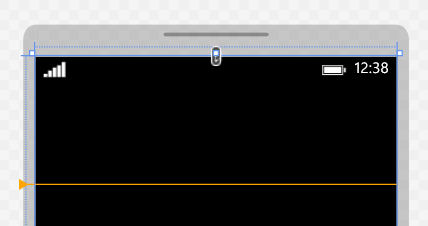
Windows Phone 8.1 App 개발 실습(ContosoCookbook)

이번 세션에서는 레이아웃 컨트롤, 앱바, 시스템 트레이, 스타일과 테마를 다루는 법에 대해 설명하는 것을 목적으로 합니다.

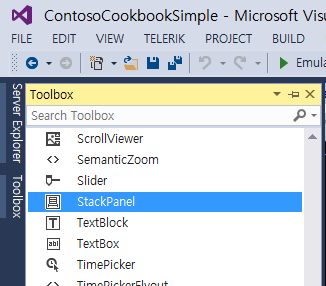
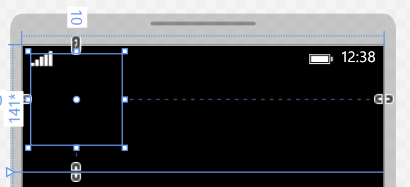
# 레이아웃 컨트롤 다뤄보기

페이지의 레이아웃을 디자인하고 Grid와 StackPanel 같은 기본 레이아웃 컨트롤에 대해 다뤄봅니다.

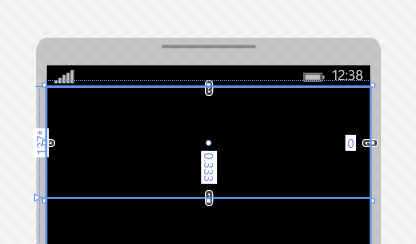
1. 2 - ContosoCookbookSimple BEGIN/ContosoCookbookSimple 폴더에서 솔루션 파일을 엽니다.
2. Solution Explorer에서 MainPage.xaml 파일을 더블클릭하여 엽니다.
3. XAML 코드 편집기에서 <Grid> 컨트롤을 선택합니다.
4. 아래 그림과 같이 Grid를 두 줄로 나누기 위해 디자이너 화면에서 Grid의 모서리를 클릭합니다. 대략 제목 부분과 컨텐츠 부분으로 나누어지도록 합니다.



1. Toolbox로부터 StackPanel 컨트롤을 마우스로 드래그앤 드랍하여 위 쪽 그리드의 좌측 상단에 위치하게 합니다.

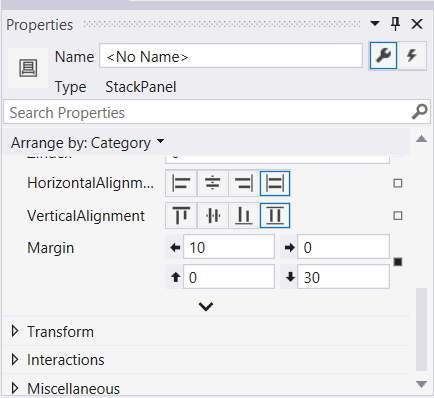
 

1. 디자이너화면의 StackPanel을 마우스 우클릭 한후 Layout > Reset All 의 순서로 선택을 합니다. 이로써 StackPanel의 여백을 없애고 StackPanel이 상단의 Grid를 가득 채우게 됩니다.

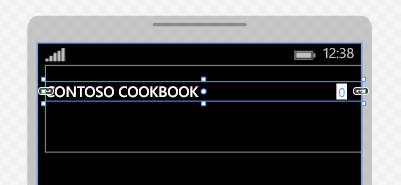


1. Properties 창에서 Layout 영역을 확장하고 Margin을 좌측 10 pixel, 하단 30 pixel로 설정합니다.

(10 pixel은 Windows XAML 앱에서 주로 사용되는 크기입니다.)



1. Toolbox로부터 TextBlock 컨트롤을 마우스로 드래그앤 드랍하여 StackPanel 영역에 추가합니다.
2. Properties 창의 Common 영역을 확장하고 Text를 **CONTOSO COOKBOOK**으로 변경합니다.
3. 디자이너 화면에서 TextBlock을 마우스 우클릭 한후 Edit Style > Apply Resource > **TitleTextBlockStyle** 의 순서로 선택을 합니다.
4. Properties 창의 Layout 영역을 확장하고 Margin을 상단 20 pixel로 설정합니다.



