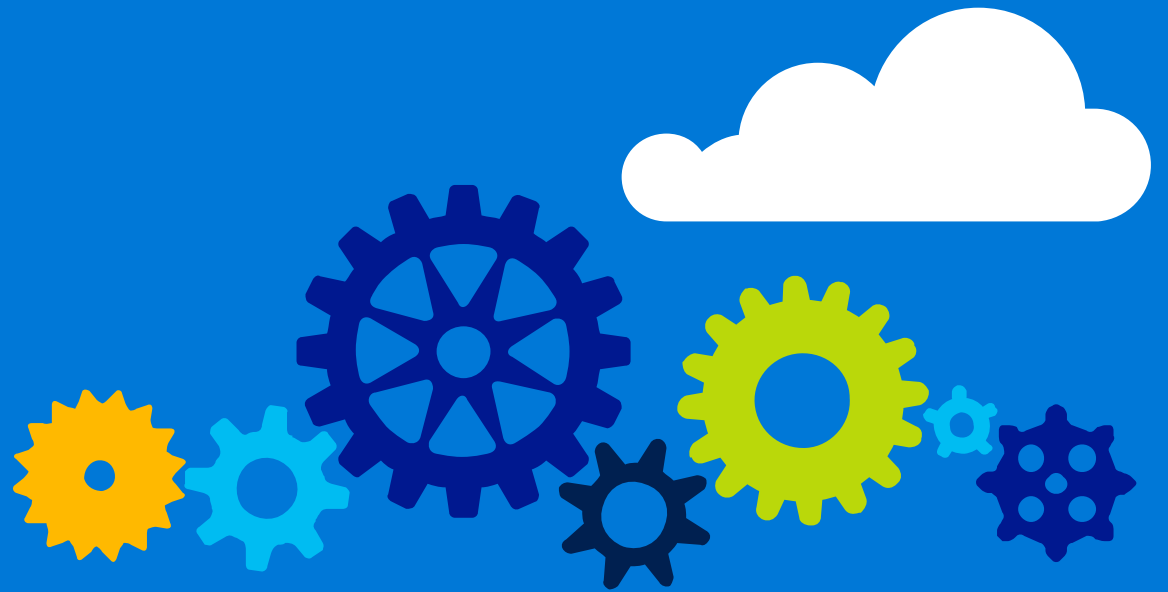


# XAM 130 – XAML in Xamarin.Forms

김은지

기술 에반젤리스트  
마이크로소프트



# Markup, Markup 언어

## Markup

- 문서의 내용 이외에 문서의 서식, 구조 등을 표현하기 위한 부가적인 정보
- 태그 등으로 불림

## Markup Language

- 마크업 정보로 표현하는 언어로 문서 중간 중간에 문장 형태의 명령어를 삽입하는 방식
- 예제
  - HTML
  - XML
  - XAML

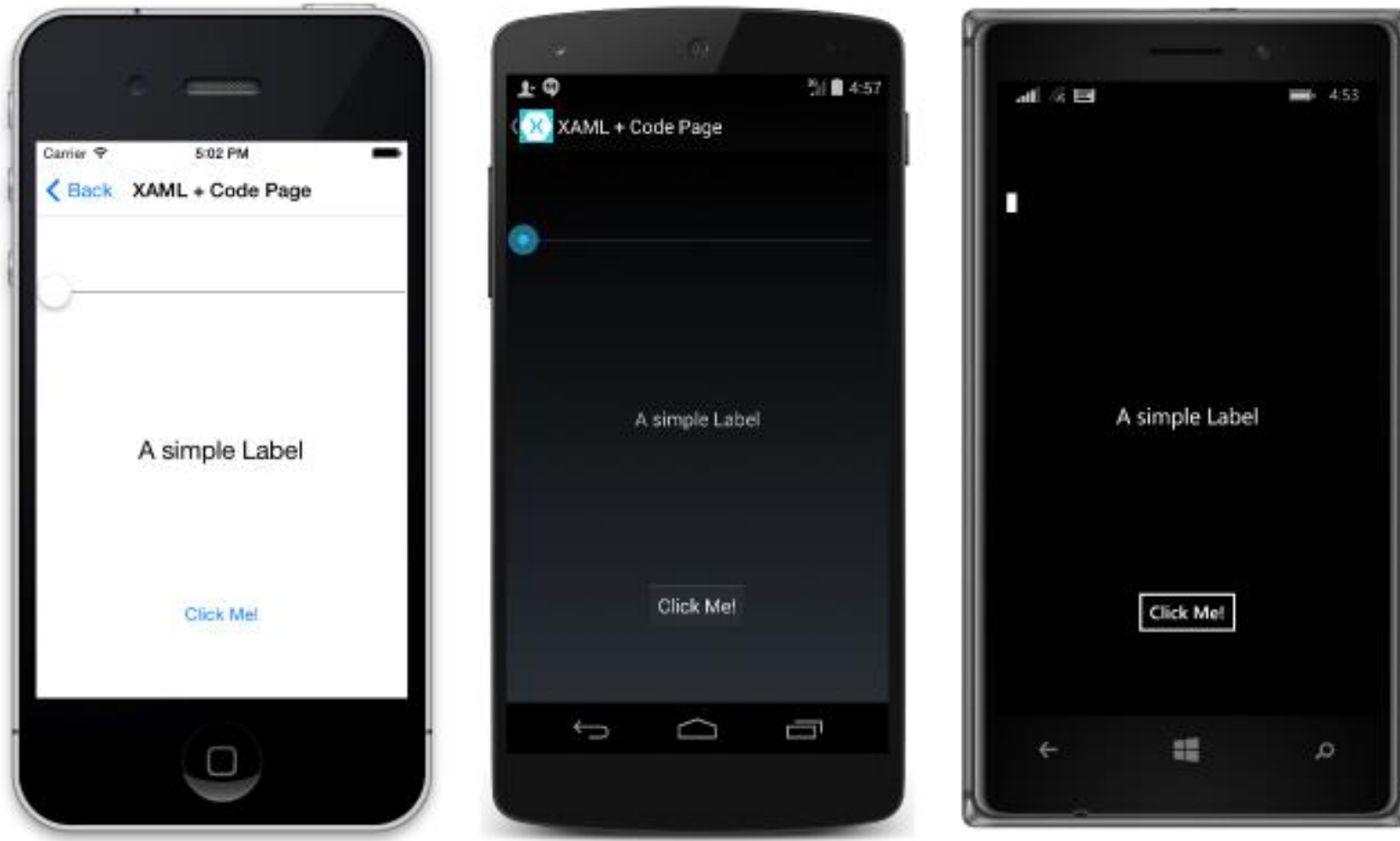
# XAML

- Extensible Application Markup Language 의 약자
- 마이크로소프트에서 만든 언어
- 주로 UI를 표현하는데 사용됨



Xamarin Forms + XAML  
= Sweetness!

