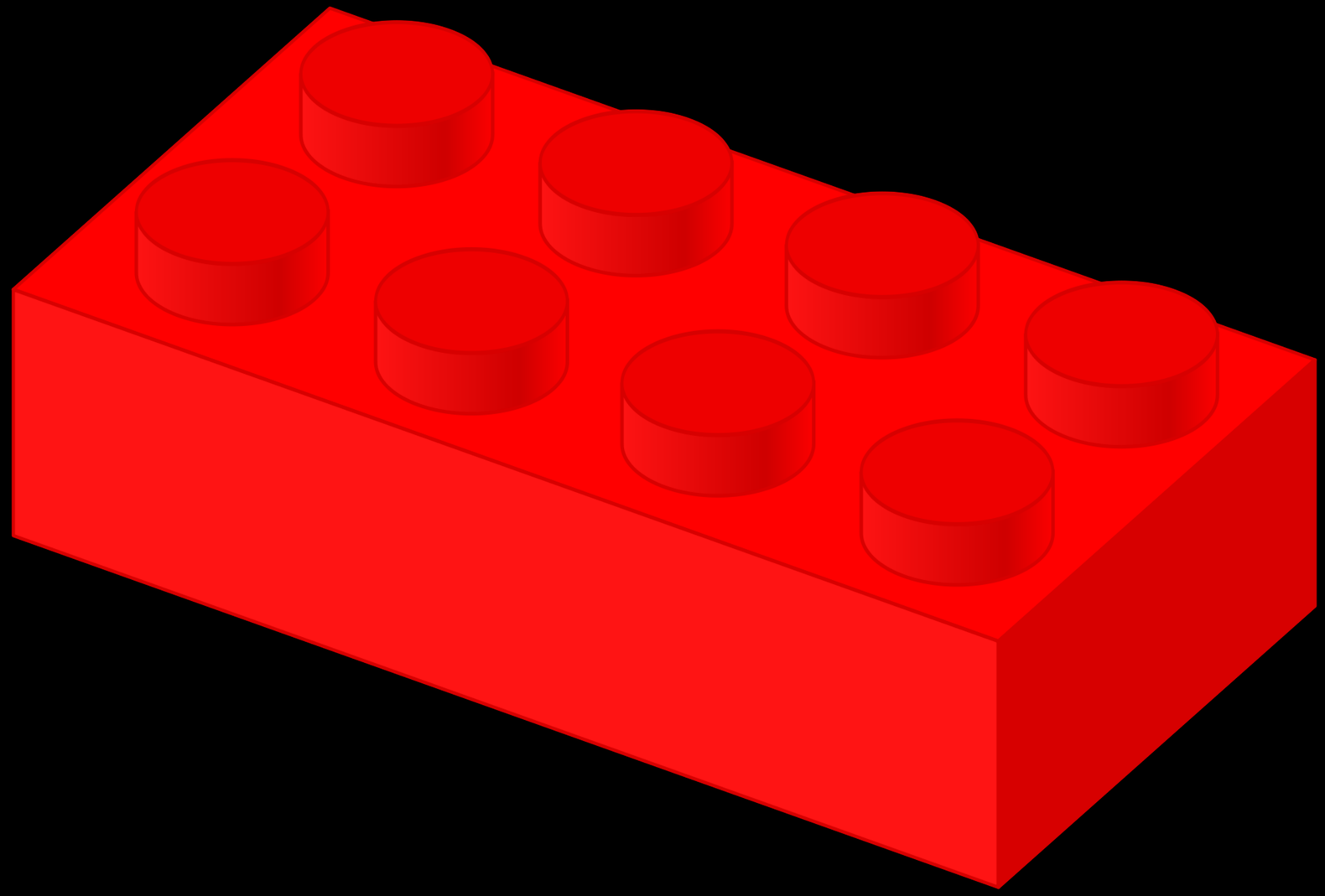


XAMARIN.FORMS FOR BEGINNERS

ABOUT ME

Tom Soderling

Sr. Mobile Apps Developer @ Polaris Industries; Ride Command

Xamarin.Forms enthusiast

DevOps hobbyist & machine learning beginner

4-year Xamarin Certified Mobile Developer

Blog: https://tomsoderling.github.io

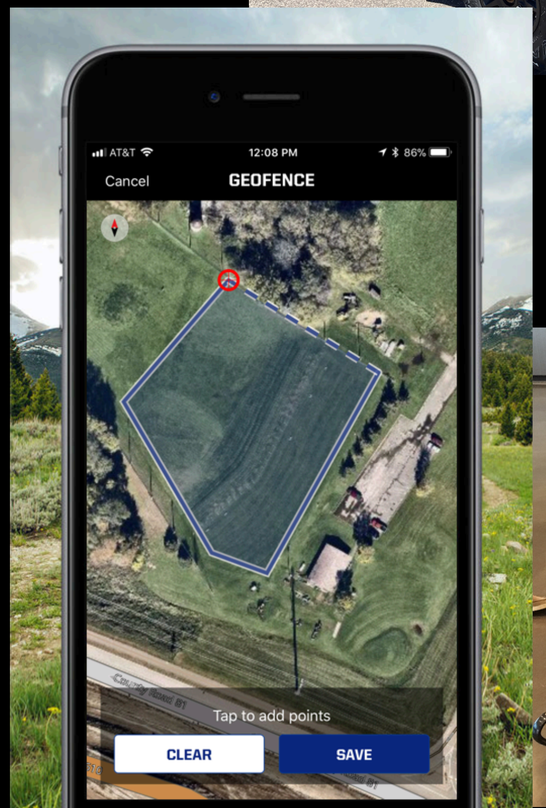
GitHub: https://github.com/TomSoderling

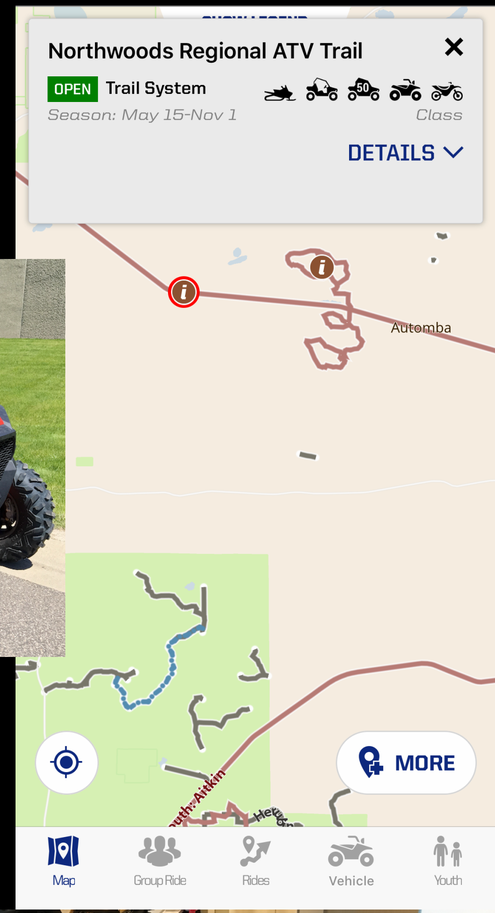
Twitter: @tomsoderling

How Deep Is It?

