바인딩 {Binding}

발표자 소개

- 이재웅 (제임스)
  오픈에스지(주) 수석 연구원

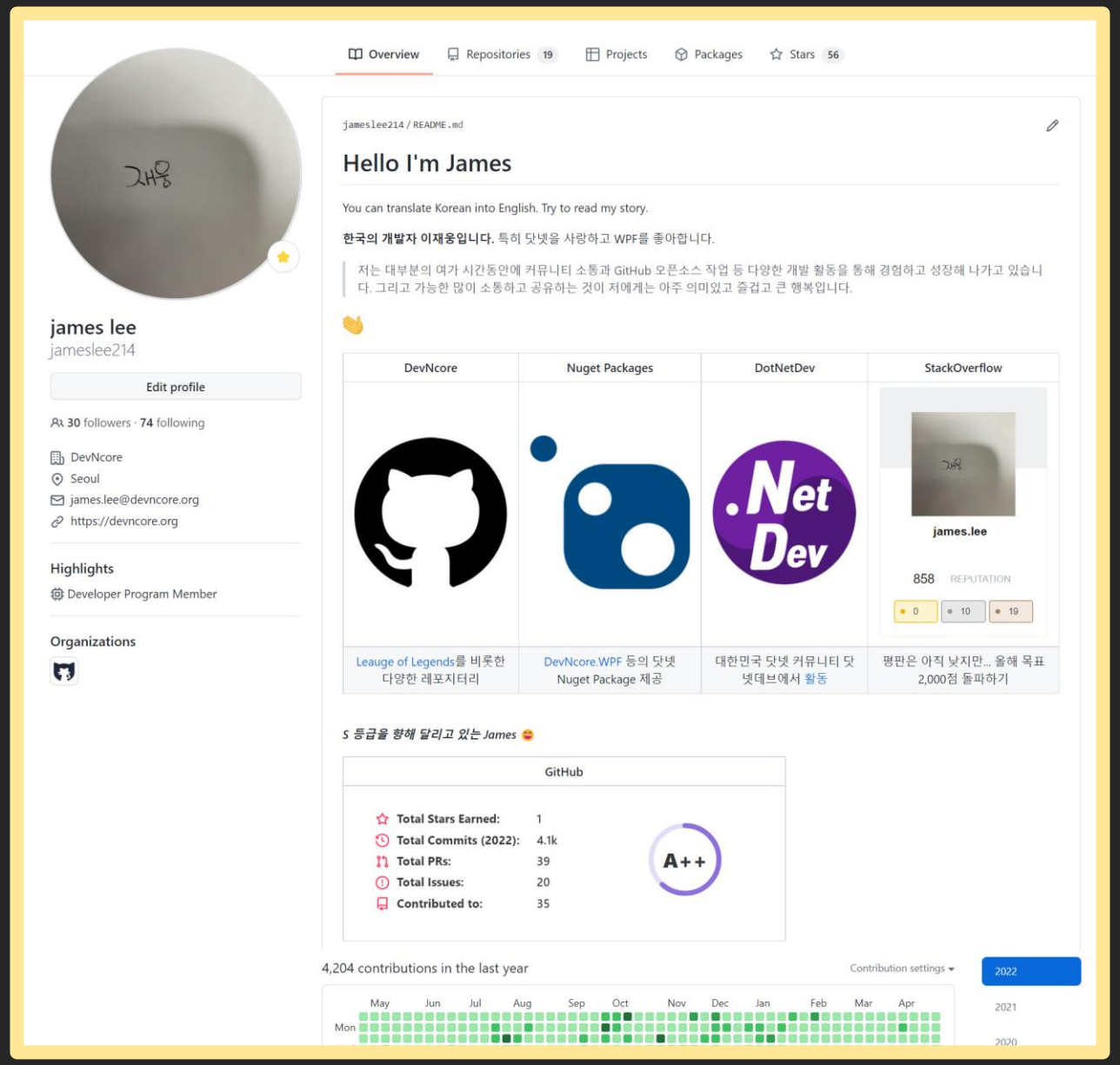
- 닷넷데브 운영진
- 데브엔코어(DevNcore) 오픈소스
  - 깃허브
  - 누겟 패키지
  - 유튜브 채널

바인딩 {Binding}

발표자 소개

- 이재웅 (제임스)
  오픈에스지(주) 수석 연구원

- 닷넷데브 운영진
- 데브엔코어(DevNcore) 오픈소스
  - 깃허브
  - 누겟 패키지
  - 유튜브 채널



바인딩 {Binding}

James (제임스)

발표 진행 순서

1. 바인딩이 왜 중요한가?
2. 바인딩 (Binding) 종류
3. 데모 리뷰 (TOSS 주식 목록)
4. 마지막 정리

바인딩 {Binding}

James (제임스)

바인딩이 왜 중요한가?

