ОБЩИЕ ХАРАКТЕРИСТИКИ IDE

1. *Определение*

PyCharm – это интегрированная среда разработки (IDE) для программирования на языке Python, разработанная чешской компанией JetBrains. У PyCharm есть две версии: PyCharm Community Edition – бесплатная версия, находится под лицензией Apache License, и PyCharm Professional Edition – профессиональная версия продукта, обладающая дополнительной функциональностью, является проприетарным ПО.

2. *Основные возможности:*

2.1 Создание проектов;

2.2 Написание кода на Python;

2.3 Запуск кода;

2.4 Отладка кода;

2.5 Тестирование кода;

2.6 Корректировка кода.

3. *Интерфейс*

При запуске PyCharm в первый раз или когда отсутствуют открытые проекты, отображается Экран приветствия (рисунок 1.1). Он предлагает основные точки входа в среду IDE: создание или открытие проекта и проверка проекта с помощью контроля версий.

При начале работы с проектом открывается главное окно, разделенное на несколько логических областей (рисунок 1.2). Ключевые элементы пользовательского интерфейса:

3.1 Заголовок окна содержит набор виджетов, которые обеспечивают быстрый доступ к наиболее популярным действиям: project widget, VCS

widget и run widget. Другие функции этой части экрана позволяют поделиться проектом (создать ссылку), открыть поиск по IDE и настроить IDE или проект.

3.2 Окно *Project Tool Window* расположено слева. В нём отображаются файлы открытого проекта.

3.3 Основной редактор (*PyCharm Editor*) расположен справа, там фактически пишется код. В верхней его части расположены вкладки открытых файлов для удобства навигации.

3.4 Контекстные меню открываются при щелчке правой кнопкой мыши по элементу интерфейса или фрагменту кода и отображают доступные действия.

3.5 *Gutter* - вертикальная полоса рядом с редактором, которая показывает имеющиеся строки кода и предоставляет удобный способ навигации по иерархии кода. Здесь также отображаются номера строк.

3.6 *Pycharm Tool Windows* - это специализированные окна, прикрепленные к нижней части и боковым сторонам рабочей области. Они предоставляют доступ к типичным задачам, таким как управление проектами, поиск исходного кода и навигация, интеграция с системами контроля версий, запуск, тестирование, отладка и так далее.

3.7 Панель навигации, которая позволяет быстро перемещаться по папкам и файлам проекта.

3.8 Строка состояния (*Status Bar*), которая показывает состояние проекта и всей среды IDE, а также отображает различные предупреждения и информационные сообщения. Оно также обеспечивает быстрый доступ к настройкам интерпретатора Python.

