Generate documentation with AI

Last modified: 08 May 2024

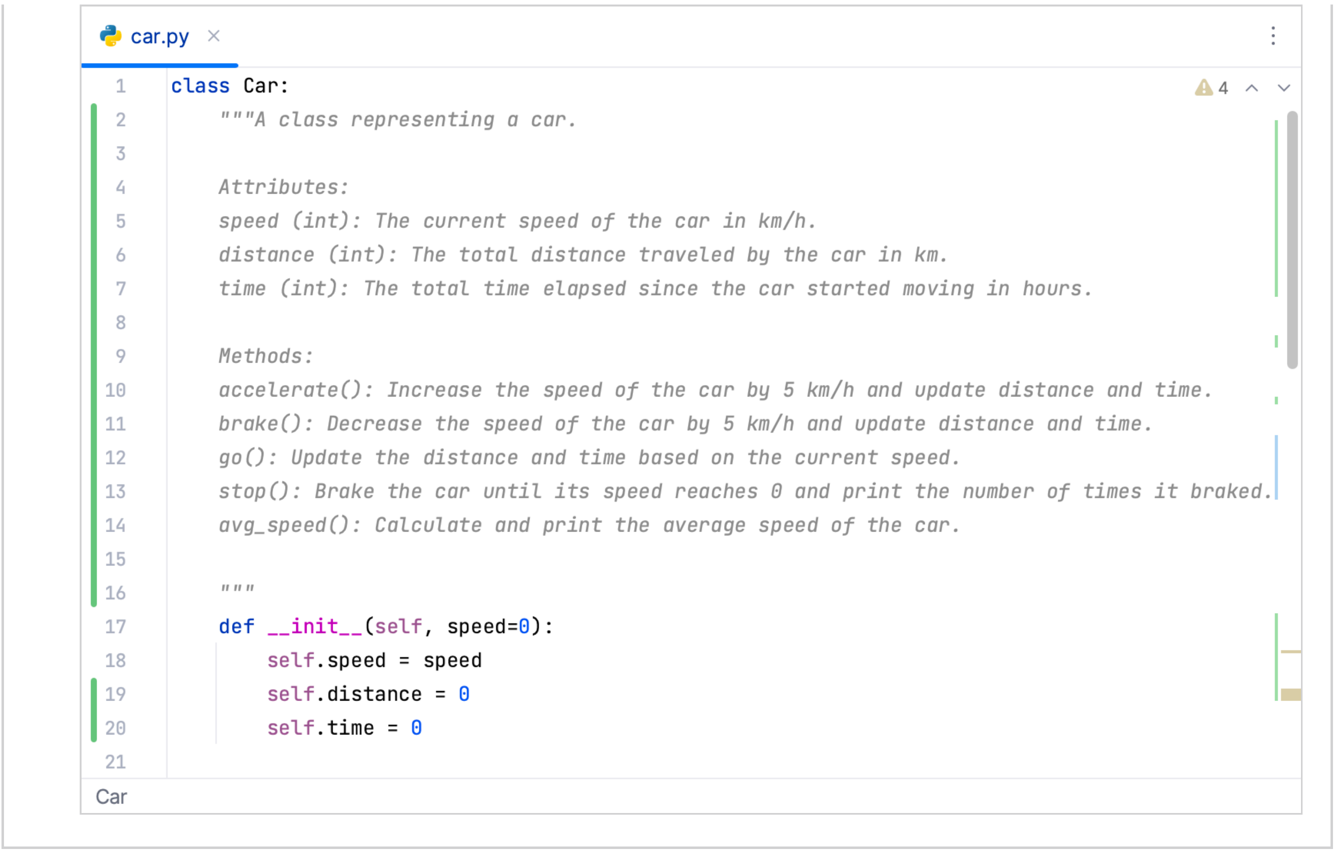
This functionality relies on the AI Assistant plugin that requires an additional license.

For more information, refer to JetBrains AI Service licensing and Enable AI Assistant plugin.

With AI Assistant, you can generate documentation for a declaration using the LLM (Large Language Model ↗).

