**Домашнее задание к лекции «Внедрение системы контроля версий»**

**Задача №1 - Демонстрация**

**Легенда**

Ребята из NeuroStartUp не используют системы контроля версий и достаточно скептически к ним относятся. Вам нужно продемонстрировать преимущества использования системы контроля версий и сервиса GitHub.com

Для этого вы должны сделать демо-проект и продемонстрировать преимущества:

* хранение истории изменений;
* видимость и авторство каждого изменения;
* «резервная копия» проекта.

**Входные данные**

*Описание проекта NeuroStartUp*:

NeuroStartUp — динамически развивающийся стартап, специализирующийся на поиске с использованием новейших технологий искусственного интеллекта. Наши преимущества:

* Высокая точность поиска
* Высокая скорость поиска
* Низкая цена

*Логотип*:

[](https://camo.githubusercontent.com/79ee96a8b8fa098c44d1ca302006f24d008408a1c22fc13260437214d705a23d/68747470733a2f2f6e65746f6c6f67792d636f64652e6769746875622e696f2f6769742d686f6d65776f726b732f696e74726f64756374696f6e2f6173736574732f6c6f676f2e706e67)

**Задача**

1. Создайте репозиторий на GitHub, склонируйте его на локальную машину. Перейдите в папку, появившуюся после клонирования командой cd имя-папки. **Переходить в папку репозитория нужно после каждого клонирования.**
2. При помощи текстового редактора в папке репозитория на локальном компьютере создайте файл README.md и добавьте в него описание в формате Markdown. *Можете скопировать разметку из подсказки и вставить её в файл README.md.* У файла README.md не должно быть никаких других расширений. Например, имя файла README.md.txt будет считаться ошибкой.
3. Сохраните файл. При помощи команды git status проверьте, видит ли Git новый файл с названием README.md.
4. Сделайте коммит изменений;
5. Запушьте все изменения на GitHub;
6. Убедитесь, что через веб-интерфейс GitHub можно посмотреть список файлов, список коммитов, изменения, сделанные каждым коммитом.

**В качестве результата пришлите проверяющему:**

1. Ссылку на ваш репозиторий на GitHub;
2. Ссылку на последний коммит. Для этого на сайте GitHub нажмите на количество коммитов (под кнопкой Code), нажмите на последний коммит и скопируйте адрес из адресной строки.

Подсказка: Markdown-разметка для файла README.md

**Задача №2 - Импорт существующего проекта**

**Легенда**

Поздравляем, вы убедили команду использовать Git и GitHub.com и они предоставили вам исходники своего текущего лендинга, но в виде zip-архива. Вам нужно перенести их в систему контроля версий Git и опубликовать на GitHub.com. Обратите внимание, что в архиве есть мусорные файлы и системные файлы (папка tmp, файлы, заканчивающиеся суффиксами \_old, \_backup, Thumbs.db и .DS\_Store) и их нужно проигнорировать, так как в противном случае получится, что вы храните ненужные файлы.

**Задача**

1. Скачайте [по ссылке](https://github.com/netology-code/git-homeworks/raw/master/introduction/src/neuro-startup.zip) архив с проектом.
2. Распакуйте из него папку 1.2. Site For Import.
   * В архиве также лежит скрытая папка \_\_MACOSX, её распаковывать не нужно. Вы можете удалить эту папку с вашего компьютера.
3. Создайте на базе папки 1.2. Site For Import Git-репозиторий.
4. Проигнорируйте файлы, находящиеся в подкаталоге tmp и файлы, заканчивающиеся суффиксами \_old, \_backup. Любые символы можно заменить символом *звёздочка* \*.
5. Добейтесь чтобы также были проигнорированы файлы Thumbs.db и .DS\_Store.
6. При помощи команды git status убедитесь, что Git не видит ненужные файлы.
7. Сделайте коммит.
8. Создайте отдельный репозиторий на GitHub'е.
9. Свяжите ваш локальный репозиторий с только что созданным удалённым репозиторием. Используйте в качестве *кодового имени* удалённого репозитория слово origin.
10. Отправьте сделанные вами изменения на GitHub.

**В качестве результата пришлите проверяющему ссылку на ваш репозиторий на GitHub**

**Задача №3 - Откат изменений**

**Легенда**

Команда из NeuroStartUp начала работать над сайтом в Git, но случилось непредвиденное: один из разработчиков допустил ряд ошибок и залил всё вместе с ошибками в репозиторий на GitHub. Вам нужно отменить его последний коммит и исправить ситуацию.

**Задача**

1. Склонируйте Git-репозиторий [по ссылке](https://github.com/netology-code/git-homeworks-neuro);
2. Отмените один последний коммит в проекте при помощи команды git revert;
3. Создайте отдельный репозиторий на GitHub'е;
4. Свяжите ваш локальный репозиторий с только что созданным удалённым репозиторием. При связывании используйте *кодовое имя* target поскольку стандартное *кодовое имя* origin уже занято.
5. Отправьте сделанные вами изменения на GitHub.

*Примечание*\*: поскольку вы использовали операцию git clone в первом пункте, то в вашем репозитории уже существует remote origin. При попытке добавления второго такого, будет ошибка. Поэтому вместо origin пишем target и push тоже делаем в target. Посмотреть все связи локального репозитория можно командой git remote -v.

**В качестве результата пришлите проверяющему ссылку на ваш репозиторий на GitHub.**

**Пожалуйста, не прикладывайте никакие файлы**

Все задачи обязательны к выполнению. Присылать на проверку можно только сразу все три задачи.

Любые вопросы по решению задач задавайте в чате учебной группы.