Extract Variable

The *Extract Variable* refactoring puts the result of the selected expression into a variable. It declares a new variable and uses the expression as an initializer. The original expression is replaced with the new variable (see the examples below) .

In Java, the type of the new variable corresponds to that returned by the expression. There is an option of declaring the new variable as `final`.

> The following is available only in the Ultimate edition of IntelliJ IDEA

To perform this refactoring, you can use:

- In-place refactoring. In this case you specify the new name right in the editor.
- Refactoring dialog, where you specify all the required information. To make such a dialog accessible, you have to clear the check box Enable in-place mode in the editor settings.

You can select the expression to be replaced with a variable yourself. You can as well use smart expression selection. In this case IntelliJ IDEA will help you select the desired expression.

This refactoring is also available for JavaScript and SASS .

Java examples

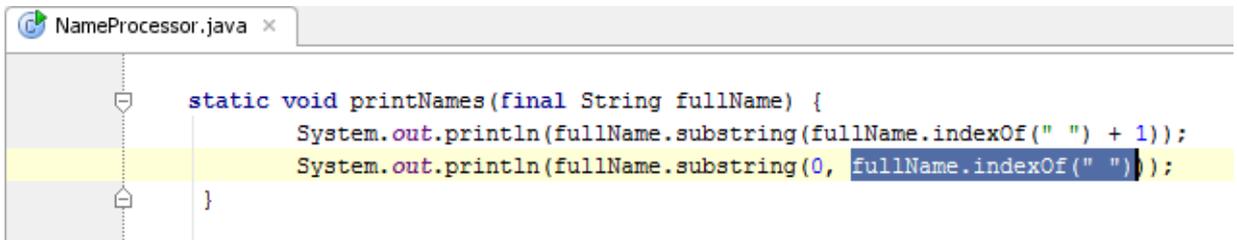| Before	
```java public void method() { int a = 1; ... int b = a + anotherClass.intValue(); int c = b + anotherClass.intValue(); } ```	```java public int a = 1; ... int number = another int b = a + number; int c = b + number; } ```
```java public void method() { int a = anotherClass.innerClass.i; int b = anotherClass.innerClass.j; } ```	```java public AnotherClass.InnerCl int a = aClass.i; int b = aClass.j; } ```
```java static void printNames(final String fullName) { System.out.println(fullName.substring(fullName.indexOf(" ") + 1)); System.out.println(fullName.substring(0, fullName.indexOf(" "))); } ```	```java static int firstNameEndInde System.out.println(f System.out.println(f } ```

To extract a variable using in-place refactoring

1. In the editor, select the expression to be replaced with a variable. You can do that yourself or use the *smart expression selection* feature to let IntelliJ IDEA help you. So, do one of the following:
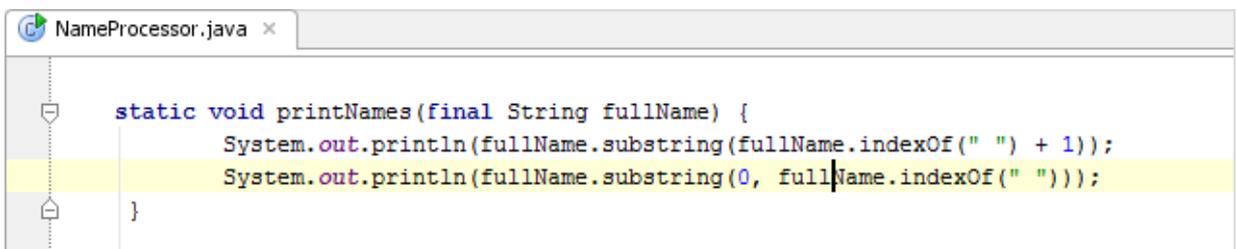
   - Highlight the expression. Then choose **Refactor | Extract | Variable** from the main or the context menu.

     Alternatively, press `Ctrl+Alt+V`.

     ```
     NameProcessor.java  ×

         static void printNames(final String fullName) {
             System.out.println(fullName.substring(fullName.indexOf(" ") + 1));
             System.out.println(fullName.substring(0, fullName.indexOf(" ")));
         }
     ```
