

PyCharm 2020.2

Work with source code

Last modified: 28 July 2020

When you work with code, PyCharm ensures that your work is stress-free. It offers various shortcuts and features to help you add, select, copy, move, edit, fold, find occurrences, and save code.

For navigation inside the editor, refer to [Editor basics](#).



Find action

- If you do not remember a shortcut for the action you want to use, press **Ctrl+Shift+A** to find any action by name.

You can use the same dialog to find classes, files, or symbols. For more information, refer to [Searching Everywhere](#).



Add a new class, file, package, or scratch file

- In the editor, press **Ctrl+Alt+Insert** to add a class, file, or package.

If the focus is inside the **Project** tool window and you want to add a new element, press **Alt+Insert**.



- To create a new *Scratch* file, press **Ctrl+Alt+Shift+Insert**.

PyCharm creates a tempo
refer to [Scratch files](#).

[X]

Cookies and IP addresses allow us to deliver and improve our web content and to provide you with a personalized experience. Our website uses cookies and collects your IP address for these purposes.

✕ Toggle read-only attribute of a file

If a file is read-only, it is marked with the closed lock icon  in the status bar, in its editor tab, or in the Project tool window. If a file is writable, it is marked with the open lock icon  in the Status bar.

1. Open file in the editor or select it in the Project tool window.
2. Do one of the following:
 - From the main menu, select **File | Make File Read-only**, or **File | Make File Writable**.
 - Click the lock icon in the status bar.

If a read-only status is set by a version control system, it's suggested that you use PyCharm version control integration features. For more information, see Version control.

✕ Select code constructs

- In the editor, place the caret at the item you want to select and press `Ctrl+W` / `Ctrl+Shift+W` to extend or shrink your selection.

For example, in a plain text file, the selection starts within the whole word then extends to the sentence, paragraph, and so on.

- If you need just to highlight your braces, place the caret immediately after the block closing brace/bracket or before the block opening brace/bracket.

✕ Select code according to capitalization

1. In the **Settings/Preferences** dialog, `Ctrl+Alt+S`, go to **Editor | General | Smart Keys**.

2. Select the **Use "CamelHu**

If you want to use double-cli

[X] Cookies and IP addresses allow us to deliver and improve our web content and to provide you with a personalized experience. Our website uses cookies and collects your IP address for these purposes.

the **Honor CamelHumps words...** checkbox is selected on the **Editor | General** page of the **Settings/Preferences** dialog `Ctrl+Alt+S`.



Select occurrences

You can quickly locate all occurrences of the selected element in a file, make a multiple selection, or select the next or the previous occurrences.

- To make a multiple selection of occurrences in a file, select the needed element in your code, then press `Alt+J`. Press the same key to find the next occurrence.

Alternatively, keeping `Shift+Alt` pressed, double-click or drag over the text you want to add to the selection.

- To move the caret to the next or the previous occurrence, press `F3` or `Shift+F3` respectively.

When you press `F3`, PyCharm automatically removes the selection of the current occurrence and selects the next one.

- To select all occurrences in the file, press `Ctrl+Alt+Shift+J`.
- To find the next or the previous occurrence of the selected identifier, press `Alt+Wheel up` or `Alt+Wheel down`.

Configure tabs and indents

- In the **Settings/Preferences** dialog `Ctrl+Alt+S`, go to **Editor | Code Style**.
- Select a language for which you want to configure the indentation.
- From the options on the right, on the **Tabs and Indents**, select the **Use tab character** for the editor to use tabs ^[X]
configure the tab size if you use spaces.

Cookies and IP addresses allow us to deliver and improve our web content and to provide you with a personalized experience. Our website uses cookies and collects your IP address for these purposes.

✕ Copy and paste code

You can use the standard shortcuts to copy `Ctrl+C` and paste `Ctrl+V` any selected code fragment. If nothing is selected, PyCharm automatically copies the whole line where the caret is located.

By default, when you paste anything in the editor, PyCharm performs "smart" paste, for example, pasting multiple lines in comments will automatically add the appropriate markers to the lines you are pasting. If you need to paste just plain text, press

`Ctrl+Alt+Shift+V`.

- Place the caret at a line or a symbol and press `Ctrl+Alt+Shift+C` / `Ctrl+V`.
- PyCharm keeps track of everything you copy to the clipboard. To paste from history, press `Ctrl+Shift+V`. In the dialog that opens, select your entry and click **Paste**.



