

Общая характеристика.

Для ускорения процесса написания программного кода удобно использовать специализированную инструментальную среду - так называемую интегрированную среду разработки (IDE, Integrated Development Environment). Эта среда включает полный комплект средств, необходимых для эффективного программирования на Python. Обычно в состав IDE входят текстовый редактор, компилятор или интерпретатор, отладчик и другое программное обеспечение. Использование IDE позволяет увеличить скорость разработки программ. В процессе разработки программных модулей удобнее работать в интерактивной среде разработки (IDE), а не в текстовом редакторе.

Для Python одним из лучших вариантов считается IDE PyCharm от компании JetBrains. Эта среда разработки доступна для Windows, Linux и macOS. Существуют два вида лицензии PyCharm: Professional (проприетарная платная версия с триальным периодом) и Community (свободно-распространяемая версия с открытым исходным кодом).

Функции.

Создание проекта.

При создании проекта появляется диалоговое окно, в котором следует указать адрес нового каталога (или согласиться с заданным по-умолчанию), создавать ли для проекта собственное виртуальное окружение.

Выбор между новой виртуальной средой (New environment using) и уже существующей (Previously configured interpreter).

В начале изучения языка Python может быть целесообразнее выбрать пункт Previously configured interpreter. После этого через список Interpreter: <No interpreter> выбрать системный интерпретатор (System Interpreter), указав его адрес.

Но, как правило, создаётся новая виртуальная среда. В поле Location можно выбрать местонахождение папки `venv`. По умолчанию папка `venv` расположится внутри папки проекта. Если необходимо создать папку с именем отличным от `venv`, то в поле Location можно стереть `venv` и вписать название папки.

Пункт Inherit global site-packages позволяет включить в виртуальную

среду все библиотеки, установленные в глобальной среде. Make available to all projects отвечает за видимость созданной виртуальной среды для других проектов. Если нужна одна виртуальная среда сразу на несколько проектов, то стоит выбрать этот пункт.

После нажатия Create запустится среда разработки, в ней будет открыт только что созданный проект.

Слева на панели Project управляют файлами проекта. Чтобы создать файл, в котором будет написана программа на Python, кликнем по этой папке правой кнопкой мыши. В контекстном меню выбираем New → Python File.

После этого в центральной части среды разработки появится небольшое окно, в которое вписываем имя файла. Нажав Enter, вы увидите файл на панели Project. Также он будет открыт в центральной части окна PyCharm.

После того, как исходный код написан, чтобы первый раз запустить программу, проще всего нажать Ctrl+Shift+F10. Внизу раскроется вкладка Run, в которой отобразиться результат выполнения.

Иногда удобнее, чтобы панель выполнения программы открывалась не снизу, а, например, справа. В этом случае в настройках панели (справа значок похожий на гайку) следует выбрать Move to → Right Top.

Внешний вид среды и множество других ее свойств, поведение настраиваются в окне Settings (меню File → Settings).

Нажатие Ctrl + D дублирует строку, в которой находится курсор. Выделенный участок можно сдвинуть вправо (сделать вложенным), нажав Tab. Смещение влево (на внешний уровень) выполняется комбинацией Shift + Tab.

Поднять/опустить (поменять местами с предшествующей/нижестоящей) строку или выделенный участок

можно с помощью сочетаний Shift + Ctrl + стрелка вверх или стрелка вниз клавиатуры.

Кодирование.

в PyCharm есть Intelligent Coding Assistance — интеллектуальный ассистент кодирования, который делает автодополнение кода, проверяет синтаксис, сообщает об ошибках и даёт рекомендации по их исправлению. В частности, если написать `main` и нажать `Tab`, PyCharm автоматически полностью завершит всю конструкцию `main`. Также если перед `if` поставить точку `.if` и нажать `Tab`, то PyCharm полностью напишет конструкцию `if`. То же самое верно для `True.while` — работает PyCharm's Postfix completions (постфиксное дополнение кода).

Форматирование кода.

Команда Ctrl + Alt + L исправляет ошибки форматирования в коде, добавляет отступы.

Отладка.

Есть три способа начать отладку:

1. Нажать Ctrl + Shift + D on Mac or Shift + Alt + F9 в Windows или Linux.
2. Щелкнуть фон правой кнопкой мыши и выбрать Debug.
3. Щелкнуть маленькую зеленую стрелку слева от предложения `__main__` и выбрать оттуда Debug.

После этого вы увидите открытое окно Debug внизу.

Запуск.

Есть три способа запуска программы:

1. Использование клавиши Ctrl+Shift+R на Mac или Ctrl+Shift+F10 на Windows или Linux.
2. Нажать правую кнопку мыши в поле редактирования и в меню выбрать Run.