주요 핵심 기술

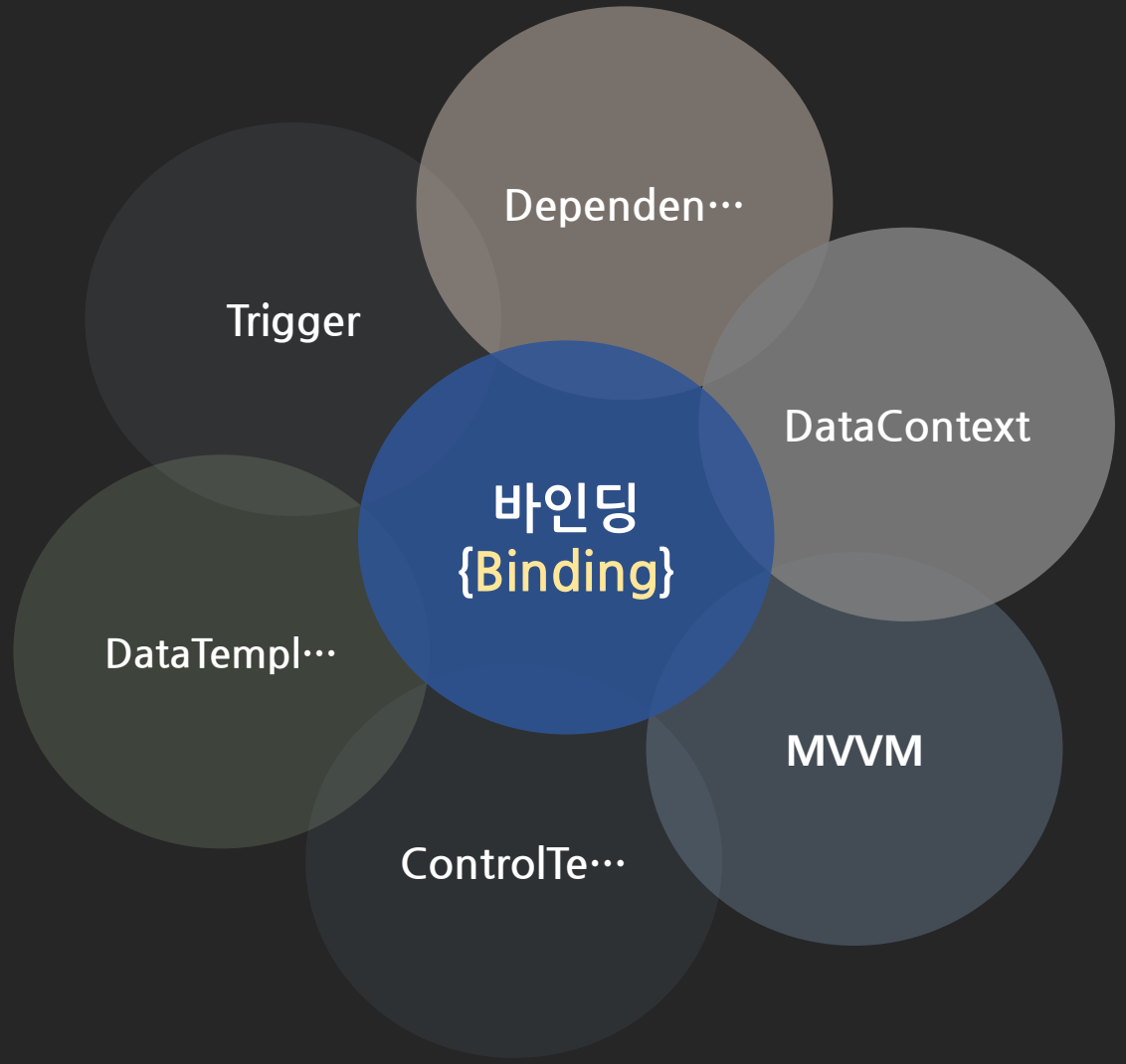
- 데이터컨텍스트 (DataContext)
- 디펜던시프로퍼티 (DependencyProperty)
- MVVM (Model View ViewModel)
- 컨트롤템플릿 (ControlTemplate)
- 데이터템플릿 (DataTemplate)
- 트리거 (Trigger)
- 컨텐트컨트롤 (ContentControl)
- 아이템스컨트롤 (ItemsControl)

바인딩 {Binding}

James (제임스)

바인딩이 왜 중요한가?