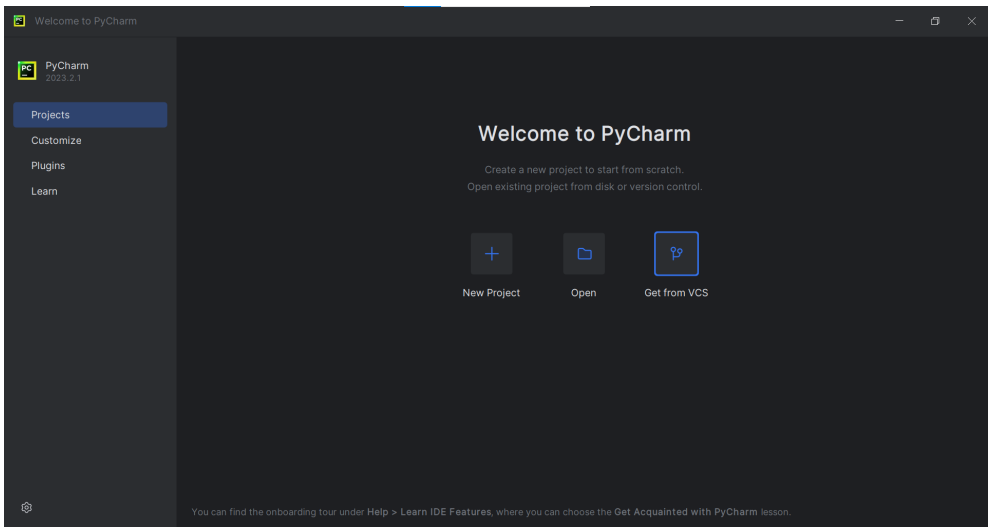


Рисунок 1.1

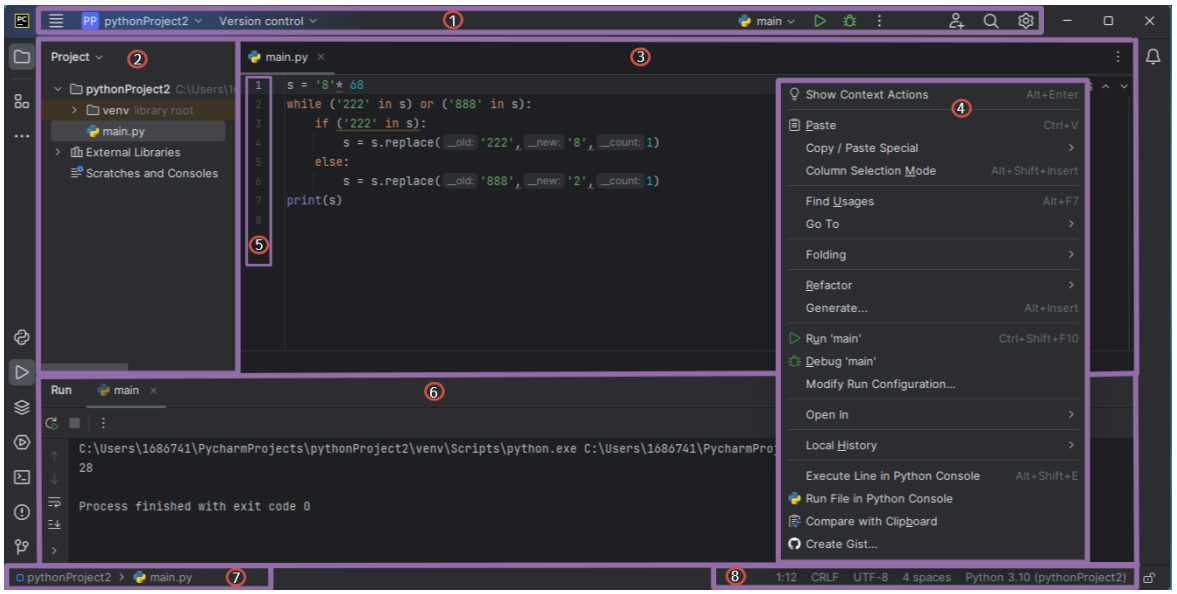


Рисунок 1.2

СИСТЕМНЫЕ ТРЕБОВАНИЯ IDE

Операционная система:

- Windows 8, 10 или 11 (64-битные версии);
- macOS 10.14 или более поздняя версия;
- Linux со средой GNOME, KDE или Unity DE; если дистрибутив не включает Glibc 2.14 или более позднюю

версию (например, RHEL 6 или CentOS 6), то PyCharm может быть для него недоступен.

Процессор: никаких специальных требований нет, но рекомендуется, чтобы он был многоядерным, так как PyCharm поддерживает многопоточность. Это позволит выполнять операции быстрее.

Оперативная память: 4 ГБ свободной RAM. Рекомендуется, чтобы общая оперативная память у устройства была 8 ГБ.

Место на диске: 2,5 ГБ и ещё 1 ГБ для кэша. Рекомендуется использовать SSD и иметь хотя бы 5 ГБ свободного пространства.

Разрешение экрана: не менее 1024 × 768 пикселей, рекомендуется 1920 × 1080.

ОСНОВНОЙ ФУНКЦИОНАЛ IDE

1) Создание проекта и запуск кода

Для создания проекта следует выбрать пункт *New Project* на Экране приветствия (рисунок 2.1) или в меню в списке *File* выбрать пункт *New Project* (рисунок 2.2). Далее откроется окно для настройки проекта (рисунок 2.3). В поле *Location* необходимо указать путь к проекту. Название папки и будет названием проекта. Кроме пути к проекту все остальные настройки можно оставить по умолчанию. Необходимо нажать на кнопку *Create*, после чего будет создан пустой проект (рисунок 2.4)