The default number of items stored in the clipboard history is 100.

- You can copy the reference to a line or a symbol.

Place the caret at a certain line of code and press `Ctrl+Alt+Shift+C` (or select **Copy Reference** from the context menu or from the **Edit** menu). PyCharm creates a string in the format that depends on a symbol at caret. For example:

`Solver.Solver.dscr` for a Python method

`Solver\Solver.py:14` for a Python file

Lines of code

PyCharm offers several useful shortcuts for manipulating code lines

[X]




If you need to undo or re

Cookies and IP addresses allow us to deliver and improve our web content and to provide you with a personalized experience. Our website uses cookies and collects your IP address for these purposes.

- To add a line after the current one, press `Shift+Enter`. PyCharm moves the caret to the next line.
- To add a line before the current one, press `Ctrl+Alt+Enter`. PyCharm moves the caret to the previous line.
- To duplicate a line, press `Ctrl+D`.
- To sort lines alphabetically in the whole file or in a code selection, from the main menu, select **Edit | Sort Lines** or **Edit | Reverse Lines**. These actions might be helpful when you work with property files, data sets, text files, log files, and so on. If you need to assign shortcuts to those actions, refer to [Configuring keyboard shortcuts](#) for more information.
- To delete a line, place the caret at the line you need and press `Ctrl+Y`.
- To join lines, place the caret at the line to which you want to join the other lines and press `Ctrl+Shift+J`. Keep pressing the keys until all the needed elements are joined.

You can also join string literals, a field or variable declaration, and a statement. Note that PyCharm checks the code style settings and eliminates unwanted spaces and redundant characters.

- To split string literals into two parts, press `Enter`.

PyCharm splits the string and provides the correct syntax. You can also use the **Break string on '\n'** intention to split string literals. Press `Alt+Enter` or click  to select this intention.

- To comment a line of code, place the caret at the appropriate line and press `Ctrl+/.`
- To move a line up or down, press `Alt+Shift+Up` or `Alt+Shift+Down` respectively.
- To move (swap) a code element to the left or to the right, place the caret at it, or select it and press `Ctrl+Alt+Shift+Left` for left or `Ctrl+Alt+Shift+Right` for right.


[X]

Cookies and IP addresses allow us to deliver and improve our web content and to provide you with a personalized experience. Our website uses cookies and collects your IP address for these purposes.

Code statements

Move statements

- In the editor, place the caret at the needed statement and press `Ctrl+Shift+Up` to move a statement up or `Ctrl+Shift+Down` to move a statement down. PyCharm moves the selected statement performing a syntax check.

 If moving of the statement is not allowed in the current context, the actions will be disabled.

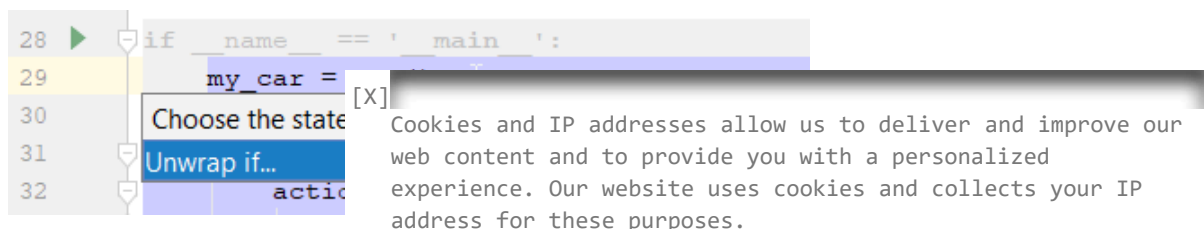
Complete current statement

- In the editor, press `Ctrl+Shift+Enter` or from the main menu select **Code | Complete Current Statement**. PyCharm inserts the required trailing comma automatically in structs, slices, and other composite literals. The caret is moved to the position where you can start typing the next statement.

Unwrap or remove statement

- Place the caret at the expression you want to remove or unwrap.
- Press `Ctrl+Shift+Delete`.

PyCharm shows a popup with all actions available in the current context. To make it easier to distinguish between statements to be extracted and statements to be removed, PyCharm uses different background colors.



