

Урок 0. Подготовка к прохождению курса

# ОГЛАВЛЕНИЕ

[ОГЛАВЛЕНИЕ 2](#_Toc87619592)

[НАСТРАИВАЕМ PyCHARM 3](#_Toc87619593)

[скачиваем установщик и устанавливаем 3](#_Toc87619594)

[скачиваем и устанавливаем плагины для корректировки стиля кода 4](#_Toc87619595)

[УСТАНАВЛИВАЕМ ГРАФИЧЕСКУЮ ОБОЛОЧКУ ДЛЯ РАБОТЫ С СИСТЕМОЙ КОНТРОЛЯ ВЕРСИЙ GIT И СЕРВИСОМ ХОСТИНГА IT-ПРОЕКТОВ GITHUB 5](#_Toc87619596)

[РЕГИСТРИРУЕМ АККАУНТ НА GITHUB 6](#_Toc87619597)

[РЕГИСТРИРУЕМ АККАУНТ В ВКОНТАКТЕ 6](#_Toc87619598)

[РЕГИСТРИРУЕМ АККАУНТ В HEROKU 7](#_Toc87619599)

[ЗАКЛЮЧЕНИЕ 8](#_Toc87619600)

[ключевые выводы 8](#_Toc87619601)



# НАСТРАИВАЕМ PyCHARM

## скачиваем установщик и устанавливаем

Нам будет достаточно версии Community, но, разумеется, если у вас будет версия Professional, это будет большим плюсом и облегчит работу с проектом.

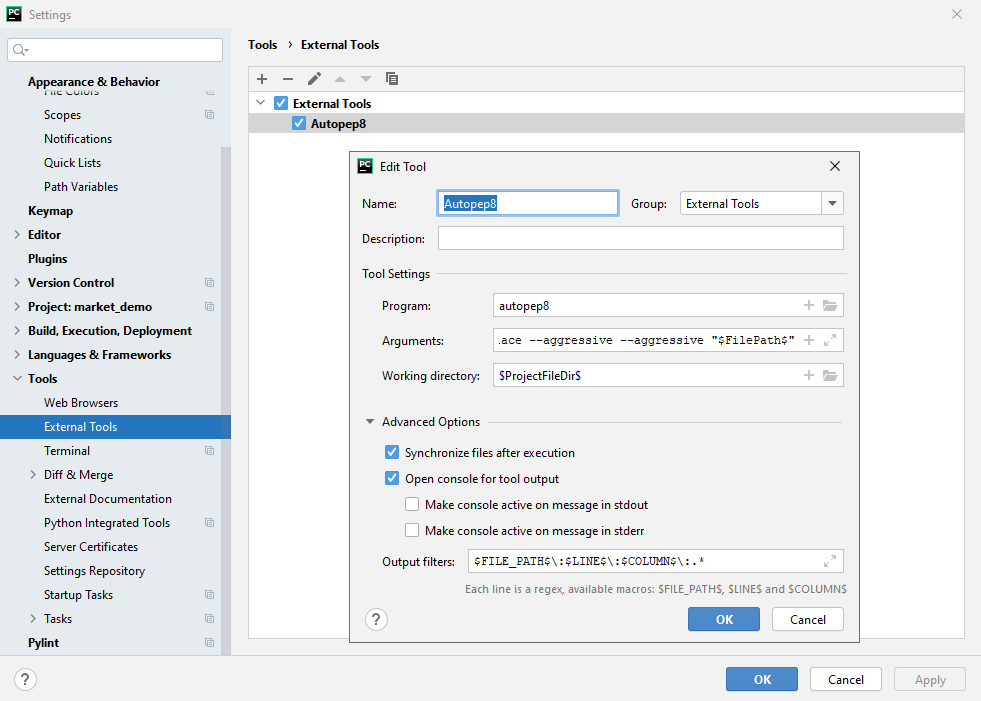
По установке версии Community вопросов не должно возникнуть. Если вы решите работать с версией Professional, то при покупке лицензии вам предоставят ключ активации и инструкцию по активации, либо можно воспользоваться инструкцией, размещенной в материалах к этому уроку.

## скачиваем и устанавливаем плагины для корректировки стиля кода

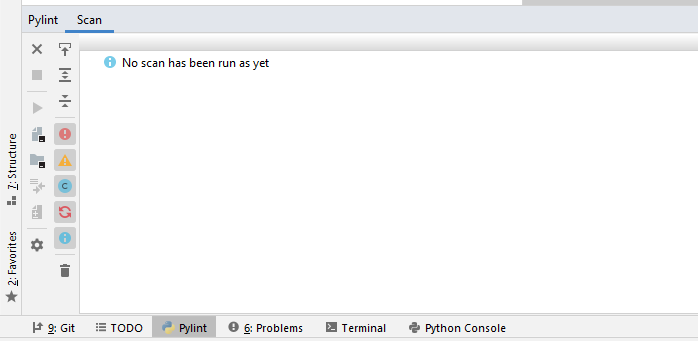
Нам в идеале при работе над проектом нужно придерживаться соглашения PEP-8, который регламентирует стиль кода при разработке проектов на Python.

