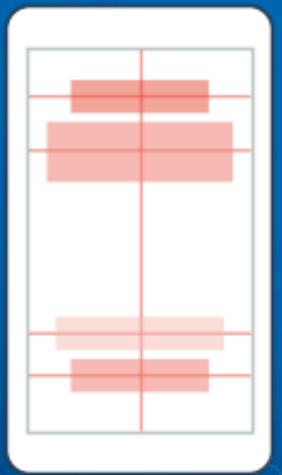


CarouselPage

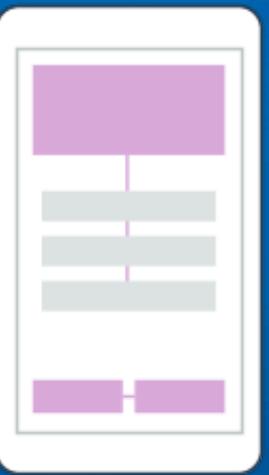
Layouts



StackLayout



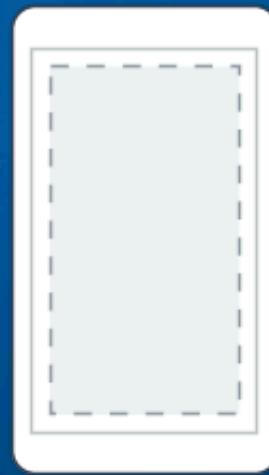
AbsoluteLayout



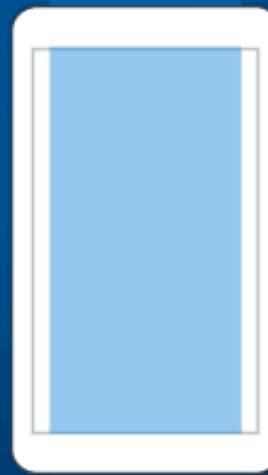
RelativeLayout



GridLayout



ContentView



ScrollView



Frame

Controls

ActivityIndicator	BoxView	Button	DatePicker	Editor
Entry	Image	Label	ListView	Map
OpenGLView	Picker	ProgressBar	SearchBar	Slider
Stepper	TableView	TimePicker	WebView	EntryCell
ImageCell	SwitchCell	TextCell	ViewCell	

Cheat Sheet

[http://tinyurl.com/x
amarin-forms](http://tinyurl.com/xamarin-forms)



