Настройка среды и основы Git

Олег Булыгин

Олег Булыгин

- Преподаватель на курсах “Основы языка программирования Python”, “Продвинутый Python”, “Python для анализа данных” в Нетологии
- Начальник бюро планирования и управления в АО “НПО автоматики”

НЕТОЛОГИЯ

О чём мы поговорим сегодня

1. Какое ПО нам понадобится для обучения?
2. Интерактивная оболочка Jupyter Notebook
3. Система контроля версий Git

Jupyter Notebook

НЕТОЛОГИЯ

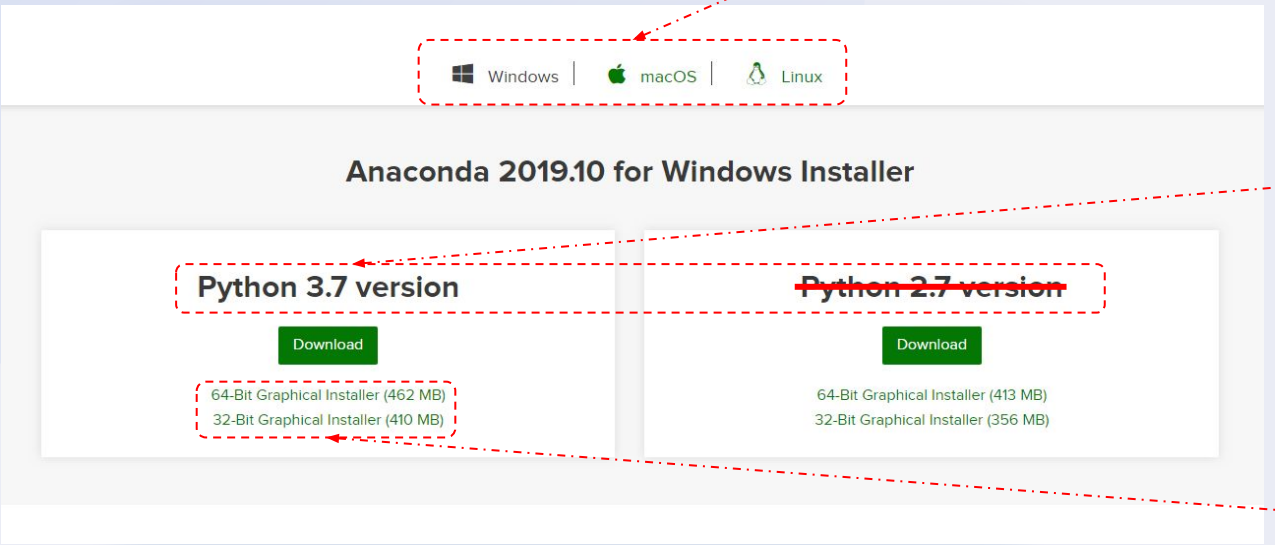
Jupyter Notebook – удобный инструмент для работы с данными, статистическим моделированием и машинным обучением.

Anaconda – дистрибутив, включающий в себя набор библиотек (не нужно устанавливать отдельно!) для научных и инженерных расчетов и интерактивную веб-оболочку Jupyter Notebook.

