

Консультация по Git и GitHub

26 января 2021 г.

Перед консультацией.

1. Установите Git на ваш компьютер.

Win установите Git for Windows Setup с [официального сайта](#). В окне «Adjusting the name of the initial branch in the new repositories» выберите «Let Git decide». В окне «Choose the default behavior of git pull» выберите «Default». В окне «Choose a credential helper» выберите «Git Credential Manager Core». В окне «Configuring experimental options» не выбирайте ничего. Проверьте, что при запуске программы Git Bash открывается тёмное окно с разноцветными надписями.

Mac откройте приложение «Терминал». Введите любую команду, связанную с Git, например, `git status`. Откроется системное окно, в котором будет предложено установить дополнительные инструменты разработчика XCode. Согласитесь и установите их.

NB!: сам XCode устанавливать не требуется, достаточно установить только дополнительные инструменты разработчика, как описано выше.

2. Зарегистрируйтесь на [GitHub](#).

Создание и управление репозиторием.

1. Команда `git init` позволяет создавать новые репозитории. Создайте папку `git_cons` в домашней папке и локальный репозиторий в ней.

NB!: для перехода между папками в терминале можно использовать команду `cd`.

NB!: кнопка TAB позволяет использовать автодополнение.

2. Создайте в репозитории текстовый файл `test.txt` с двумя (любыми) строками.

3. Команды:

- `git add <filename>` добавляет изменения в staging area.
- `git commit -m '<message>'` создаёт новый commit в репозитории.

Сделайте commit в репозиторий. Введите какое-нибудь осознанное сообщение.

NB!: `.` означает «все файлы».

4. Команда `git status` позволяет узнать, что происходит в репозитории. Используйте эту команду и прокомментируйте выдачу. Внесите какие-нибудь изменения в текстовый файл. Что происходит в репозитории теперь? Сделайте commit с вашими изменениями.
5. Команда `git help <command>` позволяет получить справку о команде. С помощью неё выясните, как создать новую ветку командой `git checkout`.

6. Выясните, в какой ветке вы находитесь при помощи команды `git branch`. Создайте новую ветку `git_test` при помощи команды из предыдущего пункта.
7. Выясните, в какой ветке вы находитесь. Внесите в текстовый файл какие-нибудь изменения и сделайте `commit` в ветке `git_test`. Убедитесь, что файл в основной ветке не был изменён.

Взаимодействие с GitHub.

1. Создайте новый удалённый публичный репозиторий на GitHub и назовите его `git_cons`. При помощи команды:
`git remote add origin <path>` ,
добавьте удалённый репозиторий для вашего локального репозитория.
NB!: название удалённого репозитория можно посмотреть при помощи команды `git remote`. В данном случае мы назвали его `origin`.
2. Перейдите на основную ветку (далее будем работать только с ней). Отправьте (`push`) локальные изменения на удалённый репозиторий при помощи команды `git push origin <main branch>`. Проверьте, что изменения появились на GitHub. Отправьте локальные изменения ветки `git_test`.
3. Измените текстовый файл вручную на GitHub. Синхронизируйте изменения на локальный компьютер при помощи команды `git pull origin <branch>`. Проверьте, что локальный файл изменился.
4. Теперь измените файл и на компьютере, и на GitHub так, чтобы изменения были противоречивыми. Сделайте `commit` в обоих местах. Используйте команду `git pull origin <branch>`. Устраните возникший конфликт.
5. Клонировать [репозиторий нашей группы](#) на локальный компьютер при помощи команды `git clone <remote>`.
6. Сделайте `fork` репозитория нашей группы и клонируйте его на компьютер. Локально добавьте в него какой-нибудь файл и сделайте `commit` и `push`.
7. Создайте `pull request` для внесения ваших изменений в репозиторий нашей группы.

Полезные ссылки.

1. [Документация Git](#).
2. [Команды с описанием](#).
3. [GitHub Desktop](#).