Pickster

Spaniel





THE PLAN

- Introduction: Why, What, and When
- Overview of Xamarin.Forms Building Blocks
- Building a Xamarin.Forms UI in XAML
- Data Binding
- View Customization
- Next Steps & Resources

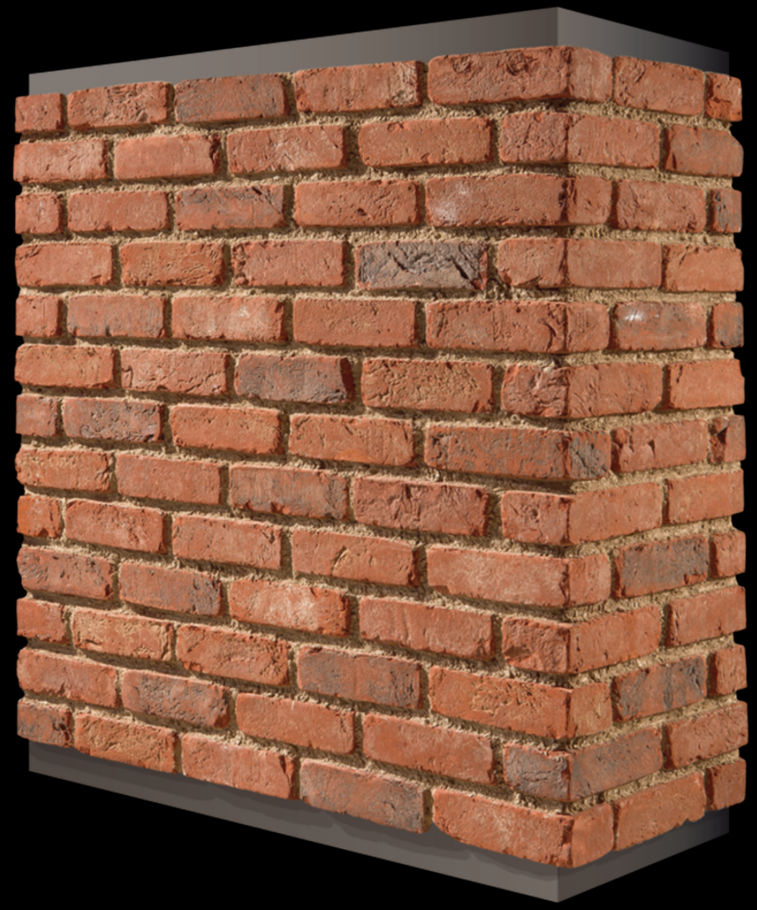
- Please ask any questions that come up!

THE PLAN

- **Introduction: Why, What, and When**
- Overview of Xamarin.Forms Building Blocks
- Building a Xamarin.Forms UI in XAML
- Data Binding
- View Customization
- Next Steps & Resources

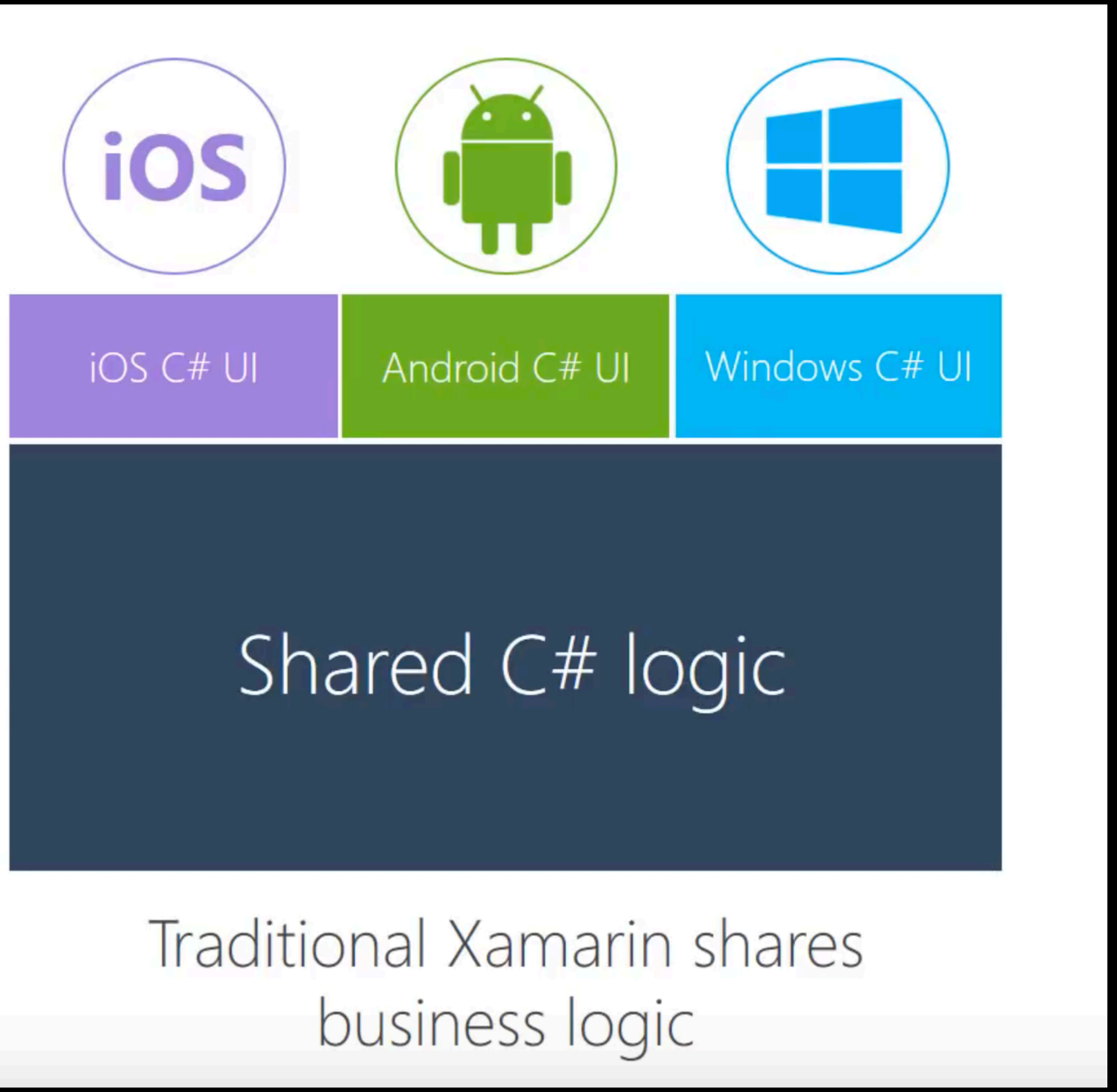
INTRODUCTION : WHY

- WET: the soggy state of mobile app development
  - **W**rite **E**verything **T**wice