1. Toolbox로부터 두 번째 TextBlock 컨트롤을 마우스로 드래그앤 드랍하여 StackPanel 영역에 추가합니다.
2. Properties 창의 Common 영역을 확장하고 Text를 **quiche Lorraine** 으로 변경합니다.

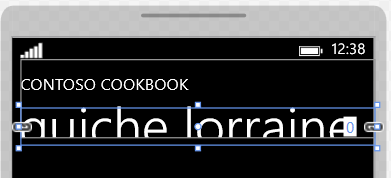
코드 편집기에서 두 번째 TextBlock의 속성값에 아래와 같이 추가합니다. 10번과 같이 디자이너 화면에서 우클릭 메뉴로 설정하는 방법으로 하셔도 됩니다.

<TextBlock TextWrapping="Wrap" Text="quiche Lorraine" Style="{StaticResource HeaderTextBlockStyle}"/>

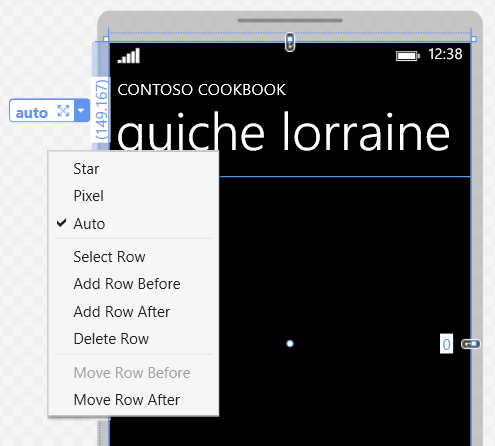
1. Properties 창의 Layout 영역을 확장하고 Margin을 상단 20 pixel로 설정합니다.
2. 코드 편집기에서 기존의 TextWrapping 속성을 “NoWrap” 으로 변경하고 TextTrimming 속성을 아래와 같이 추가합니다.

<TextBlock TextWrapping="NoWrap" TextTrimming="CharacterEllipsis" Text="quiche Lorraine" Style="{StaticResource HeaderTextBlockStyle}" Margin="0,20,0,0"/>

1. 지금까지 잘 진행하셨다면 현재 디자이너 화면의 모습은 아래와 같을 것입니다. 글씨가 잘리거나 보이지 않으실 수 있습니다. 이를 수정하기 위해서



1. 디자이너 화면이나 코드 편집기에서 Grid을 선택한 후 아래 그림과 같이 높이를 auto로 설정합니다.



1. 다음으로는 기존 Grid의 2개의 영역중 아래 영역에 Grid를 추가한 후 3개의 행으로 분할하는 작업을 할 것 입니다. 앞서 유사한 작업을 디자이너 화면에서 수행했지만 이번에는 아래 코드를 코드 편집기에서 추가하여 수행해 보겠습니다.

<!--StackPanel 종료부분-->

<Grid Grid.Row="1">

**<Grid.RowDefinitions>**

**<RowDefinition Height="109\*"/>**

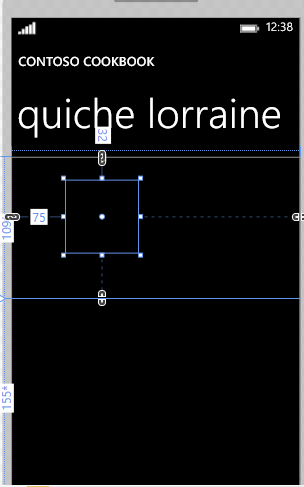
**<RowDefinition Height="155\*"/>**

**<RowDefinition Height="Auto"/>**

**</Grid.RowDefinitions>**

</Grid>

1. Toolbox로부터 Image 컨트롤을 드래그앤 드랍하여 두 번째 그리드의 첫 번째 행에 추가합니다.



1. 추가된 Image 컨트롤을 마우스 우클릭 한후 Layout > Rest All 의 순서로 선택합니다.
2. Properties 창에서 Common 영역을 확장하고 **Source** 속성에 아래의 이미지 경로를 추가합니다. Stretch 속성은 UniformToFill 로 변경합니다.