- 바인딩 (Binding)
- 데이터컨텍스트 (DataContext)
- 디펜던시프로퍼티 (DependencyProperty)
- MVVM (Model View ViewModel)
- 컨트롤템플릿 (ControlTemplate)
- 데이터템플릿 (DataTemplate)
- 트리거 (Trigger)
- 컨텐트컨트롤 (ContentControl)
- 아이템스컨트롤 (ItemsControl)



바인딩 {Binding}

James (제임스)

바인딩 (Binding) 종류

- 데이터컨텍스트 (DataContext) 바인딩
- 엘리먼트 (Element) 바인딩
- 템플릿 (TemplateBinding) 바인딩
- 언세스터타입(RelativeSource AncestorType) 바인딩
- 언세스터레벨 (RelativeSource AncestorLevel) 바인딩
- 셀프 (RelativeSource Self) 바인딩
- 템플리티드페런트 (RelativeSource TemplatedParent) 바인딩
- 멀티 (MultiBinding) 바인딩
- 컨버터 (Converter) 바인딩
- 커맨드 (Command) 바인딩
- 스테틱 (Static) 바인딩

바인딩 {Binding}

James (제임스)

모든 바인딩을 다 알아야 할까?

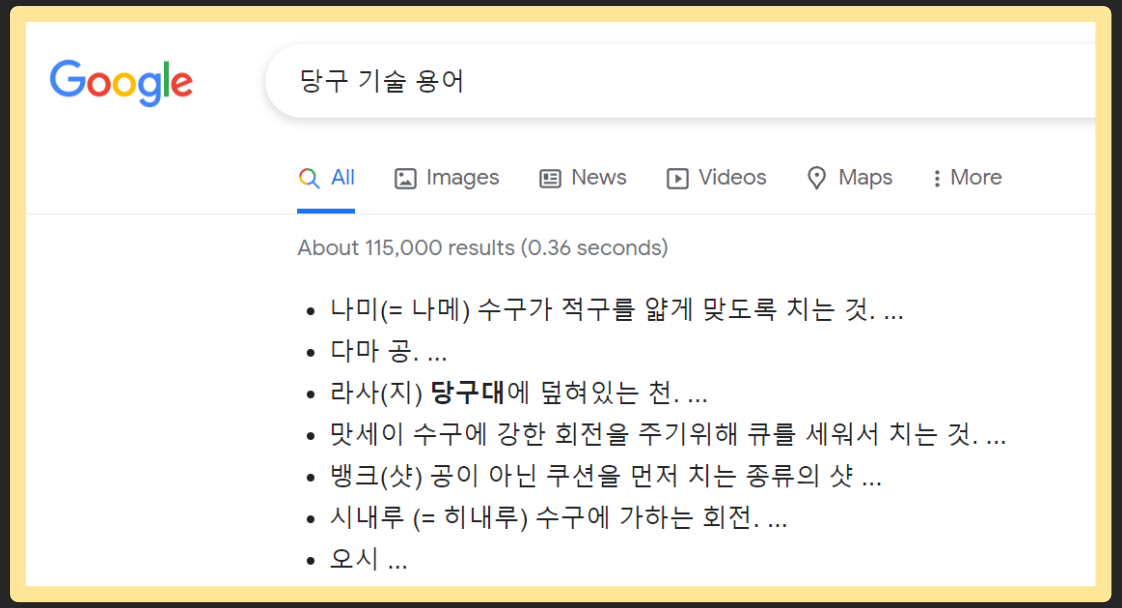
- X, 그래도 많이 알면 더 좋다.
- 그러나 숙련도를 올리는 것도 매우 중요하다.

바인딩 {Binding}

James (제임스)

모든 바인딩을 다 알아야 할까?

- X, 그래도 많이 알면 더 좋다.
- 하지만 숙련도를 올리는 것도 매우 중요하다.



당구 기술

- 다양한 기술 (밀어치기, 끌어치기, 얇게치기, 길게치기, 비껴치기, 넣어치기, 걸어치기, 뱅크샷, 뒤돌리기, 앞돌리기, 옆돌리기 대회전 등…)

기본기

- 두께, 힘 조절, 당점, 큐 걸이 자세 등… (숙련도)

바인딩 {Binding}

James (제임스)

모든 바인딩을 다 알아야 할까?