1. Select a code fragment and right-click to open the context menu.
2. Press ⌥Opt ↩Enter. Select **AI actions** and then **Write documentation.**

   AI Assistant will generate documentation for you.



```python
class Car:
    """A class representing a car.

    Attributes:
    speed (int): The current speed of the car in km/h.
    distance (int): The total distance traveled by the car in km.
    time (int): The total time elapsed since the car started moving in hours.

    Methods:
    accelerate(): Increase the speed of the car by 5 km/h and update distance and time.
    brake(): Decrease the speed of the car by 5 km/h and update distance and time.
    go(): Update the distance and time based on the current speed.
    stop(): Brake the car until its speed reaches 0 and print the number of times it braked.
    avg_speed(): Calculate and print the average speed of the car.

    """
    def __init__(self, speed=0):
        self.speed = speed
        self.distance = 0
        self.time = 0

```

File type associations

Last modified: 11 February 2024

Configure: ⌘Cmd , **Settings | Editor | File Types**

For language-specific features (such as syntax highlighting and code analysis) in files representing different languages and technologies, PyCharm maintains a list of **file types**, each of which links a language service with one or more filename patterns.

The default list of file types covers all relevant filename patterns, but you can add new file types for your custom language files and change the associated filename patterns for existing file types.

If you are working on a language that is not supported in PyCharm by default, there might be plugins ↗ supporting that language depending on the PyCharm edition.

When you open a file in the editor, PyCharm chooses the file type and the corresponding language service according to the filename pattern. If the filename doesn't match any of the patterns registered for file types, you can associate the filename pattern with a specific file type.

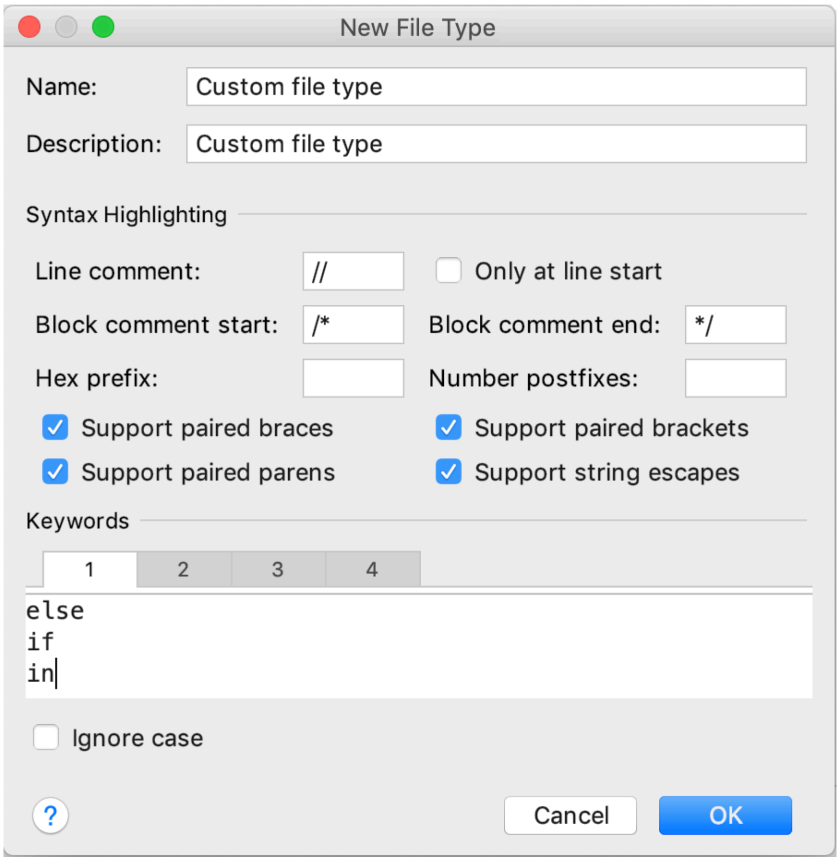
Apart from that, you can make PyCharm the default application for opening specific file types from the file manager on your operating system.

Add a custom file type

If you work on a language that is not supported by default and there are no plugins ↗ for it, you can configure a simple language service for files associated with this language — you will enjoy syntax highlighting for keywords, comments, and braces and have some basic editor helpers such as adding line/block comments with ⌘Cmd / / ⌘Cmd ⌥Opt / and extending/shrinking selection according to the structure with ⌥Opt ↑ / ⌥Opt ↓ .

