**Обзор программного продукта PyCharm**

**Что такое PyCharm?**

PyCharm — интегрированная среда разработки, с полным набором средств, предназначенных для разработки на языке Python.

Доступный как кроссплатформенное приложение, PyCharm совместим с платформами Linux, macOS и Windows. PyCharm является одной из лучших IDE Python и обеспечивает поддержку как версий Python 2 (2.7), так и Python 3 (3.5 и выше).

PyCharm поставляется с множеством модулей, пакетов и инструментов для ускорения разработки на Python, одновременно сокращая усилия, необходимые для одновременного выполнения того же самого. Кроме того, PyCharm можно настроить в соответствии с требованиями разработки и личными предпочтениями. Впервые он был выпущен для широкой публики еще в феврале 2010 года.

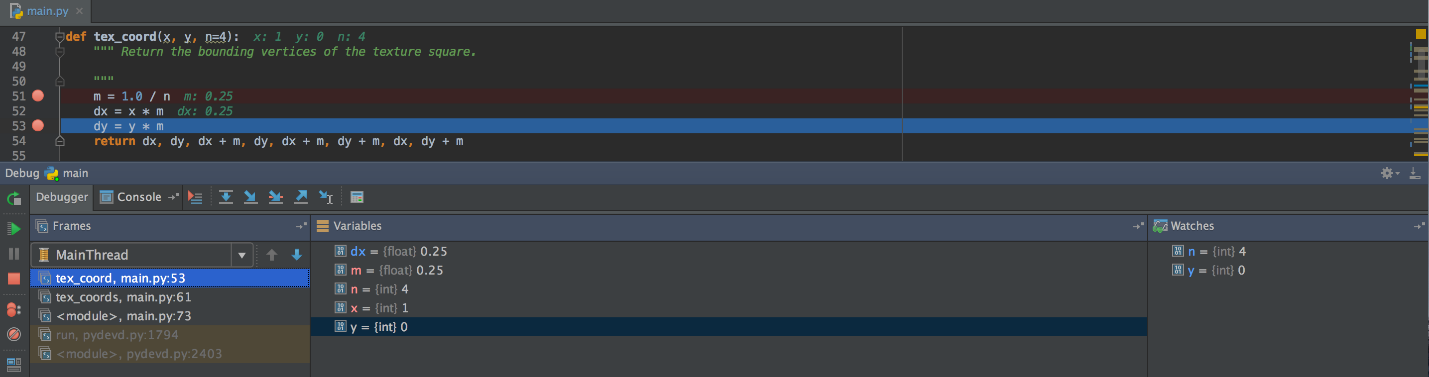
**Функции PyCharm**

PyCharm предоставляет большой набор инструментов из коробки: встроенный отладчик и инструмент запуска тестов, профилировщик Python, полнофункциональный встроенный терминал, инструменты для работы с базами данных. IDE интегрирована с популярными системами контроля версий, содержит встроенный SSH-терминал, поддерживает возможности удаленной разработки и удаленные интерпретаторы.

**Визуальный отладчик**

PyCharm предоставляет широкие возможности отладки кода на Python/Django и JavaScript:

* Расставляйте точки останова и задавайте условия их срабатывания прямо в редакторе
* Проверяйте контекстно-зависимые локальные переменные и определяемые пользователем watches, включая массивы и сложные объекты, редактируйте значения на лету



**Встроенный отладчик**

В режиме встроенной отладки значения переменных, параметров функций и других объектов доступны прямо в окне редактора. Значения переменных можно посмотреть прямо в исходном коде рядом с их использованием.

**Трассировка только по коду проекта**

Функция Step into My Code позволяет в режиме отладки осуществлять трассировку только по коду проекта, не углубляясь в библиотечные исходники.