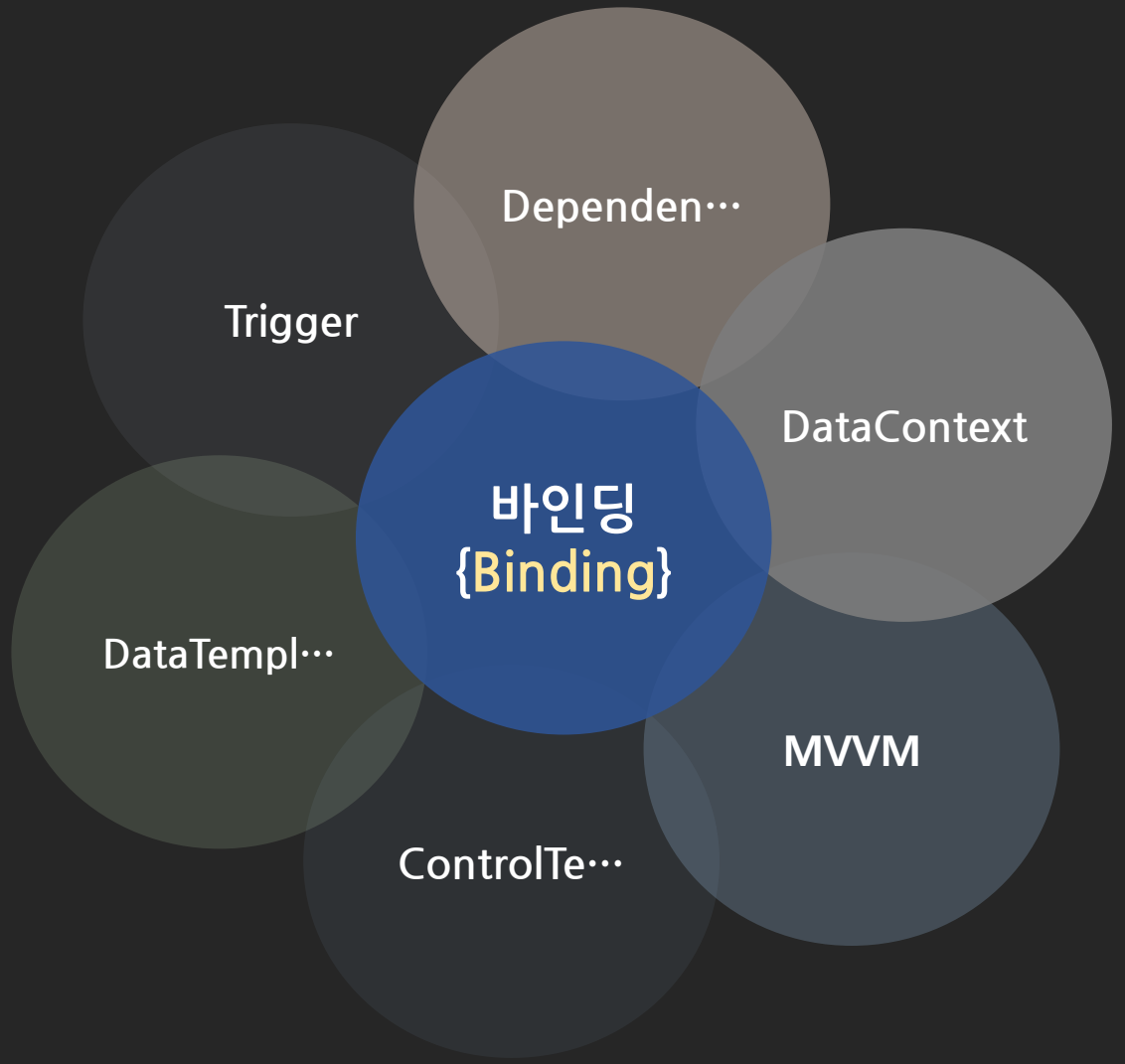
- X, 그래도 많이 알면 더 좋다.
- 하지만 숙련도를 올리는 것도 매우 중요하다.

WPF 기술

- 다양한 기술 (MVVM, 디펜던시프로퍼티, 컨트롤템플릿, 데이터템플릿, 트리거, 컨텐트컨트롤 등…)

기본기

- 바인딩, 컨버터, 커맨드 등… (숙련도)



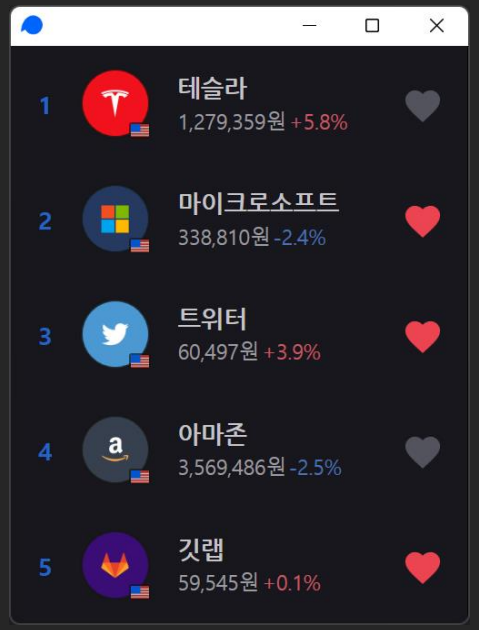
바인딩 {Binding}

James (제임스)