https://www.anaconda.com

Установка Anaconda

Выбираете свою ОС



Нам нужен Python 3.*

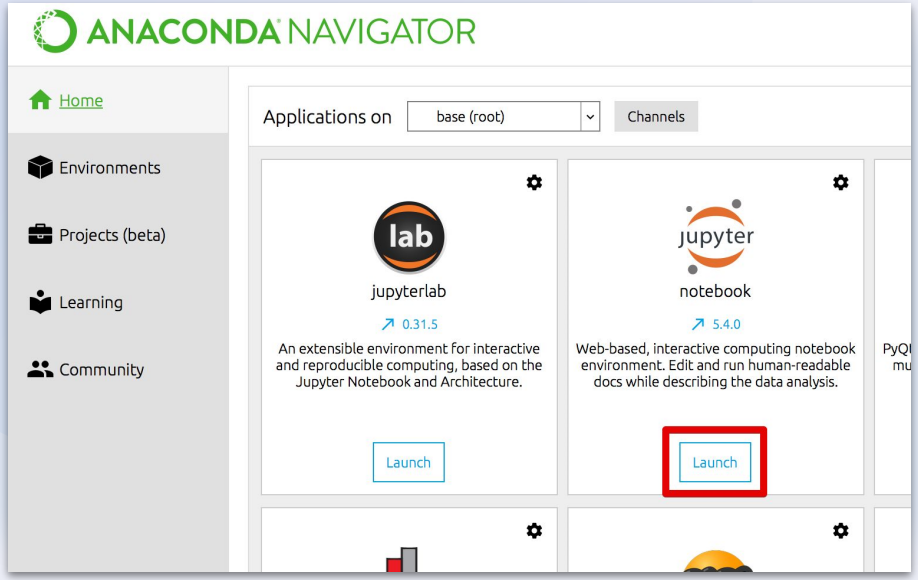
Выбираете разрядность своей системы

! При установке нужно обязательно поставить флажок "**Add python.exe to PATH**"

https://www.anaconda.com/dictribution/

НЕТОЛОГИЯ

Как запустить Jupyter Notebook?



- Anaconda Navigator;
- Anaconda prompt;
- командная строка

Git – система контроля версий

НЕТОЛОГИЯ

Git – распределённая система контроля версий, которая дает возможность отслеживать изменения в файлах и работать над ними совместно с коллегами

Преимущества Git

Бесплатный и open-source

1

Это значит, что его можно бесплатно скачать и вносить любые изменения в исходный код.

Небольшой и быстрый

2

Он выполняет все операции локально, что увеличивает его скорость. Кроме того, Git локально сохраняет весь репозиторий в небольшой файл без потери качества данных.

Резервное копирование

3

Git эффективен в хранении бэкапов, поэтому известно мало случаев, когда кто-то терял данные при использовании Git

Простое ветвление.

