



Refactor code

Last modified: 26 April 2024

[Refactoring](#) [↗] is a process of improving your source code without creating a new functionality. Refactoring helps you keep your code solid and easy to maintain.

To perform refactoring, follow these general steps

1. Select (or hover over) a symbol or code fragment to refactor.

You can select symbols in the following PyCharm components:

- Project view
- Structure tool window
- Editor
- UML Class diagram

2. Do one of the following:


- On the main **Refactor** menu or from the context menu of the selection, choose the desired refactoring or press the corresponding keyboard shortcut (if any).
- Go to **Refactor | Refactor This**, or press ⌘ **Ctrl** T, and then select the desired refactoring from the popup.

3. In the dialog that opens, specify the refactoring options.

4. To apply the changes immediately, depending on the refactoring type, click **Refactor** or **OK**.

5. For certain refactorings, there is an option of previewing the changes prior to actually performing the refactoring. In such cases the **Preview** button is available in the corresponding dialog.



To preview the potential changes and make the necessary adjustments, click **Preview**. PyCharm displays the changes that are going to be made on a dedicated tab of the [Find tool window](#).

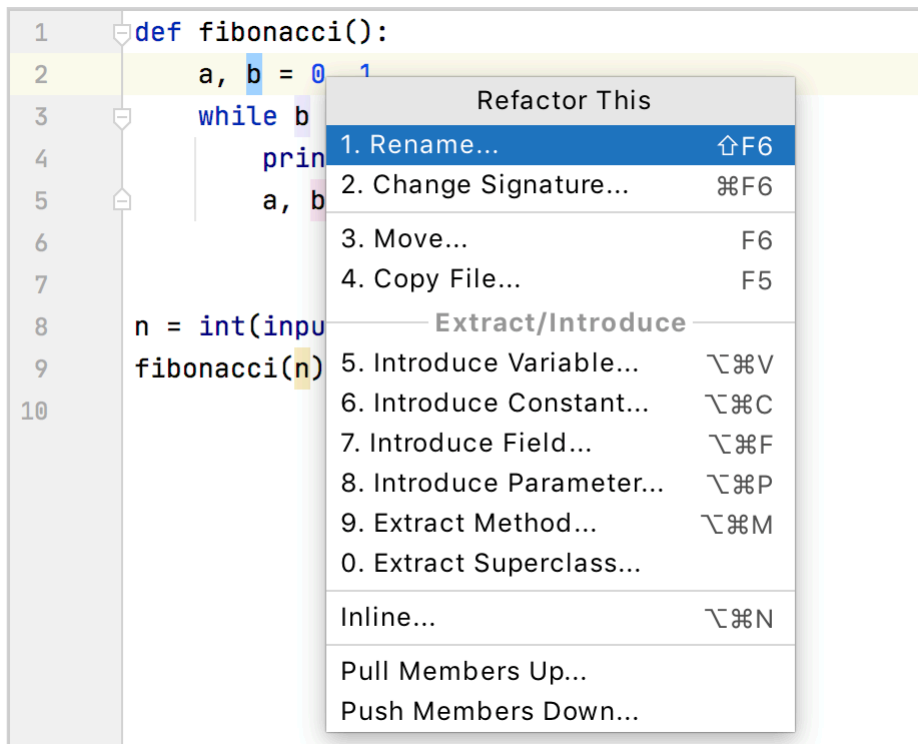
One of the possible actions at this step is to exclude certain entries from the refactoring. To do so, select the desired entry in the list and press  **Delete**.

If conflicts are expected after the refactoring, PyCharm displays a dialog with a brief description of the encountered problems. If this is the case, do one of the following:

- Ignore the conflicts by clicking the **Continue** button. As a result, the refactoring will be performed, however, this may lead to erroneous results.
 - Preview the conflicts by clicking the **Show in View** button. PyCharm shows all conflicting entries on the **Conflicts** tab in the [Find tool window](#), enabling you to navigate to the problematic lines of code and to make the necessary fixes.
 - Cancel the refactoring and return to the editor.
6. When you are satisfied with the proposed results, click **Do Refactor** to apply the changes.