데모 리뷰

- 토스앱 데모 준비 환경

- 데모 실행 스크린샷

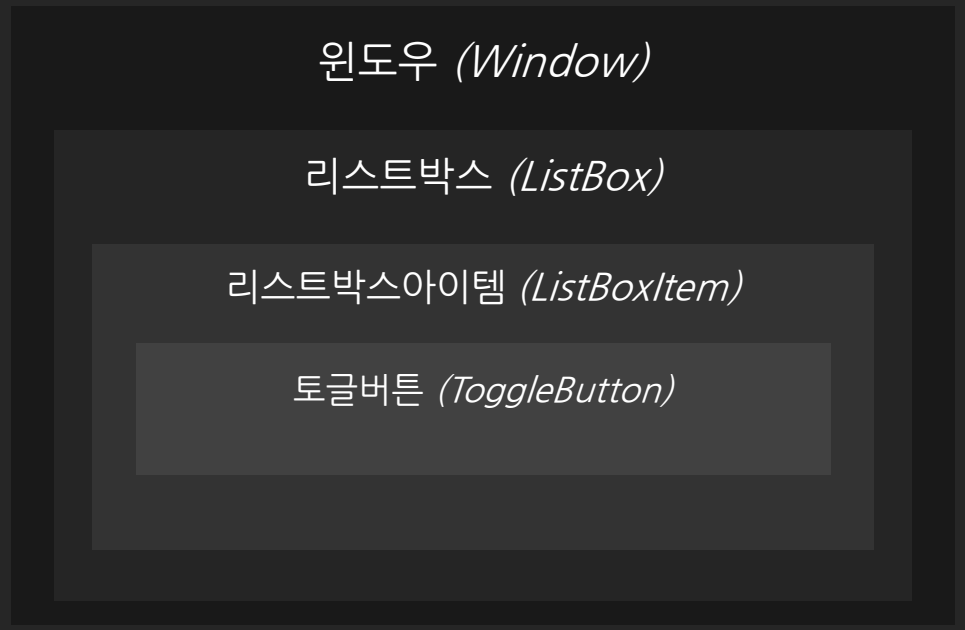


바인딩 {Binding}

James (제임스)

데모 리뷰

- 주요 UI 계층 구조



- 커스텀 클래스 구현

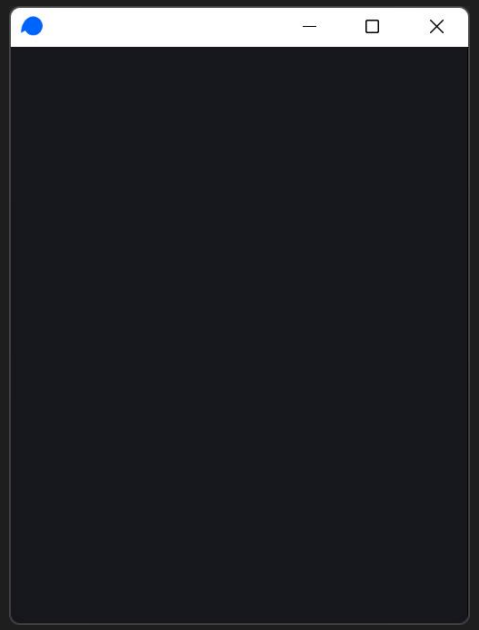
UI 클래스		커스텀 UI 클래스
Window	▶	MainWindow : Window
ListBox	▶	StockListBox : ListBox
ListBoxItem	▶	StockListItem : ListBoxItem
ToggleButton	▶	HeartSwitch : ToggleButton

바인딩 {Binding}

James (제임스)

MainWindow : Window

```
<!-- ViewModel 인스턴스 -->
<vm:MainViewModel x:Key="MVVM"/>

<!-- (커스텀컨트롤) MainWindow : Window -->
<Style TargetType="{x:Type views:MainWindow}">
    <!-- DataContext 바인딩 -->
    <Setter Property="DataContext" Value="{StaticResource MVVM}"/>
    <Setter Property="SizeToContent" Value="WidthAndHeight"/>
    <Setter Property="Background" Value="#18171D"/>
    <Setter Property="Foreground" Value="#FFFFFF"/>
    <Setter Property="Template">
        <Setter.Value>
            <ControlTemplate TargetType="{x:Type views:MainWindow}">
                <Border Background="{TemplateBinding Background}">
                    <!-- StockListBox : ListBox -->
                    <units:StockListBox ItemsSource="{Binding Stocks}"/>
                </Border>
            </ControlTemplate>
        </Setter.Value>
    </Setter>
</Style>
```

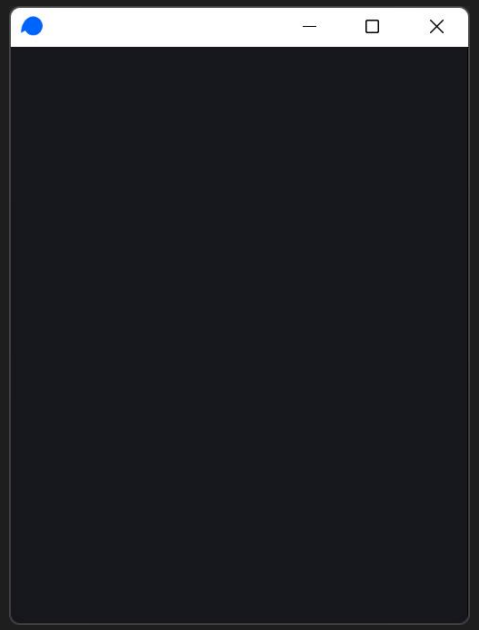


바인딩 {Binding}

James (제임스)