1. Press ⌘Cmd , to open the IDE settings and then select **Editor | File Types**.
2. In the **Recognized File Types** section, click +, specify the name of the new type, and provide a description.
3. In the **Syntax Highlighting** section, configure case sensitivity, brace matching settings, and specify ways of defining comments:
    - **Line comment**: specify characters that indicate the beginning of a single-line comment.
    - **Only at line start**: characters that indicate the beginning of a line comment are recognized as a comment if they are located at the beginning of a line.
    - **Block comment start**, **Block comment end**: specify characters that indicate the beginning and the end of a block comment.
    - **Hex prefix**: specify characters that indicate that the subsequent value is a hexadecimal number (for example, `0x`).
    - **Number postfixes**: specify characters that indicate which numeric system or unit is used. A postfix is a trailing string of characters (for example, `e-3, kg`).
    - **Support paired braces**, **Support paired brackets**, **Support paired parens**, **Support string escapes**: select these checkboxes to highlight paired braces, brackets, parentheses, and string escapes.
4. In the **Keywords** section, you can specify up to four lists of keywords. Keywords of each list will be highlighted differently in the editor and will be auto-completed.

5. The **Ignore case** checkbox indicates whether keywords in files of the custom format are case-sensitive.



Name: Custom file type
Description: Custom file type

Syntax Highlighting

Line comment: // ☐ Only at line start
Block comment start: /* Block comment end: */
Hex prefix: Number postfixes:
☑ Support paired braces ☑ Support paired brackets
☑ Support paired parens ☑ Support string escapes

Keywords

```
else
if
in
```

☐ Ignore case

6. You can customize colors for syntax highlighting of language-specific keywords, comments, and other identifiers on the **Editor | Color Scheme | User-Defined File Types** settings page.

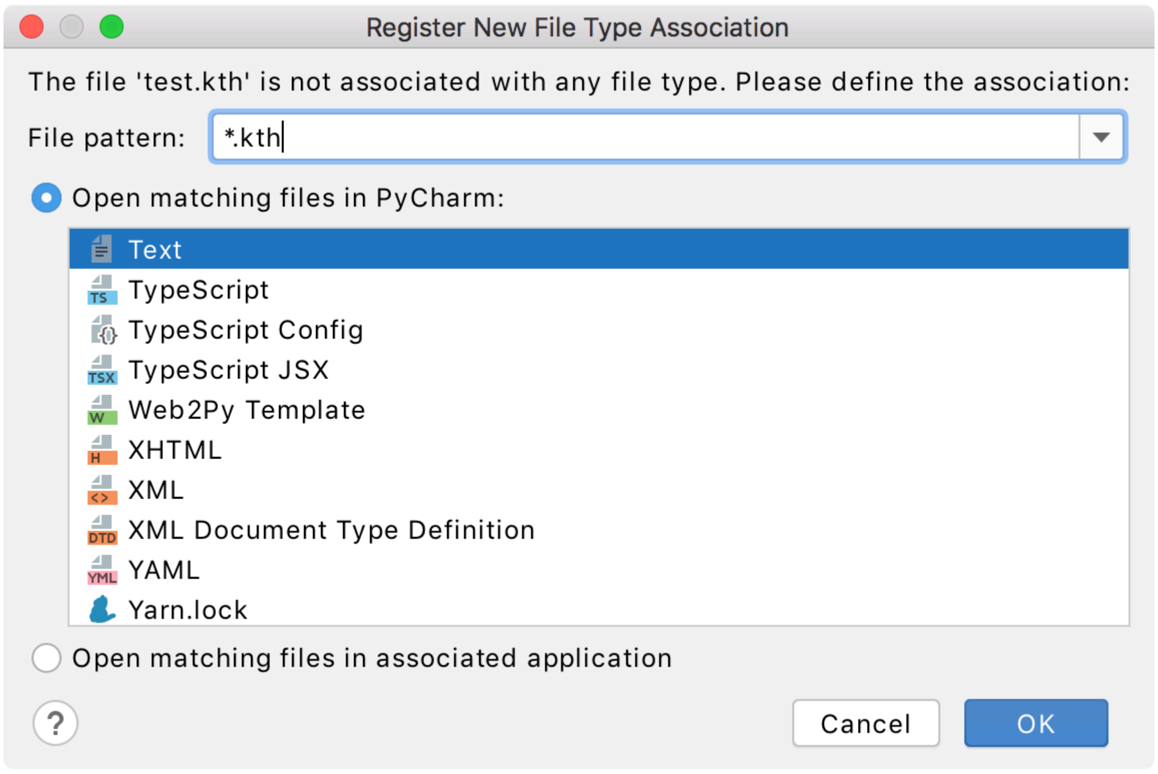
Configure associations between filename patterns and file types