3. Если в программе есть предложение `__main__`, то щелкнуй на маленькую зелёную стрелку слева от фразы `__main__` и выбрать Run.

Любой из этих вариантов приведёт к запуску программы, и появится панель «Run Tool» в нижней части окна, с выводом кода.

Компиляция.

Компиляцию можно провести с помощью модуля Pyinstaller.

В Terminal нужно перейти в папку проекта и ввести `'pyinstaller название_проекта.py'`. После нажатия Enter в папке с проектом появится несколько новых папок, в том числе dist, внутри которой и будет файл с программой формата .exe. Его можно запустить.

Публикация в репозитории на GitHub (для этого нужна учетная запись).

Алгоритм:

1. Перейти на панель VCS, которая находится в верхней части рuсharm, и нажать на нее. После нажатия выбрать “Включить интеграцию с системой управления версиями” (“Enable Version Control Intigration”).
После нажатия кнопки “Включить интеграцию с системой управления версиями” появится всплывающее окно. Нужно выбрать Git из выпадающего меню и нажать ОК. Это инициализирует проект для GitHub.
2. Следующий шаг - нажать на зеленую галочку, присутствующую в верхней части рuсharm. По сути, этот символ обозначает фиксацию, и он собирает все Unversioned файлы и подготавливает их к обновлению на GitHub.
После нажатия появится новый экран. Сначала нужно выбрать Unversioned Files, затем в окно Commit Message ввести “Initial Commit”, а затем нажать на кнопку Commit.
Теперь проект готов к загрузке на GitHub.
3. Последний шаг, который необходимо выполнить, - перейти в VCS, затем выбрать Import into Version Control, а затем нажать на Share Project on GitHub.
После нажатия появится всплывающее окно, в котором пользователь должен ввести свой идентификатор входа в

GitHub и пароль. После успешного входа в свою учетную запись появится еще одно всплывающее окно, где пользователь должен указать название и описание репозитория. Затем нужно нажать на кнопку Share и готово. Примечание: не ставьте пробелы или специальные символы при указании имени репозитория.

Дополнительные функции.

Навигация.

Умный поиск позволяет быстро перейти к любому классу, файлу или символу, а также к нужному окну или действию IDE.

Быстрые и безопасные рефакторинги.

PyCharm предоставляет широкие возможности реорганизации кода с помощью рефакторингов Rename и Delete, Extract Method, Introduce Variable, Inline Variable, Inline Method и многих других. Рефакторинги учитывают особенности конкретного языка или фреймворка, помогая вносить изменения по всему проекту.

Веб-фреймворки Python

PyCharm обеспечивает поддержку популярных веб-фреймворков, таких как Django, Flask, Google App Engine, Pyramid и web2py. Вы можете создавать и отлаживать Django-шаблоны, работать с утилитами manage.py и appcfg.py, а также использовать специфичные для фреймворков автодополнение и навигацию.

Поддержка научных библиотек

PyCharm поддерживает Pandas, Numpy, Matplotlib и другие библиотеки для научных вычислений. IDE обеспечивает умное редактирование, позволяет просматривать наборы данных в виде графиков и в табличной форме.

Необходимое программное и аппаратное обеспечение.

Официальные системные требования последней версии PyCharm выглядят так:

ОС:

Windows — Microsoft Windows 10 64-бит или Windows 8 64-бит;

macOS — macOS 10.13 или выше;

Linux — среда GNOME или KDE.

RAM: не менее 2 ГБ, но рекомендуется 8 ГБ;

Место на диске: установка потребует 2,5 ГБ, рекомендуется использование SSD;

Разрешение экрана: не менее 1024x768 пикселей;

Python: Python 2.7, Python 3.5 или более поздняя версия.

Источники:

1. Постолиит А. В. П63 Python, Django и PyCharm для начинающих. -СПб.: БХВ-Петербург, 2021. -464 с.: ил. -(Для начинающих) ISBN 978-5-9775-6779-4 v
2. <https://habr.com/ru/articles/720480/>
3. <https://younglinux.info/python/pycharm>
4. <https://waksoft.susu.ru/2019/11/07/pycharm-effektivnaya-razrabotka-na-python/#writing-code-in-pycharm>
5. https://www.codeflow.site/ru/article/pycharm-guide#_debugging_in_pycharm
6. https://translated.turbopages.org/proxy_u/en-ru.ru.1529e844-64fc523a-1a3a1885-74722d776562/https/www.geeksforgeeks.org/how-to-upload-project-on-github-from-pycharm/
7. <https://www.jetbrains.com/ru-ru/pycharm/features/>
8. <https://pythonchik.ru/osnovy/faq-po-rabote-s-pycharm>