3. Select an action and press **Enter**.

```

27
28 my_car = Car()
29 print("I'm a car!")
30 while True:
31     action = input("What should I do? [A]ccelerate,

```

Code fragments

- Move and copy code fragments by dragging them in the editor.
- To move a code fragment, select it and drag the selection to the target location.
- To copy a code selection, keeping **Ctrl** pressed, drag it to the target location.



The copy action might not be available in macOS since it can conflict with global OS shortcuts.

The drag functionality is enabled by default. To disable it, in the **Settings/Preferences** dialog **Ctrl+Alt+S**, go to **Editor | General** and clear the **Enable Drag'n'Drop functionality in editor** checkbox in the **Mouse** section.

- To toggle between the upper and lower case for the selected code fragment, press **Ctrl+Shift+U**.

Note that when you apply the toggle case action to the *CamelCase* name format, PyCharm converts the name to the lower case.

-  Block comments do not apply to Python scripts!

To comment or **uncomment** a code fragment, select it and press **Ctrl+Shift+/****.

To configure settings for commenting behavior, use options in the **Comment Code** section, on the **Code Generation** **File | Settings/Preferences**

Cookies and IP addresses allow us to deliver and improve our web content and to provide you with a personalized experience. Our website uses cookies and collects your IP address for these purposes.

Multiple carets

You can edit multiple lines simultaneously. For example, you can add comment markers to multiple lines, or remove some characters in several occurrences of a word.

 For multiple selection of occurrences, refer to the [select occurrences](#) section.

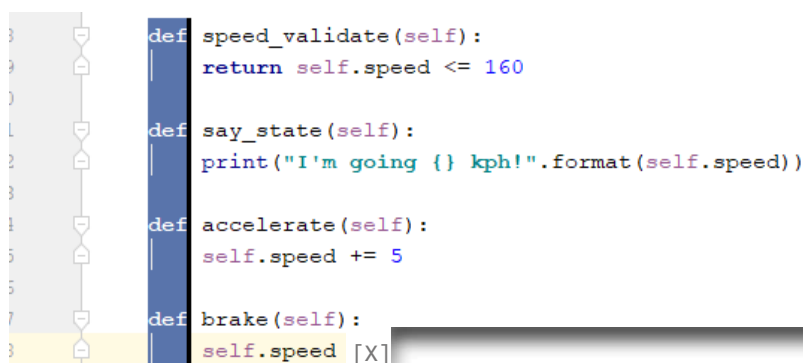
- To add or remove multiple carets, press and hold **Alt** and click at the next location of the caret.

Alternatively, to add multiple carets, you can press **Ctrl** twice, and then without releasing it, press the **Up** or **Down** arrow keys. You can also make a multiple selection by pressing **Shift+Alt** and dragging the caret.

- To delete all added carets, press **Escape**.
- You can make a multiple selection of code in columns. To enable or disable the column selection mode, press **Alt+Shift+Insert** or use the context menu of the editor.

This mode is helpful when you want to edit several lines of code simultaneously. Place the caret at a symbol you need and either drag it up or down, or select a block of code and press the specified shortcut to achieve the same result.

You can change the behavior of the caret move when you press **Alt+↑/↓**. The default behavior is to move the caret to the end of the current word. To change this setting, open settings **Ctrl+Alt+S** and navigate to **Editor | General**. In the **Caret Movement** section, select an appropriate behavior from the list.



```
3 def speed_validate(self):
4     return self.speed <= 160
5
6 def say_state(self):
7     print("I'm going {} kph!".format(self.speed))
8
9 def accelerate(self):
10    self.speed += 5
11
12 def brake(self):
13    self.speed [X]
```

Cookies and IP addresses allow us to deliver and improve our web content and to provide you with a personalized experience. Our website uses cookies and collects your IP address for these purposes.



By default, the column selection mode is disabled. When enabled, the mode affects only the current editor and is disabled if you close and reopen the editor.

Parameter hints

Parameter hints show the names of parameters in methods and functions to make your code easier to read. By default, parameter hints are enabled and shown only for values that are literals or function expressions but not for named objects.

Configure parameter hints


1. Open the **Settings/Preferences** dialog `Ctrl+Alt+S` and go to **Editor | Inlay Hints | <required language>**.
2. Select **Parameter hints** from the list, make sure the **Show parameter hints** checkbox is selected, and then specify the context where you want parameter hints shown.
3. For some methods and functions, PyCharm does not show parameter hints in any context. Click **Black list...** to view these methods and functions, possibly enable parameter hints for them, or add new items to the list.