**Первый плагин – Autopep8**. Он выполняет необходимые выравнивания, добавляет недостающие пробелы и убирает лишние.

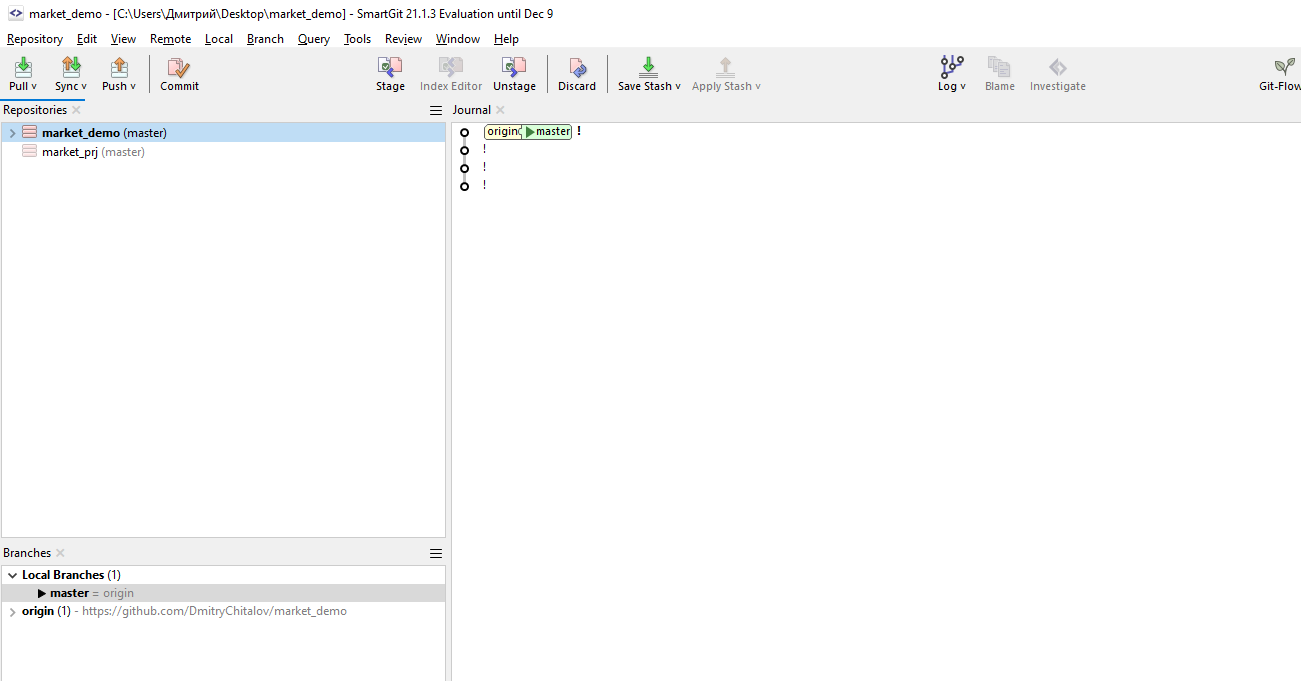
Подробнее о плагине будет рассказано в видео, настройки плагина будут приложены в материалы урока.