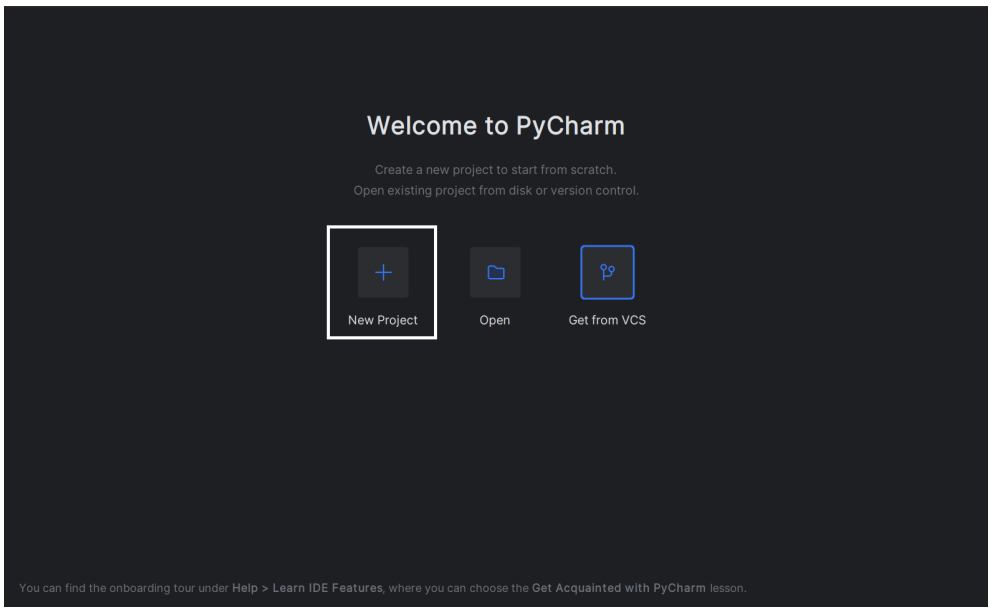


Рисунок 2.1

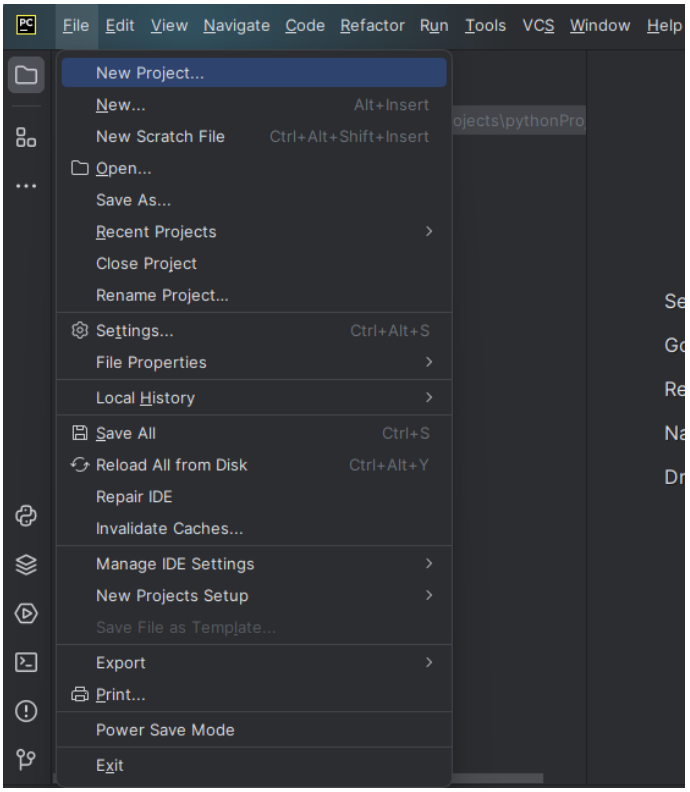


Рисунок 2.2

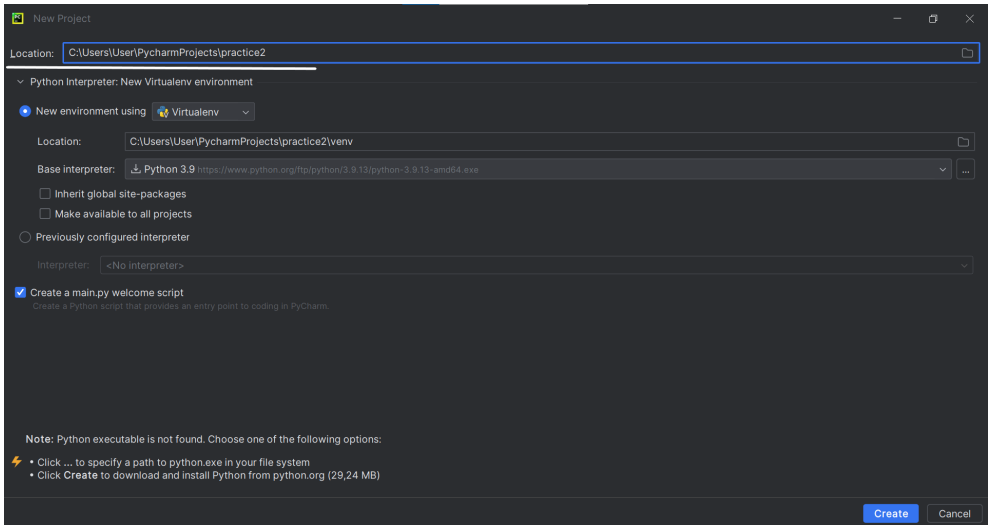


Рисунок 2.3

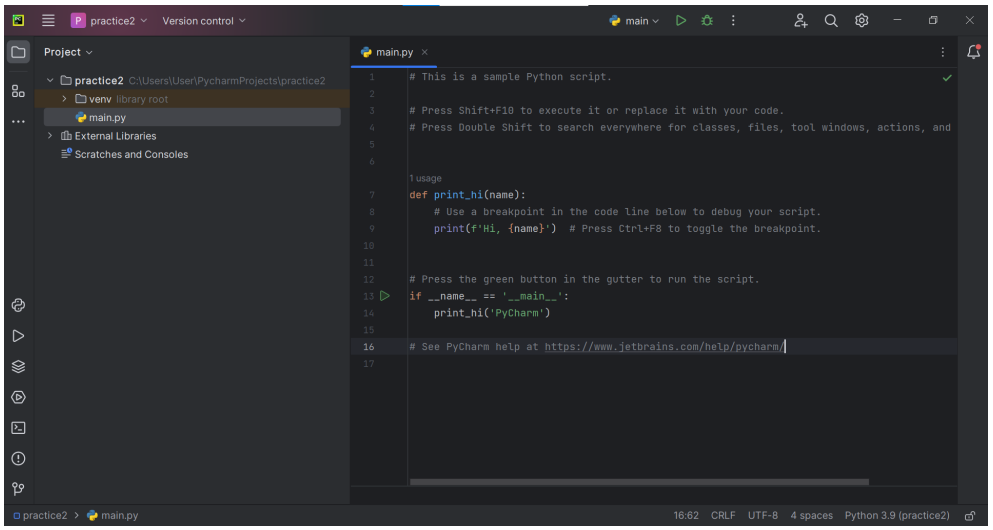
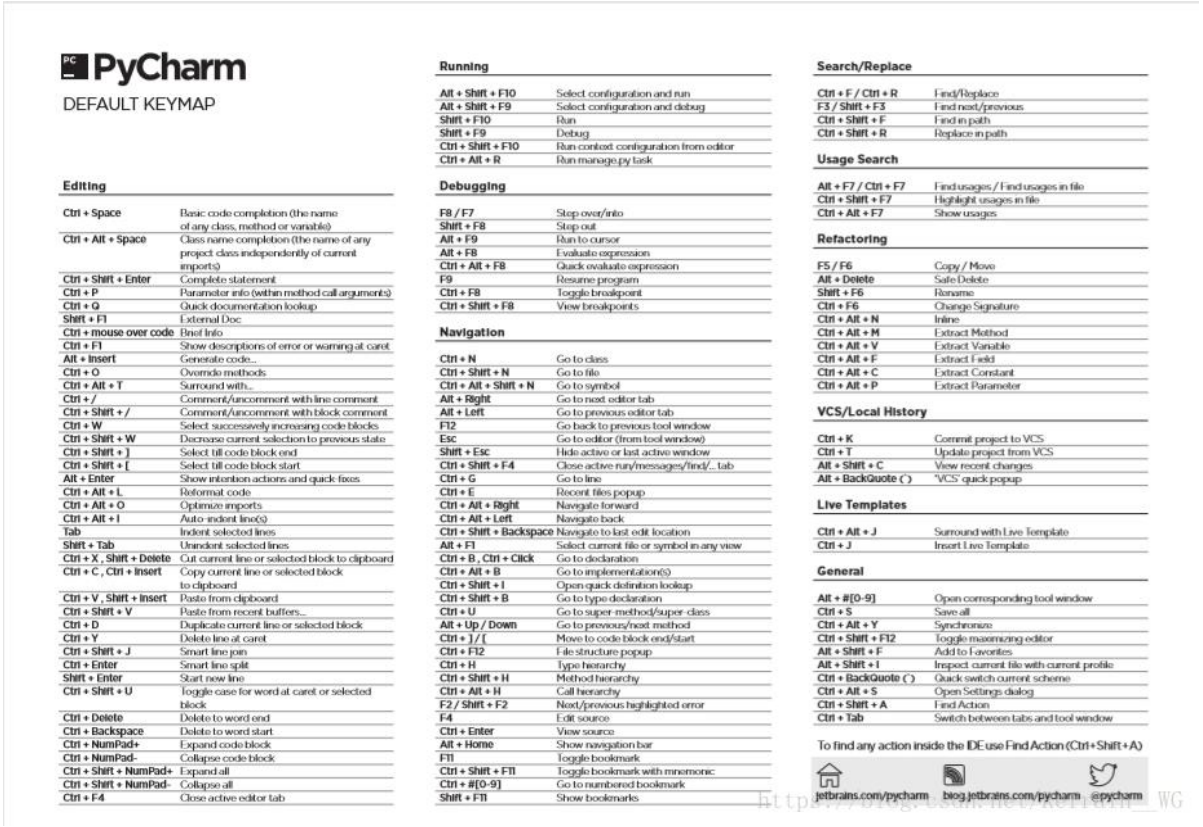


Рисунок 2.4

2) Сочетания клавиш

Использование комбинации клавиш сильно сокращает время, затрачиваемое на выполнение определенных действий. На рисунке 3.1 представлены список раскладок и все доступные сочетания клавиш.