Code folding

Folded code fragments are shown as shaded ellipses (...). If a folded code fragment contains errors, PyCharm highlights the fragment in red.

To configure the default code folding behavior, in the **Settings/Preferences** dialog `Ctrl+Alt+S`, go to **Editor | General | Code Folding** ^[X]

Cookies and IP addresses allow us to deliver and improve our web content and to provide you with a personalized experience. Our website uses cookies and collects your IP address for these purposes.

-  If PyCharm changes code in the folded fragment during the code reformatting, the code fragment will be automatically expanded.

Expand or collapse code elements

- To fold or unfold a code fragment, press `Ctrl+NumPad - / Ctrl+NumPad +`.
PyCharm folds or unfolds the current code fragment, for example, a single method.
- To collapse or expand all code fragments, press `Ctrl+Shift+NumPad - / Ctrl+Shift+NumPad +`.


PyCharm collapses or expands all fragments within the selection, or, if nothing is selected, all fragments in the current file, for example, all methods in a file.

- To collapse or expand code recursively, press `Ctrl+Alt+NumPad - / Ctrl+Alt+NumPad +`. PyCharm collapses or expands the current fragment and all its subordinate regions within that fragment.
- To fold blocks of code, press `Ctrl+Shift+.`. This action collapses the code fragment between the matched pair of curly braces `{ }`, creates a *custom folding region* for that fragment, and makes it "foldable".
- To collapse or expand doc comments in the current file, in the main menu select **Code | Folding | Expand doc comments/Collapse doc comments**.
- To collapse or expand a custom code selection, press `Ctrl+.`.

You can fold or unfold any manually selected regions in code.

Fold or unfold nested fragments

- To expand the current fragment and all the nested fragments, press

`Ctrl+NumPad *, 1`. You  [X]
level (from 1 to 5).

Cookies and IP addresses allow us to deliver and improve our web content and to provide you with a personalized experience. Our website uses cookies and collects your IP address for these purposes.

- To expand all the collapsed fragments in the file, press `Ctrl+Shift+NumPad *, 1`. You can expand the collapsed fragments up to the specified nesting level (from 1 to 5).

Use the Surround With action

You can collapse or expand code using the **Surround With** action.

1. In the editor, select a code fragment and press `Ctrl+Alt+T`.
2. In the popup menu, select **<editor-fold...> Comments** or **region...endregion Comments**.
3. If you need, name your custom region and collapse or expand the created region as usual `Ctrl+NumPad -` / `Ctrl+NumPad +`.
4. Press `Ctrl+Alt+.` to navigate to the created custom region.

Disable code folding outline

You can disable the code folding outline that appears on the left gutter.

1. In the **Settings/Preferences** dialog `Ctrl+Alt+S`, go to **Editor | General | Code Folding**.
2. Clear the **Show code folding outline** checkbox.

Autosave

PyCharm automatically saves changes to the project, such as compiling, running or a project, or quitting the IDE. But you can be sure that changes

[X]

Cookies and IP addresses allow us to deliver and improve our web content and to provide you with a personalized experience. Our website uses cookies and collects your IP address for these purposes.

-  To force saving all your files, open the **File** menu and click **Save All**, or press **Ctrl+S**.

Configure autosave behavior

1. In the **Settings/Preferences** dialog **Ctrl+Alt+S**, go to **Appearance and Behavior | System Settings**.
2. Under **Synchronization**, configure the following options:
 - **Save files on frame deactivation** (when you switch to a different application)
 - **Save files automatically if application is idle for N seconds**

Mark files with unsaved changes

1. In the **Settings/Preferences** dialog **Ctrl+Alt+S**, go to **Editor | General | Editor Tabs**.
2. Select the **Mark modified tabs with asterisk** checkbox.

Revert changes

- For the most recent changes, open the **Edit** menu and click **Undo**, or press **Ctrl+Z**.
- For a more detailed view of past changes, use [Local History](#).
- For the most robust tracking of all changes, use a [version control system supported in PyCharm](#).

[X]

Cookies and IP addresses allow us to deliver and improve our web content and to provide you with a personalized experience. Our website uses cookies and collects your IP address for these purposes.