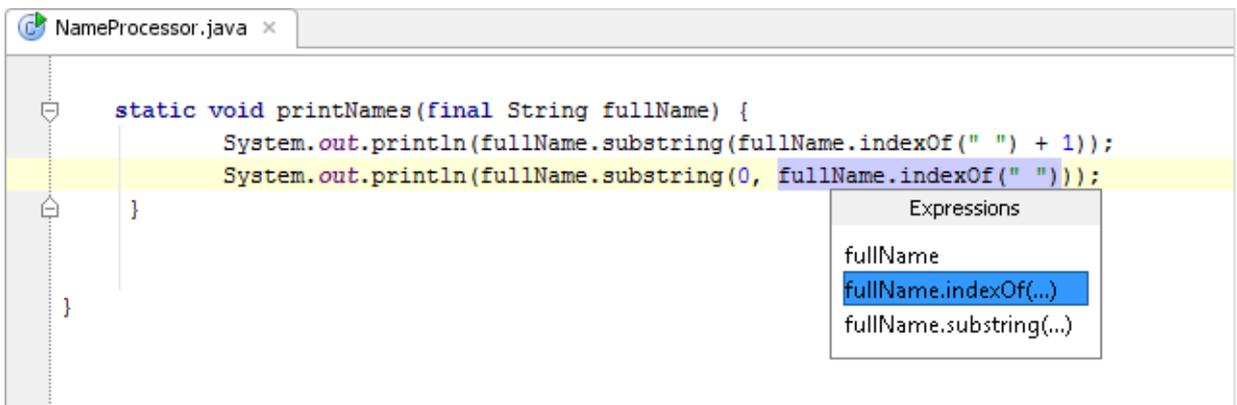
   - Place the cursor before or within the expression. Choose **Refactor | Extract Variable** from the main or the context menu, or press `Ctrl+Alt+V`.

     ```
     NameProcessor.java  ×

         static void printNames(final String fullName) {
             System.out.println(fullName.substring(fullName.indexOf(" ") + 1));
             System.out.println(fullName.substring(0, full|Name.indexOf(" ")));
         }
     ```

     In the **Expressions** pop-up menu, select the expression. To do that, click the required expression. Alternatively, use the `Up` and `Down` arrow keys to navigate to the expression of interest, and then press `Enter` to select it.

     ```
     NameProcessor.java  ×

         static void printNames(final String fullName) {
             System.out.println(fullName.substring(fullName.indexOf(" ") + 1));
             System.out.println(fullName.substring(0, fullName.indexOf(" ")));
         }
     }
                                              ┌─ Expressions ──────────┐
                                              │ fullName               │
                                              │ fullName.indexOf(...)  │
                                              │ fullName.substring(...)│
                                              └────────────────────────┘
     ```
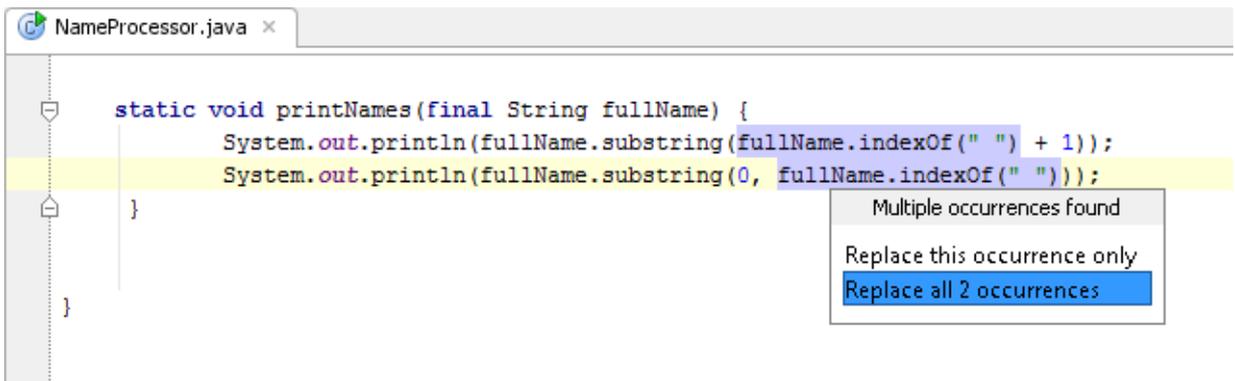
     The **Expressions** pop-up menu contains all the expressions appropriate for the current cursor position in the editor.
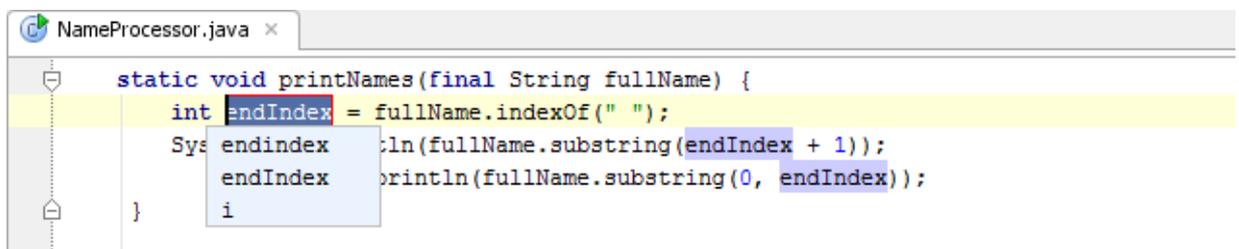
     When you navigate through the suggested expressions in the pop-up, the code highlighting in the editor changes accordingly.

2. If more than one occurrence of the selected expression is found, select **Replace this occurrence only** or **Replace all occurrences** in the **Multiple occurrences found** pop-up menu. To select the required option, just click it. Alternatively, use the Up and Down arrow keys to navigate to the option of interest, and press Enter to select it.
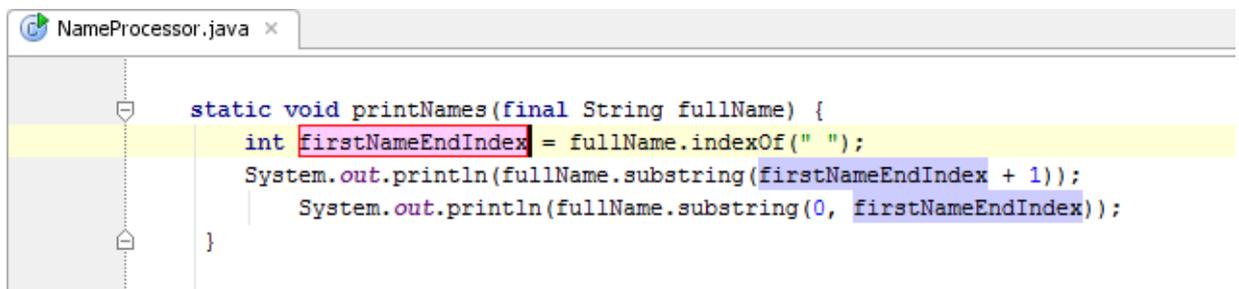

3. Specify the name of the variable. Do one of the following:

    ■ Select one of the suggested names from the pop-up list. To do that, double-click the suitable name. Alternatively, use the Up and Down arrow keys to navigate to the name of interest, and Enter to select it.
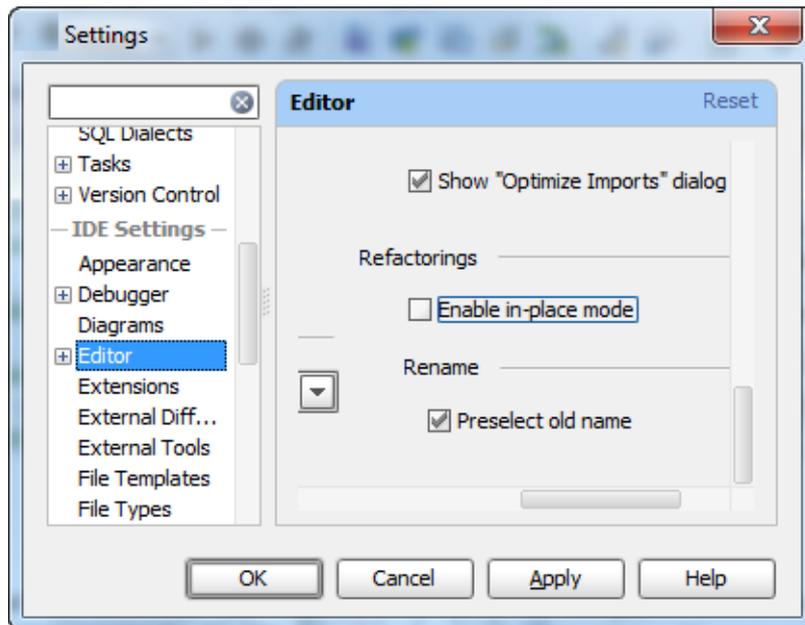

    ■ Edit the name by typing. The name is shown in the box with red borders and changes as you type. When finished, press Escape.