PyCharm

DEFAULT KEYMAP

Editing

Ctrl + Space	Basic code completion (the name of any class, method or variable)
Ctrl + Alt + Space	Class name completion (the name of any project class independently of current imports)
Ctrl + Shift + Enter	Complete statement
Ctrl + P	Parameter info (within method call arguments)
Ctrl + Q	Quick documentation lookup
Shift + F1	External Doc
Ctrl + mouse over code	Brief Info
Ctrl + F1	Show descriptions of error or warning at caret
Alt + Insert	Generate code...
Ctrl + O	Override methods
Ctrl + Alt + T	Surround with...
Ctrl + /	Comment/uncomment with line comment
Ctrl + Shift + /	Comment/uncomment with block comment
Ctrl + W	Select successively increasing code blocks
Ctrl + Shift + W	Decrease current selection to previous state
Ctrl + Shift +]	Select till code block end
Ctrl + Shift + [	Select till code block start
Alt + Enter	Show intention actions and quick fixes
Ctrl + Alt + L	Reformat code
Ctrl + Alt + O	Optimize imports
Ctrl + Alt + I	Auto-indent line(s)
Tab	Indent selected lines
Shift + Tab	Unindent selected lines
Ctrl + X , Shift + Delete	Cut current line or selected block to clipboard
Ctrl + C , Ctrl + Insert	Copy current line or selected block to clipboard
Ctrl + V , Shift + Insert	Paste from clipboard
Ctrl + Shift + V	Paste from recent buffers...
Ctrl + D	Duplicate current line or selected block
Ctrl + Y	Delete line at caret
Ctrl + Shift + J	Smart line join
Ctrl + Enter	Smart line split
Shift + Enter	Start new line
Ctrl + Shift + U	Toggle case for word at caret or selected block
Ctrl + Delete	Delete to word end
Ctrl + Backspace	Delete to word start
Ctrl + NumPad+	Expand code block
Ctrl + NumPad-	Collapse code block
Ctrl + Shift + NumPad+	Expand all
Ctrl + Shift + NumPad-	Collapse all
Ctrl + F4	Close active editor tab

Running

Alt + Shift + F10	Select configuration and run
Alt + Shift + F9	Select configuration and debug
Shift + F10	Run
Shift + F9	Debug
Ctrl + Shift + F10	Run context configuration from editor
Ctrl + Alt + R	Run manage.py task

Debugging

F8 / F7	Step over/into
Shift + F8	Step out
Alt + F9	Run to cursor
Alt + F8	Evaluate expression
Ctrl + Alt + F8	Quick evaluate expression
F9	Resume program
Ctrl + F8	Toggle breakpoint
Ctrl + Shift + F8	View breakpoints

Navigation

Ctrl + N	Go to class
Ctrl + Shift + N	Go to file
Ctrl + Alt + Shift + N	Go to symbol
Alt + Right	Go to next editor tab
Alt + Left	Go to previous editor tab
F12	Go back to previous tool window
Esc	Go to editor (from tool window)
Shift + Esc	Hide active or last active window
Ctrl + Shift + F4	Close active run/messages/find/... tab
Ctrl + G	Go to line
Ctrl + E	Recent files popup
Ctrl + Alt + Right	Navigate forward
Ctrl + Alt + Left	Navigate back
Ctrl + Shift + Backspace	Navigate to last edit location
Alt + F1	Select current file or symbol in any view
Ctrl + B , Ctrl + Click	Go to declaration
Ctrl + Alt + B	Go to implementation(s)
Ctrl + Shift + I	Open quick definition lookup
Ctrl + Shift + B	Go to type declaration
Ctrl + U	Go to super-method/super-class
Alt + Up / Down	Go to previous/next method
Ctrl +] / [	Move to code block end/start
Ctrl + F12	File structure popup
Ctrl + H	Type hierarchy
Ctrl + Shift + H	Method hierarchy
Ctrl + Alt + H	Call hierarchy
F2 / Shift + F2	Next/previous highlighted error
F4	Edit source
Ctrl + Enter	View source
Alt + Home	Show navigation bar
F11	Toggle bookmark
Ctrl + Shift + F11	Toggle bookmark with mnemonic
Ctrl + #[0-9]	Go to numbered bookmark
Shift + F11	Show bookmarks

Search/Replace

Ctrl + F / Ctrl + R	Find/Replace
F3 / Shift + F3	Find next/previous
Ctrl + Shift + F	Find in path
Ctrl + Shift + R	Replace in path

Usage Search

Alt + F7 / Ctrl + F7	Find usages / Find usages in file
Ctrl + Shift + F7	Highlight usages in file
Ctrl + Alt + F7	Show usages

Refactoring

F5 / F6	Copy / Move
Alt + Delete	Safe Delete
Shift + F6	Rename
Ctrl + F6	Change Signature
Ctrl + Alt + N	Inline
Ctrl + Alt + M	Extract Method
Ctrl + Alt + V	Extract Variable
Ctrl + Alt + F	Extract Field
Ctrl + Alt + C	Extract Constant
Ctrl + Alt + P	Extract Parameter

VCS/Local History

Ctrl + K	Commit project to VCS
Ctrl + T	Update project from VCS
Alt + Shift + C	View recent changes
Alt + BackQuote (`)	'VCS' quick popup

