PHASE 1

WEEK 1

DAY 5

ELBRUS/BOOTCAMP

План

1. Git, GitHub
2. Git commands

Git

распределённая система контроля версий

Что такое Git?

Git – технология. Распределённая система управления версиями файлов. Сайт: https://git-scm.com/

GitHub – сервис Microsoft, который позволяет публиковать git-проекты. Сайт: https://github.com/

GitLab, **Bitbucket** – сервисы наподобие GitHub для работы с git-проектами.

Git-репозиторий

Репозиторий – копия проекта.

- *локальный* - копия на вашем компьютере
- *удалённый* - копия на удалённом сервере. Например, в GitHub.
    - У локального репозитория может быть 0 или больше удалённых репозиториев (remote).
    - Обычно настраивается 1 удалённый репозиторий и называется “origin”.
    - Посмотреть список удалённых репозиториев можно командой: `git remote`

Создать репозиторий локально можно командой: `git init`

Создание Git-репозитория в GitHub

Способ 1 - новый репозиторий:

- задать имя, приватность
- можно сразу добавить файл .gitignore через интерфейс GitHub

Способ 2 - форк репозитория:

- найти нужный репозиторий, сделать форк
  (настройки приватности останутся те же, что у исходного репозитория)
- склонировать репозиторий на компьютер используя команду: git clone <url>

Способ 3 - импорт репозитория:

- можно менять настройки приватности относительно исходного репозитория

Файл .gitignore

Список файлов/папок, игнорируемых в Git.

node_modules - пример папки, которую необходимо игнорировать всегда.

Создавайте файл .gitignore сразу же после создания репозитория.

- вручную
- командой `npx create-gitignore Node`
- через интерфейс GitHub при создании репозитория

Invite a collaborator

Для добавления коллег в ваш репозиторий нужно:

- зайти в Settings > Manage access > Invite a collaborator
- указать нужный вам ник или почту коллеги
- настроить права для коллеги (Read / Write / Admin)
- приглашённый в проект проверяет почту

Git-ветки (branches)

Ветки используют для разделения разрабатываемого функционала в проекте и его тестирования.

- main (ранее master) дефолтная ветка в Git

Виды веток:

- локальная ветка – находится в вашем локальном репозитории
- удалённая ветка – находится в удалённом репозитории

Commands

Основные команды

- `git init` – создание гит-репозитория в текущей директории
- `git status` – отображает текущее состояние локального репозитория
- `git add -A` – добавляет все файлы в отслеживание
- `git commit –m "commit text"` – коммит с сообщением
- `git remote add origin <url>` – привязка локального репозитория к удалённому, где **url** адрес репозитория
- `git reset HEAD~1` – удаляет последний коммит **локально** и возвращает все изменения в рабочую область (число указывает количество коммитов которые нужно откатить)

Ветки

- `git branch` – список имеющихся у вас веток
- `git branch <name branch>` – создаёт локальную ветку
- `git branch -d <name branch>` – удаляет локальную ветку, если её содержимое влито в основную. Если использовать флаг -D (большая буква), ветка удаляется принудительно
- `git branch -M <name branch>` – переименовывает локальную ветку
- `git checkout <name branch>` – переключение на заданную ветку
- `git checkout -b <name branch>` – создание и переключение в новую ветку

Слияние веток

`git merge <target branch>` — вливает изменения из текущей ветки в заданную.

Используйте для добавления изменений из личных веток в публичные.

Настройки

- `git config --global user.name` – вывести текущее имя пользователя
- `git config --global user.name "new name"` – задать новое имя
- `git config --global user.email` – вывести текущий email пользователя
- `git config –global user.email "new@mail.ru"` – задать новый email