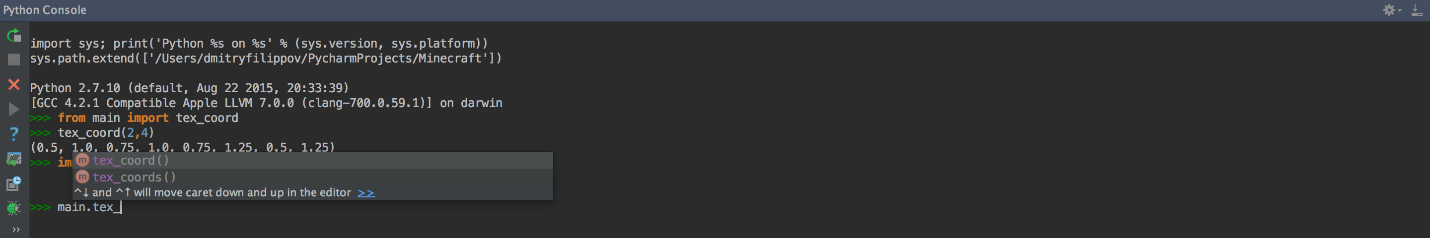
**Отладка нескольких процессов**

PyCharm умеет отлаживать приложения, которые порождают несколько процессов Python, например, приложения Django, которые не запускаются в режиме no-reload, или приложения, использующие другие веб-фреймворки, в которых реализован аналогичный подход к автоматической перезагрузке кода.

**Конфигурации Run/Debug**

При каждом выполнении скрипта/теста или отладчика создается специальная конфигурация запуска/отладки, которую можно изменить и использовать повторно. Как и настройки проекта, конфигурации Run/Debug можно использовать совместно всей командой.

**Выполнение кода**



### Запуск в любом окружении

Вы можете запустить любой файл на Python: просто кликните по нему правой кнопкой и выберите Run <your script>.

PyCharm позволяет легко настроить необходимое окружение для вашего проекта: локальное, удаленный интерпретатор или virtualenv.

### Интерактивная консоль

Вы можете запустить интерактивную консоль для Python или Django, в которой есть много преимуществ над стандартными консолями. Среди них: проверка синтаксиса на лету с помощью инспекций, сопоставление скобок и кавычек и, конечно, автодополнения. Обе консоли работают с локальными и удаленными интерпретаторами.

### Встроенный терминал

С PyCharm нет необходимости покидать IDE во время разработки. Полнофункциональный встроенный терминал работает на платформах Windows, Linux и macOS.

**Полная поддержка Python**

PyCharm поддерживает все основные реализации языка Python: Python 2.x и 3.x, Jython, IronPython, PyPy и Cython, обеспечивая:

* Подсветку синтаксиса, проверку кода на ошибки, возможности форматирования
* Автодополнение кода, учитывающее контекст
* Навигацию по коду и просмотр структуры
* Быстрый поиск вхождений и инструменты рефакторинга
* Инспекции кода и многое другое

### Юнит-тестирование

PyCharm позволяет легко выполнять модульное тестирование благодаря интеграции с популярными тестовыми фреймворками: doctests, nose и attest.

В PyCharm можно запустить тестовый файл, тест для отдельного класса или метода или все тесты из папки одновременно. Результаты тестов удобно просматривать в графическом интерфейсе инструмента запуска тестов, который отображает статистику их выполнения и позволяет быстро навигироваться по тестируемому коду.

### Покрытие кода

PyCharm интергирован с инструментом анализа покрытия кода Coverage.py. Он отслеживает, какие части программы были выполнены, и анализирует исходный код, чтобы определить, какие фрагменты кода могли быть выполнены, но этого не произошло. Результаты отображаются в наглядном формате для последующего анализа и быстрой навигации по коду.

### Python-профилировщик

PyCharm поддерживает два популярных профилировщика: yappi и cProfile. Делайте снэпшоты и собирайте статистику работы вашего приложения прямо в PyCharm — результаты отображаются в виде цветного графа вызовов функций. Вы можете посмотреть сводный отчет, также предусмотрена навигация из графа в код.

### Инструменты BDD

PyCharm поддерживает два наиболее популярных инструмента для разработки с использованием BDD — Behave и Lettuce. Вы можете писать человекопонимаемый код тестов, описывающий поведение вашего приложения. Поддержка включает: помощь в инсталляции и конфигурации, Run Configuration и Intellisense.

