

# How To WPF アプリケーション Part 1

by 中博俊



- WPFアプリって覚えることが多いでしょ？
  - 多いです。
- XAMLで書かなきゃいけないんでしょ？
  - 基本的には書かなきゃいけません。
- 大変なんでしょ？
  - ちょっと大変かもしれません。

• WPFアプリって覚えることが多いでしょ？

— 多いです。

• XAMLで書かなきゃいけないでしょ？

でも基本的にも書かなきゃいけません。  
でも楽しいんです。

• 大変なんでしょ？

— ちょっと大変かもしれませんが。

## 本日の目標

- 細かい基礎技術の説明はしません。
- そのかわり単純なアプリケーションを作れるようになりましょう。
- おまけ

Hello World 1

# とりあえず DEMO1



わんくま同盟 大阪勉強会 #15

- これだけじゃさみしいですので、やっぱりボタンを押すようにしましょう。



Hello World 2

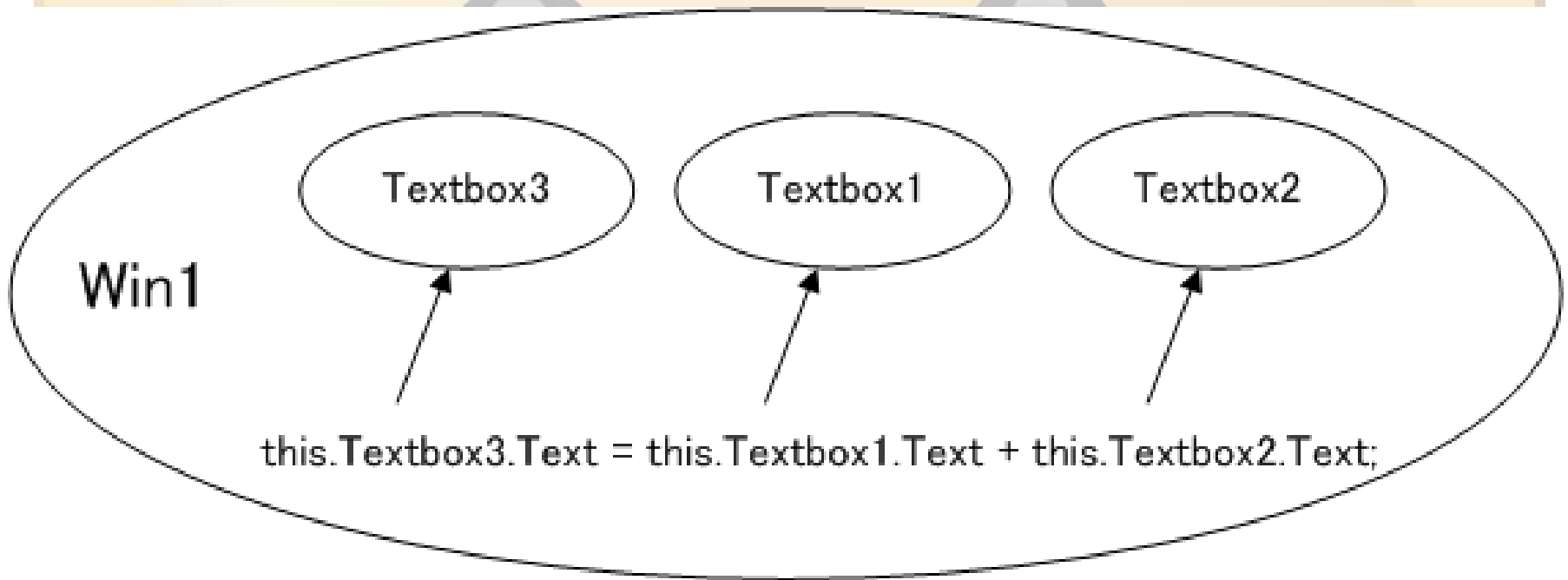
# ボタンでDEMO2



わんくま同盟 大阪勉強会 #15

## バインディング

- WPFの特徴はDataContextです。
- いままでのWindowsFormsのよくあるパターン

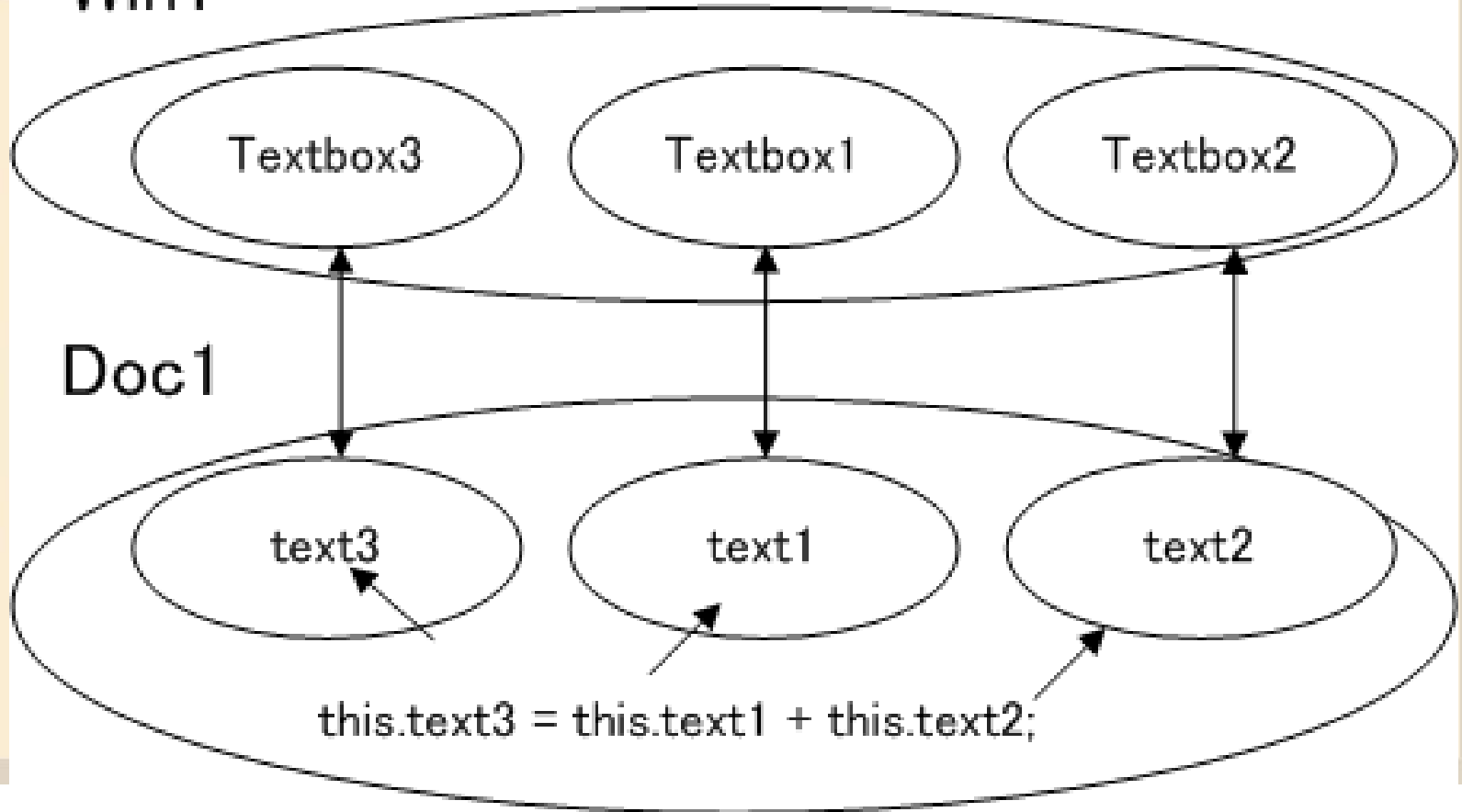




# バインディング

- WPFらしい

Win1



## バインディング

- データと画面処理の分離を行います。
- これをバインディングといいます。
- 今日はバインディングの詳細までは触れません。
- このソースとおなじ実装を試してみましょう。

# • DEMO3

## バインディング

- WPFの基本はバインディングです。
- バインディングの基本は  
INotifyPropertyChangedです。
- ドキュメントクラス共通のクラスを作っておくと  
便利です。

## バインディング

```
public class NotifyPropertyChangedBase :  
    INotifyPropertyChanged {  
  
    public event PropertyChangedEventHandler  