To extract a variable using the dialog box

If the **Enable in place refactorings** check box is cleared in the Editor settings, the **Extract Variable** refactoring is performed by means of the dialog box.

1. In the editor, select the expression to be replaced with a variable. You can do that yourself or use the *smart expression selection* feature to let IntelliJ IDEA help you. So, do one of the following:

   - Highlight the expression. Then choose **Refactor | Extract | Variable** from the main or the context menu.

     Alternatively, press `Ctrl+Alt+V`.

   - Place the cursor before or within the expression. Choose **Refactor | Extract Variable** from the main or the context menu, or press `Ctrl+Alt+V`.

     In the **Expressions** pop-up menu, select the expression. To do that, click the required expression. Alternatively, use the `Up` and `Down` arrow keys to navigate to the expression of interest, and then press `Enter` to select it.

     > The **Expressions** pop-up menu contains all the expressions appropriate for the current cursor position in the editor.
     >
     > When you navigate through the suggested expressions in the pop-up, the code highlighting in the editor changes accordingly.

2. In the Extract Variable Dialog dialog:

   1. Specify the variable name next to **Name**. You can select one of the suggested names from the list or type the name in the **Name** box.

   2. If more than one occurrence of the selected expression is found, you can select to replace all the found occurrences by selecting the corresponding check box. If you want to replace only the current occurrence, clear the **Replace all occurrences** check box.

   3. If you want to declare the new variable `final`, select the **Declare final** check box. (This option is available only for Java.)

   4. For ActionScript, you can choose to introduce a constant rather than a variable. To do that, select the **Make constant** check box.

      Note that ActionScript is not supported in IntelliJ IDEA Community Edition.

   5. Click **OK**.

Refer to the following sections for the other language- and framework-specific refactorings:

- For the JavaScript procedure, refer to the section Extract Variable in JavaScript.

- For the SASS procedure, refer to the section Extract Variable for SASS.

See Also

Procedures:

- Selecting Text in the Editor
- Rename Refactorings
- Extract Variable in JavaScript

Reference:

- Introduce Dialogs
- Extract Variable Dialog

Web Resources:

- Developer Community 🗗