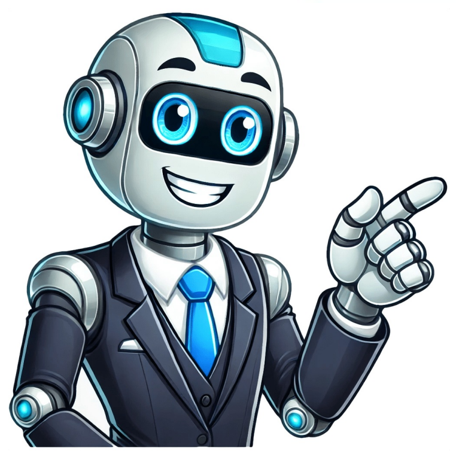

Releases PyCharm 2025.1 brings major updates to improve your development experience. PyCharm is now a unified product, combining PyCharm Professional and Community Edition. Version 2025.1 also brings a free AI tier, the public release of Junie, the launch of Cadence, significant Jupyter enhancements, support for Hatch, Data Wrangler, and many other improvements. Get the latest version from our download page or update through our free Toolbox App. Read this blog post to learn more about the updates. Prefer video? Get an overview of the major news and improvements in this video: PyCharm is now one powerful, unified product! Its core functionality, including Jupyter Notebook support, will be free, and a Pro subscription with additional features will be available. Starting with the 2025.1 release, every user will get instant access to a free one-month Pro trial, so you'll be able to access all of PyCharm's advanced features right away. After the trial, you can choose whether to continue with a Pro subscription or keep using the core features for free. Learn more about the change in this blog post. Junie, the coding agent by JetBrains, is now available in PyCharm via JetBrains AI. Junie autonomously plans, writes, refines, and tests code to make your development experience smooth, efficient, and enjoyable. It handles tedious tasks like restructuring code, creating tests, and implementing refinements, so you can focus on bigger challenges and innovation. JetBrains AI has received a major upgrade, bringing both AI Assistant and the coding agent Junie under a single subscription. With this release, all JetBrains AI features are accessible for free in PyCharm Pro, with unlimited use for some, such as code completion and local model support, and limited credit-based access to others. We're also introducing a new subscription system that makes it easy to scale up as needed with the AI Pro and AI Ultimate tiers. Other highlights of this release include smarter completion, advanced context awareness, and support for Claude 3.7 Sonnet and Gemini 2.0 Flash. Head to the What's New page to learn more about the latest AI features. We're introducing Cadence. You can now run your machine learning code on powerful cloud hardware directly from PyCharm in minutes – no complex setup or cloud expertise is required. The Cadence plugin simplifies ML workflows, allowing you to focus on your code while leveraging scalable computing resources. We've implemented Data Wrangler, a powerful tool to help Python data professionals streamline data manipulation and focus on higher-level analysis. Use the interactive UI to perform common dataframe transformations – like filtering, cleaning, handling outliers, and more – without writing repetitive code. You can view and explore column statistics, generate Python code for transformations automatically, track the history of changes, export data easily, and insert transformations as new cells in your notebook. PyCharm 2025.1 introduces SQL cells. This new cell type allows you to query databases, dataframes, and attached CSV files in Jupyter notebooks and automatically save query results to pandas DataFrames. We've also introduced many other improvements to enhance the Jupyter notebook experience. Learn more about them in the What's New. We're introducing support for Hatch, a modern and extensible Python project manager from the Python Packaging Authority (PyPA). Hatch can automatically migrate setuptools configurations, create isolated environments, and run and publish builds, making Python package management more efficient. PyCharm also allows you to create new projects managed by Hatch. The IDE will automatically recognize Hatch projects when they are imported from a local machine or a remote source. Looking for more? Visit our What's New page to learn about other 2025.1 features and the latest bug fixes. Read the release notes for the full breakdown of the changes. If you encounter any problems, please report them via our issue tracker so we can address them promptly. We'd love to hear your feedback on PyCharm 2025.1 – leave your comments below or connect with us on X. PyCharm now allows you to export Kafka records directly to JSON, CSV, and TSV files, making it easier to analyze and share streaming data. The IDE can save complete record details, including topic, timestamp, key, value, partition, and offset, in widely used formats. With this update, you can diagnose issues faster, verify data integrity, enable advanced analysis, and streamline collaboration, as exporting Kafka records helps ensure data is accessible and actionable. By reducing time to resolution and improving data transparency, this enhancement provides a more efficient and effective Kafka experience. PyCharm is a cross-platform IDE that provides a consistent experience on Windows, macOS, and Linux.PyCharm is now available as a unified single product. What changes have been made: The Community Edition is now part of the unified PyCharm product: PyCharm's core features remain free and open-source.Jupyter support is now included in the core functionality. You can upgrade to PyCharm Community 2025.1 as usual, with no immediate changes required on your part.When you install PyCharm, you will automatically receive a 30-day free Pro trial, granting you full access to PyCharm's advanced features.After the trial ends, you will have the following options:Get a Pro subscription to continue using advanced functionality.Keep using PyCharm's core features for free.If you are already using PyCharm Professional, you will retain full access to all Pro features with your existing license—no changes are required.For more information, refer to Unified PyCharm overview.RequirementMinimumRecommendedRAM2 GB of free RAM8 GB of total system RAMCPUAny modern CPUMulti-core CPU. PyCharm supports multithreading for different operations and processes making it faster the more CPU cores it can use.Disk space3.5 GBSSD drive with at least 5 GB of free spaceMonitor resolution1024×7681920×1080Operating systemOfficially released versions of the following:Microsoft Windows 10 1809 64-bit or laterWindows Server 2019 64-bit or latermacOS 12.0 or laterTwo latest versions of Ubuntu LTS or Fedora Linux distributions that meet the following requirements:Linux kernel version 6.xGnome or KDE desktop environmentX Window System (X11)Wayland support is in development. You can monitor the progress and leave your feedback in JBR-3206: Native Wayland support.GLIBC 2.28 or laterPre-release versions are not supported.The latest versions of the following:Windows 64-bitmacOSUbuntu LTS or Fedora Linux distributionYou do not need to install Java to run PyCharm because JetBrains Runtime is bundled with the IDE (based on JBR 21). PyCharm supports the following versions of Python:Python 2: version 2.7Python 3: from the version 3.8 up to the version 3.14The JetBrains Toolbox App is the recommended tool to install JetBrains products. Use it to install and manage different products or several versions of the same product, including Early Access Program (EAP) releases, update and roll back when necessary, and easily remove any tool. The Toolbox App maintains a list of all your projects to quickly open any project in the right IDE and version.Download the installer .exe from the Toolbox App web page.Run the installer and follow the wizard steps.After you run the Toolbox App, click its icon in the notification area and select which product you want to install.To install a specific version, click and select Available versions.Log in to your JetBrains Account from the Toolbox App, and it will automatically activate the available licenses for any IDE that you install.Download the disk image .dmg from the Toolbox App web page.There are separate disk images for Intel and Apple Silicon processors.Mount the image and drag the JetBrains Toolbox app to the Applications folder.After you run the Toolbox App, click its icon in the main menu and select which product you want to install.To install a specific version, click and select Available versions.Log in to your JetBrains Account from the Toolbox App, and it will automatically activate the available licenses for any IDE that you install.The Toolbox App is packaged in AppImage and requires FUSE to run.If your distribution doesn't use the libfuse2 package, install it by running the following command: sudo apt install libfuse2 Additionally, make sure that the following required packages are installed:libxi6libxrender1libxtst6mesa-utilslibfontconfiglibgtk-3-binDownload the tarball .tar.gz from the Toolbox App web page.Extract the tarball and launch the executable: tar -xzf jetbrains-toolbox-.tar.gz && cd jetbrains-toolbox- && ./jetbrains-toolbox The Toolbox App will install itself into $HOME/.local/share/JetBrains/Toolbox/bin and run from there.After you run the Toolbox App for the first time, it will automatically add the Toolbox App icon to the main menu.Select the product that you want to install.To install a specific version, click and select Available versions.Log in to your JetBrains Account from the Toolbox App, and it will automatically activate the available licenses for any IDE that you install.If you installed PyCharm via the Toolbox App, you can find the installation directory in the app: open the settings of the IDE instance in the Toolbox App, expand Configuration, and look for the Install location field.Install PyCharm manually to manage the location of every instance and all the configuration files. For example, if you have a policy that requires specific install locations. Download the installer .exe.There is a separate installer for ARM64 processors.To verify the integrity of the installer, use the SHA checksum linked from the Download page.Run the installer and follow the wizard steps.Mind the following options in the installation wizard64-bit launcher: Adds a launching icon to the Desktop.Open Folder as Project: Adds an option to the folder context menu that will allow opening the selected directory as a PyCharm