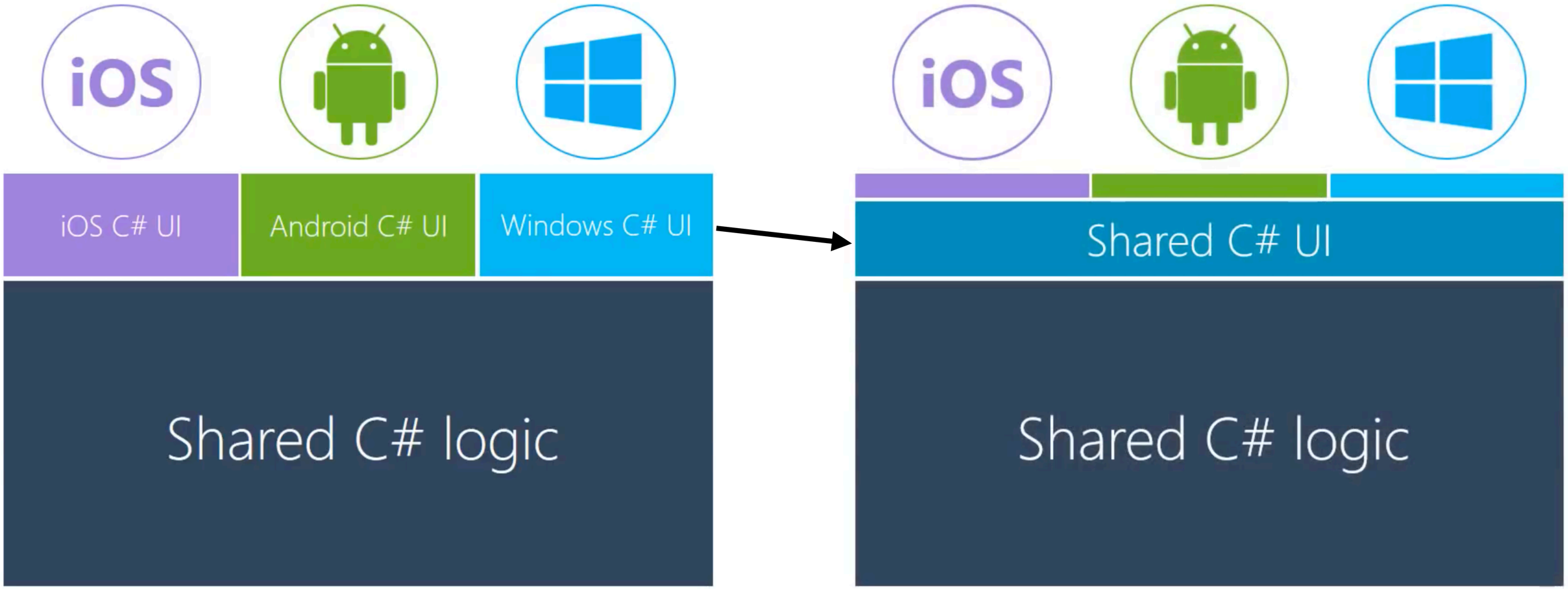


INTRODUCTION : WHY

- WET: the soggy state of mobile app development
  - ~~**W**rite **E**verything **T**wice~~



Traditional Xamarin shares business logic

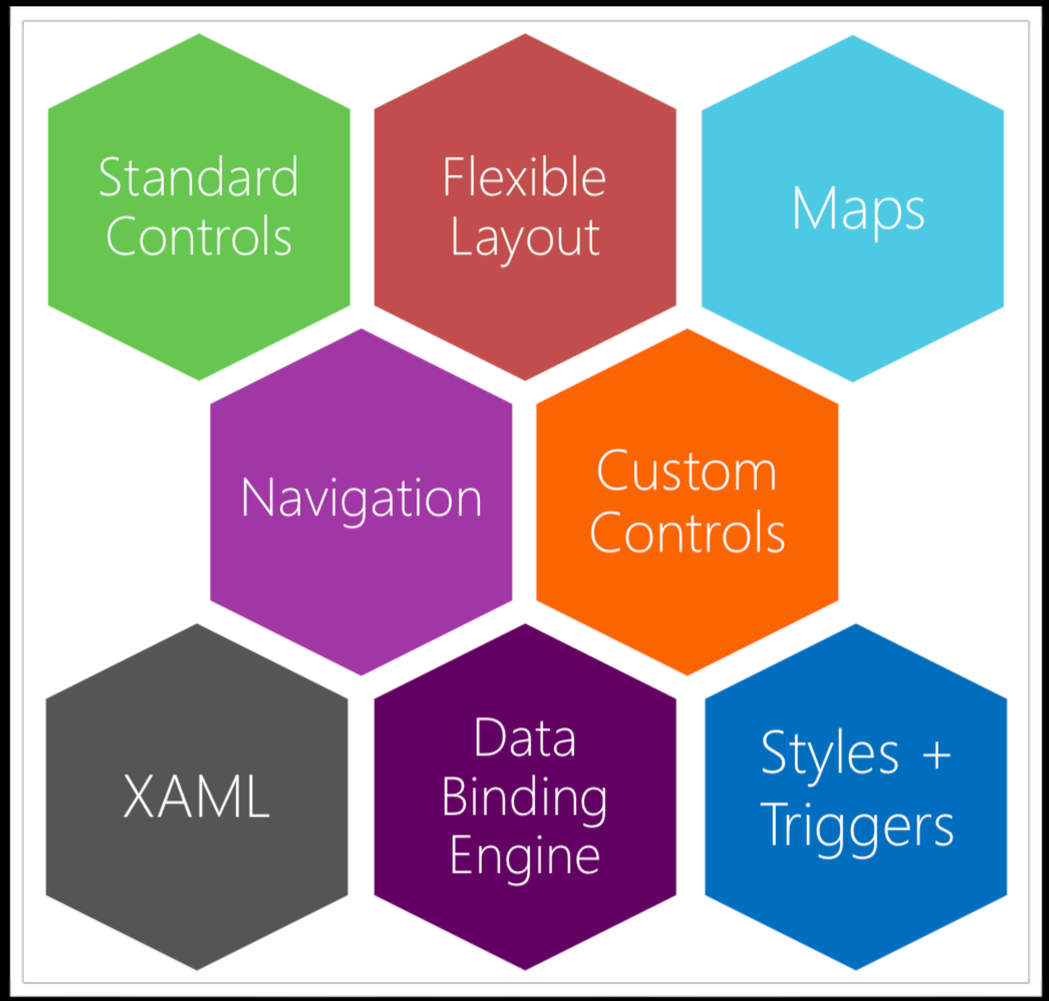


Traditional Xamarin shares business logic

Xamarin.Forms can also share the UI definition

INTRODUCTION : WHAT

- What is Xamarin.Forms?
  - Cross-platform UI framework
  - Platforms:
    - Mobile: iOS 8 and up, Android 4.0.3 (API 15)
    - Desktop: Windows 10 UWP, MacOS, WFP
    - Samsung Smart Devices: Tizen



INTRODUCTION : WHAT

- Brief History:
  - May 2011, Xamarin founded
    - MonoTouch and Mono for Android using MonoDevelop IDE
  - February 2013, release of Xamarin 2.0
    - Xamarin Studio IDE & integration with Visual Studio
    - Renamed to Xamarin.Android and Xamarin.iOS
  - May 2014, Xamarin.Forms released as part of Xamarin 3
  - February 24 2016, Xamarin acquired by Microsoft

- Owned, actively developed on, and supported by Microsoft
- Free and completely open-source on GitHub

INTRODUCTION : WHAT

- Develop on Mac or Windows
  - Visual Studio on Windows (2015 or 2017)
  - Visual Studio for Mac
- iOS development requires a Mac to build (somewhere)
  - Xamarin Mac Agent

INTRODUCTION : WHEN

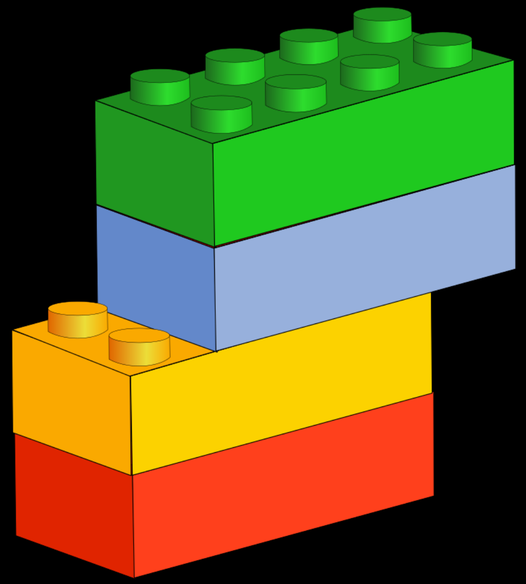
- When should I use Xamarin.Forms?
  - You or your team knows C# and .NET
  - You need apps for multiple-platforms
  - You want native app performance and/or look and feel
  - You’re okay knowing that there are cheaper ways to make an app