Invoke refactoring

1. Select an item to refactor. You can select a file/folder in the **Project** tool window or expression/symbol in the editor.
2. Press  **Ctrl**  **T** to open a list of refactorings that can be selected.



Alternatively, you can use a keyboard shortcut for a specific refactoring.

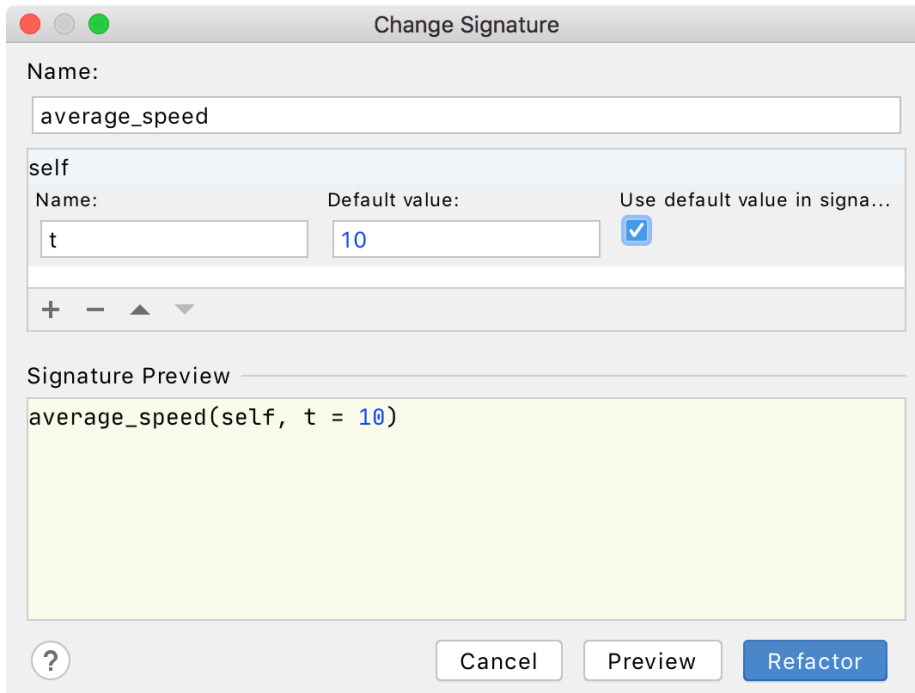
You can also make a multi-caret selection for an extended range of refactoring operations, including **Encapsulate Fields**, **Move Static Members**, **Extract Delegate**, **Extract Interface**, and **Extract Superclass**. To utilize this feature, place the caret in each member you want to apply the refactoring to, and the IDE will automatically preselect and check these members in the refactoring dialog.




If you need to undo your refactoring, press `⌘ Cmd Z`.

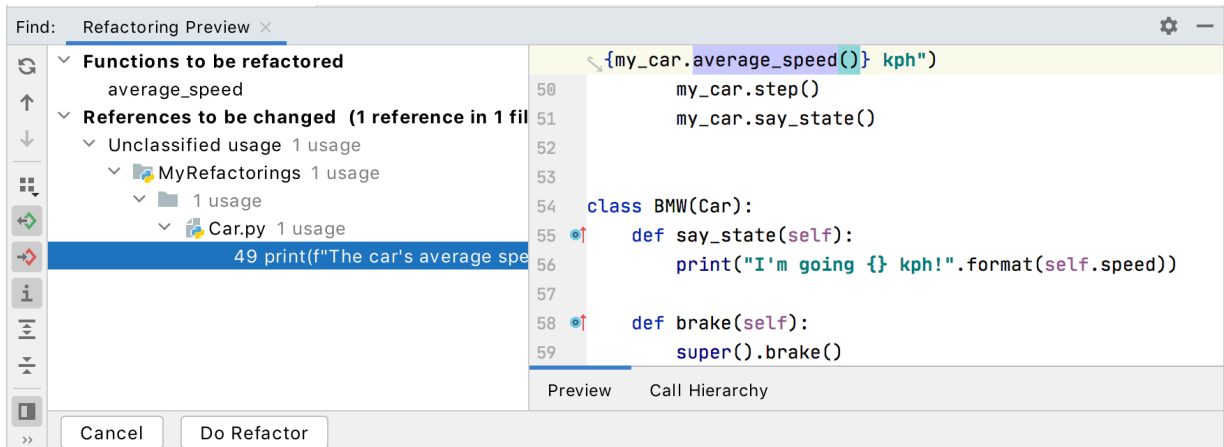
Preview refactoring changes

For some refactorings, PyCharm lets you preview the changes before applying them.