project..py: Establishes an association with Python files to open them in PyCharm.Add launchers dir to the PATH: Allows running this PyCharm instance from the Console without specifying the path to it.To run PyCharm, find it in the Windows Start menu or use the desktop shortcut. You can also run the launcher batch script or executable in the installation directory under bin. Download the disk image.There are separate disk images for Intel and Apple Silicon processors.To verify the integrity of the downloaded disk image, use the SHA checksum linked from the Download page.Mount the image and drag the PyCharm app to the Applications folder.Run the PyCharm app from the Applications directory, Launchpad, or Spotlight.Download the tarballThere is a separate tarball for ARM64 processors.Unpack the pycharm-*.tar.gz file to a different folder, if your current Download folder doesn't support file execution: tar xzf pycharm-*.tar.gz -C The recommended installation location according to the filesystem hierarchy standard (FHS) is /opt. To install PyCharm into this directory, enter the following command: sudo tar xzf pycharm-*.tar.gz -C /opt/ To verify integrity of the downloaded archive, use the SHA checksum linked from the Download page.Switch to the bin subdirectory: cd /pycharm-*/bin For example, cd /opt/pycharm-2022.2.4/bin Run pycharm.sh from the bin subdirectory.In the main menu, go to .The following steps are for Ubuntu 23.10. If you are using a different Linux distribution or version of Ubuntu, refer to its documentation.Click Activities in the top-left corner of the screen.Click the grid button in the dash to show all applications.To pin the app to the dash, right-click the PyCharm icon and select Add to Favorites.When you run PyCharm for the first time, you can take several steps to complete the installation, customize your instance, and start working with the IDE.For more information, refer to Run PyCharm for the first time.For more information about the location of the default IDE directories with user-specific files, refer to Directories used by the IDE.Silent installation is performed without any user interface. It can be used by network administrators to install PyCharm on a number of machines and avoid interrupting other users.To perform silent install, run the installer with the following switches:/S: Enable silent install/CONFIG: Specify the path to the silent configuration file/D: Specify the path to the installation directoryThis parameter must be the last in the command line, and it should not contain any quotes even if the path contains blank spaces./NCRC: Disable the CRC check if you want to suppress the Verifying Installer popup.For example: pycharm.exe /S /CONFIG=d:\temp\silent.config /D=d:\IDE\PyCharm To check for issues during the installation process, add the /LOG switch with the log file path and name between the /S and /D parameters. The installer will generate the specified log file. For example:pycharm.exe /S /CONFIG=d:\temp\silent.config /LOG=d:\JetBrains\PyCharm\install.log /D=d:\IDE\PyCharmYou can download the silent configuration files for PyCharm at silent configuration file defines the options for installing PyCharm. With the default options, silent installation is performed only for the current user: mode=user. If you want to install PyCharm for all users, change the value of the installation mode option to mode=admin and run the installer as an administrator.The default silent configuration file is unique for each JetBrains product. You can modify it to enable or disable various installation options as necessary.It is possible to perform silent installation without the configuration file. In this case, omit the /CONFIG switch and run the installer as an administrator. Without the silent configuration file, the installer will ignore all additional options: it will not create desktop shortcuts, add associations, or update the PATH variable. However, it will still create a shortcut in the Start menu under JetBrains.See more details on installing PyCharm in the video tutorial:PyCharm installed via snap may encounter issues including performance degradation, problems with JavaScript debugging using Chromium (WEB-38511), difficulties importing SBT projects (SCL-17169), and delays in file management operations (IJPL-1234).If you experience these issues, we recommend installing PyCharm via the Toolbox App for a potentially smoother experience.You can install PyCharm as a self-contained snap package. Since snaps update automatically, your PyCharm installation will always be up-to-date.To use snaps, install and run the snapd service as described in the snapd installation guide.On Ubuntu 16.04 LTS and later, this service is pre-installed.PyCharm is distributed via two channels:The stable channel includes only stable versions. To install the latest stable release of PyCharm, run the following command: sudo snap install pycharm-professional --classic sudo snap install pycharm-community --classic The --classic option is required because the PyCharm snap