# XAML 사용 예제



# XAML 사용 예제

```
Label header = new Label
{
    Text = "Button",
    Font = Font.BoldSystemFontOfSize(50),
    HorizontalOptions = LayoutOptions.Center
};

Button button = new Button
{
    Text = "Click Me!",
    Font = Font.SystemFontOfSize(NamedSize.Large),
    BorderWidth = 1,
    HorizontalOptions = LayoutOptions.Center,
    VerticalOptions = LayoutOptions.CenterAndExpand
};
button.Clicked += OnButtonClicked;

label = new Label
{
    Text = "0 button clicks",
    Font = Font.SystemFontOfSize(NamedSize.Large),
    HorizontalOptions = LayoutOptions.Center,
    VerticalOptions = LayoutOptions.CenterAndExpand
};

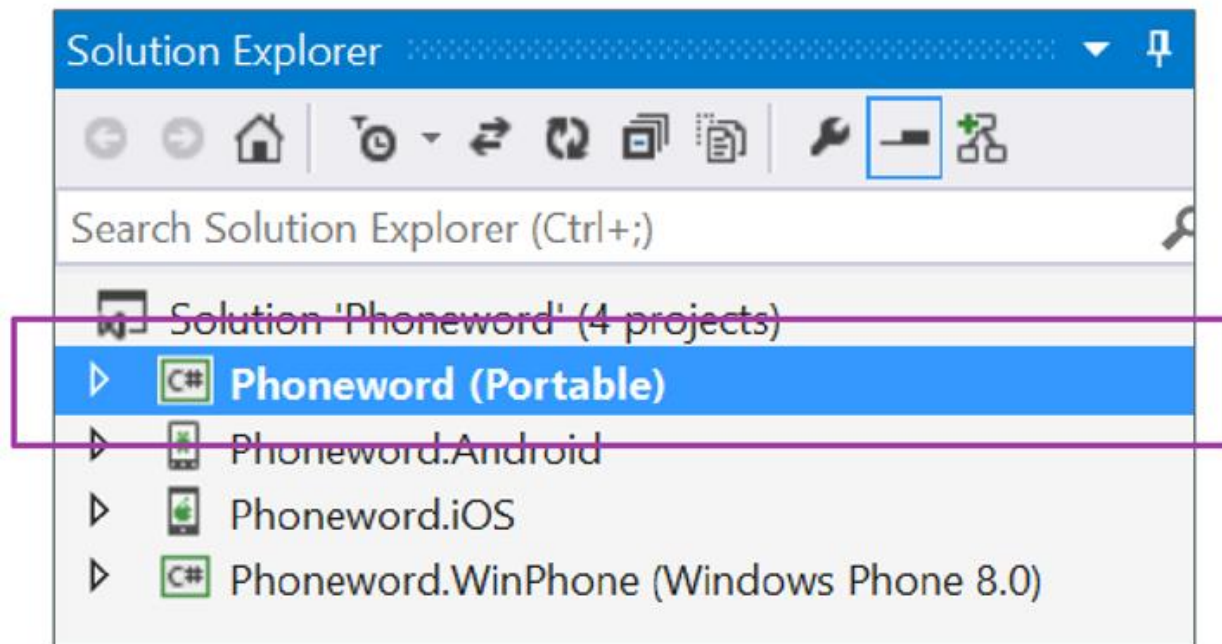
// Accomodate iPhone status bar.
this.Padding = new Thickness(10, Device.OnPlatform(20, 0, 0), 10, 5);

// Build the page.
this.Content = new StackLayout
{
    Children =
    {
        header,
        button,
        label
    }
};
```

```
<StackLayout>
    <Label Text="Button"
           VerticalOptions="Center"
           FontAttributes="Bold"
           FontSize="Large"/>
    <Button Text="Click Me!"
            FontSize="Large"
            BorderWidth="1"
            HorizontalOptions="Center"
            VerticalOptions="CenterAndExpand"
            Clicked="OnButtonClicked"/>
    <Label Text="0 button clicks"
           VerticalOptions="CenterAndExpand"
           HorizontalOptions="Center"
           FontSize="Large"/>
</StackLayout>
```