MainViewModel (뷰모델)

```
internal class MainViewModel
{
    // 하트 커맨드
    참조 1개
    public ICommand HeartCommand { get; }

    // 주식종목 리스트
    참조 1개
    public List<Stock> Stocks { get; }

    참조 0개
    public MainViewModel()
    {
        // 하트 커맨드바인딩 연결
        HeartCommand = new TossCommand<Stock>(HeartChecked);

        // 임시 데이터 생성
        List<Stock> stocks = new();
        stocks.Add(new Stock(1, "TSLA", "테슬라", "#F2121E", 1209010, 1279359, false));
        stocks.Add(new Stock(2, "MSFT", "마이크로소프트", "#263961", 347000, 338810, true));
        stocks.Add(new Stock(3, "TWTR", "트위터", "#4B98D3", 58199, 60497, true));
        stocks.Add(new Stock(4, "AMZN", "아마존", "#37404F", 3662201, 3569486, false));
        stocks.Add(new Stock(5, "GTLB", "깃랩", "#3B0E77", 59502, 59545, true));
        Stocks = stocks;
    }

    참조 1개
    private void HeartChecked(Stock item)
    {
        // 커맨드바인딩 Execute 부분
        _ = item.Heart;
    }
}
```

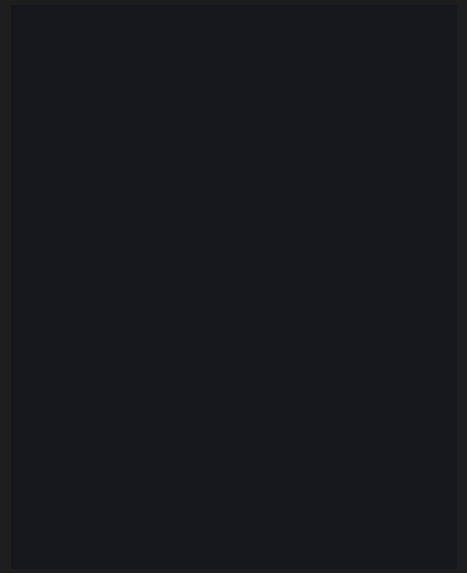


바인딩 {Binding}

James (제임스)

StockListBox : ListBox

```
<Style TargetType="{x:Type units:StockListBox}">
    <Setter Property="Template">
        <Setter.Value>
            <ControlTemplate TargetType="{x:Type units:StockListBox}">

                <!-- StockListItem 적용 부분 -->
                <ItemsPresenter/>

            </ControlTemplate>
        </Setter.Value>
    </Setter>
</Style>
```

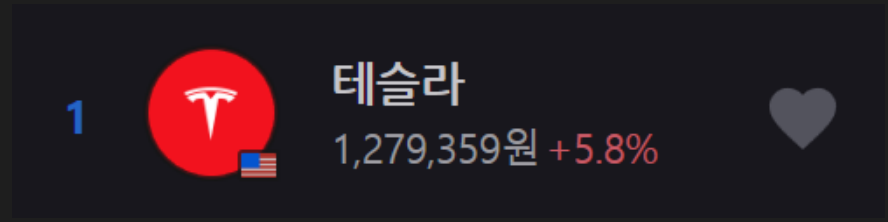


바인딩 {Binding}

James (제임스)

StockListItem : ListBoxItem

```
<Style TargetType="{x:Type units:StockListItem}">
    <Setter Property="Template">
        <Setter.Value>
            <ControlTemplate TargetType="{x:Type units:StockListItem}">
                <Border>
                    <Grid>
                        <Grid.ColumnDefinitions>
                            <ColumnDefinition Width="Auto"/>
                            <ColumnDefinition Width="Auto"/>
                            <ColumnDefinition Width="*"/>
                            <ColumnDefinition Width="Auto"/>
                        </Grid.ColumnDefinitions>

                        <!-- 순위 -->
                        <TextBlock Grid.Column="0"... />

                        <!-- 로고 배경 (동그라미) -->
                        <Ellipse Grid.Column="1"... />

                        <!-- 로고 이미지 (컨버터 사용) -->
                        <Image Grid.Column="1"... />

                        <!-- 작은 국가 이미지 -->
                        <Border Grid.Column="1"...>

                        <StackPanel Grid.Column="2"...>

                        <!-- 하트 -->
                        <units:HeartSwitch Grid.Column="3"... />
                    </Grid>
                </Border>
                <ControlTemplate.Triggers...>
            </ControlTemplate>
        </Setter.Value>
    </Setter>
</Style>
```