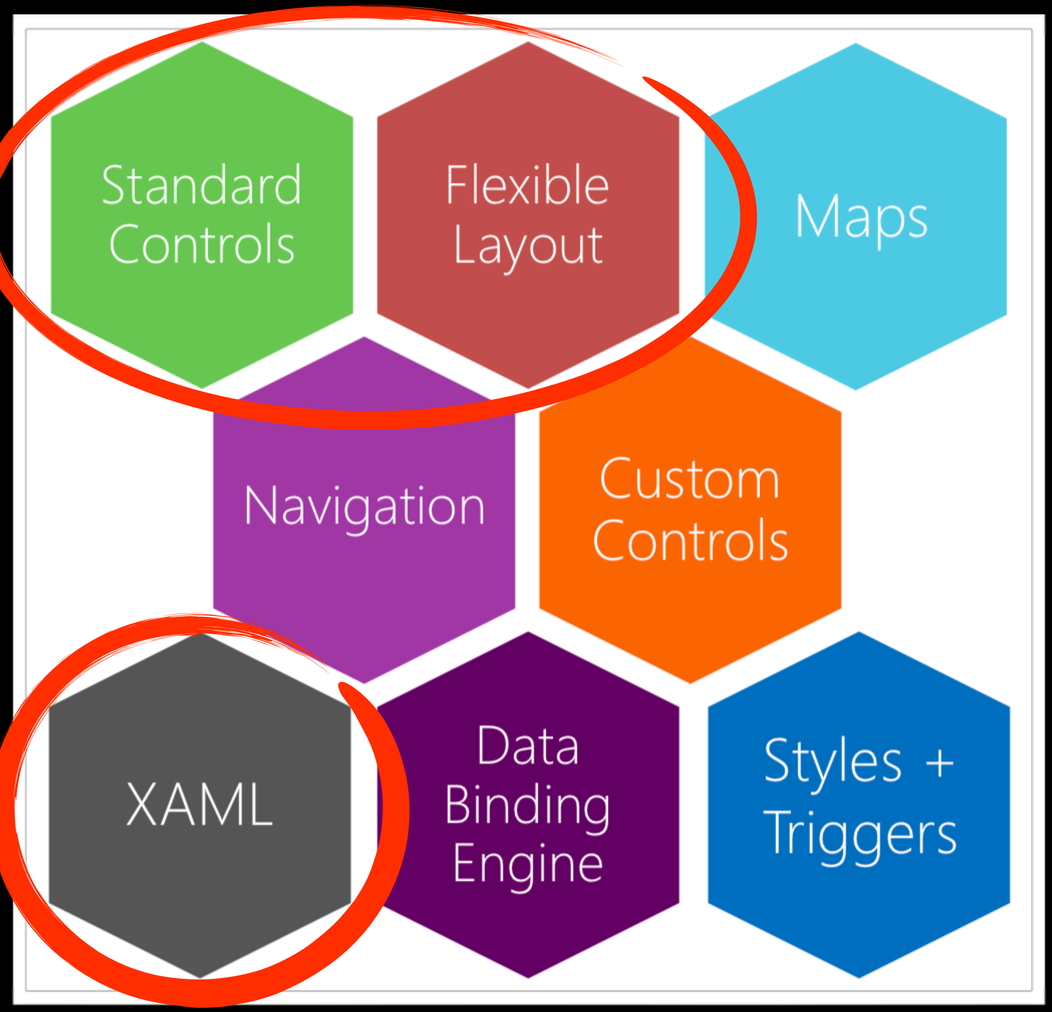
THE PLAN

- Introduction: Why, What, and When
- **Overview of Xamarin.Forms Building Blocks**
- Building a Xamarin.Forms UI in XAML
- Data Binding
- View Customization
- Next Steps & Resources

OVERVIEW OF XAMARIN.FORMS BUILDING BLOCKS

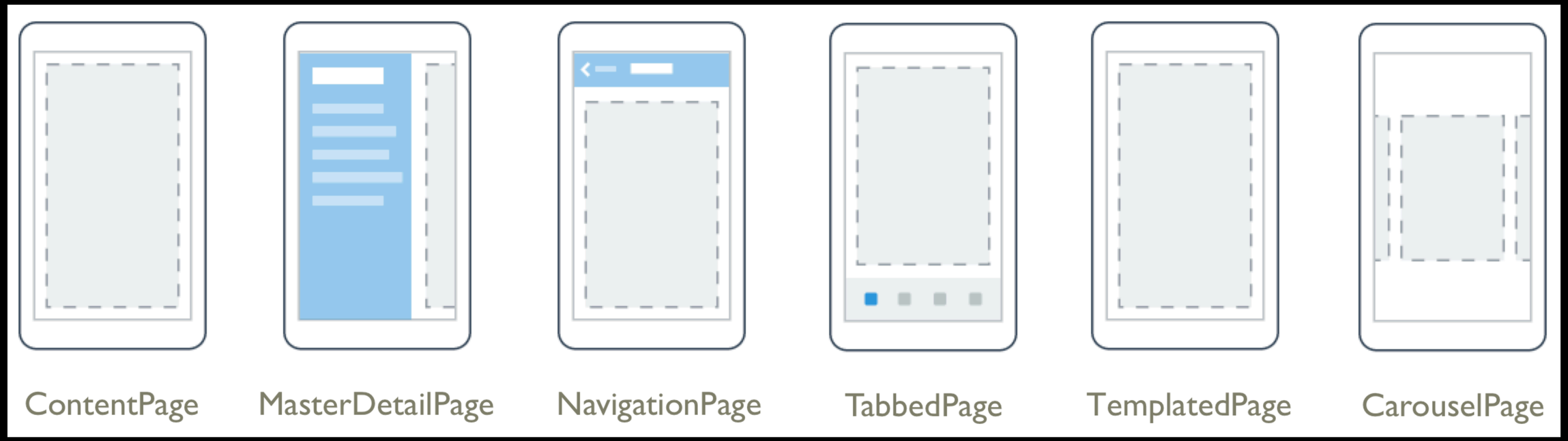
- Pages
- Layouts
- Views
- Cells





OVERVIEW OF XF BUILDING BLOCKS

- Pages



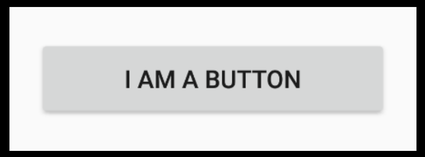
OVERVIEW OF XF BUILDING BLOCKS

- Views
  - Button
  - Label
  - Entry
  - Switch
  - ActivityIndicator

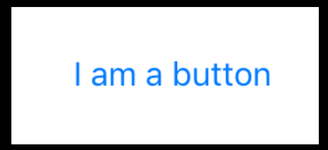
OVERVIEW OF XF BUILDING BLOCKS

- Views
  - Button
  - Label
  - Entry
  - Switch
  - ActivityIndicator

Android

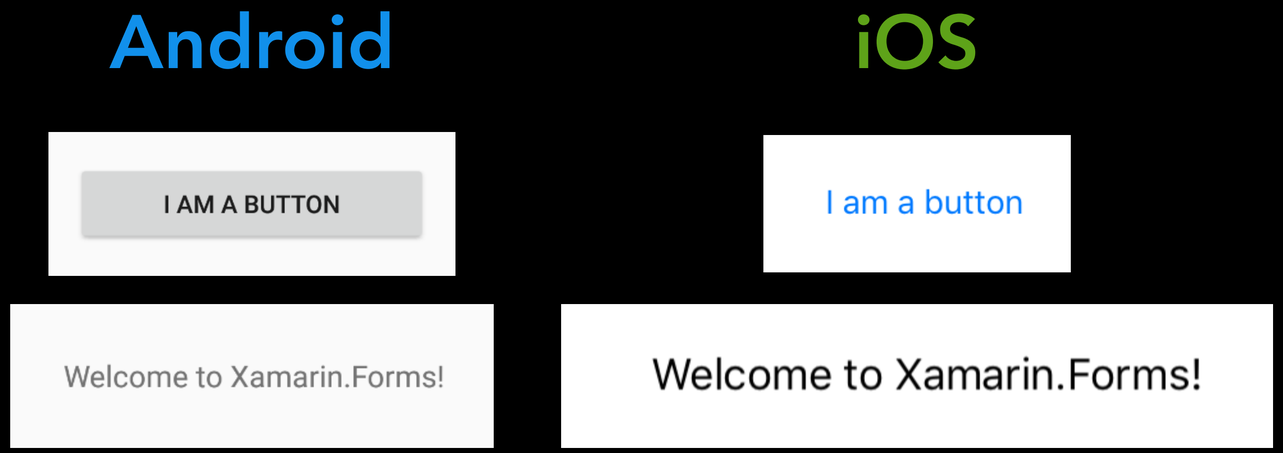


iOS



OVERVIEW OF XF BUILDING BLOCKS

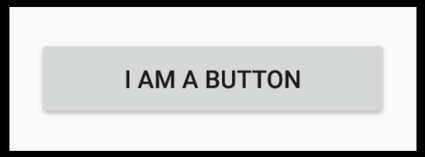
- Views
  - Button
  - Label
  - Entry
  - Switch
  - ActivityIndicator