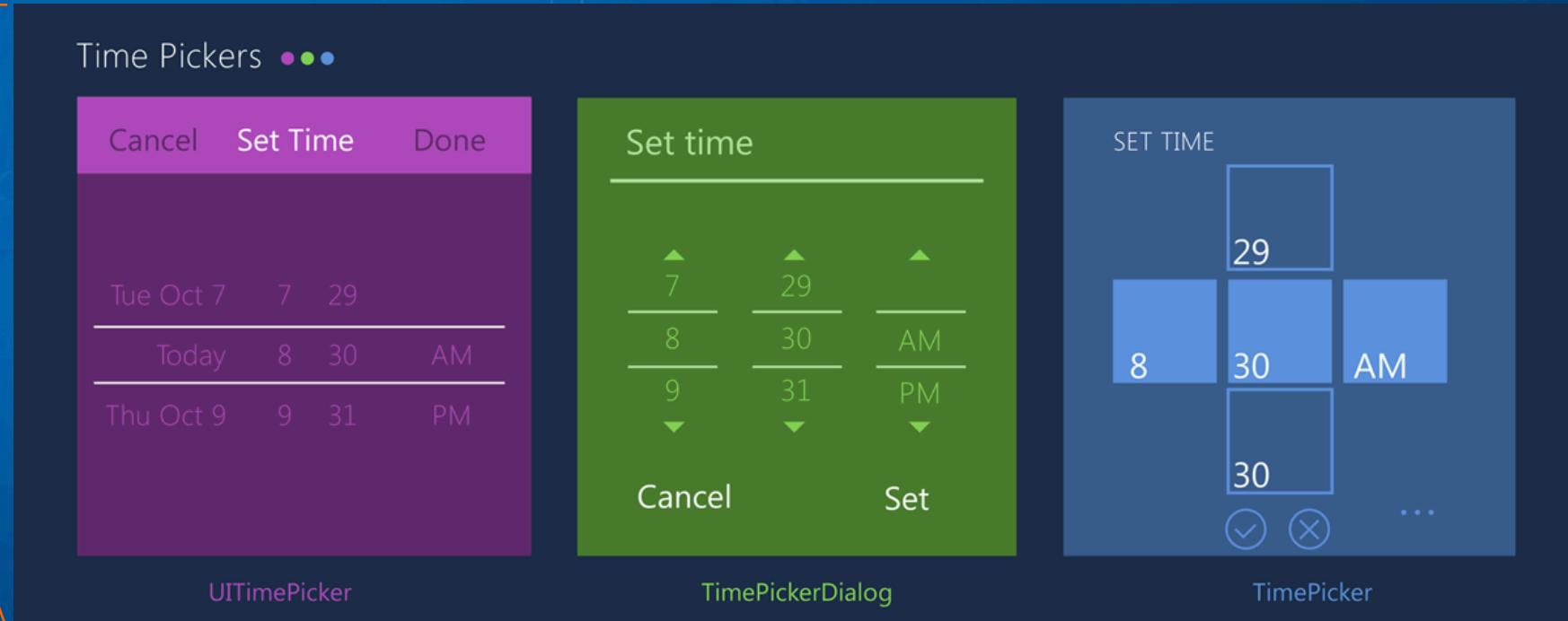
You Say ListView,
I Say UI TableView

Become Fluent In Native Mobile App Development

Different programming languages, different design philosophies, and different tools can make you feel like a foreigner in a multi-platform mobile world. This handy guide will help you speak each device platform's language.

Cheat Sheet

[http://tinyurl.com/x
amarin-forms](http://tinyurl.com/xamarin-forms)

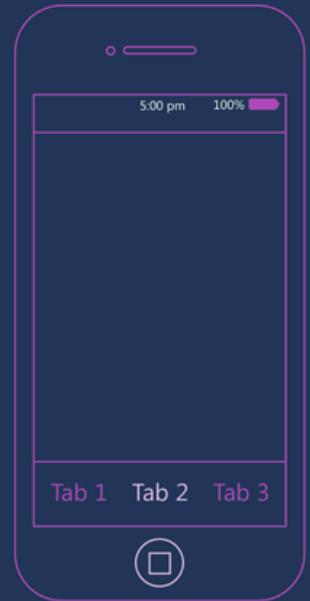


Cheat Sheet

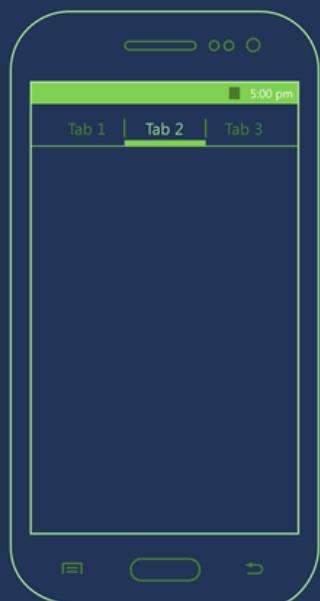
[http://tinyurl.com/x
amarin-forms](http://tinyurl.com/xamarin-forms)