Live Templates

Ctrl + Alt + J	Surround with Live Template
Ctrl + J	Insert Live Template

General

Alt + #[0-9]	Open corresponding tool window
Ctrl + S	Save all
Ctrl + Alt + Y	Synchronize
Ctrl + Shift + F12	Toggle maximizing editor
Alt + Shift + F	Add to Favorites
Alt + Shift + I	Inspect current file with current profile
Ctrl + BackQuote (`)	Quick switch current scheme
Ctrl + Alt + S	Open Settings dialog
Ctrl + Shift + A	Find Action
Ctrl + Tab	Switch between tabs and tool window

To find any action inside the IDE use Find Action (Ctrl+Shift+A)

jetbrains.com/pycharm blog.jetbrains.com/pycharm @pycharm

Рисунок 3.1

3) Форматирование кода

DataGrip автоматически отформатирует код в соответствии со стандартом PEP 8. Необходимо нажать сочетание клавиш Ctrl+Alt+L и все ошибки форматирования будут исправлены.

4) Запуск и отладка кода

Запустить код в PyCharm можно несколькими способами. Самый базовый - нажать на зеленый треугольник в заголовке окна (рисунок 4.1). Следующий способ - кликнуть по коду правой кнопкой и выбрать *Run '<имя вашего проекта>'* (рисунок 4.2). Последний способ - зажать комбинацию клавиш Ctrl + Shift + F10.

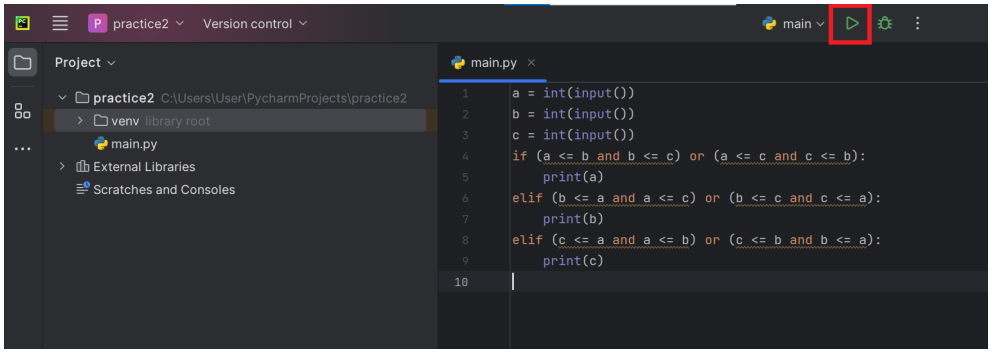


Рисунок 4.1

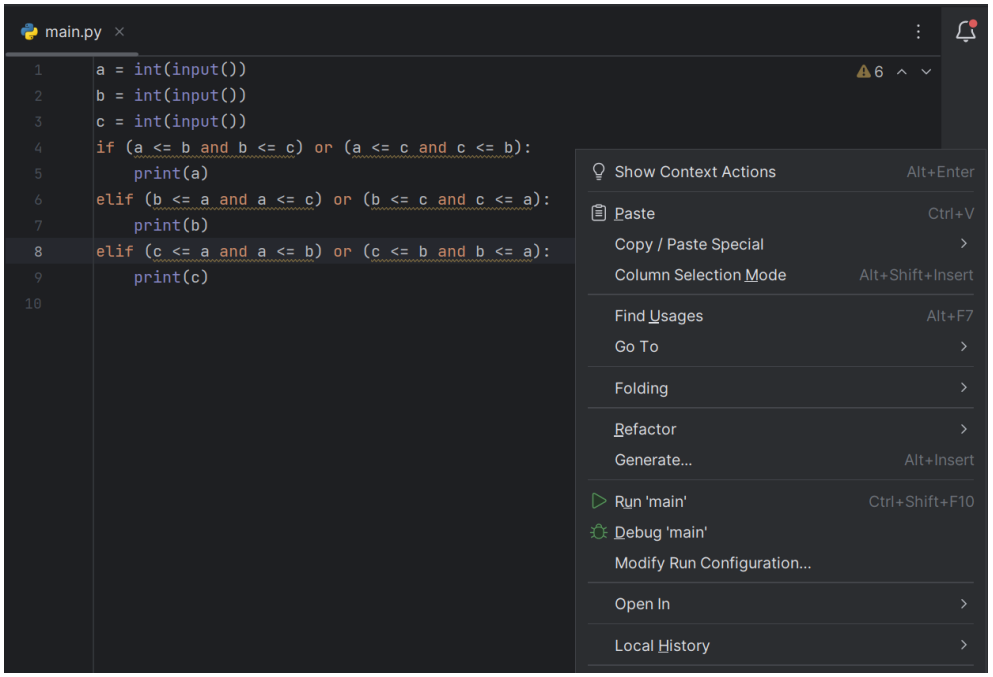


Рисунок 4.2

Графический отладчик Pycharm делает процесс максимально простым, визуализируя отладку в удобном формате. Для отладки нужно поставить брейкпоинты. Чтобы это сделать, кликните на номер строки слева от кода (Gutter). Затем запустите программу в режиме debug (рисунок 4.3). После запуска кода в режиме debug, программа остановится на первом брейкпоинте. Вы можете использовать кнопки, изображенные на рисунке 4.4:

- Step Over (F8) – выполнить текущую строку и перейти к следующей

- Step Into (F7) – зайти внутрь функции
- Step Out (Shift+F8) – выйти из функции

Также в этом режиме можно менять значения переменных, которые появляются в окне Threads&Variables (рисунок 4.5). Для этого следует дважды кликнуть по окну и ввести новое значение.

