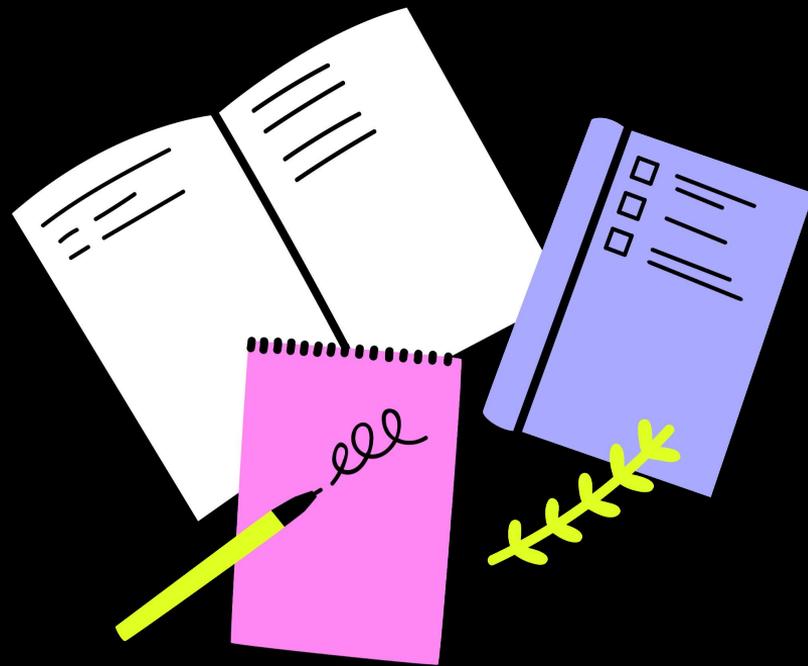




Введение в контроль версий

Семинар 2

Работа с ветками в Git.



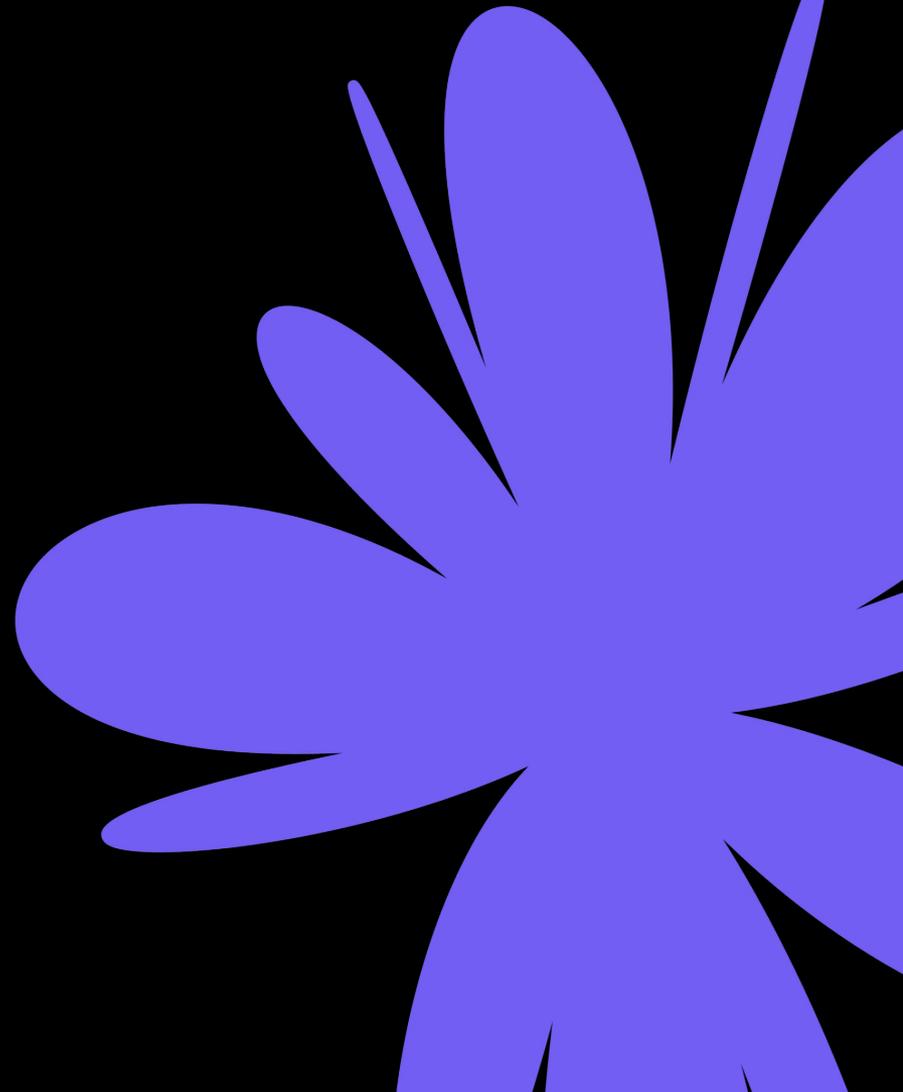
Что будет сегодня

-  Kahoot!
Ознакомительная интерактивная викторина
-  Составление инструкции по работе с Git, используя ветвление.
Продолжение.
-  Домашнее задание





Quiz!



Как вывести список всех имеющихся веток на экран?

1. `git branches`
2. `git branch`
3. `git branch all`
4. `git print branch`



Как вывести список всех имеющихся веток на экран?