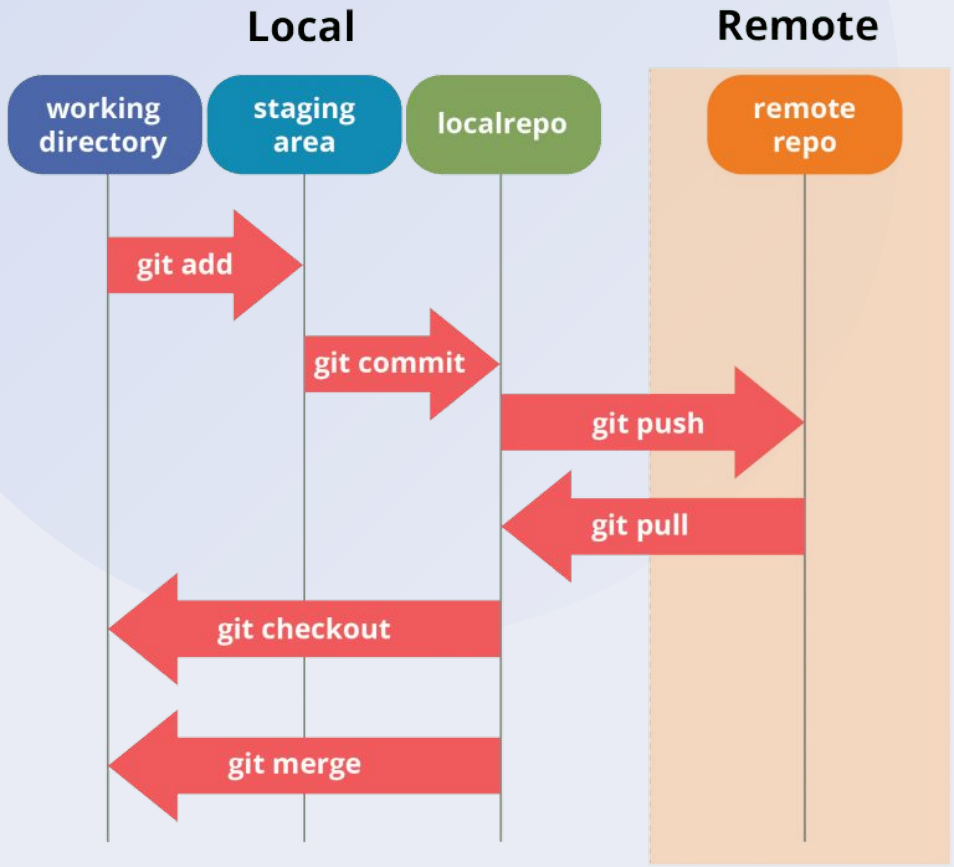
4

В других VSC создание веток— утомительная и трудоемкая задача, так как весь код копируется в новую ветку. В Git управление ветками реализовано гораздо проще и эффективнее.

Что требуется для начала?

Установить git
https://git-scm.com/download/

Зарегистрироваться на https://github.com/



Основные термины

- **Репозиторий (repository)** – место, где хранятся и поддерживаются какие-либо данные;
- **Рабочая директория (working directory)** – это файлы в корневой директории проекта, тот код с которым вы работаете;
- **Область подготовленных изменений (staging area)** – область в которой находятся подготовленные файлы которые могут быть включены или не включены в коммит;
- **Локальный репозиторий (local repo)** — она же директория .git. В ней хранятся коммиты и другие объекты;
- **Удаленный репозиторий (remote repo)** – репозиторий, в который вы можете передать свои коммиты из локального репозитория, чтобы коллеги могли их увидеть.

Создание/получение нового репозитория

Создание нового репозитория

1

Для того чтобы создать новый репозиторий git, необходимо открыть папку где вы хотите его разместить и выполнить команду git init

Получение репозитория

2

Создать локальную рабочую копию удаленного репозитория можно командой git clone <*url репозитория*>