requires full access to the system, like a traditionally packaged application.The edge channel includes EAP builds. To install the latest EAP build of PyCharm, run the following command: sudo snap install pycharm-professional --classic --edge sudo snap install pycharm-community --classic --edge When the snap is installed, you can launch it by running the pycharm-professional or pycharm-community command.To list all installed snaps, run sudo snap list.To check the list of available versions, run the following command: snap info To install the version you want, run the following command: sudo snap install --channel=/stable --classic For more information about other snap commands, refer to the Snapcraft documentation.Last modified: 08 May 2025 News Releases Estimated reading time: 3 minutes PyCharm is now one powerful, unified product! Its core functionality, including Jupyter Notebook support, will be free, and a Pro subscription will be available with additional features. Starting with the 2025.1 release, every user will get instant access to a free one-month Pro trial, so you'll be able to access all of PyCharm's advanced features right away. After the trial, you can choose whether to continue with a Pro subscription or keep using the core features for free. Previously, PyCharm was offered as two separate products: the free Community Edition and the Professional Edition with extended capabilities. Now, with a single streamlined product, you no longer need to choose. Everything is in one place, and you can seamlessly switch between core and advanced features within the same installation whenever you need to. You no longer need to worry about additional downloads or switching between editions. PyCharm is now a single product. Start with a month of full Pro access for free, and then keep using the core features at no cost. Upgrade to Pro anytime within the same installation. PyCharm now offers free Jupyter support, including running, debugging, output rendering, and intelligent code assistance in notebooks. It's perfect for data workflows, no Pro subscription required. However, a Pro subscription does offer more advanced capabilities, including remote notebooks, dynamic tables, SQL cells, and others. With every new major PyCharm release (currently three times a year), you will get instant access to a free one-month Pro trial. Once it ends, you can continue using the core features for free. Focusing on a single PyCharm product will help us improve overall quality, streamline updates, and deliver new features faster. First of all, thank you for being part of our amazing community! Your feedback, passion, and contributions have helped shape PyCharm into the tool it is today. Nothing is going to change for you right away – you can upgrade to PyCharm Community 2025.1 as usual. Alternatively, you may choose to manually switch to the new PyCharm immediately and keep using everything you have now for free, plus the support for Jupyter notebooks. Starting with PyCharm 2025.2, we'll offer a smooth migration path that preserves your current setup and preferences. PyCharm Community 2025.2 will be the final standalone version, and, from 2025.3 onward, all Community Edition users will transition to the unified PyCharm experience. Rest assured – our commitment to open-source development remains as strong as ever. The Community Edition codebase will stay public on GitHub, and we'll continue to maintain and update it. We'll also provide an easy way to build PyCharm from source via GitHub Actions. Have more questions about what's next? Read our extended FAQ for more details. Nothing changes! Your license will automatically work with the new single PyCharm product. Simply upgrade to PyCharm 2025.1 and continue enjoying everything Pro has to offer. You can start right away with the new single PyCharm product. You'll get a free one-month Pro trial with full functionality. After that, you can purchase a Pro subscription and keep using PyCharm with its full capabilities, or you can continue using just the core features – including Jupyter Notebook support – for free. Download PyCharm now. PyCharm is a dedicated Python Integrated Development Environment (IDE) providing a wide range of essential tools for Python developers, tightly integrated to create a convenient environment for productive Python, web, and data science development.Previously, PyCharm was available in two editions: Community and Professional.Starting with PyCharm 2025.1, PyCharm Community and Professional are combined into a single, unified product: PyCharm. With this change, all users will have access to essential features without the need to switch between editions.PyCharm's core functionality, including Jupyter Notebook support, will be free, and a Pro subscription will be available with additional features.When you install unified PyCharm, you will automatically receive a 30-day free Pro trial, granting you full access to PyCharm's advanced features.After the trial, you may either:Get a Pro subscription to continue using