1. Click **Preview** in the **Refactoring Preview** dialog to see potential changes (the list of usages where the refactoring will be performed).



2. In the Find tool window that opens, check the changes that are going to be made. You can exclude  Delete or remove  Cmd  changes that you consider unnecessary.



3. Click **Do Refactor** to proceed with the changes.

Configure refactoring settings

1. In the **Settings** dialog ( Cmd ), select **Editor | Code Editing**.


2. On the **Code Editing** page, in the **Refactorings** section, adjust the refactoring options and click **OK**.



You can also adjust the refactoring intentions, in **Editor | Intentions**.

The most popular refactorings supported in PyCharm

Safe Delete	 Cmd  Delete	Makes sure that you do not delete files that are referenced in your source code.	Safe delete
Copy/Move	F5 / F6	Copies/Moves an element.	Copy and Move Refactorings
Extract Method	 Cmd  Opt M	These refactoring actions help you reduce the code duplication.	Extract method
Extract Constant	 Cmd  Opt C		Extract constant
Extract Field	 Cmd  Opt F		Extract field
Extract Parameter	 Cmd  Opt P		Extract parameter
Introduce Variable	 Cmd  Opt V		Extract/Introduce variable
Rename	 Shift F6	Renames an element.	Rename refactorings
Inline	 Cmd  Opt N	Inlines an element. Acts as the opposite of extracting.	Inline

Change signature	 Cmd F6	Changes the call signature of a method or class.	<u>Change signature</u>
------------------	--	--	-------------------------



PyCharm 2024.1 ▼

[Configure projects in PyCharm](#) / [Configure a Python interpreter](#) / [Configure an interpreter using SSH](#)

Professional

Configure an interpreter using SSH

Last modified: 09 May 2024

Available only in PyCharm Professional: [download](#) ↗ to try or [compare editions](#) ↗



You cannot use a Windows machine as a remote host when configuring SSH interpreters.

Configuring remote Python interpreters via SSH

1. Ensure that there is an SSH server running on a remote host, since PyCharm runs remote interpreters via ssh-sessions.
2. Do one of the following:
 - Click the [Python Interpreter selector](#) and choose **Add New Interpreter**.
 - Press `⌘ Cmd` `,` to open **Settings** and go to **Project: <project name> | Python Interpreter**. Click the **Add Interpreter** link next to the list of the available interpreters.

- Click the [Python Interpreter selector](#) and choose **Interpreter Settings**. Click the **Add Interpreter** link next to the list of the available interpreters.

3. Select **On SSH**.

4. Select an option to create a new SSH connection, then specify server information (host, port, and username).

The screenshot shows a macOS-style dialog box titled "New Target: SSH". Below the title bar, it says "1/4. Connecting to SSH server". There are two radio buttons: "New" (selected) and "Existing". Below these are three input fields: "Host:" with the value "10.211.55.255", "Port:" with the value "22", and "Username:" with the value "jetbrains". At the bottom, there is a question mark icon, a "Cancel" button, and a "Next" button.

Alternatively, you can select **Existing** and choose any available SSH configuration from the list. To create a new SSH configuration, follow the steps below:

Creating an SSH configuration

5. In the next dialog window, provide the authentication details to connect to the target server.

The screenshot shows the next step in the "New Target: SSH" dialog box. The title bar is the same. Below the title bar, it says "Authentication options for connecting to jetbrains@10.211.55.255:22". There are two radio buttons: "Password:" (selected) and "Key pair: OpenSSH or PuTTY". The "Password:" option has a text input field with masked characters "....." and a checked "Save password" checkbox. The "Key pair:" option has a "Private key:" text input field with a folder icon, a "Passphrase:" text input field, and a checked "Save passphrase" checkbox. At the bottom, there is a question mark icon, a "Cancel" button, a "Previous" button, and a "Next" button.