Tabs



UITabBar

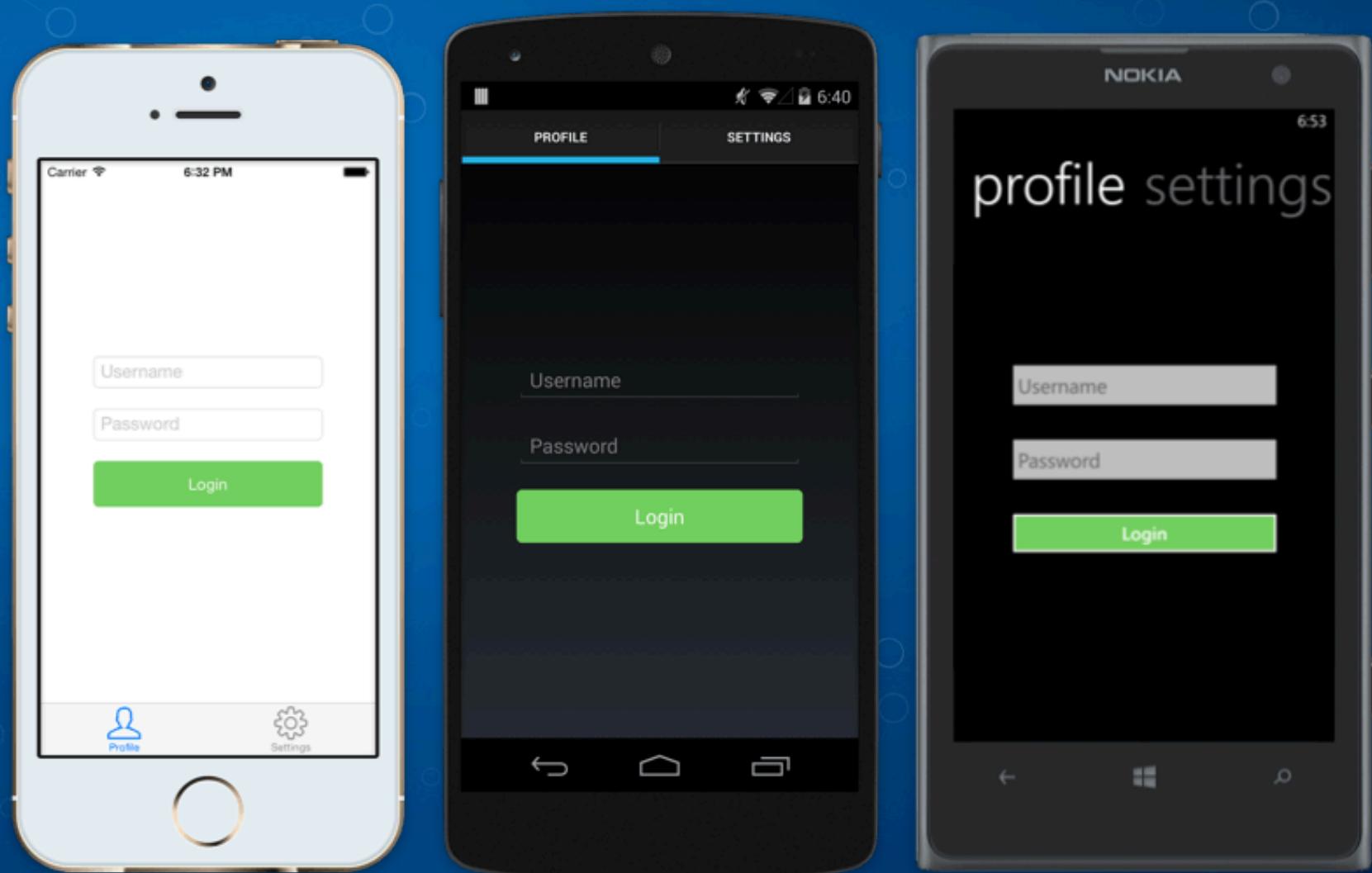


ActionBar Tabs



Pivot

Shared UI



But how? C#

```
67 private ListView CreateListView(IEnumerable itemsSource) {
68     return new ListView () {
69         ItemsSource = itemsSource,
70         ItemTemplate = new DataTemplate(() =>
71             {
72                 var nameLabel = new Label();
73                 nameLabel.SetBinding(Label.TextProperty, "Name");
74                 nameLabel.SetBinding(Label.TextColorProperty, "Color");
75
76                 return new ViewCell
77                 {
78                     View = new StackLayout
79                     {
80                         Padding = new Thickness(0, 5),
81                         Orientation = StackOrientation.Horizontal,
82                         Children =
83                         {
84                             nameLabel
85                         }
86                     }
87                 };
88             });
89         };
90     };
}
```