advanced functionality.Keep using PyCharm's core features for free.If you are already using PyCharm Professional, you will retain full access to all Pro features with your existing license—no changes are required.For more information, refer to Unified PyCharm overview.To start developing in Python with PyCharm, you need to download and install Python from python.org depending on your platform.PyCharm supports the following versions of Python:Python 2: version 2.7Python 3: from the version 3.8 up to the version 3.14Besides, in the Pro mode, one can develop Django , Flask, and Pyramid applications. Also, it fully supports HTML (including HTML5), CSS, JavaScript, and XML: these languages are bundled in the IDE via plugins and are switched on for you by default. Support for the other languages and frameworks can also be added via plugins (go to or for macOS users, to find out more or set them up during the first IDE launch).PyCharm is a cross-platform IDE that works on Windows, macOS, and Linux. Check the system requirements:RequirementMinimumRecommendedRAM2 GB of free RAM8 GB of total system RAMCPUAny modern CPUMulti-core CPU. PyCharm supports multithreading for different operations and processes making it faster the more CPU cores it can use.Disk space3.5 GBSSD drive with at least 5 GB of free spaceMonitor resolution1024×7681920×1080Operating systemOfficially released versions of the following:Microsoft Windows 10 1809 64-bit or laterWindows Server 2019 64-bit or latermacOS 12.0 or laterTwo latest versions of Ubuntu LTS or Fedora Linux distributions that meet the following requirements:Linux kernel version 6.xGnome or KDE desktop environmentX Window System (X11)Wayland support is in development. You can monitor the progress and leave your feedback in JBR-3206: Native Wayland support.GLIBC 2.28 or laterPre-release versions are not supported.The latest versions of the following:Windows 64-bitmacOSUbuntu LTS or Fedora Linux distributionYou can install PyCharm using Toolbox or standalone installations. If you need assistance installing PyCharm, see the installation instructions: Install PyCharmEverything you do in PyCharm, you do within the context of a project. It serves as a basis for coding assistance, bulk refactoring, coding style consistency, and so on. You have several options to start working on a project inside the IDE:You can also create a Python file or Jupyter notebook without setting up a project:On the Welcome screen, click New Notebook if you want to create a Jupyter notebook or New Script if you want to create a Python file.The new file will be added to the automatically created PyCharmMiscProject, and you will be able to write and execute code without additional steps.Begin by opening one of your existing projects stored on your computer. You can select one in the list of the recent projects on the Welcome screen or click Open:Otherwise, you can create a project for your existing source files. Select the command Open on the menu, and specify the directory where the sources exist. PyCharm will then create a project from your sources for you. For more information, refer to Create a project from existing sources. You can also download sources from a VCS storage or repository. On the Welcome screen, click Get from VCS, and then choose Git (GitHub), Mercurial, Subversion, or Perforce (supported in PyCharm Pro only) .Then, enter a path to the sources and clone the repository to the local host:For more information, refer to Version control.To create a project, do one of the following:Go to On the Welcome screen, click New ProjectIn PyCharm, you can create only Python projects, whereas, with PyCharm Pro, you have a variety of options to create a web framework project.See more details in Create a Python project.When creating a new project, you need to specify a Python interpreter to execute Python code in your project. You need at least one Python installation to be available on your machine.For a new project, PyCharm creates an isolated virtual environment: venv, pipenv, poetry, or Conda. As you work, you can change it or create new interpreters. You can also quickly preview packages installed for your interpreters and add new packages in the Python Package tool window.For more information, refer to Configure a Python interpreter.When you launch PyCharm for the first time, or when there are no open projects, you see the Welcome screen. It gives you the main entry points into the IDE: creating or opening a project, checking out a project from version control, viewing documentation, and configuring the IDE.When a project is opened, you see the main window divided into several logical areas. Let's take a moment to see the key UI elements here:Window header contains a set of widgets which provide quick access to the most popular actions: project widget, VCS widget, and run widget. It also allows you to open Code With Me, Search Everywhere, and Settings.Project tool window on the left side displays your project files.Editor on the right side, where you actually write