Associate a filename pattern with specific file type

1. If PyCharm cannot identify the type of the file that you are trying to open or create, it displays the **Register New File Type Association** dialog where you can choose the way you want to process this file.

If the dialog doesn't appear automatically, right-click the file in the **Project** tool window and select **Associate with File Type** from the context menu or choose **File | File Properties | Associate with File Type** from the main menu.

2. In the **Register New File Type Association** dialog, select the necessary options:



The file 'test.kth' is not associated with any file type. Please define the association:

File pattern: *.kth

[✓ Open matching files in PyCharm:]

- Text
- TypeScript
- TypeScript Config
- TypeScript JSX
- Web2Py Template
- XHTML
- XML
- XML Document Type Definition
- YAML
- Yarn.lock

Open matching files in associated application

3. From the **File pattern** list, select whether you want to specify a type for the current file (`file.extension`) or for all files with this extension (`*.extension`).

4. Select one of the following options:

   - **Open matching files as text and auto-detect file type by content**: open the file without an extension as a text file and identify its type by the content, for example, by the shebang line.

   - **Open matching files in PyCharm**: associate the file with one of the existing file types. You can change this association later in the settings.

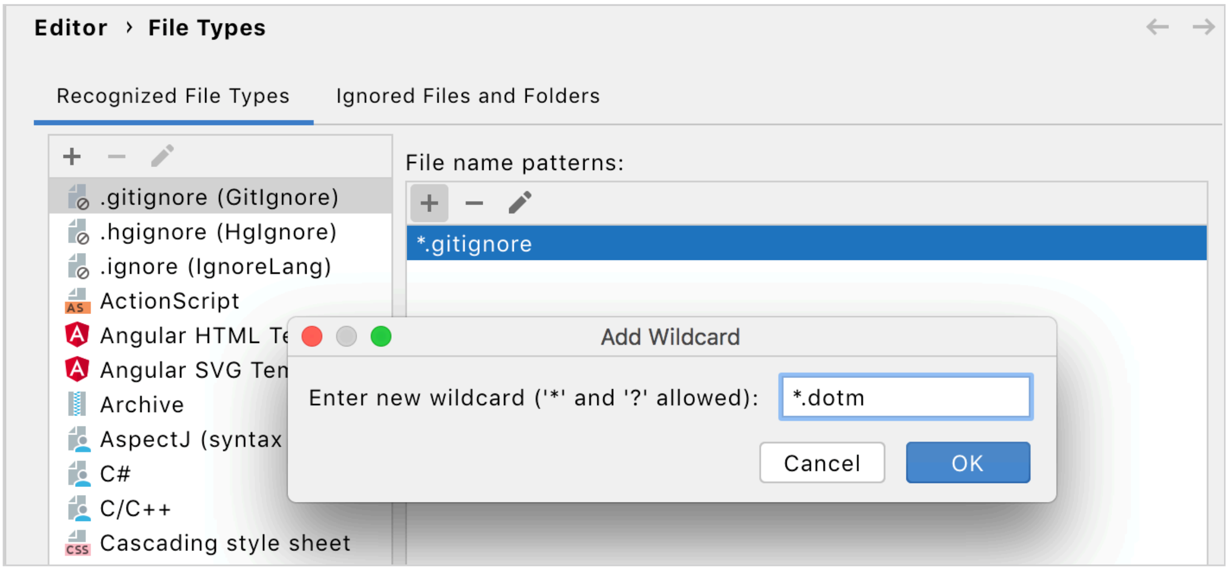
   - **Open matching files in associated application**: open the file in the default system application configured in your operating system. For example, `.pdf` files are opened in the default PDF viewer.

     If necessary, you can check and configure all filename patterns associated with system applications.