        PropertyChanged;  
  
    protected void FirePropertyChanged(string propertyName)  
    {  
        if (this.PropertyChanged != null) {  
            this.PropertyChanged(this,  
                new PropertyChangedEventArgs(propertyName));  
        }  
    }  
}
```



Id?

- いまままでのXAMLなどでおかしいところがありませんでしたか？
- `<TextBox Text="{Binding Path=text1}"/>`

**IDがない!!!**

## ID?

- そもそもオブジェクトをツリー形式で保持するのには参照さえあれば実現可能です。
- 逆になぜ今までIDを必要としたのでしょうか？
- IDをつけてコントロールを操作していたため、コントロールの操作が少なければIDなど不要なのです。
- とはいえ・・・つけなければいけない場合もあるので `<TextBox x:Name="t1"/>`

# バリエーションの DEMO4

- 解決策1
- 計算のもとになっている値の変更時に ButtonEnabledも変更通知を上げてもらう。
- 解決策2
- 変更通知を利用して、変更通知を上げる。
- 関数型言語っぽく利用するにはすこし手間が必要ですね。



- HTMLのように<TAG></TAG>でくくられた中身は Tag.Contentと等価です。
- Tag.Contentには1つの何かを置くことができます。
- ちょっとボタンにお遊びしてみましよう。

# おまけのDEMO5

## 今回のまとめ

- WPFらしいDocument-Viewの関係を構築するとWindowsFormsの考え方から切り替えよう。
- INotifyPropertyChangedをモノにしよう  
NotifyPropertyChangedBaseというようなドキュメントの共通ベースクラスを作るといい
- 複数の値から導き出すプロパティはひと工夫が必要

Enjoy WPF