

## Plan

# Задачи на урок:

1. Изучить основы работы с Git

## Theory

## Ссылки

- [Git: введение](#)
- [Git: зачем?](#)

## GIT - система контроля версий

- **Git** - вспомогательная программа (инструмент) любого разработчика
- **GitHub** - интернет-площадка (хостинг проектов)
- **GitHub Pages** - сервис GitHub (хостинг статических файлов)

## Необходимые знания для командной работы

1. Базовый порядок работы с Git
2. Механизм ветвления
3. Модели совместной разработки
4. Их особенности и порядок действий

## Виды программных интерфейсов

1. **GUI** (Graphical User Interface) - графический интерфейс пользователя
2. **CLI** (Command Line Interface) - интерфейс командной строки

## Этапы работы

### 1. Подготовительный этап

- (только для Windows) установка GIT на компьютер <https://git-scm.com/downloads>
- настройка Git на компьютере:
  - `git config --global -l` посмотреть текущие настройки
  - `git config --global user.name "Ivanov Ivan"` настройка имени
  - `git config --global user.email "my_email@example.org"` настройка email
- наличие аккаунта на GitHub
- выгрузить публичный ключ на GitHub
  - просмотр публичного ключа (если ошибка, то надо сгенерировать)
    - `cat ~/.ssh/id_ed25519.pub`
    - `cat ~/.ssh/id_rsa.pub`

- `ssh-keygen -t ed25519 -C 'email@example.org'` сгенерировать пару ключей (ВЫПОЛНИТЬ ТОЛЬКО В НАЧАЛЕ КАРЬЕРЫ)
- скопировать публичный ключ на GitHub
- `ssh -T git@github.com` проверить, что ключ выгружен на GitHub

## Проверка текущей версии Git

```
git -v
```

## Основные способы авторизации в Интернет

1. По паролю
2. По **RSA/SSH**-ключам (текстовые файлы)
  - используют в профе. среде
  - позволяет выполнять авторизацию без пароля (прозрачно)
  - подразумевает наличие пары ключей
    - публичный (public) - можно делиться со всеми
    - приватный (private) - храним в секрете

## Базовый (основной) порядок работы с GIT

1. `git init` создать пустой репо (выполняется один раз, для нового проекта)
2. `**git add .**` поставить все файлы проекта на контроль
3. `**git commit -m 'описание изменений'**` выполнить коммит (сохранение)
4. Создание нового репо на GitHub (один проект - один репо)
5. `git remote add origin <ссылка>` выполнить привязку LOCAL ↔ REMOTE (делается один раз для нового проекта)
6. `git push -u origin main` выгрузить на GitHub
  1. `git push` (короткий вариант для следующих коммитов)

### Homework

## Homework

### Задание 1

1. Выполнить несколько коммитов и выгрузить их на GitHub

### Code