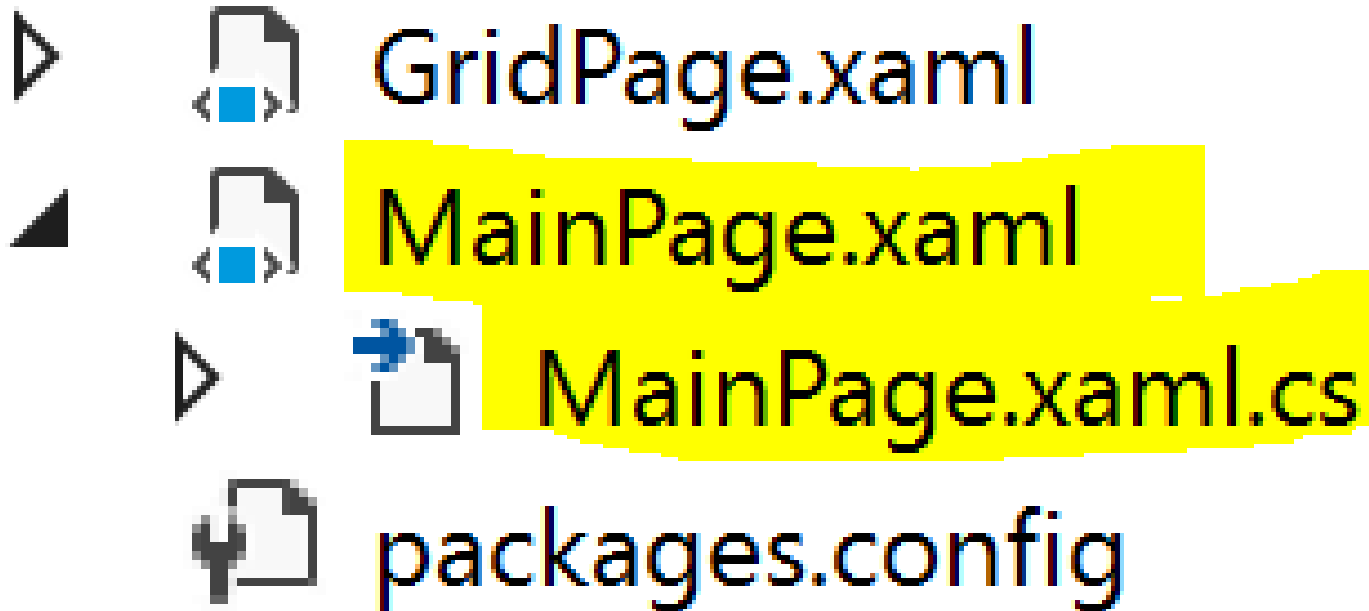
# XAML 페이지가 어디에 포함되어야 하는가

- XAML을 이용해 공통으로 사용할 수 있는 UI를 작성하는 것이기 때문에 PCL프로젝트에 포함되어야 한다




# 어라? 1개가 아니네요?

- 사실상 2개가 생성됩니다 – xaml과 xaml.cs 라는 파일 입니다



# XAML 파해치기

- 노란색 표기가 되어있는 부분은 C#의 object에 해당합니다



```
<?xml version="1.0" encoding="UTF-8" ?>
<ContentPage ...>
    <StackLayout Padding="20" Spacing="10">
        <Label Text="Enter a Phoneword:" />
        <Entry Placeholder="Number" />
        <Button Text="Translate" />
        <Button Text="Call" IsEnabled="False" />
    </StackLayout>
</ContentPage>
```