### Визуализация параллелизма потоков

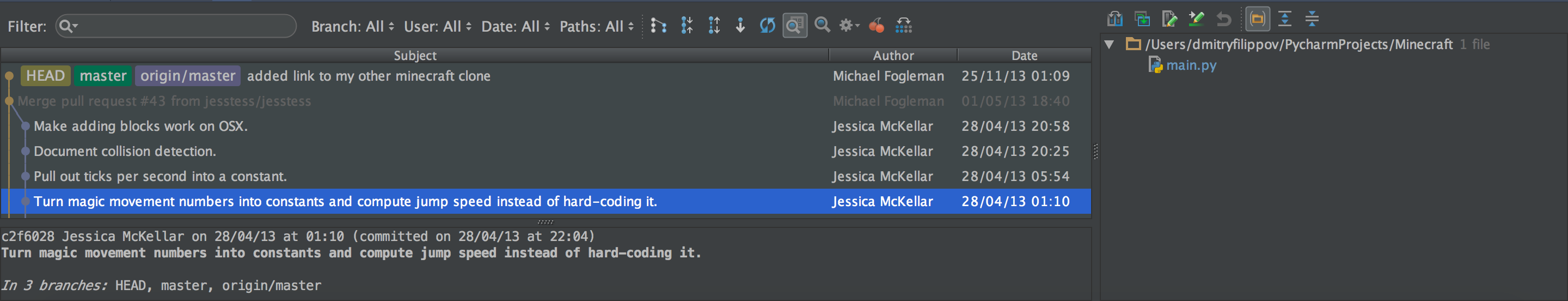
Визуализация параллельных потоков помогает контролировать многопоточные приложения. Вы можете запустить ваше приложение в режиме Concurrency Diagram и проверять состояния потоков внутри запущенного процесса на диаграмме в реальном времени.

**Системы контроля версий**

Поддержка Git, SVN, Mercurial, Perforce и других систем контроля версий помогает управлять локальными изменениями и производить сложные операции с ветками. Все однообразные задачи (добавление и удаление файлов) выполняются автоматически.

Следующие функции доступны независимо от того, какую систему контроля версий вы используете:

* Интегрированные списки изменений — группируйте изменения по разным спискам для удобства организации
* Отложенные изменения — сохраните отдельно изменения, которые вы хотите внести позднее
* Просмотр изменений репозитория — отслеживайте, какие изменения внесли ваши коллеги
* Просмотр входящих изменений — просматривайте изменения, которые еще не были внесены в локальную копию
* Уведомление о неактуальных изменениях — получите предупреждение о том, что файл, с которым вы работаете, был изменен после последней синхронизации



### Локальная история

PyCharm отслеживает любые изменения в исходных файлах, защищает от случайной потери данных и не дает другим приложениям вносить изменения в проект. Вы всегда можете просмотреть историю изменений файла или каталога и откатиться к любой из предыдущих версий.

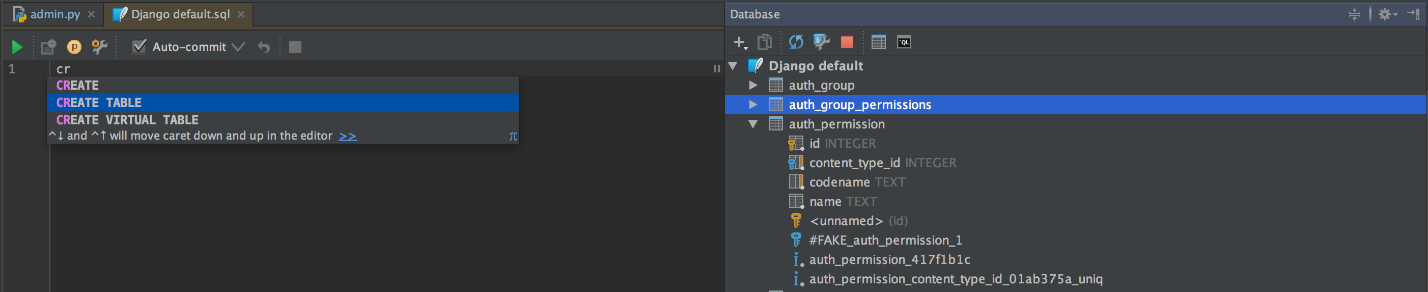
### Встроенный инструмент слияния

Встроенный инструмент visual merge легко и быстро разрешает конфликты слияния. Локальные изменения подсвечиваются при вводе в редакторе, обеспечивая интуитивно понятную навигацию и возможность отменить отдельное изменение в два клика.