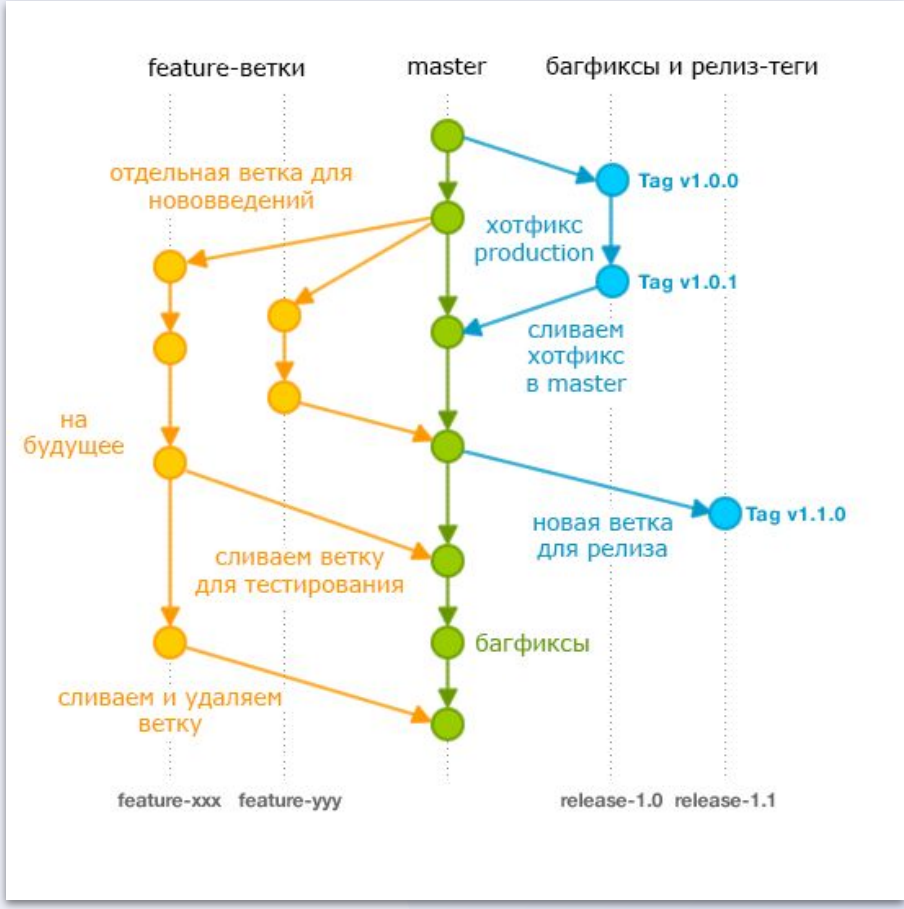
Основные команды

- git add(reset) *<имя файла / имя директории>* – добавление (удаление) файлов и директорий под версионный контроль;
- git commit –m “*комментарий*”- команда создающая слепок текущего состояния проекта и файлов в нем;
- git remote add origin <url удаленного репозитория> – команда, подключающая ваш локальный репозиторий к удаленному;
- git remote -v – список удаленных репозиториев;
- git status – просмотр состояния файлов в области подготовленных изменений (staging area);
- git push *<remote>* *<branch>* – запушить коммиты в удаленный репозиторий <remote> в ветку <branch>.

Ветвление

Ветки (branches) используются для разработки функциональности, изолированной от остальной. Ветка **master** используется по-умолчанию, когда вы создаете репозиторий. После завершения работы над функционалом производят слияние ветки с master.

- git branch *<имя ветки>* – создание новой ветки;
- git checkout *<имя ветки>* – переключение на ветку;
- git checkout -b *<имя ветки>* – создание новой ветки и переключение на нее;
- git branch -d *<имя ветки>* – удаление ветки;
- git push origin *<имя ветки>* – отправить ветку в удаленный репозиторий.



Обновление и слияние

- git pull – изменения из удаленного репозитория обновляют ваш локальный репозиторий;
- git merge <имя ветки> – слияние выбранной ветки с активной (отличия одной ветки от другой можно посмотреть командой git diff <имя ветки1> <имя ветки2>).

Автоматические изменения в результате этих команд не всегда возможны, т.к. может возникнуть конфликт. Его нужно разрешить путем ручного редактирование файлов и их git add *<имя файла>*.

Полезные ссылки

01 **Интерактивный тренажер git**
https://learngitbranching.js.org/

02 **Pro Git Book**
https://git-scm.com/book/en/v2

03 **Краткое руководство по GitKraken**
https://github.com/StriderAJR/StudentCpp/blob/master/manuals/GitKraken%20manual.md

Как справляться с проблемами?

1) Мы, к сожалению, не сможем вам предоставить **всю** информацию, т.к.:
    a) мир постоянно меняется;
    b) постоянная актуализация своих знаний – неотъемлемый атрибут любого успешного специалиста.
2) Лекции + практические задания дают основную канву и принципы;
3) Для успешного выполнения домашних заданий иногда придется расширять эти знания и искать ответы в интернете (практикующие специалисты так делают постоянно);
4) Выигрышная стратегия:
    a) поискать самостоятельно;
    b) не получилось – спросить в группе;
    c) опять не получилось – спросить у преподавателя.

Как общаться в slack и задавать вопросы?

1) Отвечайте на сообщения не отдельными сообщениями, а в тредах, так проще следить за беседой. А новые вопросы задавайте отдельными сообщениями, так будет потом проще искать ответы.
2) Задавайте вопросы в общем чате, а не в личку. Так больше людей сможет вам помочь, а ответы могут помочь другим.
3) Чем конкретнее сформулированы вопросы, тем проще на них ответить. Формулируйте вопрос по алгоритму:
    a) что я хотел получить?
    b) что я для этого сделал?
    c) чем результат отличается от моих ожиданий?
4) Всегда к вопросу прилагайте код (ссылкой на github, либо сниппетом).

Настройка среды и основы Git

Вопросы?

Олег Булыгин

Соцсеть
fb.com/obulygin91

Почта
obulygin91@ya.ru