



PyCharm 2024.3 ▼

[Getting started](#) / [Learn keyboard shortcuts](#)

# PyCharm keyboard shortcuts

Last modified: 06 November 2024










PyCharm has keyboard shortcuts for most of its commands related to editing, navigation, refactoring, debugging, and other tasks. Memorizing these hotkeys can help you stay more productive by keeping your hands on the keyboard.

All default shortcuts are configurable and you can also assign shortcuts to any action that does not have them by default.



If your keyboard does not have an English layout, PyCharm may not detect all the shortcuts correctly.

The following table lists some of the most useful shortcuts to learn:

Shortcut	Action
Double 	<u>Search Everywhere</u>  Quickly find any file, action, class, symbol, tool window, or setting in PyCharm, in your project, and in the current Git repository.
  	<u>Find Action</u>  Find a command and execute it, open a tool window, or search for a setting.
 	<u>Run/Rerun</u>  Run and Rerun entire applications as well as particular scripts.
Double 	<u>Run Anything</u>  Execute commands, such as opening a project, launching a run/debug configuration, running a command-line utility, and so on. The available commands depend on the set of plugins and tools you have configured for your project.
 	<u>Show Context Actions</u>  Quick-fixes for highlighted errors and warnings, intention actions for improving and optimizing your code.

Shortcut	Action
<div>F2</div> <div>Shift F2</div>	<div><u>Navigate between code issues</u></div> <div>Jump to the next or previous highlighted error.</div>
<div>Ctrl E</div>	<div><u>View recent files</u></div> <div>Select a recently opened file from the list.</div>
<div>Ctrl W</div> <div>Ctrl Shift W</div>	<div><u>Extend or shrink selection</u></div> <div>Increase or decrease the scope of selection according to specific code constructs.</div>
<div>Ctrl /</div> <div>Ctrl Shift /</div>	<div><u>Add/remove line or block comment</u></div> <div>Comment out a line or block of code.</div>
<div>Alt F7</div>	<div><u>Find Usages</u></div> <div>Show all places where a code element is used across your project.</div>

## Choose the right keymap

To view the keymap configuration, open the **Settings** dialog 

Ctrl

Alt

S

 and select **Keymap**.






Enable function keys and check for possible conflicts with global OS shortcuts.

## Use a predefined keymap

PyCharm automatically suggests a predefined keymap based on your environment. Make sure that it matches the OS you are using or select the one that matches shortcuts from another IDE or editor you are used to (for example, Emacs).

When consulting this page and other pages in PyCharm documentation, you can see keyboard shortcuts for the keymap that you use in the IDE — choose it using the selector at the top of a page.

## Install a keymap from a plugin

Besides the default set of keymaps, you can install keymaps from plugins (such as, keymaps for GNOME and KDE): open the **Settings** dialog   , select **Plugins**, switch to the **Marketplace** tab and search for **keymap**.

## Tune your keymap

You can modify a copy of any predefined keymap to assign your own shortcuts for commands that you use frequently.

## Import custom keymap

If you have a customized keymap that you are used to, you can transfer it to your installation.



If your keymap stopped working after an update, it is likely that the keymap is not available by default in the new version of PyCharm. Find this keymap as a plugin and install it on the Plugins page as described in Install plugins.

# Print your keymap

If you prefer a hard copy, download the default PyCharm keymap reference and print it out as a card:

- [Default PyCharm keymap for macOS ↗](#)
- [Default PyCharm keymap for Windows and Linux ↗](#)

# Learn shortcuts as you work

PyCharm provides several possibilities to learn shortcuts:

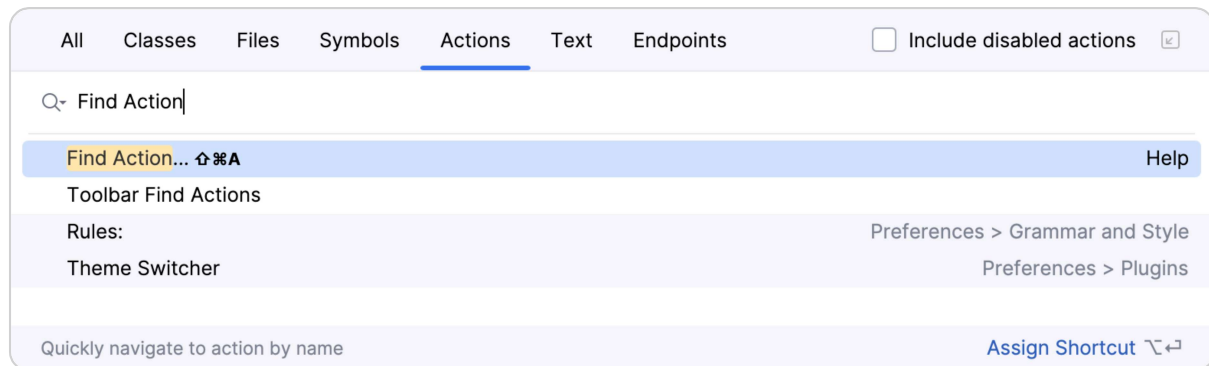
- PyCharm includes a built-in IDE Features Trainer. It provides interactive training courses to help you learn how to do some common tasks, such as refactoring code, navigating through projects, debugging, and working with version control.

Make sure the IDE Features Trainer plugin is enabled in the **Installed** tab of the **Plugins** page in the IDE settings.

To open the **Learn** tool window and start your exercises, select **View | Tool Windows | Learn** or **Help | Learn IDE Features** from the main menu. The tool window lists the available tutorials.

- The [Find Action](#) lets you search for commands and settings across all menus and tools.

Press **Ctrl** **Shift** **A** and start typing to get a list of suggested actions. Then select the necessary action and press **Enter** to execute it.



To add or change the shortcut for any action, press **Alt** **Enter** when it is selected in the list.

- [Key Promoter X ↗](#) is a plugin that shows a popup notification with the corresponding keyboard shortcut whenever a command is executed using the mouse. It also suggests creating a shortcut for commands that are executed frequently.
- If you are using one of the predefined keymaps, you can print the [default keymap reference card ↗](#) and keep it on your desk to consult it if necessary. This cheat sheet is also available under **Help | Keyboard Shortcuts PDF**.
- To print a non-default or customized keymap, use the [Keymap exporter plugin ↗](#).

## Use advanced features

You can further improve your productivity with the following useful features:

### Quick Lists

If there is a group of actions that you often use, create a [quick list](#) to access them using a custom shortcut. For example, you can try using the following predefined quick lists:

- **Refactor this** **Ctrl** **Alt** **Shift** **T**
- **VCS Operations** **Alt** **`**

**Smart Keys**

PyCharm provides a lot of typing assistance features, such as automatically adding paired tags and quotes, and detecting **CamelHump** words.

**Speed search**

When the focus is on a tool window with a tree, list, or table, start typing to see matching items.

**Press twice**

Some actions in PyCharm provide more results when you execute them multiple times. For example, when you invoke basic code completion with **Ctrl Space** on a part of a field, parameter, or variable declaration, it suggests names depending on the item type within the current scope. If you invoke it again, it will include classes available through module dependencies. When invoked for the third time in a row, the list of suggestions will include the whole project.

**Resize tool windows**

You can adjust the size of tool windows without a mouse:

- To resize a vertical tool window, press **Ctrl Alt Shift ←** and **Ctrl Alt Shift →**
- To resize a horizontal tool window, press **Ctrl Alt Shift ↑** and **Ctrl Alt Shift ↓**