### Diff Viewer

С помощью встроенного средства просмотра изменений Diff Viewer вы можете легко сравнить два файла или две версии одного файла (просмотреть локальные изменения или изменения в локальных файлах и их версиях в удаленном репозитории).

**Инструменты для работы с базами данных**



### Редактирование баз данных

Доступ к Oracle, SQL Server, PostgreSQL, MySQL и другим базам данным осуществляется прямо из IDE. PyCharm помогает редактировать SQL-код, просматривать данные, изменять схемы и таблицы, выполнять запросы и анализировать схемы с помощью UML-диаграмм.

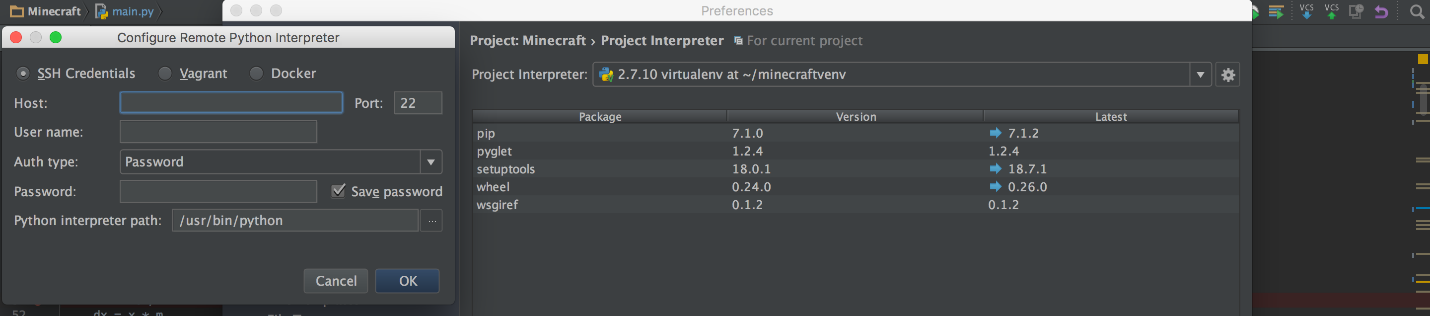
### Вставки SQL-кода

Код на SQL можно вставить в файлы на других языках и работать с ним в SQL-редакторе, который предоставляет подсветку синтаксиса, автодополнение, анализ кода на лету, форматирование и удобную навигацию.

### Поддержка SQLAlchemy

PyCharm производит анализ кода библиотеки SQLAlchemy, на основании которого предлагает варианты автодополнения. IDE также поддерживает визуализацию данных при помощи диаграмм SQLAlchemy.

**Возможности удаленной разработки**



### Инструменты удаленного запуска

Встроенная SSH-консоль позволяет подключаться к любому удаленному компьютеру и выполнять различные действия через SSH. Кроме того, можно настроить SSH Remote Tools для запуска любого удаленного инструмента прямо из IDE одним нажатием клавиши.

### Удаленный интерпретатор

Использование удаленного интерпретатора Python вместо локального позволяет запускать, отлаживать и профилировать приложения в рабочих условиях или тестовой среде на обычном или виртуальном сервере с помощью Vagrant или Docker.

### Vagrant

Vagrant — это удобный инструмент, встроенный в PyCharm, который помогает создавать воспроизводимое виртуальное окружение для разработки. Он позволяет поделиться с другими членами команды простым скриптом, который содержит в себе всю информацию о виртуальной машине. Другие разработчики смогут запустить идентичное виртуальное окружение, выполнив команду vagrant up.

### Docker

PyCharm интегрирован с Docker, популярной платформой для распределенных приложений и контейнерной виртуализации. Выберите Python-интерпретатор из Docker-контейнера и используйте его в проекте для запуска, отладки и профилирования приложений.

**Интеграция с баг-трекерами**

Если вы используете систему отслеживания ошибок, подключите PyCharm к вашему баг-трекеру, чтобы работать с задачами прямо из IDE. Для этого нужно просто указать сервер вашего баг-трекера.