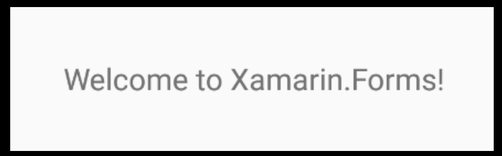


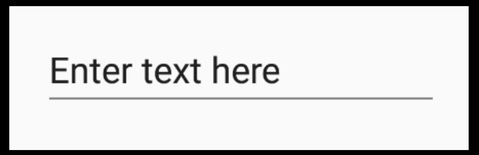
OVERVIEW OF XF BUILDING BLOCKS

- Views
  - Button
  - Label
  - Entry
  - Switch
  - ActivityIndicator

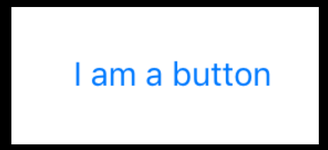
Android

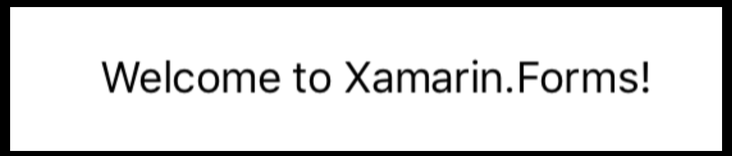


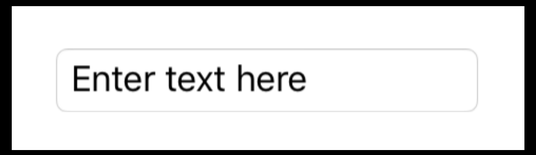




iOS



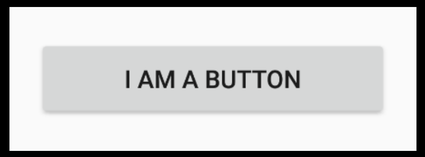


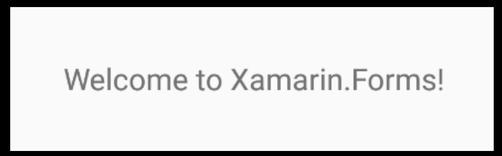


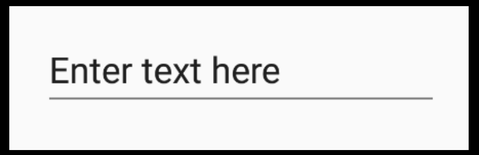
OVERVIEW OF XF BUILDING BLOCKS

- Views
  - Button
  - Label
  - Entry
  - Switch
  - ActivityIndicator

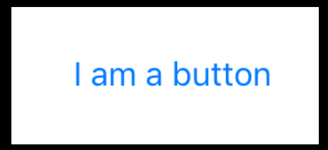
Android

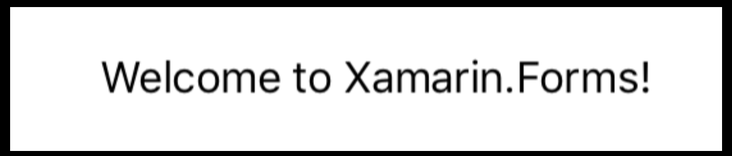


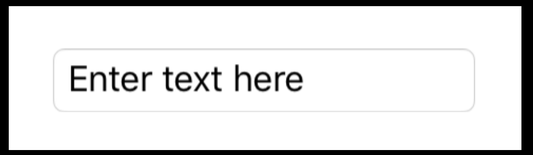




iOS

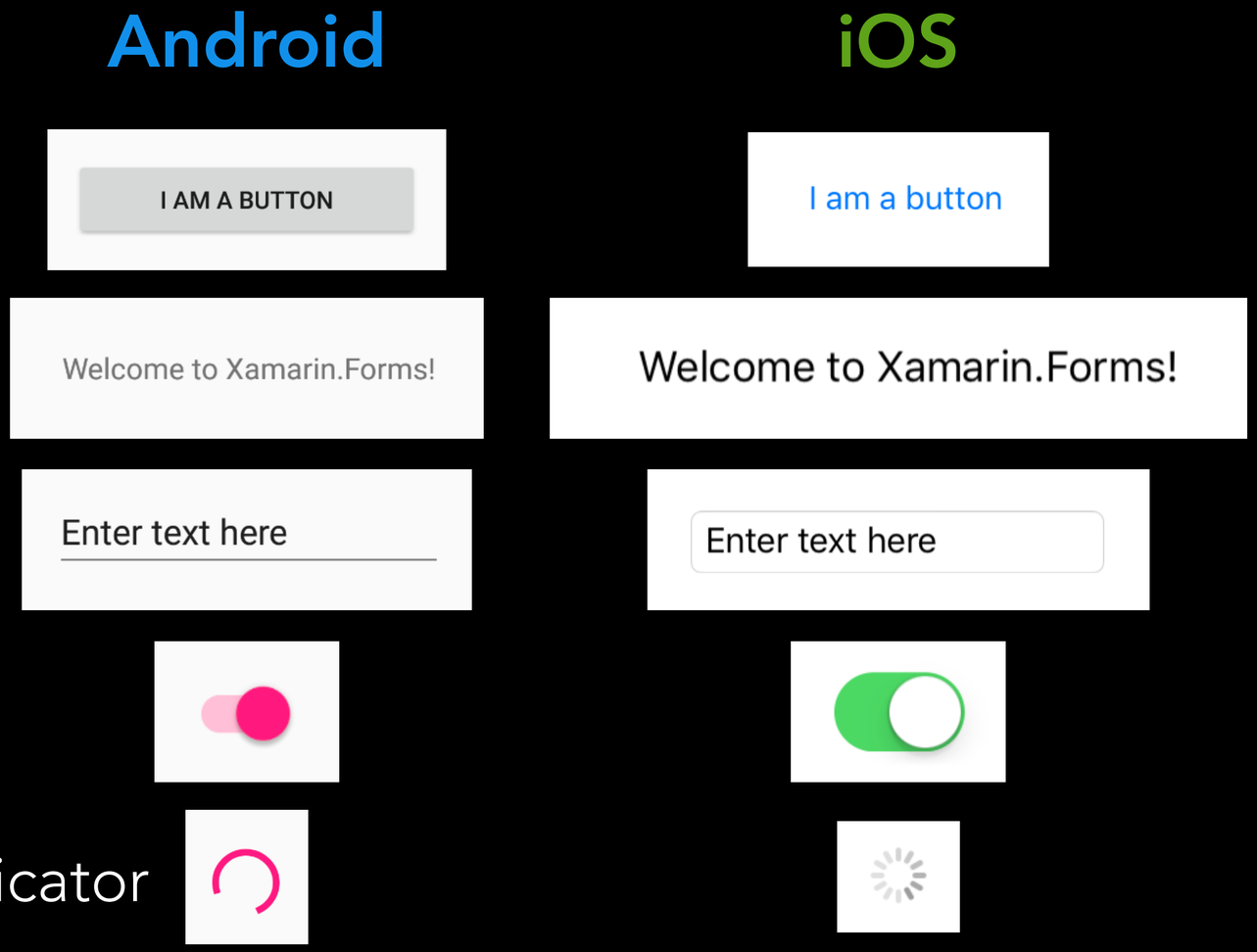






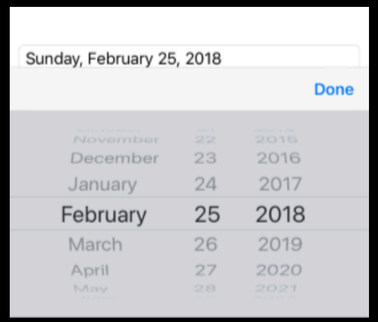
OVERVIEW OF XF BUILDING BLOCKS

- Views
  - Button
  - Label
  - Entry
  - Switch
  - ActivityIndicator

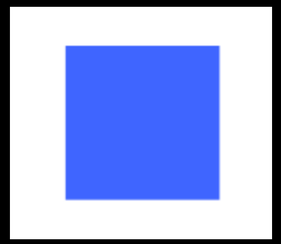


OVERVIEW OF XF BUILDING BLOCKS

- Views
  - ListView
  - TableView
  - Image
  - Slider
  - Picker
  - DatePicker
  - Editor