**Images/french/French\_1\_600\_C.jpg**

1. Toolbox로부터 ScrollViewer 컨트롤을 드래그앤 드랍하여 이미지 아래 두 번째 행에 추가합니다.
2. 추가된 Scrollviewer 컨트롤을 마우스 우클릭 한후 Layout > Rest All 의 순서로 선택합니다. 코드 편집기의 코드는 아래와 같을 것입니다.

<ScrollViewer Grid.Row="1"/>

1. 위 코드를 보면 ScrollViewer 컨트롤의 시작과 끝이 한 줄로 선언되어 있습니다. 이를 아래와 같이 세 줄로 바꿔줍니다. 앞 뒤 코드 가운데 비어있는 줄을 한 줄 넣습니다.

<ScrollViewer Grid.Row="1">

</ScrollViewer>

1. 위 코드에서 두 번째 줄(비어 있는 영역)을 클릭하고 마우스 우클릭 후 Insert Snippet 메뉴를 선택한 후 **JumpStart Contoso Cookbook Instructions TextBlock** 코드스니펫을 추가합니다. 완료시 디자이너의 편집기의 모습은 아래와 같습니다.



1. Image 컨트롤의 좌우측 Margin을 10으로 설정합니다.
2. ScrollViewer의 상단과 좌우측 Margin을 10로 설정합니다.



1. 코드 편집기에서 ScrollViewer 종료 부분(</ScrollViewer>) 아랫줄에 빈 영역을 만듭니다.
2. 빈 영역을 선택 후 마우스 우클릭 > Insert Snippet 메뉴 > **JumpStart Contoso Cookbook Preptime** 을 선택하여 아래와 같이 코드를 추가되는지 확인합니다.

</ScrollViewer>

**<StackPanel Grid.Row="2" Orientation="Horizontal">**

**<TextBlock Text="Prep time:" Style="{StaticResource BodyTextBlockStyle}"**

**Foreground="{StaticResource PhoneAccentBrush}"/>**

**<TextBlock x:Name="PrepTimeTextBlock" Text="120" Style="{StaticResource BodyTextBlockStyle}" Margin="10,0,0,10"/>**

**</StackPanel>**

1. 완료 시 디자이너 화면의 모습은 아래와 같습니다. 애플리케이션을 실행시켜 시뮬레이터에서도 동일한 화면이 나오는지 확인합니다.

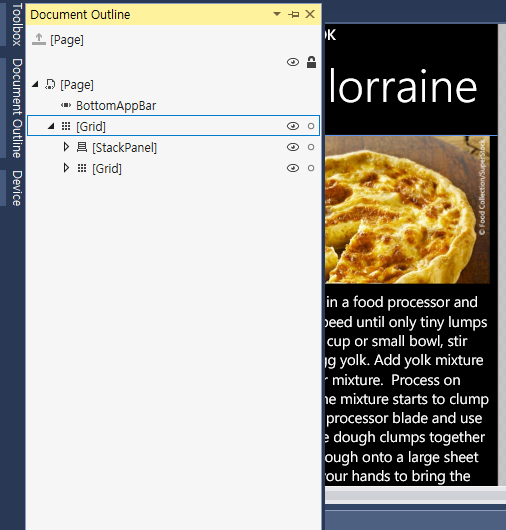


이제 ContosoCookbookSimple BEGIN 솔루션 파일을 저장하고 Session 2 Demo 폴더내 2-ContosoCookbookSimple END 폴더의 솔루션 파일을 열어서 비교해 봅니다.

# 앱 바 추가하기

앞서 만들어둔 애플리케이션에 앱 바를 추가해 보도록 합니다.

1. Session 2 폴더 내 **3A - AppBar BEGIN** 폴더의 ContosoCookbookSimple 솔루션 파일을 열거나, 지금까지 작업했던 솔루션에서 계속해서 작업합니다.
2. 좌측의 **Document Outline** 창을 클릭하여 엽니다.