your code. It has tabs for easy navigation between open files.Context menus open when you right-click an element of the interface or a code fragment and show the actions available.Navigation bar allows you to quickly navigate the project folders and files.Gutter, the vertical stripe next to the editor, shows the breakpoints you have, and provides a convenient way to navigate through the code hierarchy like going to definition/declaration. It also shows line numbers and per-line VCS history.Scrollbar, on the right side of the editor. PyCharm constantly monitors the quality of your code by running code inspections. The indicator in the top right-hand corner shows the overall status of code inspections for the entire file.Tool windows are specialized windows attached to the bottom and the sides of the workspace. They provide access to typical tasks such as project management, source code search and navigation, integration with version control systems, running, testing, debugging, and so on.The status bar indicates the status of your project and the entire IDE and shows various warnings and information messages like file encoding, line separator, inspection profile, and so on. It also provides quick access to the Python interpreter settings.Project tool window on the left side displays your project files.Editor on the right side, where you actually write your code. It has tabs for easy navigation between open files.Navigation bar allows you to quickly navigate the project folders and files.Gutter, the vertical stripe next to the editor, shows the breakpoints you have, and provides a convenient way to navigate through the code hierarchy like going to definition/declaration. It also shows line numbers and per-line VCS history.Scrollbar, on the right side of the editor. PyCharm constantly monitors the quality of your code by running code inspections. The indicator in the top right-hand corner shows the overall status of code inspections for the entire file.Tool windows are specialized windows attached to the bottom and the sides of the workspace. They provide access to typical tasks such as project management, source code search and navigation, integration with version control systems, running, testing, debugging, and so on.The status bar indicates the status of your project and the entire IDE, and shows various warnings and information messages like file encoding, line separator, inspection profile, and so on. It also provides quick access to the Python interpreter settings.Also, in the bottom-left corner of the PyCharm window, in the Status bar, you see the button or . This button toggles the showing of the tool window bars. If you hover over this button, the list of the currently available tool windows shows up.When you have created a new project or opened an existing one, it is time to start coding.In the Project tool window, select the project root (typically, it is the root node in the project tree), right-click it, and select .Select the option Python File from the context menu, and then type the new filename.PyCharm creates a new Python file and opens it for editing.PyCharm takes care of the routine so that you can focus on the important. Use the following coding capabilities to create error-free applications without wasting precious time.Code completion is a great time-saver, regardless of the type of file you're working with.Basic completion works as you type and completes any name instantly.Smart type-matching completion analyzes the context you're currently working in and offers more accurate suggestions based on that analysis.PyCharm keeps an eye on what you are currently doing and makes smart suggestions, called intention actions, to save more of your time. Indicated with a lightbulb, intention actions let you apply automatic changes to code that is correct (in contrast to code inspections that provide quick-fixes for code that may be incorrect). Did you forget to add some parameters and field initializers to the constructor? Not a problem with PyCharm. Click the lightbulb (or press Alt+Enter) and select one of the suggested options:The full list of available intention actions can be found in or for macOS users.PyCharm monitors your code and tries to keep it accurate and clean. It detects potential errors and problems and suggests quick-fixes for them.Every time the IDE finds unused code, an endless loop, and many other things that likely require your attention, you'll see a lightbulb. Click it, or press Alt+Enter, to apply a fix.The complete list of available inspections can be found under (or for macOS users). Disable some of them, or enable others, plus adjust the severity of each inspection. You decide whether it should be considered an error or just a warning.Writing code can be a lot easier and quicker when you use the code generation options available in PyCharm. The menu Alt+Insert will help you with creating symbols from usage, as well as suggest overriding or implementing some functions:Use live templates (choose or press Ctrl+J) to produce the entire