

---

# Обзор PyCharm



---

*Подготовили студенты 2 курса*

*Нечаева Наталья Андреевна*

*Фирсов Кирилл Александрович*

*Яблонская Евгения Дмитриевна*

---

# IDE (Integrated Development Environment) – интегрированная среда разработки.

В состав IDE обычно входят текстовый редактор, компилятор или интерпретатор, отладчик и другое программное обеспечение.

Одной из самых удобных IDE при работе с Python считается PyCharm (JetBrains).

---

---

# PyCharm

**Windows, Linux, MacOS**

Professional	Community
Проприетарная платная версия с триальным периодом. Предназначена для профессиональных разработчиков, более широкая функциональность.	Свободно распространяемая версия. Подойдет для обучения или для небольших проектов.

# Необходимое программное и аппаратное обеспечение

Операционная система	Microsoft Windows 8, 10 64-bit MacOS 10.13 и выше Для Linux - среды GNOME или KDE
Оперативная память	Не менее 2 Гб, рекомендуемая - 8 Гб
Место на диске	Установка потребует 2.5 Гб, рекомендуется использование SSD
Разрешение экрана	Не менее 1024x768 пикселей
Версии Python	2.7, 3.5 и более поздние

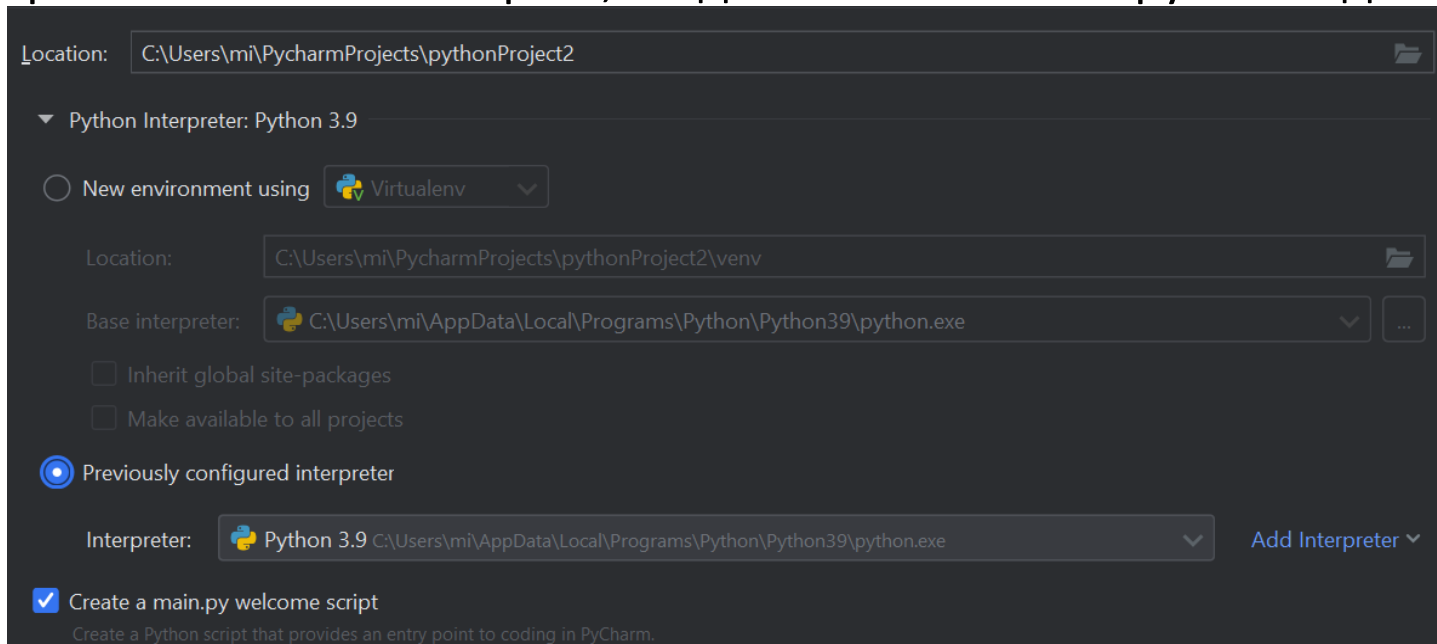
---

# Основные функции

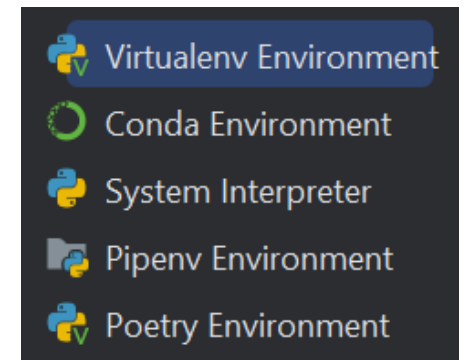
- Создание проекта;
- Кодирование;
- Форматирование кода;
- Отладка;
- Запуск;
- Компиляция;
- Публикация в репозитории на GitHub.