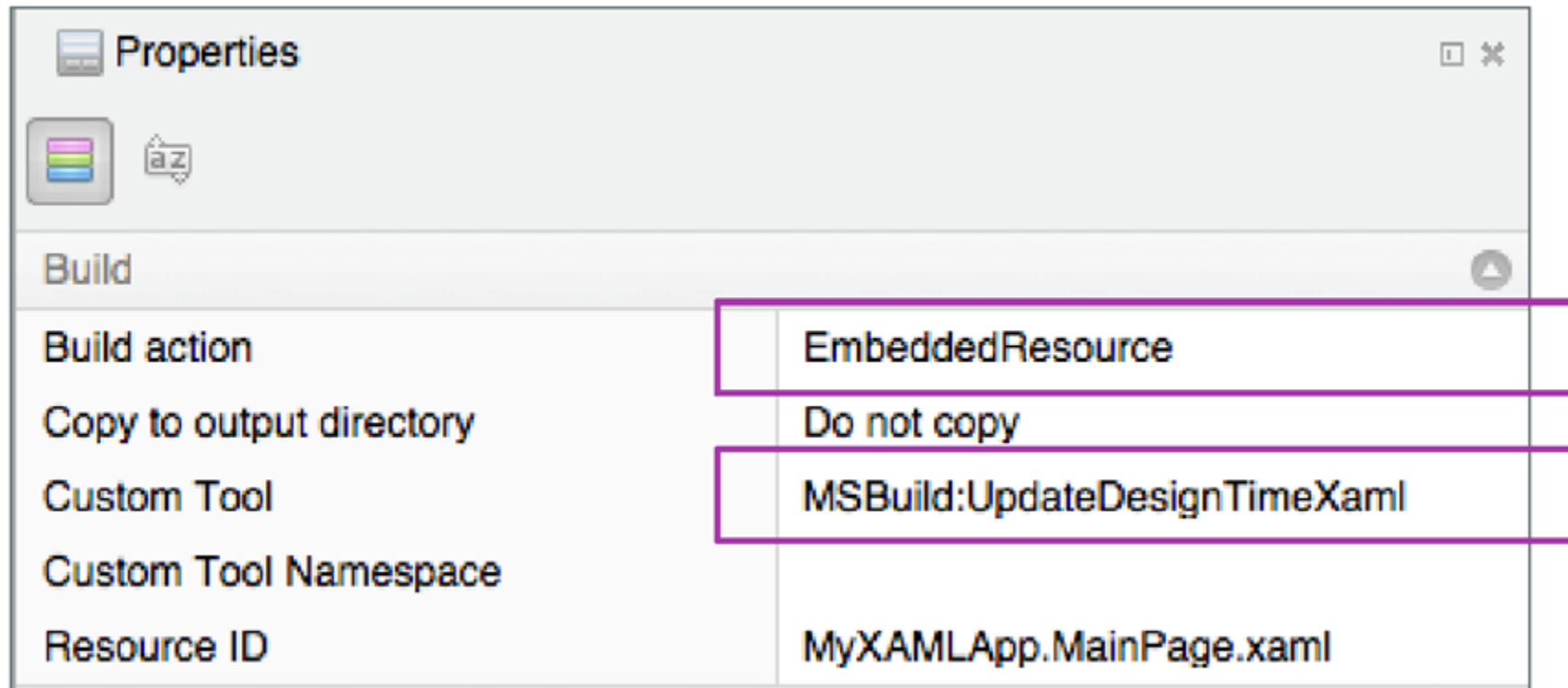
# XAML 파해치기

- 노란색 표기가 되어있는 부분은 C#의 attribute에 해당하는 것으로 property나 event를 생성합니다

```
<?xml version="1.0" encoding="UTF-8" ?>
<ContentPage ...>
    <StackLayout Padding="20" Spacing="10"> ←
        <Label Text="Enter a Phoneword:" />
        <Entry Placeholder="Number" />
        <Button Text="Translate" />
        <Button Text="Call" IsEnabled="False" />
    </StackLayout>
</ContentPage>
```

# XAML 빌드 타입

- XAML 파일은 빌드될 때 embedded resources 로 특수한 빌드 타입인 “MSBuild:UpdateDesignTimeXaml”를 취합니다



# XAML + Code Behind

- XAML 과 Code Behind 파일은 사실 하나의 파일입니다  
(하나의 클래스가 두개로 나뉘어져있다가 나중에 한개로 합쳐집니다)


```
<?xml version="1.0" encoding="UTF-8" ?>  
<ContentPage x:Class="Phoneword.MainPage" ...>
```

```
namespace Phoneword  
{  
    public partial class MainPage : ContentPage  
    {  
        ...  
    }  
}
```

# XAML 초기화

- Code Behind 파일의 생성자는 InitializeComponent() 메소드를 호출합니다
- XAML을 로딩하고 Object를 생성하는 역할을 합니다

```
public partial class MainPage : ContentPage
{
    public MainPage ()
    {
        InitializeComponent ();
    }
}
```



# Property 변환

- XML attributes는 문자열 값만 가능하기 때문에 원래 가진 타입 값으로 변환이 됩니다

```
<Label Text="This is a Label" isVisible="True" opacity="0.75"  
      FontAttributes="Bold,Italic" FontSize="Large"  
      Margin="5,20,5,0" TextColor="#fffc0d34" />
```

String 그대로  
유지됩니다

# Property 변환

- XML attributes는 문자열 값만 가능하기 때문에 원래 가진 타입 값으로 변환이 됩니다