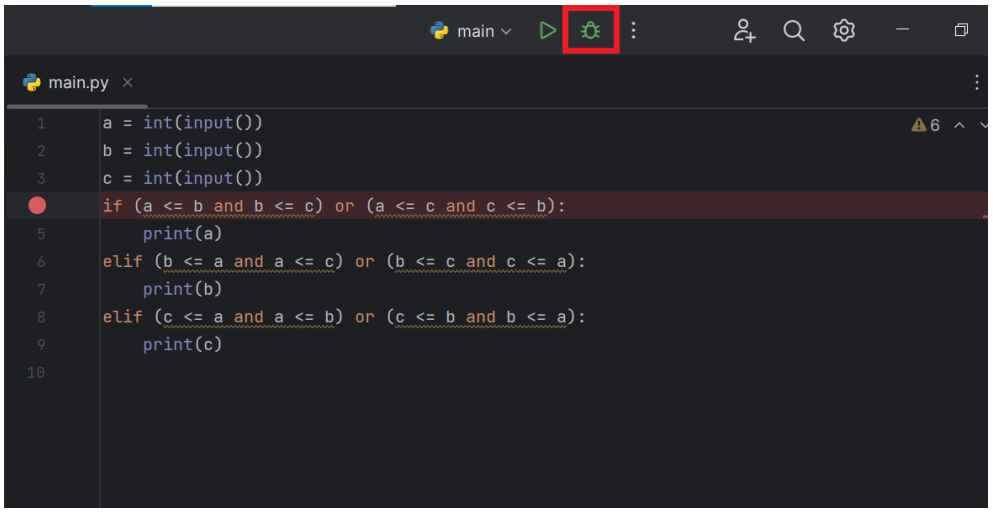


Рисунок 4.3

Рисунок 4.4

Рисунок 4.5

5) Подсветка синтаксиса

Редактор PyCharm распознает и выделяет ключевые слова, комментарии, параметры, типы данных и другие элементы (рисунок 5.1).

Рисунок 5.1

Конкретные цвета выделения определяются в диалоговом окне настроек.

В новых версиях PyCharm поддерживает более специфические типы и языковые структуры, например, специфичное для Python 3.10 сопоставление с образцом (рисунок 5.2).

Рисунок 5.2

6) Работа с Git

В PyCharm с помощью встроенных функций можно проделывать все основные действия с Git. Чтобы добавить свой проект на GitHub нужно проделать последовательность действий:

1) Создать новый репозиторий на GitHub;
2) Создать проект в PyCharm;
3) В заголовке окна выбирать *VCS -> Create Git Repository* (рисунок 6.1);
4) В GitHub на странице проекта скопировать ссылку в формате HTTPS;
5) Вместо *VCS* в заголовке окна появился раздел *Git*. В нем выбрать *Manage Remotes*. Нажать "+". В "*URL*" вставить ссылку на репозиторий GitHub, которая была скопирована заранее. Нажать "ОК";
6) В разделе Git выбрать *Commit*. Если нужно - написать сообщение. Выбрать все файлы и нажать "*Commit*" (рисунок 6.2);
7) Чтобы проект оказался на GitHub, в разделе Git следует выбрать *Push* и нажать "ОК";
8) В репозитории на GitHub появится ветка *master*. Там и находится созданный проект с остальными добавленными файлами.

В дальнейшем, чтобы добавлять проекты в этот же репозиторий, у другого проекта в заголовке окна необходимо выбрать *VCS -> Share on GitHub*. Там нажать "*Share*". После обновления страницы на GitHub появится проект.