5. Click **OK** to apply the settings.

Change filename patterns associated with file type

1. Press ⌘Cmd , to open the IDE settings and then select **Editor | File Types**.
2. From the **Recognized File Types** list, select the file type that you want to associate with other filename patterns.
3. Use the **File name patterns** section to make the necessary changes. You can add a new pattern (+), remove an existing one (—), or modify an existing pattern (✎).



If your project contains files in proprietary formats, such as `.pdf` and `.docx`, PyCharm will open these files using the default application configured in your operating system. When a specific proprietary file format is not recognized, or you just want to open certain files with the system application, you can add the necessary associations.

Configure filename patterns associated with system applications

1. Press ⌘Cmd , to open the IDE settings and then select **Editor | File Types**.
2. In the **Recognized File Types** list, select **Files opened in associated applications**.
3. In the **File name patterns** section on the right, click + and specify a filename pattern that should be associated with an external application.

If a file is correctly associated with a specific file type by its filename pattern, but you want to process this file differently, you can override the file type association for this file only — other files matching that pattern will not be affected.

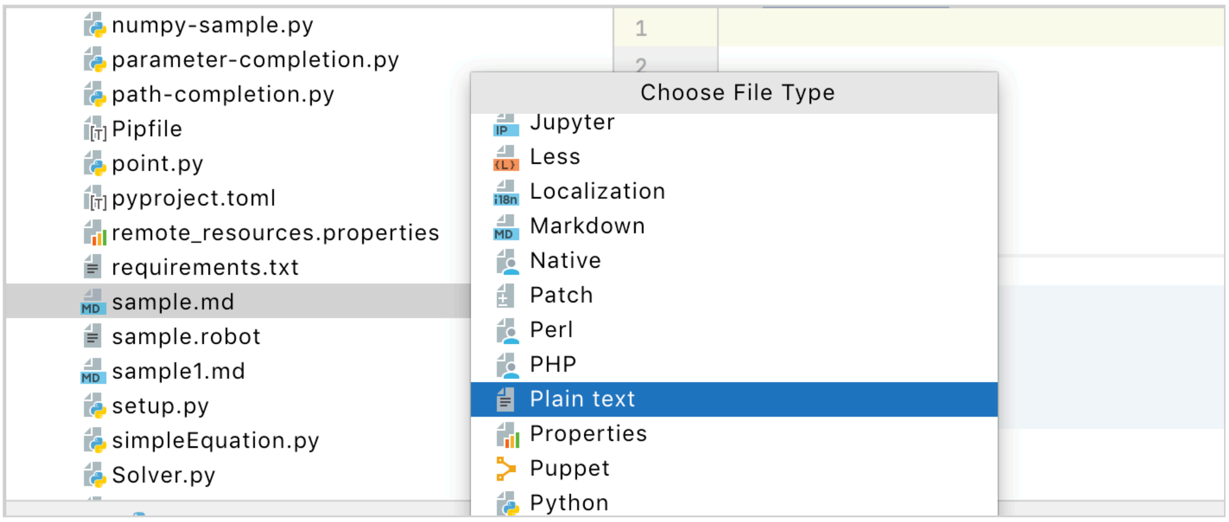
Override file type for specific file

1. In the **Project** tool window (⌘Cmd 1) , select one or more files that should have another file type association, right-click the selection and choose **Override File Type**.

2. From the list that opens, select a new file type.

   Use speed search to find the required file type faster.