# Создание проекта

При создании проекта откроется окно, в котором нужно будет указать имя нового проекта, его расположение и выбрать, создавать ли новое окружение для него.



Для выбора существующего окружения необходимо выбрать *Previously configured interpreter*, после чего выбрать *Add Interpreter*. В открывшемся окне выбрать желаемое.



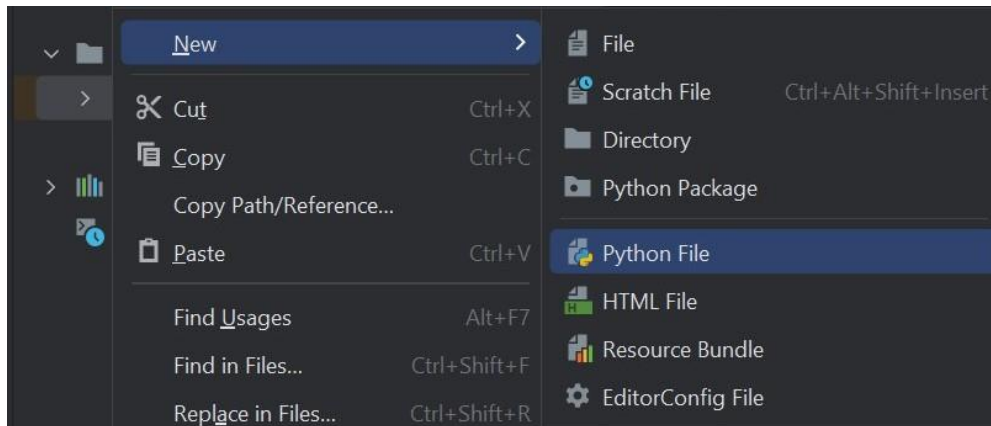
Но, как правило, создается новое окружение.

---

# Создание проекта

Все файлы созданного проекта перечислены на панели слева.

Для того, чтобы создать файл для написания программы, нужно правой кнопкой мыши нажать на корневую папку проекта, после кликнуть на *New* и выбрать необходимое.



После выбора откроется новое окно, в которое нужно вписать имя нового файла.

После создания файла, он появится на панели слева.

В центре окна PyCharm откроется окно редактирования кода.

---

---

# Кодирование

**Intelligent Coding Assistance** — интеллектуальный ассистент кодирования, делает автодополнение кода, проверяет синтаксис, сообщает об ошибках и даёт рекомендации по их исправлению.

Например, если написать *main* и нажать Tab, Space или Enter (Windows и MacOS), PyCharm автоматически завершит конструкцию *main*.

PyCharm's Postfix completions — постфиксное дополнение кода.



# Форматирование кода

Ctrl+Alt+L (Windows) или Cmd+Opt+L (MacOS) - команда, исправляющая ошибки форматирования в коде и добавляющая отступы.

```
main.py x
1 name = input("Введите свое имя: ")
2 coding = input(f"Скажи {name} когда тебе комфортно писать код, ночью или днем? ")
3 if coding == "днем":
4     print("Хорошо, я тоже часто пишу код днем")
5     elif coding == "ночью":
6         print("Ночью наступает полная концентрация")
7     else:
8         print("Введено неверно слово, повтори ввод")
```



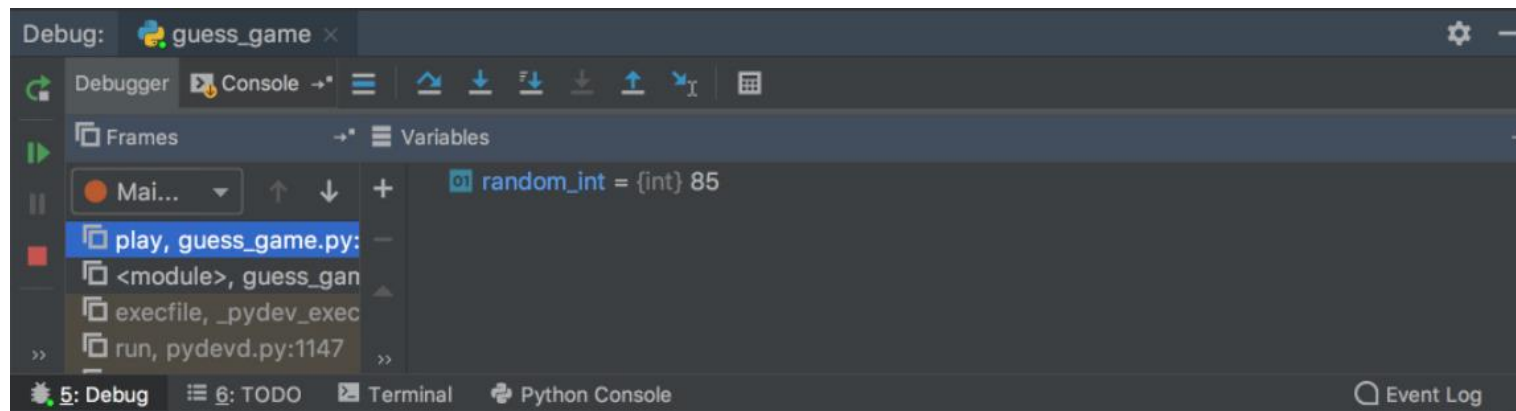
```
main.py x
1 name = input("Введите свое имя: ")
2 coding = input(f"Скажи {name} когда тебе комфортно писать код, ночью или днем? ")
3 if coding == "днем":
4     print("Хорошо, я тоже часто пишу код днем")
5 elif coding == "ночью":
6     print("Ночью наступает полная концентрация")
7 else:
8     print("Введено неверно слово, повтори ввод")
```