* Выберите задачи из отфильтрованного списка
* Изменения в VCS автоматически группируются по задачам
* Перед коммитом имя/ID задачи заполняются автоматически
* При переключении между зачами IDE сохранит контекст (открытые владки редактора, точки останова и тд.)

PyCharm поддерживает следующие системы отслеживания ошибок: Atlassian JIRA, JetBrains YouTrack, Lighthouse, Pivotal Tracker, GitHub, Redmine и Trac.

**Необходимое программное и аппаратное обеспечение**

PyCharm кроссплатформенна и совместима с Windows, Linux и MacOS. Она поддерживает, как

вторую, так и третью версию Питона и, кроме всего прочего, имеет красивый и функциональный

UI.

ОС:

• Windows — Microsoft Windows 10 64-битили 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;

**Создание проекта**

1. Если вы находитесь на экране приветствия, нажмите Новый проект. Если у вас уже открыт какой-либо проект, выберите Файл | Новый проект в главном меню.
2. Выберите местоположение проекта. Нажмите кнопка Обзоркнопку рядом с полем Location и укажите каталог для вашего проекта.
3. Кроме того, снимите флажок Создать main.py флажок приветствия скрипта установлен, потому что вы создадите новый файл Python для этого руководства.
4. Лучшая практика Python - создать virtualenv для каждого проекта. В большинстве случаев PyCharm автоматически создает новую виртуальную среду, и вам не нужно ничего настраивать. Тем не менее, вы можете просмотреть и изменить параметры venv. Разверните узел Интерпретатор Python: новая среда Virtualenv и выберите инструмент, используемый для создания новой виртуальной среды. Давайте выберем Virtualenv tool и укажем расположение среды и базовый интерпретатор Python, используемый для новой виртуальной среды.

При настройке базового интерпретатора вам необходимо указать путь к исполняемому файлу Python. Если PyCharm не обнаруживает Python на вашем компьютере, он предоставляет следующие опции:

* Укажите путь к исполняемому файлу Python (в случае нестандартной установки)
* Загрузите и установите последние версии Python с python.org
* Установите Python с помощью инструментов разработчика командной строки (только для macOS).

**Кодирование**

Умный редактор PyCharm предназначен для максимально продуктивной разработки на Python, JavaScript, CoffeeScript, TypeScript, CSS и популярных языках шаблонов.

* В PyCharm код читается легко благодаря возможностям настройки цветов подсветки синтаксиса Python и шаблонов Django. Просто выберите одну из предложенных цветовых схем.
* PyCharm предлагает варианты автодополнения для ключевых слов, классов и переменных, когда вы печатаете код. Они также доступны по нажатию Ctrl+Space. Автодополнение учитывает контекст и предлагает наиболее подходящие варианты.
* Выберите фрагмент кода и выделите выражение, строку или логический блок, используя сочетания клавиш. Добавить/удалить комментарий к выделенному фрагменту можно одним нажатием.
* Ошибки отображаются, когда вы набираете код. Встроенная проверка орфографии предупредит об опечатках в текстах идентификаторов и комментариев.
* Многочисленные инспекции проверяют код прямо в режиме редактирования, а также позволяют проанализировать весь проект на наличие ошибок и проблем в структуре кода.
* Для большинства инспекций доступны быстрые исправления, которые позволяют откорректировать код мгновенно. Нажмите Alt+Enter, чтобы просмотреть список возможных исправлений для любой инспекции.
* С помощью умного детектора дубликатов PyCharm проверяет код на наличие дублирующихся фрагментов. IDE предложит список фрагментов, которые следует преобразовать, а рефакторинги помогут избавиться от повторяющегося кода.
* Редактируйте код, написанный не на Python, внутри строковых литералов. При этом доступны функции автодополнения, подсветки ошибок и другие возможности IDE.

**Форматирование кода**

В PyCharm есть встроенный форматер кода, отвечающий рекомендациям стандарта PEP-8 для Python, а также других стандартов для поддерживаемых языков. Отступы добавляются автоматически в начале новой строки. Проверка корректности отступов и автоматическое переформатирование производятся в соответствии с настройками стиля кода проекта.