**컨텐츠 (Content)**

**컨텐츠의 정의** : WPF는 새로운 컨텐츠 모델로 설계되어 요소들 안에 여러 요소를 배치할 수 있도록 해준다.

이 방법을 통해 버튼과 같은 수많은 간단한 컨트롤 안에 그림이나 벡터 도형, 레이아웃 컨테이너까지 배치 가능하다.

- 컨텐츠 모델은 WPF 를 매우 유연하게 만들어 주는 특징 중 하나.

- 컨텐츠 모델을 지원하는 ContentControl 기본 클래스에서 파생되는 특수한 작식 클래스들을 사용하여, 패널을 스크롤 할 수 있도록 하거나 아니면 접거나 펼칠 수 있도록 해준다.

**1. 컨텐츠 프로퍼티(Content Property)**

1. 컨텐츠 프로퍼티는 window 에서 가장 중요한 프로퍼티 중에 하나이다.
2. 컨텐츠 프로퍼티는 object로 정의되어있다.  
   즉, window를 제외한 어떠한 객체라도 할당할 수 있다(window 객체는 반드시 트리의 Root가 되어야 한다(자식이 될수없다))
3. 컨텐츠 프로퍼티에는 오직 하나의 객체만 할당할 수 있다.
4. 100개가 넘는다.

**2. 컨텐츠 프로퍼티의 분류**

(1). UIElement를 상속받은 것

-OnRender 메소드가 호출됨  
(객체의 그래픽 표시를 위한 UIElement에 정의된 메소드)

(2). UIElement를 상속받지 않은 것

-ToString()으로 출력됨 (문자열로 출력됨)

※Content는 비록 하나의 객체 밖에 할당할 수 없지만, 레이아웃을 이용하여 여러가지 객체를 담을 수 있다. 예를 들어 Grid, StackPanel, WrapPanel, Canvas Panel, Dock Panel등 이 있다.

이 판넬들을 컨텐츠로 사용 할 때를 생각해보면 어려 객체를 담을 수 있다는 결론이 나온다. (참조: System.Windows.Controls.Panel)

***특수한 컨테이너 목록 (이런 것이 있다는 것만 알아두기)***

-Scroll Viewer

-Custom Scroll

-GroupBox & TabItem

-Expander

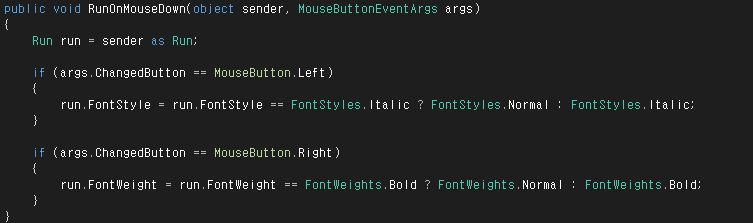
-Decorator

-Viewbox

**3. 폰트 패밀리(font family)**

- 타입 패밀리(type family) 라고도 부르며, 대부분의 UI 요소를 같은 Button 및 TextBlock, 텍스트 콘텐츠 컨트롤에 대한 글꼴을 지정 하는 데 사용 할 수 있는 속성으로, 폰트 패밀리와 이를 이용한 변화를 조합한 개념 (ex. Ascender / descender)

EX)



**4. 엘리먼트(Element)**

1. WPF의 최소 작업 단위로 선, 도형, 점 등 요소를 나타낸다.

2. 기존의 사용하던 컨트롤과 비슷한 개념이지만 WPF에서는 이보다 더 큰 범위의 기능을 제공한다.

3. 자체적으로 Hit Test를 지원하기 때문에 마우스의 모든 이벤트를 사용 좀더 정확히 구현 할 수 있다.

4. 윈도우 상에서 그려진 이미지가 자동으로 상태관리가 된다.

5. 하나의 Element는 그 모양에 관한 정보들과 이벤트 작업속성 등을 담을 수 있게 구성되어 있다.

**5. 이미지 할당**

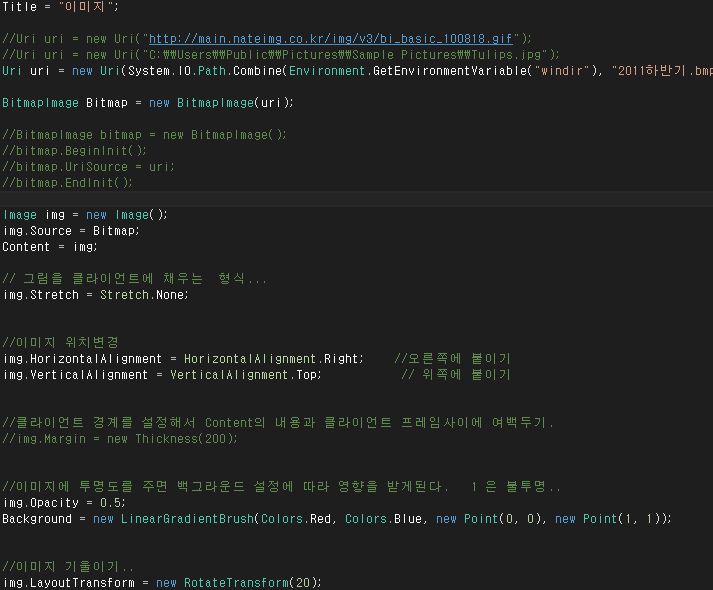
- 이미지 출력 순서 : Uri 객체 생성 🡪 위치지정 🡪 BitmapImage객체생성

🡪 Image객체에 등록 🡪 Content에 대입

1. 디렉토리의 이미지 로딩하기 : Uri객체 생성자에 FullPath를 인자로 넘긴다.

2. Image 객체의 각종 프로퍼티를 이용해 이미지조정

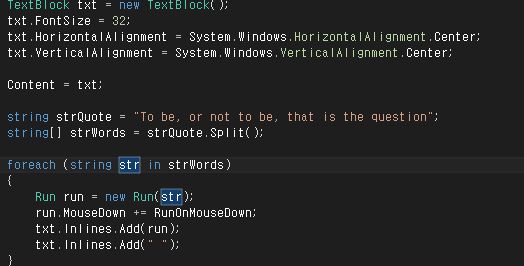
3. Foreground 프로퍼티는 이미지 사용시에는 아무런 효과가 없다.

Ex) 

**6. 텍스트에 각각 다른 속성 주기.**

**-** TextBlock 객체의 Inlines.Add 매서드를 사용해 문장과 스타일을 각각 추가한다.

Ex)



**7. OnRender를 이용한 그리기.**

1) OnRender 는 UiElement를 상속받은 모든 Element에 존재하며   
그리기에 관한 정보를 담고 있다.

2) OnRender 를 오버라이딩 하면 DrawingContext 객체를 매개 수로 받는다. 이 객체에 미리 정의된 매서드를 이용해 각종 그리기에 대한 속성을 저장한다. Memdc와 비슷한 역할

3) 최종적으로 Content에 할당 시에 OnRender를 호출하여 저장된 그림들을 그려 주게 된다.  
EX)