# Отладка

Есть несколько способов начать отладку:

1. Ctrl+Shift+D (MacOS) или Shift+Alt+F9 и Alt+5 (Windows/Linux).
2. После щелчка правой кнопкой мыши в открывшемся меню выбрать Debug <имя файла>.
3. Слева на панели нажать на зеленую стрелку и выбрать Debug <имя файла>

После этого внизу откроется окно Debug.



---

# Запуск программы

1. Ctrl+Shift+R (MacOs) или Ctrl + Shift + F10 (Windows/Linux)
2. После щелчка правой кнопкой мыши в открывшемся меню выбрать Run <имя>.
3. В главном меню выбрать Run.
4. Если в программе есть предложение `__main__`, то щёлкнуть на маленькую зелёную стрелку слева от фразы `__main__` и выбрать Run.

Любой из этих вариантов приведет к запуску программы и открытию в нижней части окна панели «Run Tool» с выводом кода.

---

---

# Компиляция

Компиляцию можно провести с помощью модуля Pyinstaller.

Для этого в поле Terminal (Alt+F12 либо View→Tool Windows→Terminal) после пути проекта ввести «pyinstaller \*название проекта\*.py».

После нажатия Enter в папке проекта появятся новые папки, в том числе папка «dist», в который будет лежать файл с тем же названием, что и проект, но с форматом «.exe». Его и нужно запустить.

---

# Публикация в GitHub

Для публикации своего проекта в GitHub необходимо иметь учетную запись.

1. VCS (в главном меню) → «Enable Version Control Integration». После открытия всплывающего окна выбрать GitHub, это инициализирует репозиторий git.
2. После этого на верхней панели появится зелёная галочка (обозначает Commit). После нажатия откроется новое окно, в котором нужно выбрать Unversioned Files → ввести «Initial Commit» (окно Commit Message) → Commit.
3. Перейти в VCS → Import into Version Control → Share Project on GitHub.
4. В всплывшем окне ввести свои данные GitHub, после чего указать название и описание нового репозитория, в котором расположится проект из PyCharm, и нажать Share.

---

# Дополнительные функции

---

В PyCharm много дополнительных функций, значительно упрощающих работу с ним.

# Навигация

В PyCharm имеется умный поиск, осуществляемый с помощью горячих клавиш. Благодаря ему можно перейти к любому классу, файлу, символу, а также окну или действию.

Горячие клавиши для MacOS и Windows отличаются.

Некоторые примеры:

Действие	MacOS	Windows
Сдвинуть курсор к началу строки	Cmd + ←	Home
Недавний файл	Cmd + E	Ctrl + E
Показать все закладки	Cmd + F3	Shift + F11
Перейти к строчке/столбцу...	Cmd + L	Ctrl + G

---

# Рефакторинги и веб-фреймворки Python

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

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



---

# Библиотеки

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

# Плагины

Для PyCharm существуют плагины для улучшения работы с кодом.

Название	Функционал
<i>Tabnine</i>	<ul style="list-style-type: none"><li>• автоматически дополняет код</li><li>• предлагает закончить функцию при её объявлении</li><li>• генерирует коды блока по запросу из комментариев</li></ul>
<i>PyLint</i>	<ul style="list-style-type: none"><li>• проверяет код на соответствие стандартам (длина строки, имена переменных)</li><li>• подсвечивает ошибки в коде, анализирует его без запуска</li></ul>
<i>SonarLint</i>	В реальном времени подсвечивает ошибки, предоставляет комментарий с причиной ошибки, риском и возможностями её устранения
<i>Rainbow Brackets</i>	Красит скобки в различные цвета, чтобы было проще отслеживать их иерархию
<i>Python Smart Execute</i>	Позволяет запустить отдельный блок кода в консоли.