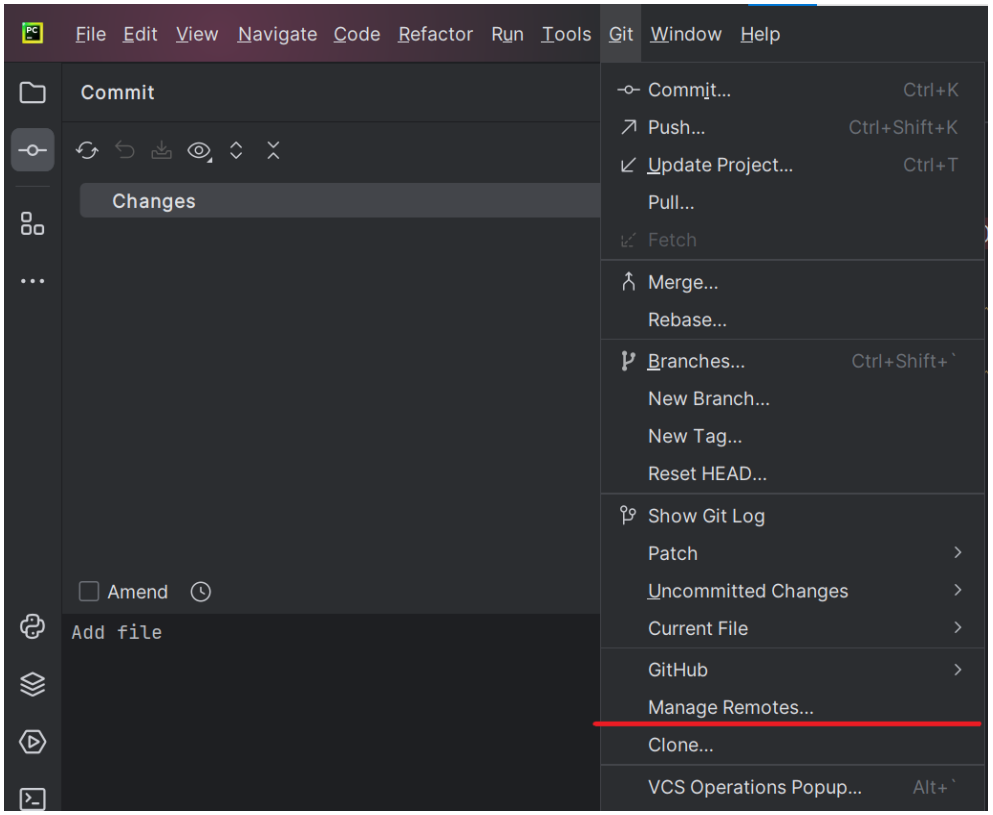

File Edit View Navigate Code Refactor Run Tools Git Window Help
Commit
Changes
Amend
Add file
Commit... Ctrl+K
Push... Ctrl+Shift+K
Update Project... Ctrl+T
Pull...
Fetch
Merge...
Rebase...
Branches... Ctrl+Shift+`
New Branch...
New Tag...
Reset HEAD...
Show Git Log
Patch
Uncommitted Changes
Current File
GitHub
Manage Remotes...
Clone...
VCS Operations Popup... Alt+`

Рисунок 6.1

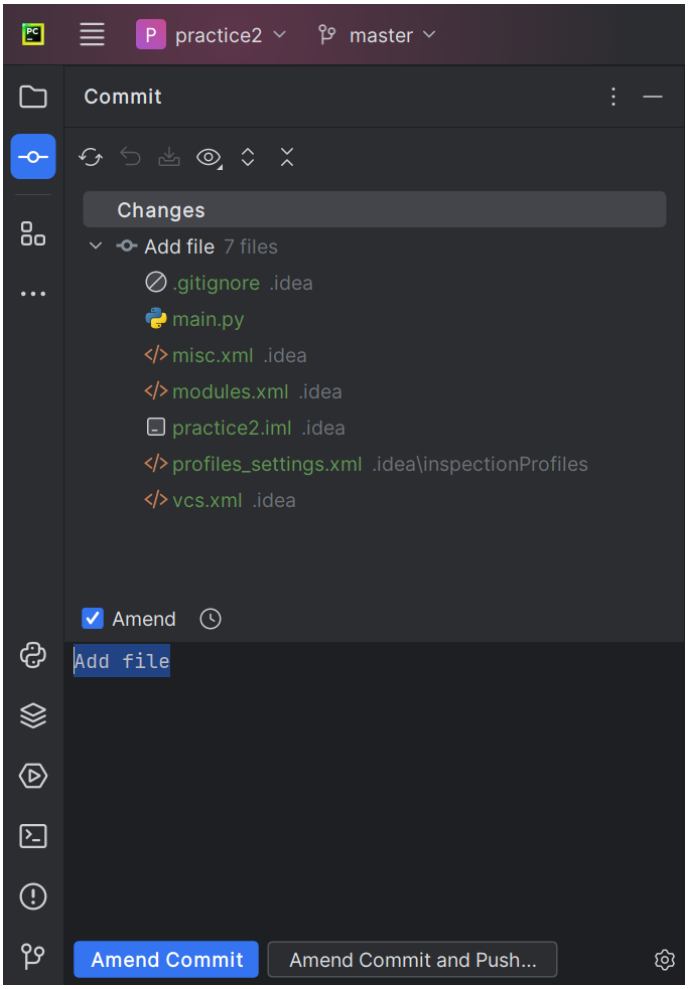

practice2
master
Commit
Changes
Add file 7 files
.gitignore .idea
main.py
misc.xml .idea
modules.xml .idea
practice2.iml .idea
profiles_settings.xml .idea\inspectionProfiles
vcs.xml .idea
Amend
Add file
Amend Commit
Amend Commit and Push...

Рисунок 6.2