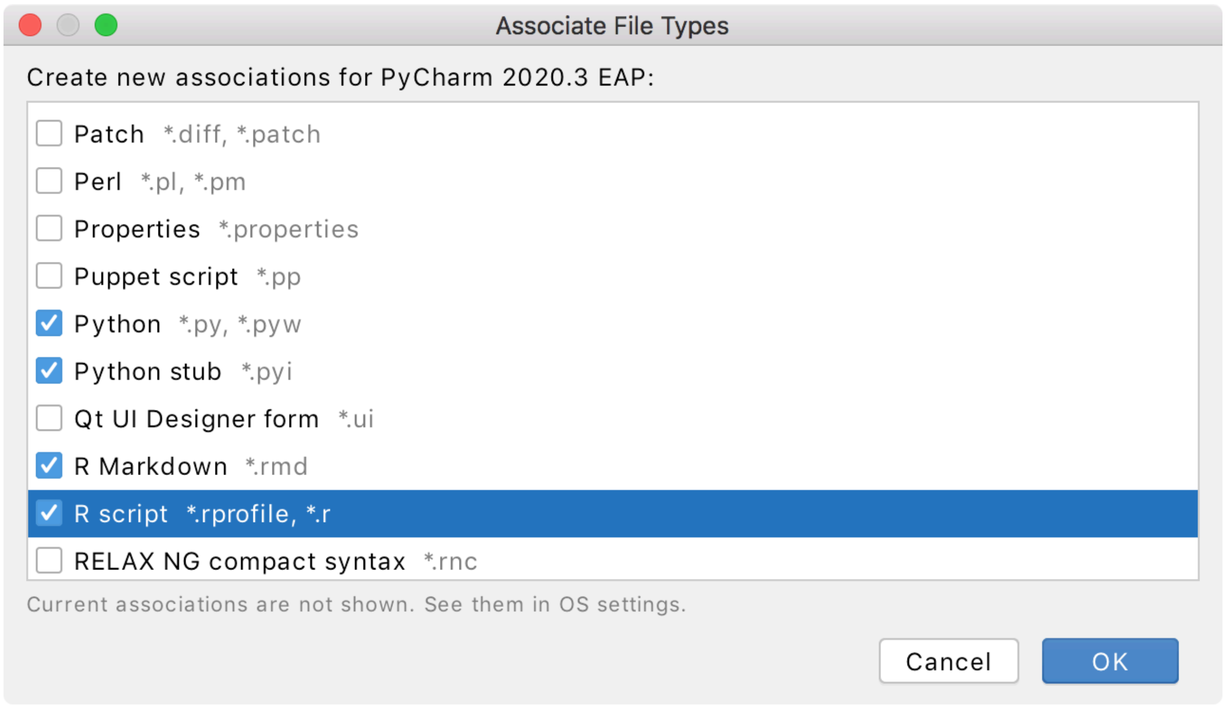
3. To restore the original file type association according to the filename pattern, right-click the file or files again and select **Revert File Type Override** from the context menu.



Make PyCharm the default app for specific file types

You can make PyCharm the default application for opening specific file types from the default file manager on your operating system.

1. Press ⌘Cmd , to open the IDE settings and then select **Editor | File Types**.
2. Click **Associate File Types with PyCharm** and select the file extensions you want to open with the IDE.



Create new associations for PyCharm 2020.3 EAP:

- [ ] Patch *.diff, *.patch
- [ ] Perl *.pl, *.pm
- [ ] Properties *.properties
- [ ] Puppet script *.pp
- [✓] Python *.py, *.pyw
- [✓] Python stub *.pyi
- [ ] Qt UI Designer form *.ui
- [✓] R Markdown *.rmd
- [✓] R script *.rprofile, *.r
- [ ] RELAX NG compact syntax *.rnc

Current associations are not shown. See them in OS settings.

3. Click **OK** and close the dialog.

If you're using macOS, restart your computer to apply the changes.

Ignore files and folders

PyCharm also maintains a list of files and folders that are completely excluded from any kind of processing. Out of the box, this list includes temporary files, service files related to version control systems, and so on:

```
*.pyc;*.pyo;*.rbc;*.yarb;*~;.DS_Store;.git;.hg;.svn;CVS;__pycache__;_
```

Modify the list of ignored files and folders

1. Press ⌘Cmd , to open the IDE settings and then select **Editor | File Types**.
2. Switch to the **Ignored Files and Folders** tab.

   You can add a new extension (+), remove an existing one (—), or modify an existing extension (✎).
3. Apply the changes and close the dialog.

Configure shebang commands for file types

PyCharm can recognize file types by the path specified on the shebang line. A **shebang** is a combination of characters in a script file followed by a path to the interpreter program that should execute this script. It starts with `#!` and it's always located on the first line of a script file.

1. Press ⌘Cmd , to open the IDE settings and then select **Editor | File Types**.
2. From the **Recognized File Types** list, select the file type for which you want to configure a command.
3. In the **HashBang patterns** area, click + (**Add HashBang Pattern**).
4. In the dialog that opens, specify the pattern that the IDE will use to recognize a file type, then click **OK**.
5. Apply the changes and close the dialog.