Editor

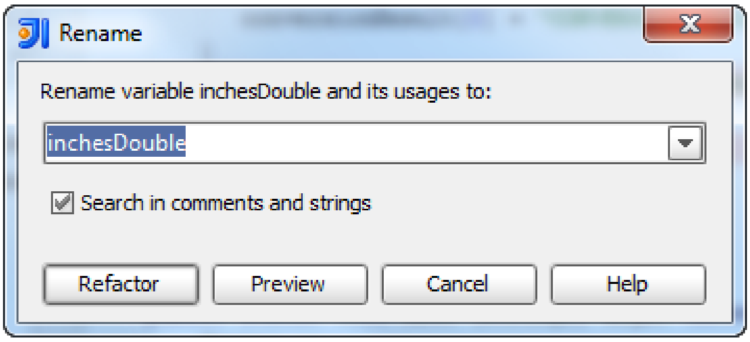
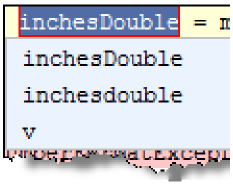
File | Settings | IDE Settings | Editor

Use the Editor page of the Settings dialog to configure the editor's behaviour and customize its view.

Item	Description
Mouse	
Honor "CamelHumps" word settings when selecting using double click	Select this check box to have IntelliJ IDEA invoke the CamelHumps selection whe words are selected by double-clicking. This feature works only if the Use 'CamelHumps' words option is enabled.
Change font size (Zoom) with Ctrl+Mouse Wheel	If this check box is selected, the particular editor font size can be changed by r the mouse wheel while holding the `Ctrl` key. If this check box is not selected, rolling mouse wheel while holding the `Ctrl` key scrolls the editor.
Enable Drag'n'Drop functionality in editor	If this check box is selected, you can drag code fragments in the editor. Refer t section Using Drag-and-Drop in the Editor.
Scrolling	
Smooth scrolling	Select this check box to enable smooth scrolling in the editor.
Prefer scrolling editor canvas to keep caret line centered	Click this option to choose scrolling editor canvas and keeping the caret in place Keeping the caret in place and scrolling the editor canvas can be helpful in cours debugging session. As you step through the lines of code, the editor canvas scr while the line at caret is always in the center of the screen.
Prefer moving caret line to minimize editor scrolling	Click this option to choose moving the caret. When you step through the lines of code during the debugging session, the care moves down, and the editor canvas doesn't scroll until the caret line reaches th bottom of the screen.
Limits	
Maximum number of contents to keep in clipboard	In this text box, specify how many code blocks can be kept in clipboard.
Recent file limit	In this text box, specify how many file names can be included in the list of recer files.
Console commands history size	In this text box, specify how many console commands will be included in the con history and can be browsed through.
Error Highlighting	

Item	Description
Error stripe mark min height (pixels)	In this text box, specify the minimum size of the error and warning stripes.
Autoreparse delay (ms)	In this text box, specify the time period after when IntelliJ IDEA starts reparsing entered text.
'Next Error' action goes to errors first	Select this check box to have IntelliJ IDEA pass through the errors and then thro the warnings. Clear this check box to have IntelliJ IDEA pass through the errors and warnings sequentially.
Suppress with @SuppressWarnings (for 5.0 only)	Select this check box to have `@SuppressWarnings` implemented as an annotation. Clear this check box to have `@SuppressWarnings` implemented as a JavaDoc comm
Other	
Strip trailing spaces on Save	From this drop-down list, select the mode in which IntelliJ IDEA will handle trailin spaces in the end of lines on file saving: - *Modified lines* - Strips trailing spaces only in the end of modified lines. - *All* - Strips trailing spaces in all lines. - *None* - Does not strip trailing spaces.
Ensure line feed at file end on Save	Select this check box to have IntelliJ IDEA automatically add an empty line in th of a file during the save procedure.
Show quick doc on mouse move	Select this check box to show quick documentation for the symbol at caret. The quick documentation pop-up window appears after the specified delay.
Virtual Space	
Use soft wraps in the editor	If this check box is selected, the *soft wraps* (or *word wraps*) are used in the edi Horizontal scroll bar in general is not shown, when this option is enabled. Howeve certain cases, when a line cannot be "soft-wrapped", the horizontal scroll bar st appears (for example, a line consists of a single string that is wider than the visi area.)
Use soft wraps in the console	If this check box is selected, the *soft wraps* (or *word wraps*) are used in the cor
Use custom soft wrap indent	Select this check box to use custom indentation for soft wraps on resizing the e Specify the indent value in the text field to the right. This field is only available when at least one of the **Use soft wraps...** check bo is selected.
Show all soft wraps	If this check box is selected, the soft wrap characters ↲ ↳ will be always shown a the end of each line, and at the beginning of each next line. Otherwise, the soft wrap characters will be shown in the active logical line only.

Item	Description
Allow placement of caret after end of line	If this check box is cleared, the caret never rests after the last symbol in a line.
Allow placement of caret inside tabs	Select this check box to allow placing the caret inside tab characters. The reaso that each tab character shows in the editor as a set of 'virtual' space character
Show virtual space at file bottom	If this check box is selected, the currently edited line (even if it is the final line) be scrolled to the top of the screen. IntelliJ IDEA adds the necessary amount of virtual lines.
Highlight on Caret Movement	
Highlight matched brace	Select this check box to have IntelliJ IDEA highlight pairs of opening/closing brac when you position the caret right before the opening or right after the closing o also works for HTML and XML tags.
Highlight current scope	Select this check box to have IntelliJ IDEA highlight the available scope for the c typed in the current caret location.
Highlight usages of element at caret	Select this check box to have IntelliJ IDEA highlight all usages of the element at which the caret is currently positioned.
Formatting	
Show "Reformat Code" dialog	If this check box is selected, the Reformat Code dialog will appear every time yo to reformat code, where you can specify the scope of reformatting. Otherwise, IntelliJ IDEA will reformat code silently.
Show "Optimize Imports" dialog	If this check box is selected, the Optimize Imports dialog will appear every time try to optimize import statements. Otherwise, IntelliJ IDEA will optimize imports silently.
Refactorings	
Enable in-place mode	Select or clear this check box to enable or disable in-place refactorings for Java. The *in-place* in connection with the refactorings means specifying all or most of information necessary for the refactoring by typing, right in the editor. All the affected code fragments are highlighted and change as you type. If appropriate, additional refactoring options are selected in corresponding option boxes. The *in-place refactoring mode* is available for the following refactorings: ▪ Extract Constant ▪ Extract Field ▪ Extract Parameter ▪ Extract Variable ▪ Rename If this check box is not selected, the refactoring settings for all of the refactorin are specified in the corresponding dialogs.

Item	Description
Preselect old name	If this option is on, the old name of a symbol is selected when the Rename refactoring is invoked for that symbol.  Rename Rename variable inchesDouble and its usages to: inchesDouble ☑ Search in comments and strings  If option is off, the symbol being renamed is not selected.

See Also

Procedures:

- Advanced Editing Procedures

Getting Started:

- Familiarize Yourself with IntelliJ IDEA Editor

Web Resources:

- Developer Community