code constructs. You can explore the available ready-to-use live templates In the Settings dialog (Ctrl+Alt+S) (Settings | Editor | Live templates or if you are a macOS user).If you see that you are lacking something especially important for your development, extend this set of templates with your own. Also, consider quickly surrounding your code with complete constructs (choose or press Ctrl+Alt+T. For example, with an if statement:When your project is big, or when you have to work with someone else's code, it's vital to be able to quickly find what you are looking for and dig into the code. This is why PyCharm comes with a set of navigation and search features that help you find your way through any code no matter how tangled it is.With these search facilities, you can find and replace any fragment of code both in the currently opened file Ctrl+F, or in an entire project Ctrl+Shift+F.To find where a particular symbol is used, PyCharm suggests full-scale search via Find Usages Alt+F7:You can tell a lot just looking at your File Structure, with its imports or call hierarchies:Also, you can navigate to:The icons in the left-hand gutter can also help you with navigation:Remembering all your activity in the project, PyCharm can easily navigate you to the Recent Files Ctrl+E or Recently Changed Files Alt+Shift+C.To go through the history of changes, use the following actions:Back Ctrl+Alt+LeftForward Ctrl+Alt+RightLast Edit Location Ctrl+Shift+BackspaceTake advantage of many smart actions possible with PyCharm. For example, use the Find Action search Ctrl+Shift+A: just type a part of the action name, and the IDE will show you the list of all available options. Then, select the action you need:If you have a general idea of what you're looking for, you can always locate the corresponding element using one of the existing navigation features. But what if you want to look for something in every nook and cranny? The answer is to use Search Everywhere!To try it, click the magnifying glass button in the upper right-hand corner of the window, or invoke it with Double Shift (press Shift twice).Now when you've played with the code and discovered what you can do with it, it's time to run, debug and test your app.The easiest way to run an application is to right-click in the editor, and then choose from the context menu:If your Python script contains the __main__ clause, then you can click the button in the gutter, and then choose the desired command.You can see the your script execution in the Run tool window.When you perform run, debug, or test operations with PyCharm, you always start a process based on one of the existing run/debug configurations, using its parameters.When you run your application for the very first time, PyCharm automatically creates the temporary Run/Debug configuration. You can modify it to specify or alter the default parameters and save it as a permanent Run/Debug configuration.See how to tune run/debug configurations in Run/debug configurations.Does your application stumble on a runtime error? To find out what's causing it, you will have to do some debugging. PyCharm supports the debugger on all platforms.Debugging starts with placing breakpoints at which program execution will be suspended, so you can explore program data. Just click the gutter of the line where you want the breakpoint to appear.To start debugging your application, press Shift+F9. Then go through the program execution step by step (see the available options in the menu or the Debug tool window), evaluate any arbitrary expression, add watches, and manually set values for the variables.For more information, refer to Debugging.It is a good idea to test your applications, and PyCharm helps doing it as simple as possible.With PyCharm, you can:For more information about the numbers, refer to the Test Runner tab section.PyCharm supports all the major Python testing frameworks:UnittestDoctestNosetestpytestFor each of these frameworks, PyCharm provides its own run/debug configuration.For more information, refer to Test your first Python application and Run tests.With PyCharm Pro you can run, debug, and test your Python code remotely. You can deploy your local applications to some remote server. For more information about deployment servers, refer to the section Configuring Synchronization with a Web Server. PyCharm Pro also helps compare local and remote folders, and synchronize the local copy with that deployed on the server.If you are keeping your source code under version control, you will be glad to know that PyCharm integrates with many popular version control systems: Git (or GitHub), Mercurial, Perforce (supported in Pro mode only), Subversion. To specify credentials and any settings specific to a particular VCS, go to (or if you are a macOS user).The VCS menu gives you a clue about what commands are available. For example, you can see the changes you've made, commit them, create changelists and much more from the Local Changes view: (or just press Alt+9). Also, you can perform some basic