**Windows. ShowDialog() method**

* Opens a window and returns only when the newly opened window is closed
* A [Nullable<T>](https://docs.microsoft.com/en-us/dotnet/api/system.nullable-1?view=windowsdesktop-6.0) value of type [Boolean](https://docs.microsoft.com/en-us/dotnet/api/system.boolean?view=windowsdesktop-6.0) that specifies whether the activity was accepted (true) or canceled (false). The return value is the value of the [DialogResult](https://docs.microsoft.com/en-us/dotnet/api/system.windows.window.dialogresult?view=windowsdesktop-6.0" \l "system-windows-window-dialogresult) property before a window closes.

**Window.DialogResult Property**

* Gets or sets the dialog result value, which is the value that is returned from the [ShowDialog()](https://docs.microsoft.com/en-us/dotnet/api/system.windows.window.showdialog?view=windowsdesktop-6.0" \l "system-windows-window-showdialog) method.
* A [Nullable<T>](https://docs.microsoft.com/en-us/dotnet/api/system.nullable-1?view=windowsdesktop-6.0) value of type [Boolean](https://docs.microsoft.com/en-us/dotnet/api/system.boolean?view=windowsdesktop-6.0). The default is false.

**Window.ContentRendered Event**

* Occurs after a window's content has been rendered.
* [EventHandler](https://docs.microsoft.com/en-us/dotnet/api/system.eventhandler?view=windowsdesktop-6.0) Type
* If the window has no content, this event is not raised.

**TextBox.SelectAll Method**

* Selects the entire contents of the text box.

**TextBoxBase.SelectionChanged Event**

* Occurs when the text selection has changed.
* [RoutedEventHandler](https://docs.microsoft.com/en-us/dotnet/api/system.windows.routedeventhandler?view=windowsdesktop-6.0)

**Popup.PlacementTarget Property**

* Gets or sets the element relative to which the [Popup](https://docs.microsoft.com/en-us/dotnet/api/system.windows.controls.primitives.popup?view=windowsdesktop-6.0) is positioned when it opens.
* You can position a popup by setting the [PlacementTarget](https://docs.microsoft.com/en-us/dotnet/api/system.windows.controls.primitives.popup.placementtarget?view=windowsdesktop-6.0), [PlacementRectangle](https://docs.microsoft.com/en-us/dotnet/api/system.windows.controls.primitives.popup.placementrectangle?view=windowsdesktop-6.0), [Placement](https://docs.microsoft.com/en-us/dotnet/api/system.windows.controls.primitives.popup.placement?view=windowsdesktop-6.0), [HorizontalOffset](https://docs.microsoft.com/en-us/dotnet/api/system.windows.controls.primitives.popup.horizontaloffset?view=windowsdesktop-6.0), and [VerticalOffset](https://docs.microsoft.com/en-us/dotnet/api/system.windows.controls.primitives.popup.verticaloffset?view=windowsdesktop-6.0) properties. For more information, see [Popup Placement Behavior](https://docs.microsoft.com/en-us/dotnet/framework/wpf/controls/popup-placement-behavior).

**GridViewColumn.DisplayMemberBinding Property**

* Gets or sets the data item to bind to for this column.
* ([BindingBase](https://docs.microsoft.com/en-us/dotnet/api/system.windows.data.bindingbase?view=windowsdesktop-6.0)) The specified data item type that displays in the column. The default is null.

**Selector.SelectionChanged Event**

* Occurs when the selection of a [Selector](https://docs.microsoft.com/en-us/dotnet/api/system.windows.controls.primitives.selector?view=windowsdesktop-6.0) changes.
* [SelectionChangedEventHandler](https://docs.microsoft.com/en-us/dotnet/api/system.windows.controls.selectionchangedeventhandler?view=windowsdesktop-6.0)

**TextPointer Class**

* Represents a position within a [FlowDocument](https://docs.microsoft.com/en-us/dotnet/api/system.windows.documents.flowdocument?view=windowsdesktop-6.0) or [TextBlock](https://docs.microsoft.com/en-us/dotnet/api/system.windows.controls.textblock?view=windowsdesktop-6.0).
* The [TextPointer](https://docs.microsoft.com/en-us/dotnet/api/system.windows.documents.textpointer?view=windowsdesktop-6.0) class introduces the following terminology:

1. Position - Inherently, a [TextPointer](https://docs.microsoft.com/en-us/dotnet/api/system.windows.documents.textpointer?view=windowsdesktop-6.0) always points to a position in content. Such positions either fall between characters in the content, or between flow content element tags that define structure for the content.
2. Current Position - Because a [TextPointer](https://docs.microsoft.com/en-us/dotnet/api/system.windows.documents.textpointer?view=windowsdesktop-6.0) always indicates a position, and because many of the operations that can be performed through a [TextPointer](https://docs.microsoft.com/en-us/dotnet/api/system.windows.documents.textpointer?view=windowsdesktop-6.0) are relative to the position currently pointed to by the [TextPointer](https://docs.microsoft.com/en-us/dotnet/api/system.windows.documents.textpointer?view=windowsdesktop-6.0), it makes sense to simply refer to the position indicated by a [TextPointer](https://docs.microsoft.com/en-us/dotnet/api/system.windows.documents.textpointer?view=windowsdesktop-6.0) as the current position.
3. Insertion Position - An insertion position is a position where new content may be added without breaking any semantic rules for the associated content. In practice, an insertion position is anywhere in content where a caret may be positioned. An example of a valid [TextPointer](https://docs.microsoft.com/en-us/dotnet/api/system.windows.documents.textpointer?view=windowsdesktop-6.0) position that is not an insertion position is the position between two adjacent [Paragraph](https://docs.microsoft.com/en-us/dotnet/api/system.windows.documents.paragraph?view=windowsdesktop-6.0) tags (that is, between the closing tag of the preceding paragraph and the opening tag of the next paragraph).
4. Symbol - For the purposes of [TextPointer](https://docs.microsoft.com/en-us/dotnet/api/system.windows.documents.textpointer?view=windowsdesktop-6.0) operations that involve symbols, any of the following is considered to be a symbol:
   * An opening or closing tag for a [TextElement](https://docs.microsoft.com/en-us/dotnet/api/system.windows.documents.textelement?view=windowsdesktop-6.0) element.
   * A [UIElement](https://docs.microsoft.com/en-us/dotnet/api/system.windows.uielement?view=windowsdesktop-6.0) element contained within an [InlineUIContainer](https://docs.microsoft.com/en-us/dotnet/api/system.windows.documents.inlineuicontainer?view=windowsdesktop-6.0) or [BlockUIContainer](https://docs.microsoft.com/en-us/dotnet/api/system.windows.documents.blockuicontainer?view=windowsdesktop-6.0). Note that such a [UIElement](https://docs.microsoft.com/en-us/dotnet/api/system.windows.uielement?view=windowsdesktop-6.0) is always counted as exactly one symbol; any additional content or elements contained by the [UIElement](https://docs.microsoft.com/en-us/dotnet/api/system.windows.uielement?view=windowsdesktop-6.0) are not counted as symbols.
   * Each 16-bit Unicode character inside of a text [Run](https://docs.microsoft.com/en-us/dotnet/api/system.windows.documents.run?view=windowsdesktop-6.0) element.
5. Text Container - A text container is the element that forms the ultimate border for the flow content at hand; the position indicated by a [TextPointer](https://docs.microsoft.com/en-us/dotnet/api/system.windows.documents.textpointer?view=windowsdesktop-6.0) always falls within a text container. Currently, a text container must be either a [FlowDocument](https://docs.microsoft.com/en-us/dotnet/api/system.windows.documents.flowdocument?view=windowsdesktop-6.0) or a [TextBlock](https://docs.microsoft.com/en-us/dotnet/api/system.windows.controls.textblock?view=windowsdesktop-6.0). Generally speaking, operations between [TextPointer](https://docs.microsoft.com/en-us/dotnet/api/system.windows.documents.textpointer?view=windowsdesktop-6.0) instances in different text containers are not supported.
6. Document - The content in a text container is referred to as a document, as in the [IsInSameDocument](https://docs.microsoft.com/en-us/dotnet/api/system.windows.documents.textpointer.isinsamedocument?view=windowsdesktop-6.0) method and the [DocumentStart](https://docs.microsoft.com/en-us/dotnet/api/system.windows.documents.textpointer.documentstart?view=windowsdesktop-6.0) and [DocumentEnd](https://docs.microsoft.com/en-us/dotnet/api/system.windows.documents.textpointer.documentend?view=windowsdesktop-6.0) properties.

**TextRange Class**

* Represents a selection of content between two [TextPointer](https://docs.microsoft.com/en-us/dotnet/api/system.windows.documents.textpointer?view=windowsdesktop-6.0) positions.
* The [TextRange](https://docs.microsoft.com/en-us/dotnet/api/system.windows.documents.textrange?view=windowsdesktop-6.0) class introduces the following terminology.

1. **Selection** - A [TextRange](https://docs.microsoft.com/en-us/dotnet/api/system.windows.documents.textrange?view=windowsdesktop-6.0) is a selection of content between two positions indicated by **TextPointers**. One of these positions a fixed anchor with respect to the selection, while the other position is movable. This is similar to how a selection made by a user using the mouse or keyboard behaves.
2. **Current Selection** - Because a [TextRange](https://docs.microsoft.com/en-us/dotnet/api/system.windows.documents.textrange?view=windowsdesktop-6.0) always indicates a selection in content, it makes sense to simply refer to the selection indicated by a [TextRange](https://docs.microsoft.com/en-us/dotnet/api/system.windows.documents.textrange?view=windowsdesktop-6.0) as the current selection.
3. **Text Container** - A text container is the element that forms the ultimate border for the flow content at hand; the selection indicated by a [TextRange](https://docs.microsoft.com/en-us/dotnet/api/system.windows.documents.textrange?view=windowsdesktop-6.0) always falls within a text container. Currently, a text container must be either a [FlowDocument](https://docs.microsoft.com/en-us/dotnet/api/system.windows.documents.flowdocument?view=windowsdesktop-6.0) or a [TextBlock](https://docs.microsoft.com/en-us/dotnet/api/system.windows.controls.textblock?view=windowsdesktop-6.0).
4. **Document** - The content collectively contained within a text container is referred to as a document.

**Environment.NewLine Property**

* Gets the newline string defined for this environment.