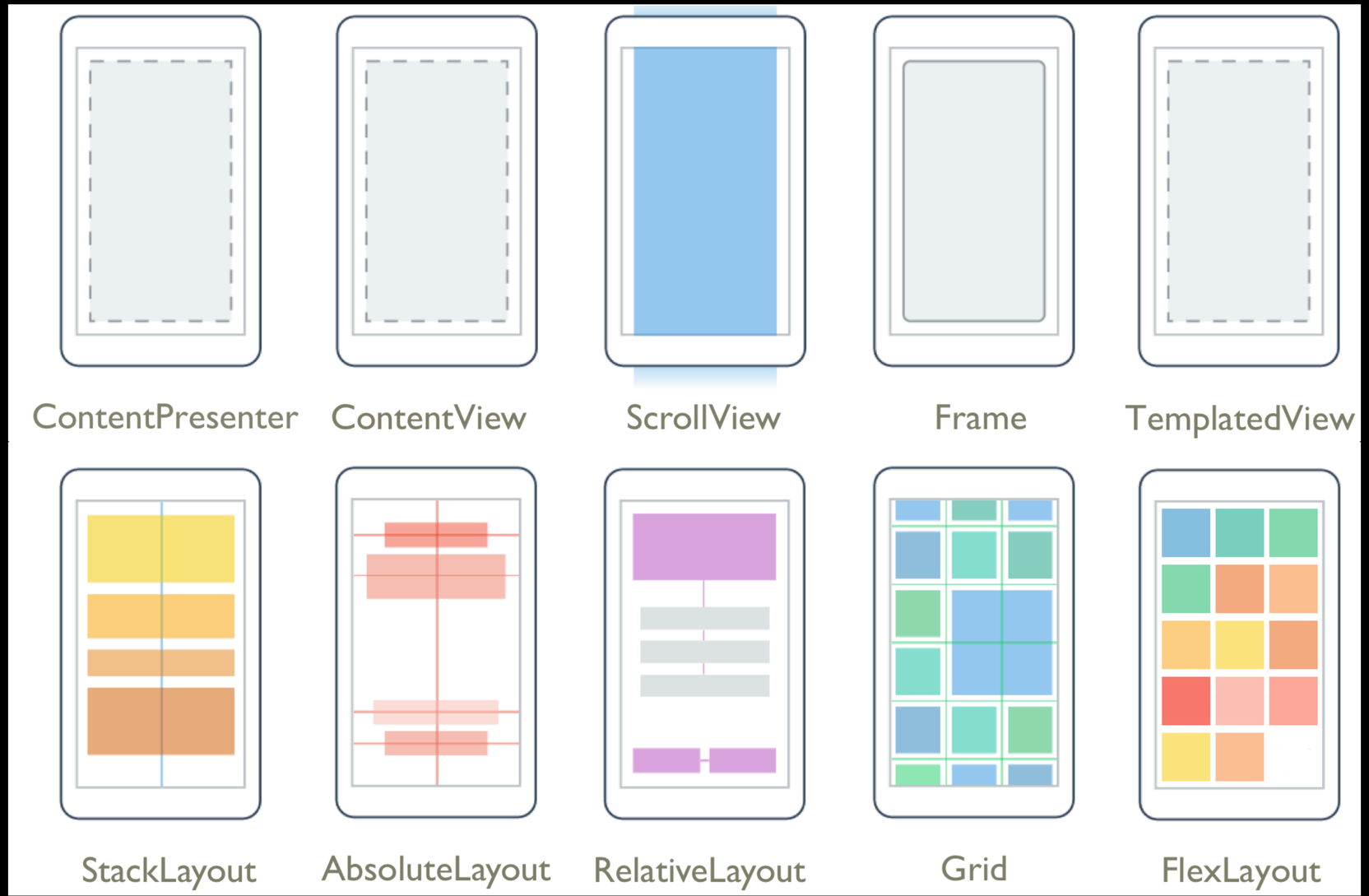


- ProgressBar
- SearchBar
- Map
- WebView
- OpenGLView
- Frame
- BoxView



OVERVIEW OF XF BUILDING BLOCKS

- Layouts



THE PLAN

- Introduction: Why, What, and When
- Overview of Xamarin.Forms Building Blocks
- **Building a Xamarin.Forms UI in XAML**
- Data Binding
- View Customization
- Next Steps & Resources

BUILDING A XAMARIN.FORMS UI IN XAML

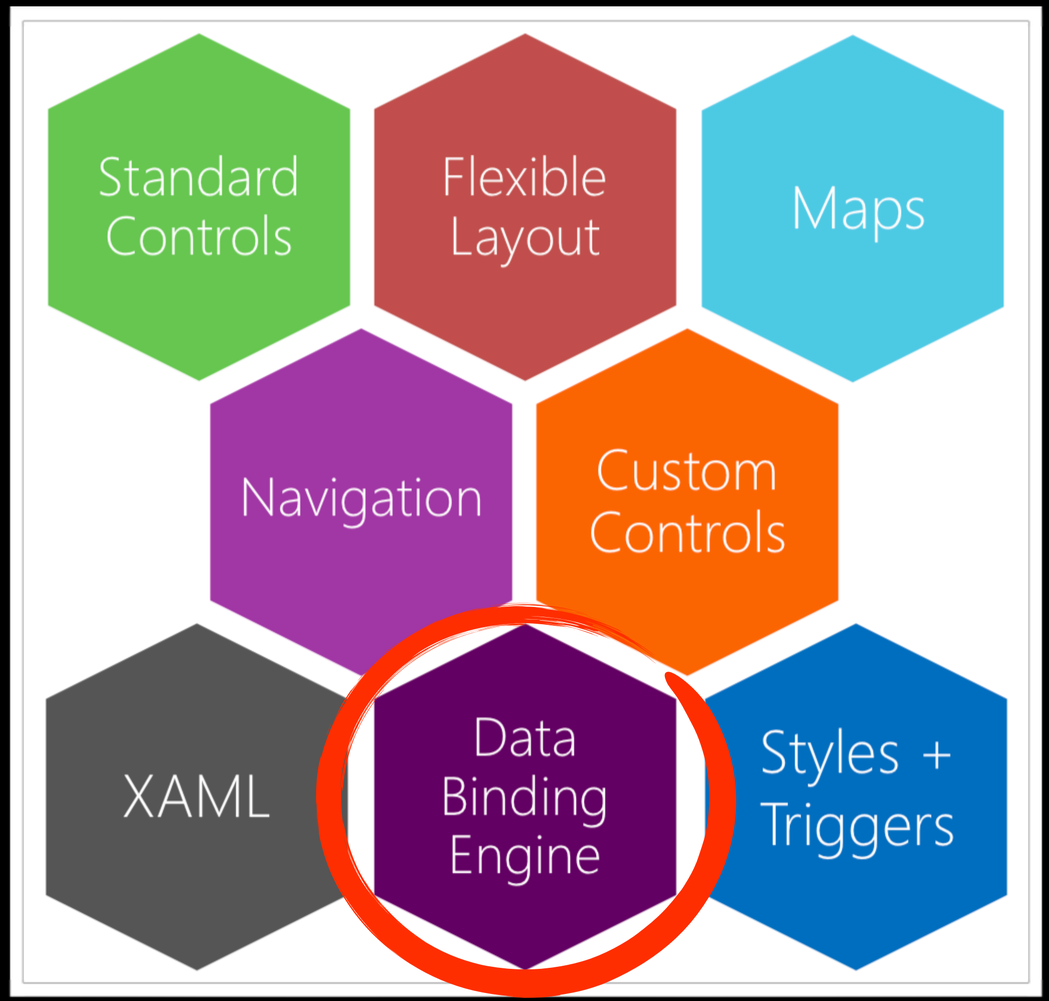
- **Let's look at some code!**
  - Anatomy of a Xamarin.Forms app
  - ContentPage
  - Views
    - Styles
  - Layouts
    - StackLayout
    - Grid
    - AbsoluteLayout