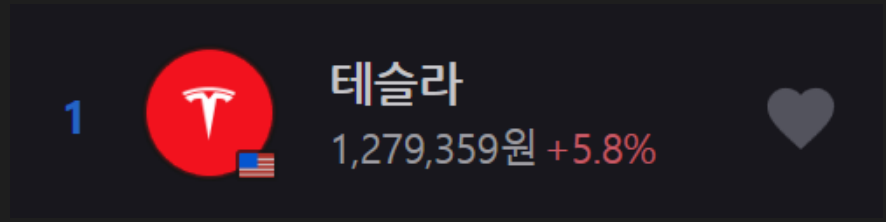


바인딩 {Binding}

James (제임스)

Stock (모델)

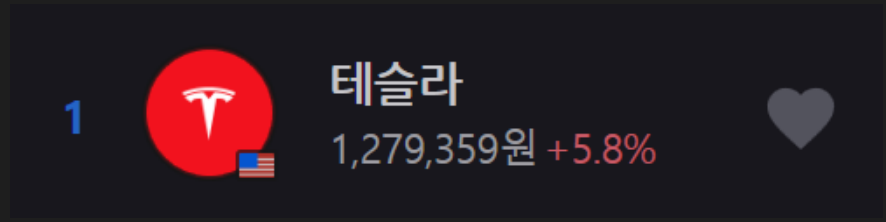
```
internal class Stock
{
    // 순위 (1, 2, ...)
    참조 1개
    public int Rank { get; set; }
    // 코드 (주식종목코드: TSLA, MSFT, TWTR, AMZN, GTLB)
    참조 1개
    public string Code { get; set; }
    // 이름 (테슬라, 마이크로소프트, ...)
    참조 1개
    public string Name { get; set; }
    // 시그니쳐 컬러코드 (#F2121E, #263961, ...)
    참조 1개
    public string Color { get; set; }
    // 주식 현재 가격
    참조 3개
    public double LiveValue { get; set; }
    // 주식 어제 가격
    참조 5개
    public double PastValue { get; set; }
    // 하트 (즐겨찾기 여부: True, False)
    참조 2개
    public bool Heart { get; set; }
```



바인딩 {Binding}

James (제임스)

Stock (모델)

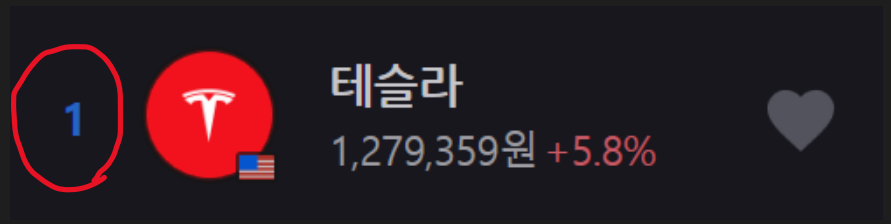



```
// 임시 데이터 생성
List<Stock> stocks = new();
stocks.Add(new Stock(1, "TSLA", "테슬라", "#F2121E", 1209010, 1279359, false));
stocks.Add(new Stock(2, "MSFT", "마이크로소프트", "#263961", 347000, 338810, true));
stocks.Add(new Stock(3, "TWTR", "트위터", "#4B98D3", 58199, 60497, true));
stocks.Add(new Stock(4, "AMZN", "아마존", "#37404F", 3662201, 3569486, false));
stocks.Add(new Stock(5, "GTLB", "깃랩", "#3B0E77", 59502, 59545, true));
Stocks = stocks;
```

바인딩 {Binding}

James (제임스)

StockListItem : ListBoxItem



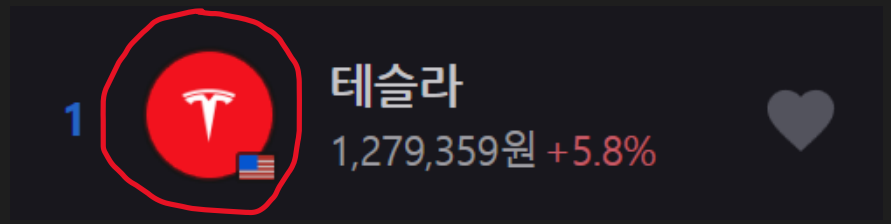

```
<!-- 순위 -->
<TextBlock Grid.Column="0"
           Text="{Binding Rank}"
           HorizontalAlignment="Center"
           VerticalAlignment="Center"
           Foreground="#2464C6"
           FontWeight="Bold"
           FontSize="18"
           Margin="20"/>
```

바인딩 {Binding}

James (제임스)