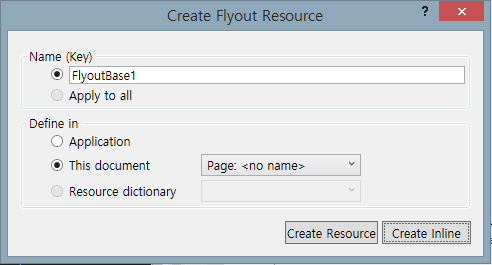
1. **BottomAppBar** 를 마우스 우 클릭한 후 **Add CommandBar** 메뉴를 선택합니다.
2. 디자이너 화면에서 하단에 앱 바와 두개의 버튼이 생성되었음을 확인합니다. 본 세션에서는 하나의 버튼만 필요하므로 둘중 하나를 선택 한 후 지워줍니다.



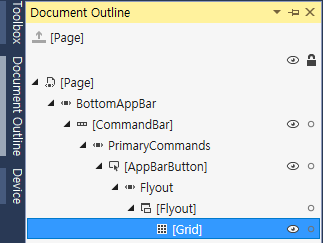
1. 남은 하나의 버튼을 선택한 후 Properties 창의 Icon 영역을 확장합니다. Symbol 속성의 드랍다운 메뉴를 클릭한 후 List를 선택합니다.
2. Label 속성의 값을 **ingredients** 로 입력한 후 잘 변경되었는지 확인합니다.

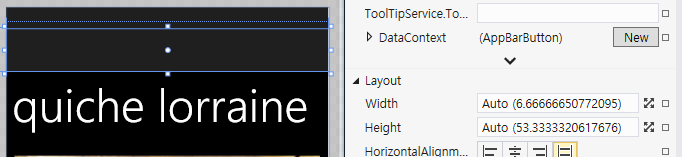


1. Document Outline 창을 다시 열고 AppBarButton을 마우스 우클릭 한 후 **Add Flyout** 메뉴를 선택합니다. 이 후 열리는 창에서 Create Inline 버튼을 클릭합니다.



1. Document Outline 창에서 Flyout을 확장 합니다. Flyout 아래의 Grid를 선택한 후 Properties 창에서 Width와 Height 값을 Auto로 변경합니다.





현재까지 잘 수행하셨다면 코드 편집기의 모습은 아래와 같을 것입니다.

<Page.BottomAppBar>

<CommandBar>

<AppBarButton Icon="List" Label="ingredients">

<AppBarButton.Flyout>

<Flyout>

<Grid/>

</Flyout>

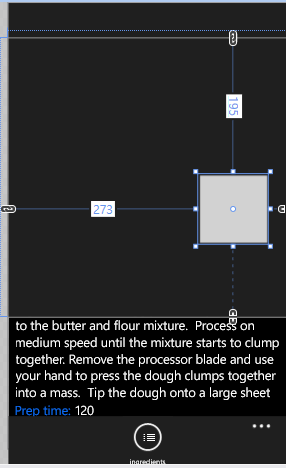
</AppBarButton.Flyout>

</AppBarButton>

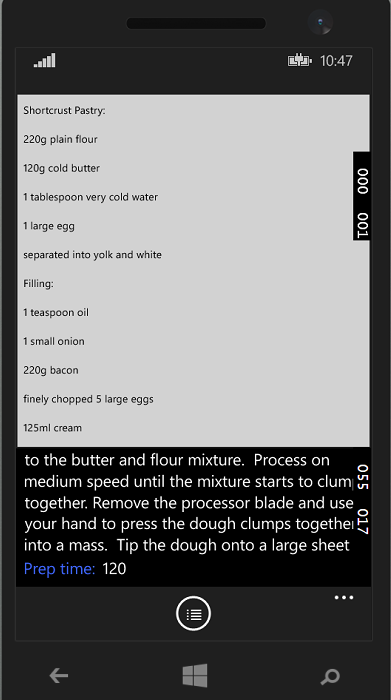
</CommandBar>

</Page.BottomAppBar>

1. 코드 편집기에서 <Grid/> 라인을 클릭합니다. Properties 창의 Layout영역을 확장하여 Height 속성값을 400, Margin은 상단 25로 설정합니다
2. Toolbox로 부터 ListBox 컨트롤을 Grid위에 드래그앱 드랍 하여 추가합니다.