THE PLAN

- Introduction: Why, What, and When
- Overview of Xamarin.Forms Building Blocks
- Building a Xamarin.Forms UI in XAML
- **Data Binding**
- View Customization
- Next Steps & Resources

DATA BINDING

- How do we show information to the user?
- How do we get data from user input?

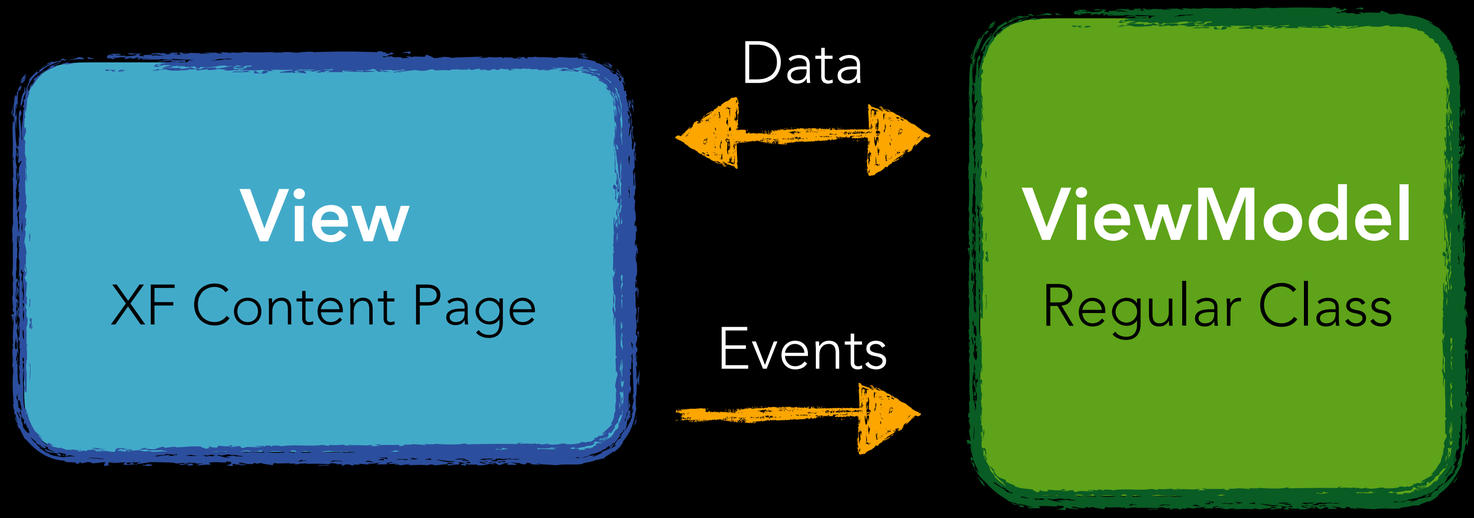


DATA BINDING

- Some basics of Model-View-ViewModel architecture (MVVM)
  - **View**: knows **how** to display data

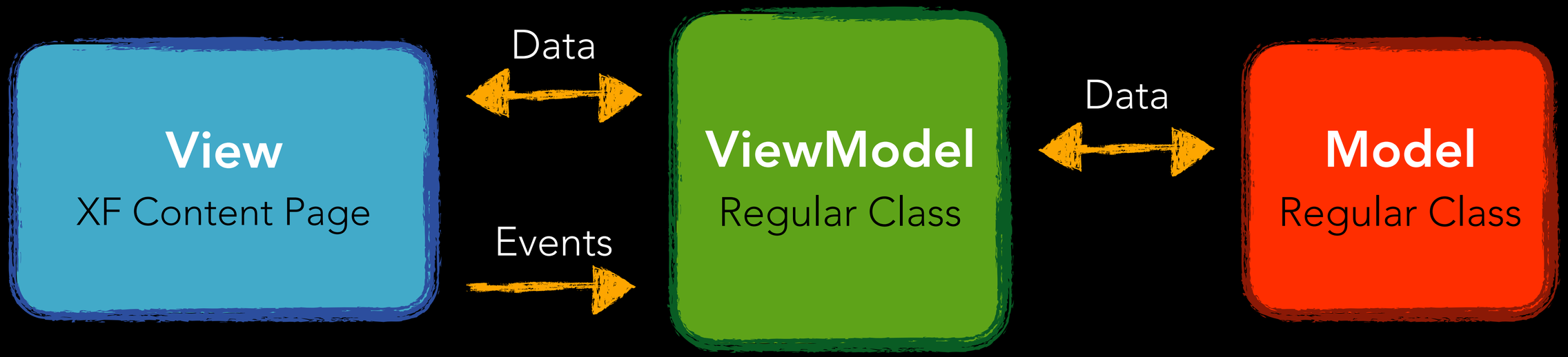
DATA BINDING

- Some basics of Model-View-ViewModel architecture (MVVM)
  - **View**: knows **how** to display data
  - **ViewModel**: knows **what** data to display



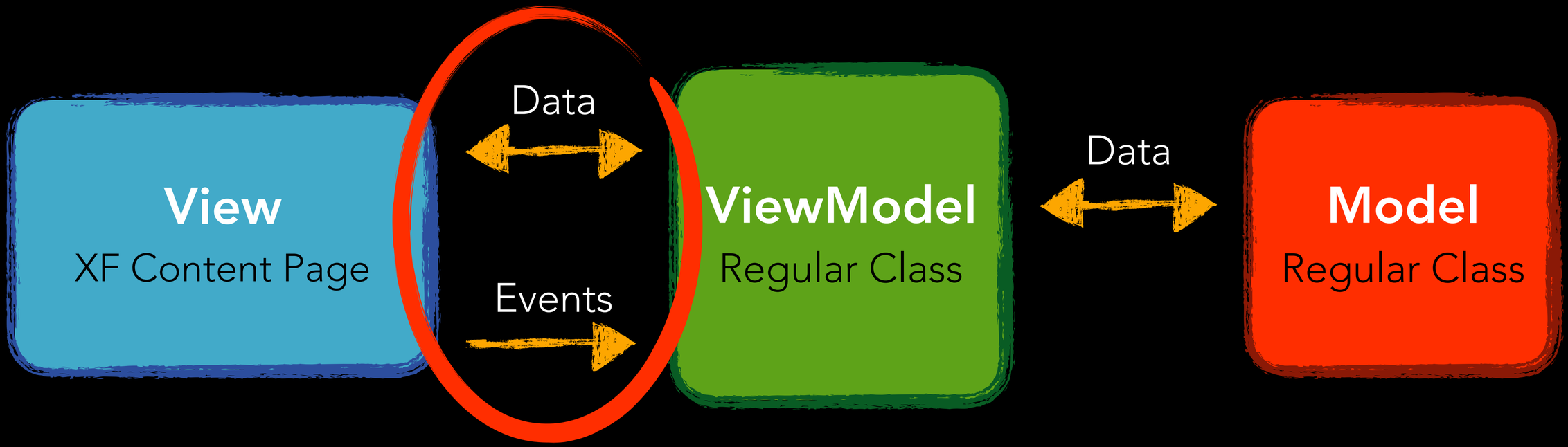
DATA BINDING

- Some basics of Model-View-ViewModel architecture (MVVM)
  - **View**: knows **how** to display data
  - **ViewModel**: knows **what** data to display
  - **Model**: The nouns of the system. Data objects