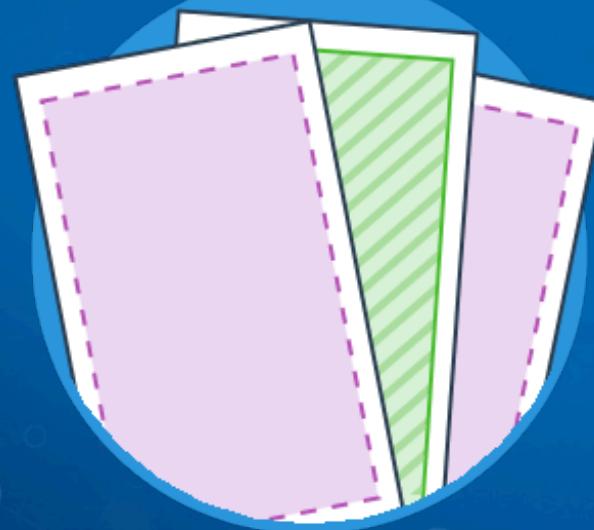
But how? XAML

```
8 | <Label Text="XAML" HorizontalOptions="Center" Font="Bold,50" />
9 | <Label Text="{Binding CompletedTasksLabel}" HorizontalOptions="Center" Font="Micro" />
10|
11| <ListView x:Name="DataListView" ItemsSource="{Binding Data}" ItemTapped="OnItemTapped">
12|   <ListView.ItemTemplate>
13|     <DataTemplate>
14|       <ViewCell>
15|         <StackLayout Padding="0,5" Orientation="Horizontal">
16|           <Label Text="{Binding Name}" TextColor="{Binding Color}" />
17|         </StackLayout>
18|       </ViewCell>
19|     </DataTemplate>
20|   </ListView.ItemTemplate>
21| </ListView>
```

