

Resource Dictionary

```
<ContentPage.Resources>
  <ResourceDictionary>
    <x:Int32 x:Key="borderRadius">20</x:Int32>
    <x:String x:Key="...">...</x:String>
    <Color x:Key="...">...</Color>
  </ResourceDictionary>
</ContentPage.Resources>
```

If resources need to be shared across multiple-pages, you should add them to the App.

To reference resources:

```
<Button BorderRadius="{StaticResource borderRadius}" />
```

Dynamic Resources

If the value of the resource changes, the references get updated.

Explicit Styles

```
<ContentPage.Resources>
    <ResourceDictionary>
        <Style x:key="button" TargetType="Button">
            <Setter Property="BackgroundColor" Value="Gray" />
            ...
        </Style>
    </ResourceDictionary>
</ContentPage.Resources>
```

Using a style:

```
<Button Style="{StaticResource button}" />
```

Style Inheritance

```
<Style x:key="primaryButton" TargetType="Button"
    BasedOn="{StaticResource button}" />
```

Implicit Styles

Remove the key. The style gets applied automatically based on TargetType:

Beyond the Basics

By: Mosh Hamedani

```
<Style TargetType="Button">  
    <Setter Property="BackgroundColor" Value="Gray" />  
</Style>
```

Messaging Center

In the publisher:

```
MessagingCenter.Send(this, "SliderValueChanged", e.NewValue);
```

In the subscriber:

```
MessagingCenter.Subscribe<TargetPage, double>  
    (this, "SliderValueChanged", (source, arg) => { });
```

We can also unsubscribe (in the subscriber):

```
MessagingCenter.Unsubscribe<MainPage>  
    (this, "SliderValueChanged");
```

Accessing Common Mobile Device Functionality

<http://github.com/xamarin/XamarinComponents>