DATA BINDING

- Some basics of Model-View-ViewModel architecture (MVVM)
  - **View**: knows **how** to display data
  - **ViewModel**: knows **what** data to display
  - **Model**: The nouns of the system. Data objects



DATA BINDING

- **Let's look at some code!**
  - Bindable Properties
  - Binding Context
  - Converters

THE PLAN

- Introduction: Why, What, and When
- Overview of Xamarin.Forms Building Blocks
- Building a Xamarin.Forms UI in XAML
- Data Binding
- **View Customization**
- Next Steps & Resources

VIEW CUSTOMIZATION

- What if a Xamarin.Forms View doesn't look or behave how I want?
  - Platform-Specifics
  - Effects
  - Behaviors
  - Custom Renderers

VIEW CUSTOMIZATION

- Platform-Specifics
  - Use functionality that's only available on a specific platform
  - Some examples:

iOS	Android
VisualElement.BlurEffect	VisualElement.Elevation
VisualElement.IsShadowEnabled	Button.UseDefaultPadding
Entry.AdjustsFontSizeToFitWidth	Button.UseDefaultShadow
Entry.CursorColor	Entry.ImeOptions (set user action button)
ListView.SeparatorStyle	ListView.IsFastScrollEnabled
NavigationPage.HideNavigationBarSeparator	NavigationPage.BarHeight

VIEW CUSTOMIZATION

- Effects
  - AKA “Custom Renderers Lite”
  - Allow the native controls on each platform to be customized
  - Typically used for small styling changes
  - Benefits:
    - Simplify the customization of a control
    - Are reusable
    - Can be passed parameters to further increase reuse

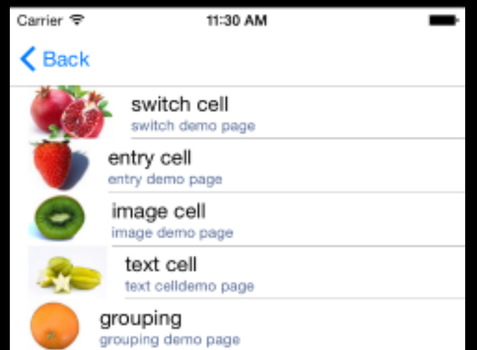
VIEW CUSTOMIZATION

- Behaviors
  - Attach additional functionality to any Xamarin.Forms View
  - Examples:
    - Only allow X number of characters to be entered into an Entry
    - Only allow integers to be entered into an Entry

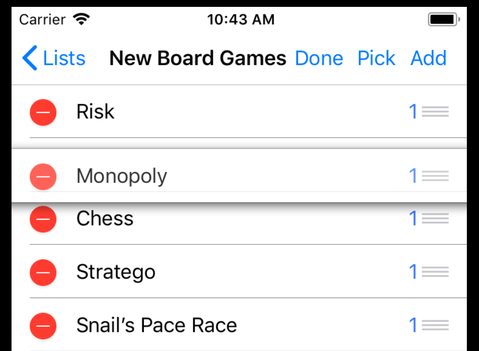
VIEW CUSTOMIZATION

- Custom Renderers
  - Let developers override the out-of-the-box renderers to customize the appearance and behavior of Xamarin.Forms controls on each platform
  - Think “can I do this with an Effect?” first
  - Extend a Xamarin.Forms View
  - Required when there's a need to override methods of a platform-specific control

out of the box



custom renderer



VIEW CUSTOMIZATION

- **Let's look at some code!**
  - Platform-Specifics
  - Effects
  - Behaviors
  - Custom Renderers

THE PLAN

- Introduction: Why, What, and When
- Overview of Xamarin.Forms Building Blocks
- Building a Xamarin.Forms UI in XAML
- Data Binding
- View Customization
- **Next Steps & Resources**

NEXT STEPS

- Using mobile device hardware features
  - Accelerometer, Barometer, Battery, Compass, Connectivity state, Device Display Information, GPS, Gyroscope, Magnetometer, Phone Dialer, Power, Secure Storage, Text-to-Speech, Vibrate, many more!
  - Using Plugins
    - NuGet packages that implement the platform features and provide an API for developers to use from shared code!
    - Xamarin.Essentials: a kit of essential API's for your apps (in preview)

NEXT STEPS

Xamarin University

New
Aug 28, 2017

FREE

Unlimited training
Live classes via Go to Meeting
1-on-1 office hours

Xamarin Certified Mobile Professional

To achieve the Mobile Professional certification, complete the list of required courses in green below and pass the Professional Certification exam.

- XAM101 - Getting Started with Xamarin
- AND101 - Introduction to Xamarin.Android
- IOS102 - Introduction to the Xamarin Designer for iOS
- AND102 - Activities and Intents
- IOS101 - Introduction to Xamarin.iOS
- XAM110 - Introduction to Cross-Platform Mobile Development
- XAM150 - Consuming REST-based Web Services
- XAM160 - SQLite and Mobile Data
- XAM220 - Preparing for Publishing
- XAM120 - Introduction to Xamarin.Forms
- XAM130 - XAML in Xamarin.Forms
- XAM140 - Resources and Styles in Xamarin.Forms
- XAM135 - Layout in Xamarin.Forms
- AND110 - ListViews and Adapters in Android
- XAM250 - Patterns for Cross Platform Mobile Development
- IOS110 - Fundamentals of TableViews
- XAM270 - Data Binding in Xamarin.Forms
- XAM280 - Using ListView in Xamarin.Forms
- IOS115 - Customizing TableViews
- AND205 - Android Navigation
- IOS205 - Navigation Patterns
- XAM370 - Diagnosing Memory Management Issues
- XAM330 - Xamarin.Forms Effects
- AND180 - Toolbar and App Bar
- XAM301 - Mobile Application Architecture

Xamarin Certified Mobile Developer

To achieve the Mobile Developer certification, complete all of the required courses listed above and pass both the Professional Certification and Developer Certification exams. Note: You must be a Xamarin Certified Mobile Professional prior to gaining eligibility to take the Developer Certification Exam.

NEXT STEPS

- Documentation
  - All Xamarin documentation has moved to Microsoft Docs. Very well done
  - https://docs.microsoft.com/en-us/xamarin
- Xamarin Slack Channel
  - https://xamarinchat.herokuapp.com
- Look into using the MVVM pattern and a good framework
  - Help you build loosely coupled, maintainable, and testable apps
  - Prism MVVM Library (many others out there)
  - http://prismlibrary.github.io

RESOURCES

- Installing Xamarin: https://docs.microsoft.com/en-us/xamarin/cross-platform/get-started/installation/index
- Xamarin Documentation: https://docs.microsoft.com/en-us/xamarin
- XAM120 class - Intro to Xamarin.Forms https://university.xamarin.com/videos/xam120-intro-to-xamarinforms
- Building your first app with Xamarin.Forms: https://www.youtube.com/watch?v=NGvn-pGZFPA
- Xamarin.Forms Feature Roadmap: https://github.com/xamarin/Xamarin.Forms/wiki/Feature-Roadmap

- Reveal https://revealapp.com
- MFractor https://www.mfractor.com
- Prism MVVM Library http://prismlibrary.github.io

- All code, slides, resources: https://github.com/TomSoderling/XF-for-Beginners