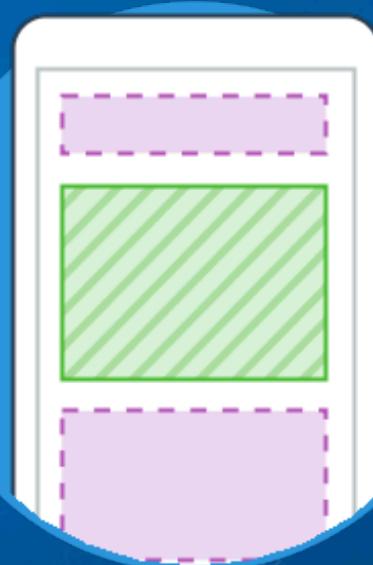




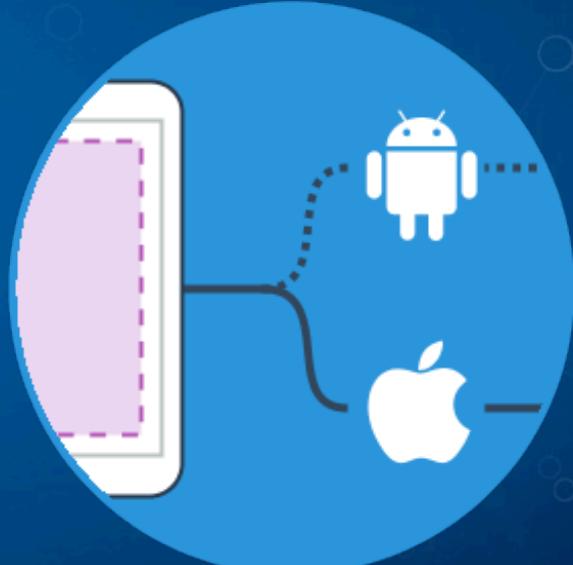
Mix-and-match



Page-By-Page



Custom Views



Shared Services

In Depth Presentation

<http://tinyurl.com/iska-xamarin-forms>

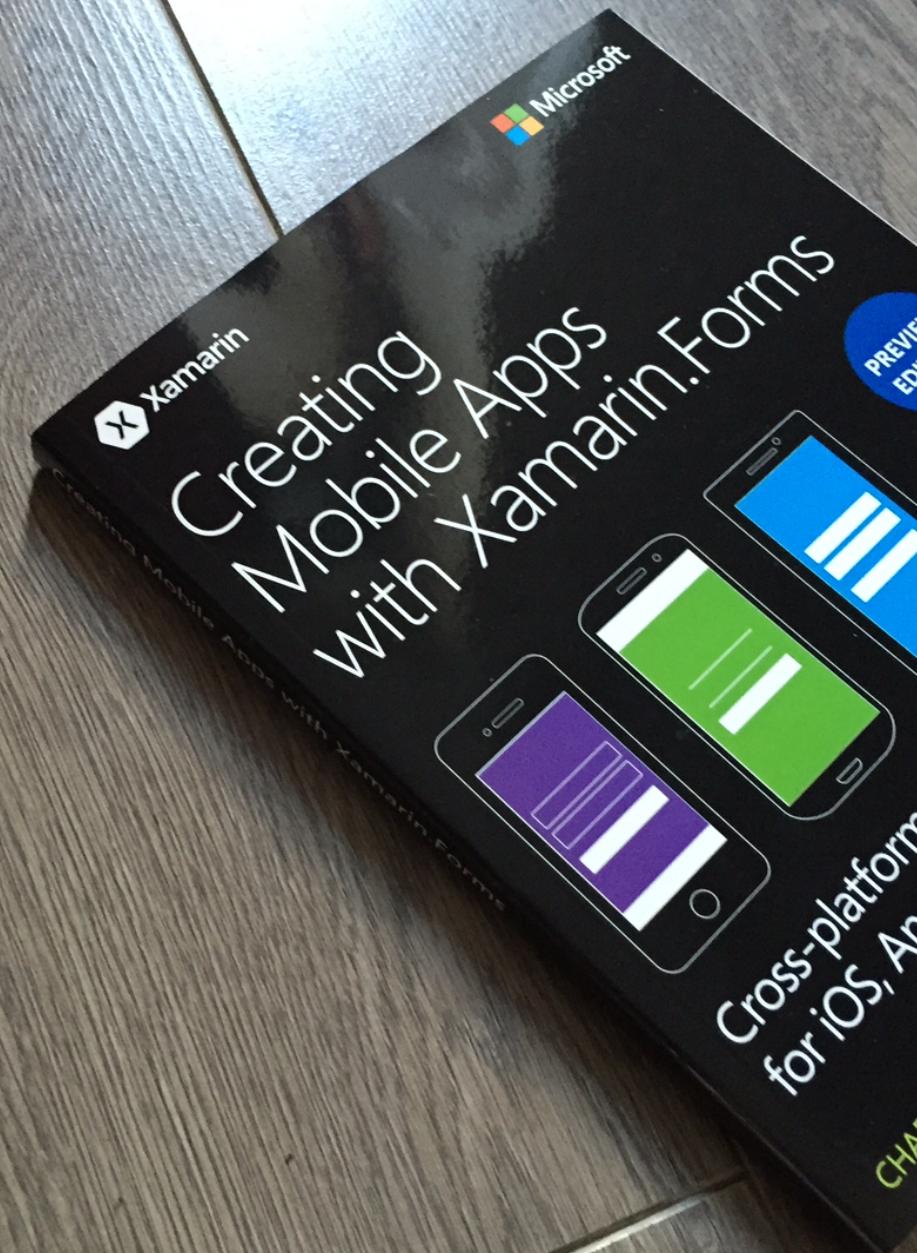
InfoSupport Kennisavond



*xamarin.Forms
ies*

Free E-book

<http://tinyurl.com/ebook-xamarin-forms>



Silver bullet (?)



Xamarin approach



Xamarin.Forms is best for:

Data entry apps

Prototypes and proofs-of-concept

Apps that require little platform-specific functionality

Apps where code sharing is more important than custom UI

Xamarin.iOS / Xamarin.Android is best for:

Apps that require specialized interactions

Apps with highly polished design

Apps that use many platform-specific APIs

Apps where custom UI is more important than code sharing