

ВВЕДЕНИЕ В GIT

Всем программистам и системным администраторам необходимо уметь работать с git в командной строке. GIT – децентрализованная система контроля версий файлов. Каждая копия (клон) репозитория является независимой от всех прочих и может обмениваться с другими репозиториями историей изменений файлов.

GIT создал Линус Торвалдс для координации работы над ядром Linux. GitHub, GitLab, BitBucket и т.д. это веб-интерфейсы, дополнительные надстройки и платные сервисы для более удобной работы с git в команде.

Клонирование репозитория:

```
git clone https://gitlab.com/s.sinita/web1.git
```

Вместо s.sinita подставить логин на гитлабе, вместо web1 – название проекта. Создается каталог web1 с полной копией всех файлов и историей всех коммитов. В каталоге .git хранится сам репозиторий, остальное – файлы рабочей копии проекта.

Все дальнейшие команды надо выполнять в каталоге проекта (web1).

Репозиторий «помнит» откуда он клонирован и git может «подтянуть новые коммиты» (загрузить изменения) или отправить сделанные в рабочей копии коммиты с помощью команд **git pull** и **git push** соответственно.

Команда **git status** показывает изменения в файлах по сравнению с текущим коммитом в текущей ветке.

По умолчанию текущая ветка master. Рекомендуется делать всю работу по учебным задачам в ней.

Если в рабочей копии изменить файлы, то для фиксации изменений в git необходимо добавить измененные файлы в коммит:

```
git add *
```

Создать коммит с этими изменениями и отправить его в репозиторий GitLab:

```
git commit -m "Комментарий"  
git push
```

Команда **git log** покажет историю коммитов. Каждый коммит имеет уникальный идентификатор (хеш).

Команда **git revert хеш_комита** откатит изменения коммита и создаст новый коммит.

Более полная инструкция к использованию git совместно с gitlab:
<https://docs.gitlab.com/ee/gitlab-basics/start-using-git.html>