1. `git branches`
2. `git branch`
3. `git branch all`
4. `git print branch`



Как создать новую ветку?

1. `git clone`
2. `git newbranch branch_name`
3. `git create branch`
4. `git branch branch_name`



Как создать новую ветку?

1. `git clone`
2. `git newbranch branch_name`
3. `git create branch`
4. `git branch branch_name`



Как перейти на другую ветку?

1. `git pull branch_name`
2. `git checkout branch_name`
3. `git push branch_name`
4. `git branch branch_name`



Как перейти на другую ветку?

1. `git pull branch_name`
2. `git checkout branch_name`
3. `git push branch_name`
4. `git branch branch_name`



Какая команда позволяет слить ветки?

1. `git clone branch_name`
2. `git push branch_name`
3. `git merge branch_name`
4. `git branch branch_name`



Какая команда позволяет слить ветки?

1. `git clone branch_name`
2. `git push branch_name`
3. `git merge branch_name`
4. `git branch branch_name`



Как работает git merge lists?

1. В ветку list добавить всю информацию из текущей ветки
2. В текущую ветку добавит информацию из ветки lists



Как работает git merge lists?

1. В ветку list добавить всю информацию из текущей ветки
2. В текущую ветку добавит информацию из ветки lists



Как удалить уже слитую ветку?

1. `git checkout -d branch_name`
2. `git branch -d branch_name`
3. `git merge branch_name`
4. `git branch branch_name`



Как удалить уже слитую ветку?

1. `git checkout -d branch_name`
2. `git branch -d branch_name`
3. `git merge branch_name`
4. `git branch branch_name`



Как увидеть лог коммитов с визуализацией между ними?

1. `git log - - graph`
2. `git log`
3. `git log - - connections`
4. `git log - - graf`



Как увидеть лог коммитов с визуализацией между ними?

1. `git log - - graph`
2. `git log`
3. `git log - - connections`
4. `git log - - graf`



Когда возникают конфликты при merge?

1. Когда пользователь совершает ошибку
2. Когда в разных ветках одна и та же строка написана по-разному



Когда возникают конфликты при merge?

1. Когда пользователь совершает ошибку
2. Когда в разных ветках одна и та же строка написана по-разному



Введение в контроль версий. Kahoot!

Для присоединения к игре необходимо перейти по ссылке kahoot.it или открыть приложение на смартфоне.

Ввести пин-код игры и своё имя (ник).



Основные команды Git

- ✦ **git init** – инициализация локального репозитория
- ✦ **git status** – получить информацию от git о его текущем состоянии
- ✦ **git add** – добавить файл или файлы к следующему коммиту
- ✦ **git commit -m "message"** – создание коммита.
- ✦ **git log** – вывод на экран истории всех коммитов с их хеш-кодами
- ✦ **git branch** – посмотреть список веток в репозитории
- ✦ **git branch <название ветки>** – создать новую ветку
- ✦ **git checkout <название ветки>** – переход к другой ветке
- ✦ **git branch -d <название ветки>** – удалить ветку



СИНТАКСИС ЯЗЫКА Markdown

Справочник по Markdown от Microsoft:

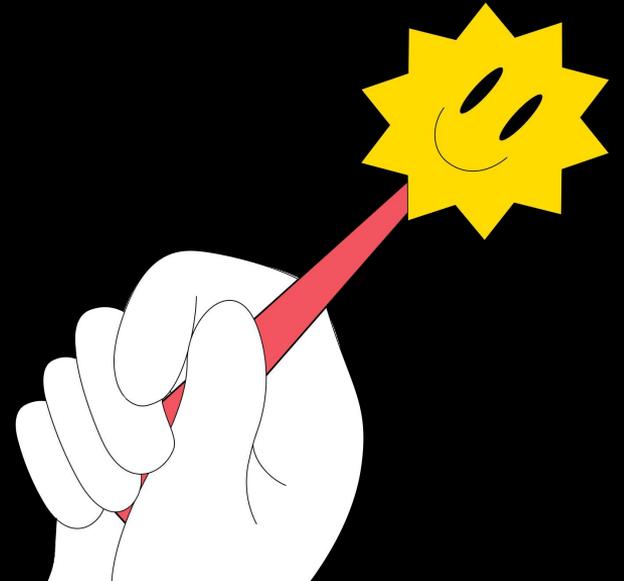
<https://docs.microsoft.com/ru-ru/contribute/markdown-reference>

- ◆ **# Заголовок** – выделение заголовков. Количество символов “#” задаёт уровень заголовка (поддерживается 6 уровней).
- ◆ **= ИЛИ -** – подчёркиванием этими символами (не менее 3 подряд) выделяют заголовки первого (“=”) и второго (“-”) уровней.
- ◆ **** Полужирное начертание**** или **__ Полужирное начертание __**
- ◆ ****Курсивное начертание**** или ***_Курсивное начертание_***
- ◆ ******Полужирное курсивное начертание******
- ◆ **~~Зачёркнутый текст~~**
- ◆ *** Строка** – нумерованные списки, символ “*” в начале строки
- ◆ **1, 2, 3 ...** – нумерованные списки





Домашнее задание



Введение в контроль версий. Домашнее задание.

Продолжить работу с файлом, которую начали на семинаре. Создать и слить как минимум 4 ветки, обязательно создать конфликт и разрешить его. Приложить свой проект в заархивированном виде (всю папку целиком).





Спасибо 
за внимание