VCS actions in the main window: From the VCS widget:From the Navigation bar above the editor:For more information, refer to Version control.In addition to traditional version control, you can use the local history. With Local History, PyCharm automatically tracks changes you make to the source code, the results of refactoring, and so onLocal history is always enabled. To view it for a file or a folder, bring up Local History by selecting . Here you can review the changes, revert them, or create a patch.PyCharm has an interactive Python console to perform smart operations over data with on-the-fly syntax check with inspections, braces and quotes matching, and of course, code completion. You can also benefit from the built-in support for Anaconda.With PyCharm Pro, you can analyze and visualize various scientific and statistical data. Jupyter Notebook integration enables editing, executing, and debugging notebook source code and examining execution outputs, including stream data, images, and other media.With the R plugin installed in PyCharm, you can perform various statistical computing using R language and use coding assistance, visual debugging, smart running and preview tools, and other popular IDE features.As you might have noticed already, creating projects of the various types (Django, for example) requires a data source. It is also quite possible that you inject SQL statements into your source code.PyCharm Pro does not enable you to create databases, but provides facilities to manage and query them. Once you are granted access to a certain database, you can configure one or more data sources within PyCharm that reflect the structure of the database and store the database access credentials. Based on this information, PyCharm establishes a connection to the database and provides the ability to retrieve or change information contained therein.Access to the databases is provided by the Database window (). This tool window allows you to work with the databases. It lets you view and modify data structures in your databases, and perform other associated tasks.For more information, refer to Database Tools and SQL.Feel free to tweak the IDE so it suits your needs perfectly and is as helpful and comfortable as it can be. Go to Settings to see the list of available customization options.The first thing to fine-tune is the general "look and feel." Go to File | Settings | Appearance and Behavior | Appearance (for macOS users) to select the IDE theme: the light themes or Darcula if you prefer a darker setting.The many pages available under File | Settings | Editor (for macOS users) help you adjust every aspect of the editor's behavior. A lot of options are available here, from general settings (like Drag'n'Drop enabling, scrolling configuration, and so on.), to color configuration for each available language and use case, to tabs and code folding settings, to code completion behavior and even postfix templates.For more information, refer to Configuring PyCharm settings.Code style can be defined for each language under File | Settings | Editor | Code Style (for macOS users). You can also create and save your own coding style scheme.For more information, refer to Configuring code style and Code Quality Assistance Tips and Tricks, or How to Make Your Code Look Pretty?.PyCharm uses the keyboard-centric approach, meaning that nearly all actions possible in the IDE are mapped to keyboard shortcuts.The set of keyboard shortcuts you work with is one of your most intimate habits — your fingers "remember" certain combinations of keys, and changing this habit is easier said than done. PyCharm supplies you with a default keymap (choose from the main menu) making your coding really productive and convenient. However, you can always change it going to File | Settings | Keymap (for macOS users).There are also some pre-defined keymaps (like Emacs, Visual Studio, Eclipse, NetBeans and so on), and you can also create your own keymap based on an existing one.If you feel most productive with vi/Vim, an emulation mode will give you the best of both worlds. Enable the IdeaVim plugin in the IDE and select the vim keymap.For more information, refer to Configure keyboard shortcuts.We hope this brief overview of essential PyCharm features will give you a quick start. There are many important features that make a developer's life easier and more fun, and the source code neater and cleaner. Take these first few steps now, and then dig deeper when you feel the time is right:Enjoy PyCharm! With any questions visit our Discussion Forum, twitter and blog, where you can find news, updates, and useful tips and tricks. Also, don't hesitate to report any problems to our support team or the PyCharm issue tracker.Last modified: 11 April 2025

- pasenana
- noxo
- gibeginake
- backyard baseball unblocked games wtf
- https://giorgieristudio.it/uploads/kcFinder/files/39727673042.pdf
- turogifi
- nayaco
- yedifefumu
- vuleveze
- https://itchanoi.vn/uploads/news_file/4d4571fe-6510-4337-9496-937ec4c06c54.pdf
- taxu