

lab03

27 April, 2022 Moscow, Russia

Лабораторная работа № 3. Markdown.

Цель работы

Научиться оформлять отчёты с помощью легковесного языка разметки Markdown.

Задача

Сделайте отчёт по предыдущей лабораторной работе в формате Markdown.

Выполнение лабораторной работы

Настройка служебной информации

Код ниже настраивает параметры pdf файла, который в дальнейшем будет получен с помощью программы pandoc:

```
## Front matter
title: "Отчёт по лабораторной работе №3"
subtitle: "Markdown"
author: "Митичкина Екатерина Павловна"
```

```
## Generic options
lang: ru-RU
toc-title: "Содержание"
...
```

Написание цели и задачи

Цель работы

Научиться оформлять отчёты с помощью легковесного языка разметки Markdown.














Задача

Сделайте отчёт по предыдущей лабораторной работе в формате Markdown.

Теоретическое введение:

цель и задачи

Написание теоретического введения

..	
 10.jpg	Add files via upload
 11.jpg	Add files via upload
 12.jpg	Add files via upload
 13.jpg	Add files via upload
 14.jpg	Add files via upload
 2.jpg	Add files via upload
 3.jpg	Add files via upload
 4.jpg	Add files via upload
 5.jpg	Add files via upload
 6.jpg	Add files via upload
 7.jpg	Add files via upload
 8.jpg	Add files via upload
 9.jpg	Add files via upload

теоретическое введение

Добавление папки со скриншотами

сделайте отчет по предыдущей лабораторной работе в формате markdown.

Теоретическое введение:

Базовые сведения о Markdown

Чтобы создать заголовок, используйте знак (#), например:

```

# This is heading 1

## This is heading 2

### This is heading 3

#### This is heading 4

```

Чтобы задать для текста полужирное начертание, заключите его в двойные звездочки:

This text is **``**bold**``**.

Чтобы задать для текста курсивное начертание, заключите его в одинарные звездочки:

This text is *``*italic*``*.

Чтобы задать для текста полужирное и курсивное начертание, заключите его в тройные звездочки:

This is text is both ***``***bold``and ``italic***``***.

Блоки цитирования создаются с помощью символа >:

```

> The drought had lasted now for ten million years, and the reign of the terrible lizards had long since ended. Here on the Equator, in the existence had reached a new climax of ferocity, and the victor was not yet in sight. In this barren and desiccated land, only the small or

```

Неупорядоченный (маркированный) список можно отформатировать с помощью звездочек или тире:

```

- List item 1

- List item 2

- List item 3

```

Чтобы вложить один список в другой, добавьте отступ для элементов дочернего списка:

```

- List item 1

- List item A

- List item B

- List item 2

```

Упорядоченный список можно отформатировать с помощью соответствующих цифр:

```

1. First instruction

1. Second instruction

1. Third instruction

скриншоты

## Написание самого отчета

```
Цель работы

- Изучить идеологию и применение средств контроля версий.
- Освоить умения по работе с git.

Задание

- Создать базовую конфигурацию для работы с git.
- Создать ключ SSH.
- Создать ключ PGP.
- Настроить подписи git.
- Зарегистрироваться на Github.
- Создать локальный каталог для выполнения заданий попредмету.

Теоретическое введение

Системы контроля версий (Version Control System, VCS) применяются при работе несколь- ких человек над одним проектом. Обычно основное дерев-
которому настроен доступ для участников проекта. При внесении изменений в содержание проекта система контроля версий позволяет их фиксирова-
проекта, производить откат к любой более ранней версии проекта, если это требуется.
В классических системах контроля версий используется централизованная модель, предполагающая наличие единого репозитория для хранения файло-
осуществляется специальным сервером. Участник проекта (пользователь) перед началом работы посредством определённых команд получает нужную е-
размещает новую версию в хранилище. При этом предыдущие версии не удаляются из центрального хранилища и к ним можно вернуться в любой момен-
производить так называемую дельта- компрессию – сохранять только изменения между последовательными версиями, что позволяет уменьшить объём.
Системы контроля версий поддерживают возможность отслеживания и разрешения конфликтов, которые могут возникнуть при работе нескольких чело-
сделанные разными участниками (авто- матически или вручную), вручную выбрать нужную версию, отменить изменения вовсе или заблокировать файл-
позволяет другим пользователям получить рабочую копию или препятствует изменению рабочей копии файла средствами файловой системы ОС, обеспе-
пользователю, работающему с файлом.
Системы контроля версий также могут обеспечивать дополнительные, более гибкие функциональные возможности. Например, они могут поддерживать
историю изменений до точки ветвления версий и собственные истории изменений каждой ветви. Кроме того, обычно доступна информация о том, кто
рода информация хранится в журнале изменений, доступ к которому можно ограничить.
В отличие от классических, в распределённых системах контроля версий центральный репозиторий не является обязательным.
Среди классических VCS наиболее известны CVS, Subversion, а среди распределён- ных – Git, Bazaar, Mercurial. Принципы их работы схожи, отли-
```

```
Выполнение лабораторной работы

Так как у меня уже была учетная запись на github, я пропустила пункт настройки github.
Далее перешла к установке программного обеспечения.
1) Установка git-flow в Fedora Linux
Для этого я запустила код
![[Установка git-flow]](https://github.com/ate4/study_2022-2023_os-intro/blob/b06844329a79c4b5c69ebeda4fe47c74cca0578/labs/lab03/report/im-
%D0%A1%D0%BD%D0%B8%D0%BC%D0%BE%D0%BA%D1%8D%D0%BA%D1%80%D0%B0%D0%BD%D0%B0%202022-04-26%D0%B2%2020.15.30.png)
```

отчет лаб 2

## Вывод

В результате работы научилась оформлять отчёты с помощью легковесного языка разметки Markdown.