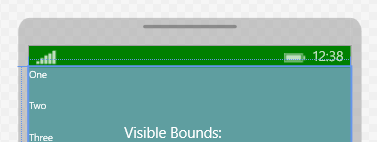
1. 추가된 Listbox 컨트롤을 선택한 상태로Properties창에서 Width와 Height를 Auto로 Margin을 모두 0으로 변경합니다. 이 후 Properties 창 상단의 Name 속성을 IngredientsListBox로 변경합니다.
2. 추가된 Listbox 컨트롤을 마우스 우클릭 한 후 View Code 메뉴를 선택합니다. 조금 전 Flyout 내에 디자인한 Listbox에 데이터를 보여주는 소스코드를 코드 스니펫을 통해 추가할 것입니다.
3. OnNavigatedTo 함수에는 주석을 제외하곤 내용이 들어 있지 않습니다. 함수 내에서 마우스 우클릭 후 C# JumpStart > JumpStart Contoso Cookbook Ingredients List 코드 스니펫을 선택합니다.
4. 애플리케이션을 실행 한 후 앱바의 버튼을 클릭했을 때 아래와 같이 Listbox가 잘 나오는지 확인합니다.



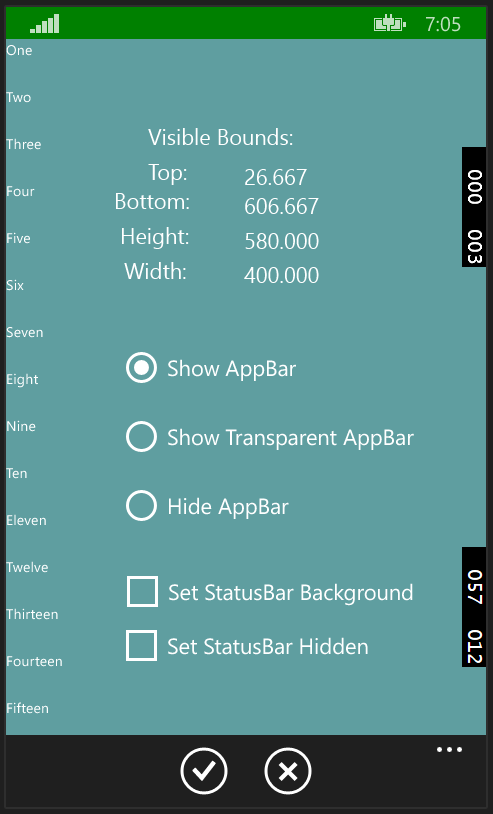
# Visible Bounds 값 나타내 보기

AppBar와 SystemTray 영역을 다루는 방법에 대해 알아 봅니다.

1. **3B – System UI – VisibleBoundsDemo** 폴더의 솔루션을 엽니다.
2. MainPage.xaml 파일을 더블클릭하여 엽니다. (디자이너 화면이 나오지 않는 경우 Visual Studio 메뉴의 Build > Rebuild Solution 을 선택하여 다시 빌드해 줍니다.)
3. 디자이너 화면에서 여러 컨트롤들을 확인해 봅니다. 최 상단 StatusBar의 경우 녹색으로 표시되어 있는데, 본래 기본값으로 투명하게 되어있지만 배경색을 녹색으로 지정했기 때문에 그렇습니다.



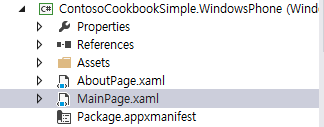
1. 애플리케이션을 실행합니다. 화면에 보이는 Visible Bounds 숫자들은 앱의 푸른색 영역의 Margin과 크기를 나타내고 있습니다.



1. 위 숫자들을 계산하는 함수를 소스코드에서 확인하여 봅니다. Solution Explorer에서 MainPage.xaml.cs 파일을 더블클릭한 후 검색 메뉴(Ctrl + F)를 실행합니다.
2. 검색 키워드로 ReportVisibleBounds()를 입력하여 함수의 내용을 확인합니다.
3. 실행중인 에뮬레이터에서 Show Transparent AppBar 버튼을 선택합니다. Visible Bounds 값들은 변하지 않고 배경색만 바뀌게 됩니다.
4. Hide AppBar 버튼을 선택한 후 Visible Bounds 값들이 바뀌는 것을 확인합니다.
5. Set StatusBar Background 체크박스를 선택합니다. StatusBar의 배경색을 변경할 수 있으며 디자이너 화면에서는 구현이 불가능하고 소스코드를 추가하여 변경할 수 있습니다.
6. Set StatusBar Hidden 체크박스를 선택합니다. StatusBar를 감출 수 있고, 이 때 Visible Bounds 값들이 바뀌는 것을 확인할 수 있습니다.