StockListItem : ListBoxItem

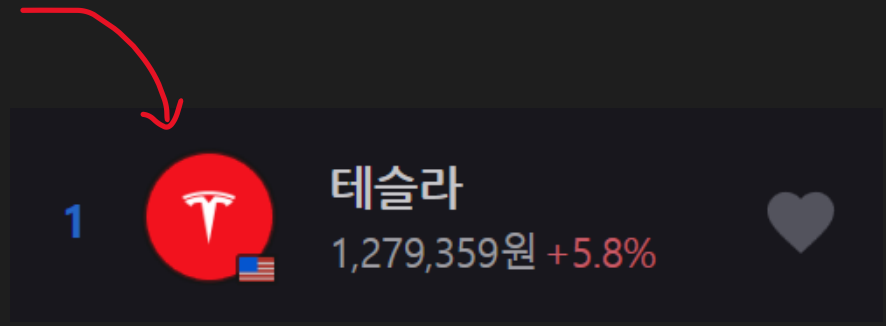
```
<!-- 로고 배경 (동그라미) -->
<Ellipse Grid.Column="1"
         Width="48" Height="48"
         Fill="{Binding Color}"/>
```



바인딩 {Binding}

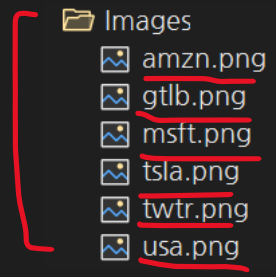
James (제임스)

StockListItem : ListBoxItem




```
<!-- 로고 이미지 (컨버터 사용) -->
<Image Grid.Column="1"
       Width="20" Height="20"
       Source="{Binding Code, Converter={StaticResource CodeToBrandConverter}}"
       RenderOptions.BitmapScalingMode="HighQuality"/>
```

```
public object Convert(object value, Type targetType, object parameter, CultureInfo culture)
{
    // TSLA, MSFT, TWTR, AMZN, GTLB
    return $"/Toss;component/Images/{value}.png";
}
```



바인딩 {Binding}

James (제임스)

StockListItem : ListBoxItem

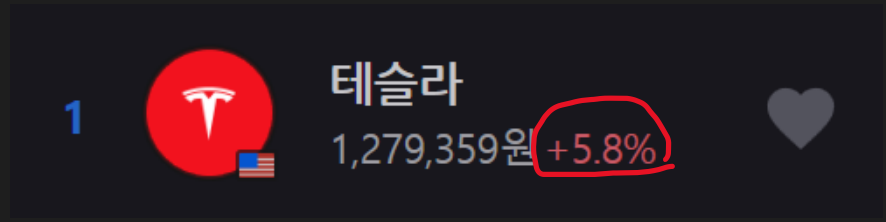

```
<!-- 현재 가격 (3자리 쉼표, 원 표시) -->
<TextBlock Text="{Binding LiveValue, Converter={StaticResource ValueFormatConverter}}"
           Foreground="#9D9CA2"
           FontSize="15"/>
```

```
public object Convert(object value, Type targetType, object parameter, CultureInfo culture)
{
    return $"{(double)value:N0}원";
}
```

바인딩 {Binding}

James (제임스)

StockListItem : ListBoxItem




```
<!-- 전일대비 Rate% (어제보다 올랐는지 내렸는지, 소숫점 1자리까지, +/- 표시) -->
<TextBlock x:Name="rate"
           Text="{Binding Path=DataContext, RelativeSource={RelativeSource Self}, Converter={StaticResource RealtimeValueConverter}}"
           Foreground="#CA5A66"
           FontSize="15"
           Margin="2 0 0 0"/>
```

```
public object Convert(object value, Type targetType, object parameter, CultureInfo culture)
{
    Stock stock = (Stock)value;
    double rate = (stock.LiveValue - stock.PastValue) / stock.PastValue * 100;
    return $"{rate:+0.0;-#.0}%";
}
```

바인딩 {Binding}

James (제임스)

데모 실행

닷넷데브 커뮤니티 또는 깃허브에서 내려받을 수 있습니다.

https://dotnetdev.kr

https://github.com/devncore/toss

바인딩 {Binding}

James (제임스)

마지막 정리

제가 준비한 내용은 여기까지 입니다. 혹시 도움이 되셨을까요?

사실 바인딩에 정답이란 없는 것 같습니다.
그리고 MVVM 같은 디자인 패턴을 사용할 때도 마찬가지입니다.

하지만 Template과 ViewModel 사이를 유연하고 강력하게 구현하기 위해서는 바인딩 기술을 제대로 이해하고 상황에 맞게 필요한 바인딩을 사용하는 것이 무엇보다 중요한 사실입니다.

바인딩은 WPF의 기초체력과도 같은 중요한 기반이기 때문에 저도 계속해서 바인딩에 변화를 주며 연구해 나아가고 있습니다.

바인딩 {Binding}

이재웅 (제임스)

James.lee@devncore.org
https://devncore.org

커뮤니티 (질문)

https://forum.dotnetdev.kr

토스 샘플 오픈소스 (깃허브)

https://github.com/devncore/toss

감사합니다.