

# Самостоятельная работа по Git

## Цель работы:

Понять основные принципы работы с системой контроля версий Git и закрепить навыки использования основных команд.

---

## Задание 1: Установка и настройка Git

1. Установите Git на свой компьютер.
2. Настройте имя пользователя и email:

```
git config --global user.name "Ваше Имя"  
git config --global user.email "ваш.email@example.com"
```

3. Проверьте настройки командой:

```
git config --list
```

4. Напишите в отчёте команды, которые вы использовали, и результат их выполнения.
- 

## Задание 2: Работа с локальным репозиторием

1. Создайте папку с именем `git-practice` и инициализируйте в ней репозиторий.

```
git init
```

2. Создайте файл `README.md` и напишите в нём краткое описание вашего проекта.
3. Добавьте файл в индекс и создайте коммит:

```
git add README.md  
git commit -m "Добавил файл README.md с описанием проекта"
```

4. Создайте ещё один файл `index.html` с базовой HTML-разметкой. Добавьте и закоммитьте его.
  5. Напишите в отчёте последовательность команд, которые вы использовали.
- 

## Задание 3: Работа с ветками

1. Создайте новую ветку `feature/header` и переключитесь на неё:

```
git checkout -b feature/header
```

2. В этой ветке создайте файл `header.html`, добавьте в него базовую структуру для шапки сайта.
3. Закоммитьте изменения.
4. Вернитесь в основную ветку `main` и объедините ветку `feature/header` с основной веткой:

```
git merge feature/header
```

5. Удалите ветку `feature/header` после слияния:

```
git branch -d feature/header
```

6. Напишите в отчёте, какие команды вы использовали, и покажите вывод команды:

```
git log --oneline
```

---

## Задание 4: Работа с удалённым репозиторием

1. Зарегистрируйтесь или войдите на [GitHub](#).
2. Создайте новый репозиторий с именем `git-practice`.
3. Свяжите локальный репозиторий с удалённым:

```
git remote add origin <URL вашего репозитория>
```

4. Отправьте изменения на GitHub:

```
git push -u origin main
```

5. Добавьте в отчёт ссылку на ваш удалённый репозиторий.
- 

## Задание 5: Решение конфликта

1. Создайте две ветки: `feature/footer` и `feature/content`.
2. В ветке `feature/footer` создайте файл `footer.html` и добавьте базовую структуру футера. Закоммитьте изменения.
3. Переключитесь на ветку `feature/content`, создайте файл `content.html` с базовой структурой контента, но измените файл `footer.html` (например, добавьте в него текст).
4. Закоммитьте изменения в ветке `feature/content`.
5. Вернитесь в основную ветку `main` и поочерёдно слейте обе ветки. Если возникнет конфликт, разрешите его вручную, закоммитьте результат.
6. Напишите в отчёте, как вы решили конфликт, и приложите вывод команды:

```
git log --graph --oneline
```

---

## Дополнительное задание (по желанию)

1. Настройте `.gitignore` и добавьте в него файлы, которые не нужно отслеживать (например, временные файлы или папку `node_modules`).
2. Просмотрите статус:

```
git status
```

3. Закоммитьте изменения в `.gitignore`.
- 

## Требования к отчёту

1. Кратко опишите выполнение каждого задания.
2. Приложите используемые команды и их результаты.
3. Добавьте скриншоты работы в Git и ссылку на ваш удалённый репозиторий.