# 스타일과 테마 지정하기

1. **4 – Styles and Themes** 폴더의 ContosoCookbookSimple 솔루션을 엽니다. 이번 솔루션은 Universal App이므로 Windows 8.1 프로젝트와 Windows Phone 8.1 프로젝트, Shared 프로젝트로 구성되어 있습니다.
2. ContosoCookbookSimple.WindowsPhone 프로젝트의 MainPage.xaml 파일을 더블클릭하여 엽니다.



1. 요리법의 설명이 있는 TextBlock (아래 참조) 의 Foreground 속성을 추가하고 White로 설정합니다.

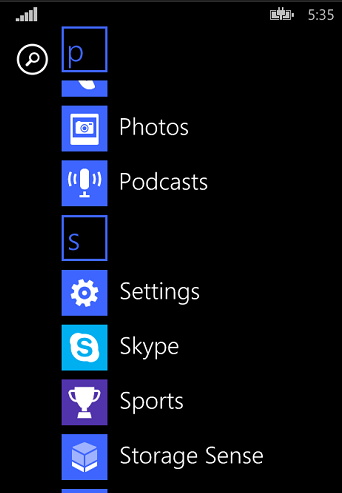
<TextBlock x:Name="InstructionsTextBlock" TextWrapping="Wrap"

Style="{StaticResource BodyTextBlockStyle}"

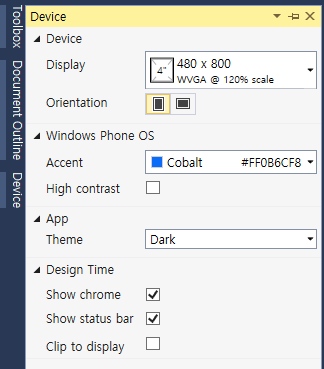
Text="{Binding Directions}"

Foreground="White"/>

1. Solution Explorer에서 CotosoCookbookSimple.WindowsPhone프로젝트를 우측 클릭한 후 Set as Startup Project로 시작프로젝트로 설정한 후 애플리케이션을 실행합니다. Windows Phone 에뮬레이터에서 앱이 실행되는 것을 확인합니다.
2. 에뮬레이터에서 윈도우키를 눌러 시작화면으로 간뒤 하단의 -> 버튼을 클릭하여 앱 목록으로 이동합니다.
3. 하단의 Settings 앱을 실행 시킨 후 Start+Theme 메뉴를 선택합니다.



1. Background 값이 dark 로 설정되어있습니다. light 로 바꿔준 후 Visual Studio로 돌아와서 애플리케이션 실행을 정지합니다.
2. 애플리케이션을 다시 한번 실행합니다. 하얀색 배경으로 변경되어 이제는 하얀 글씨의 요리법 설명이 보이지 않습니다.
3. 좀 전에 변경한 Background 설정값은 물론이고 강조색, 고대비 설정, 디바이스 방향, 스크린사이즈등을 아래와 같이 Device 창에서도 변경할 수 있습니다.



1. 코드 편집기에서 **Foreground=”White”** 부분을 삭제합니다.
2. 디자이너 화면에서 InstructionsTextBlock (요리법 설명이 있는) 을 선택 후 마우스 우클릭 > Edit Style > ApplyResource 의 순으로 선택하여 스타일을 변경해 봅니다.

스타일들은 (e.g. {StaticResource BodyTextBlockStyle} ) 코드 편집기에서 마우스 우클릭하면 각각이 어떻게 정의되어 있는지를 볼 수 있습니다. 또한 아래의 예제 코드와 같이 개발자가 자신만의 스타일을 생성하고, BasedOn 속성을 사용하여 상속받고 오버라이드 할수도 있습니다.

<Style x:Key="BodyTextBlockStyle" TargetType="TextBlock" BasedOn="{StaticResource BaseTextBlockStyle}">

<Setter Property="LineHeight" Value="24" />

</Style>

이러한 커스텀 스타일들은 App.xaml 파일에 넣을 수 있고, 새로운 xaml 파일을 생성하여 넣을수도 있습니다.