```
<Label Text="This is a Label" isVisible="True" Opacity="0.75"  
FontAttributes="Bold,Italic" FontSize="Large"  
Margin="5,20,5,0" TextColor="#fffc0d34" />
```

Bool 타입으로  
변환됩니다

# Identifying Types

기본 네임스페이스로 대부분의 Xamarin.Forms의 타입들을 포함합니다



```
<ContentPage ...  
  xmlns="http://xamarin.com/schemas/2014/forms"  
  xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml">  
  
  <StackLayout ... />  
  
</ContentPage>
```

The diagram shows a code block with XAML code. A purple arrow points from the text above to the first namespace URI, and another purple arrow points from the text below to the 'x' namespace URI.


x: 네임스페이스는 XAML 타입과 CLR 타입들을 포함합니다

# XAML에서 변수명 정하고 싶을때..

MainPage.xaml

```
<Entry x:Name="PhoneNumber"  
        Placeholder="Number" />
```

x:Name 을 이용하여 변수명을  
정할 수 있습니다



```
public partial class MainPage : ContentPage  
{  
    private Entry PhoneNumber;  
  
    private void InitializeComponent() {  
        this.LoadFromXaml(typeof(MainPage));  
        PhoneNumber = this.FindByName<Entry>(  
            "PhoneNumber");  
    }  
}
```

MainPage.xaml.g.cs



# XAML에서 이벤트 처리

- XAML에 이벤트를 선언해놓고 Code behind 클래스에 이벤트를 구현합니다

```
<Entry Placeholder="Number" TextChanged="OnTextChanged" />
```

```
public partial class MainPage : ContentPage
{
    ...
    void OnTextChanged(object sender, TextChangedEventArgs e) {
        ...
    }
}
```

# 실습 130 – Lab1, Lab2

계산기 만들어 보기

