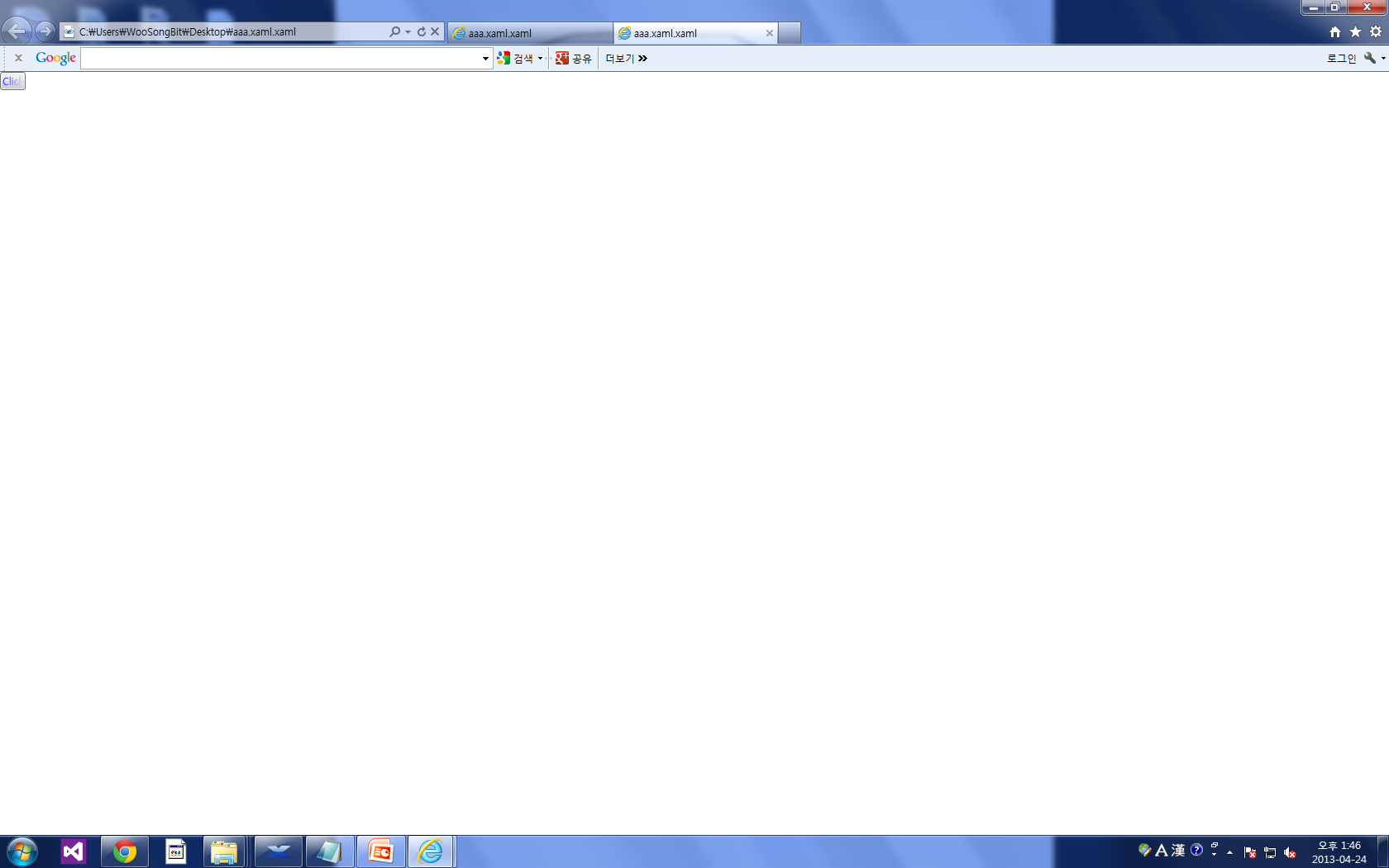


<Button Content = "Click" Foreground ="Blue"></Button>



<Button Content = "Click">

<Button.Foreground> ProPerty Element를 이용하려면

<엘리먼트 이름. 프로퍼티이름>

<요소 이름. 속성이름>

<LinearGradientBrush>

<GradientStop Offset="0" Color = "Blue"/>

<GradientStop Offset="1" Color = "White"/>

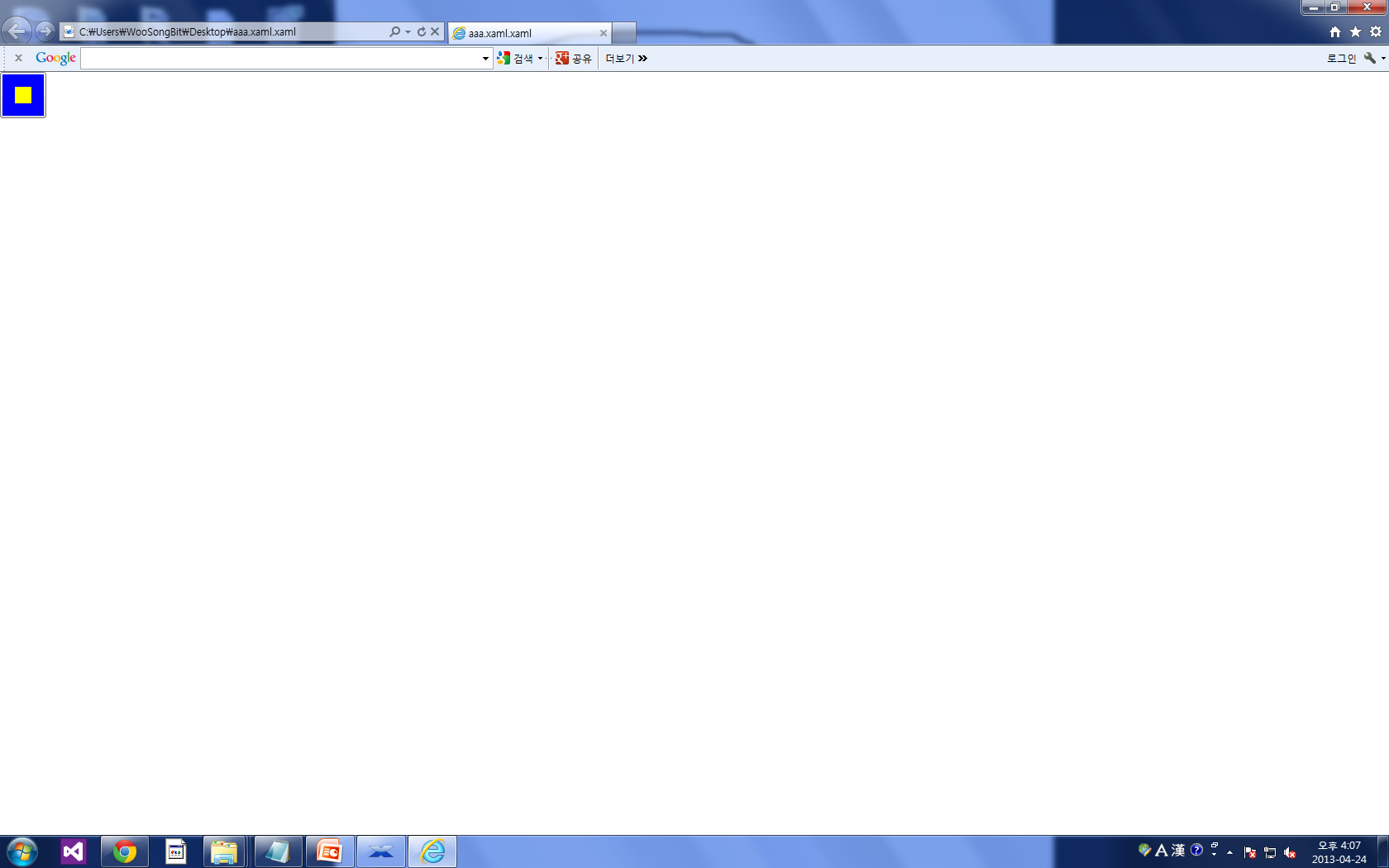
</LinearGradientBrush>

</Button.Foreground>

</Button>

이 코드는 Foreground 프로퍼티에 LinearGradientBrush object를 생성해

각각의 속성을 대입하는 코드



<Button>

<Grid>

<Rectangle Width = "50" Height = "50" Fill="Blue"/>

<Rectangle Width = "20" Height = "20" Fill="yellow"/>

</Grid>

</Button>

Content프로퍼티는 Object로 정의

Content프로퍼티는 오직 하나의 자식만 할당가능

<Button Content = "Click">

<Button.Foreground> ProPerty Element를 이용하려면

<엘리먼트 이름. 프로퍼티이름>

<요소 이름. 속성이름>

<LinearGradientBrush>

<GradientStop Offset="0" Color = "Blue"/>

<GradientStop Offset="1" Color = "White"/>

</LinearGradientBrush>

</Button.Foreground>

</Button>

Button button = new Button {Foreground = new  LinearGradientBrush(Colors. Blue) };

위 코드처럼 보통의 코드들은 내부적으로 new 연산을 수행 고로 이미 존재하는 object에 대한

사용은 어려움 기존에 존재하는 Object의 참조자를 Property에 대입하고 싶을 때

기존의 XML문법으로는 이 방법을 고안하기가 쉽지 않다.

Markup Extension은 문법에서 확장했다는 의미가 포함 되어 있고

'{'로 시작하고 '}'로 끝나게 됩니다.

Markup Extension은 자신에게 필요한 것이 없다면 본인 스스로 확장도 가능하다.