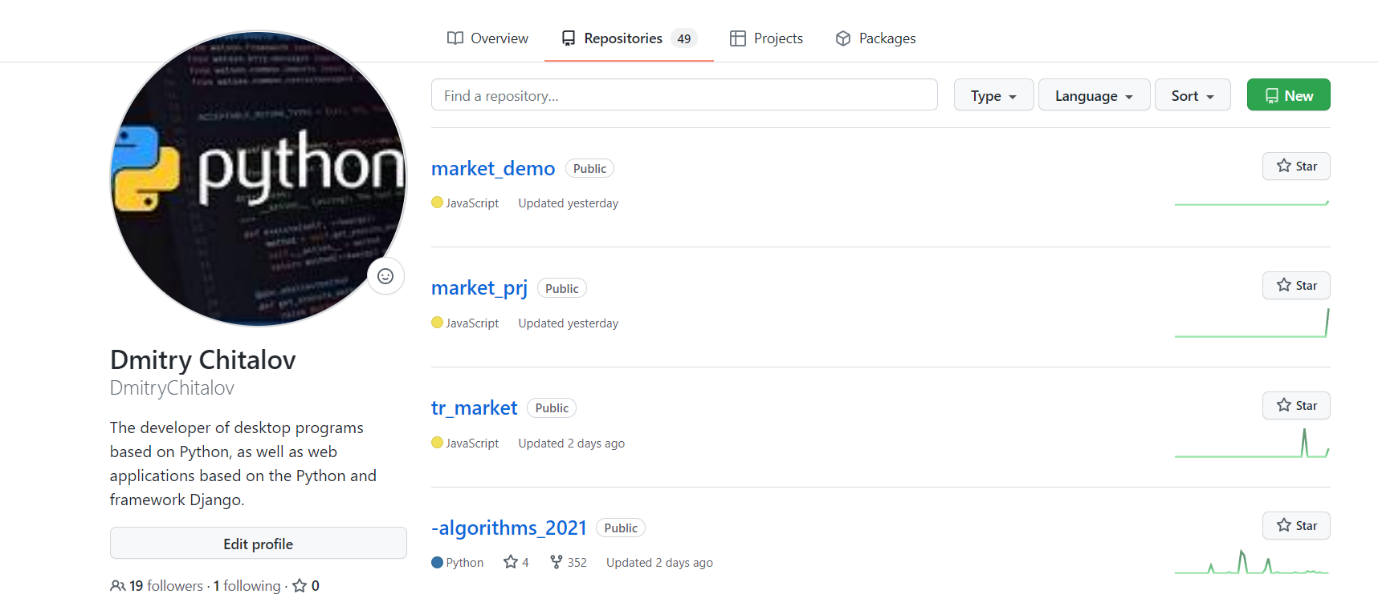
**Второй плагин – pylint**. Это линтер или другими словами «анализатор кода». Он не приводит код к правильному стилю автоматически, а лишь дает подсказки разработчику о предлагаемых исправлениях. Здесь важно слово «предлагаемых», поскольку не все замечания **pylint** нужно срочно бежать и выполнять. В компаниях, специализирующихся на разработке программного обеспечения, зачастую существуют свои соглашения по стилю кода и нужно ориентироваться на них.

Об установке и настройке **pylint** также будет рассказано в видео, а настройки для **pylint** приложены к материалам урока.

# УСТАНАВЛИВАЕМ ГРАФИЧЕСКУЮ ОБОЛОЧКУ ДЛЯ РАБОТЫ С СИСТЕМОЙ КОНТРОЛЯ ВЕРСИЙ GIT И СЕРВИСОМ ХОСТИНГА IT-ПРОЕКТОВ GITHUB

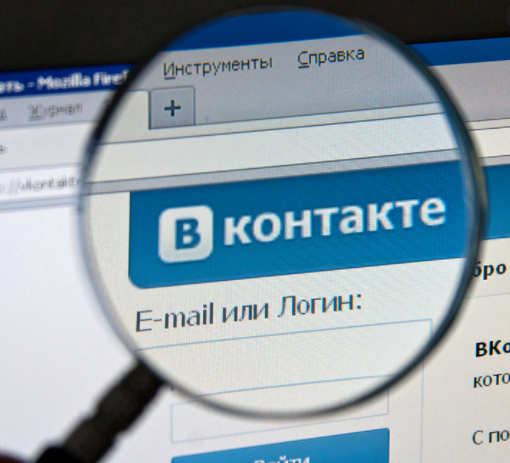
Работу с Git и Github можно осуществлять из консоли, но это требует хорошего знания консольных команд. Поэтому можно воспользоваться графической оболочкой, например, SmartGit. Подробнее о ее установке будет рассказано в уроке 8.

# РЕГИСТРИРУЕМ АККАУНТ НА GITHUB

Это сервис, обеспечивающий возможности хостинга исходного кода проектов, т.е. мы делаем код нашего проекта доступным для других разработчиков, кроме того, наличие ПРОЕКТА НА GITHUB УПРОЩАЕТ ЕГО РАЗВЕРТЫВАНИЕ В СЕТИ.

# РЕГИСТРИРУЕМ АККАУНТ В ВКОНТАКТЕ

В нашем курсе рассматривается возможность реализации аутентификации пользователя через социальные сети, в частности, через ВКонтакте, а значит нам будет нужен аккаунт в этой социальной сети.



# РЕГИСТРИРУЕМ АККАУНТ В HEROKU

Это облачный сервис, предоставляющий услуги хостинга ваших проектов. Подробнее работа с Heroku будет освещена на уроке 8.

# Вид сверху на рукопожатие над деловыми документамиЗАКЛЮЧЕНИЕ

Друзья, перед тем, как «бросаться в бой» и создавать проект, нужно сперва основательно подготовиться – установить и настроить необходимый инструментарий, т.е. среду разработки и т.д. Правильный выбор инструментария определяет удобство разработчика, скорость разработки и даже в целом ее возможность.

## ключевые выводы

* **PyCharm** версии Community или Professional – среда разработки проекта
* Плагины для PyCharm – обеспечивают соответствие кода соглашению PEP-8.
* **SmartGit** – графическая оболочка для работы с системой контроля версий Git и сервисом хостинга IT-проектов GitHub.
* **GitHub** – аккаунт в этом сервисе нам необходим для демонстрации кода проекта и обеспечения его развертывания в сети.
* **Heroku** – облачный сервис, предоставляющий услуги хостинга проектов в сети.