

1.1

```
git config --global user.name "Ivan "
```

```
git config --global user.email "ivan456754324567654324567@whatever.com"
```

1.2

```
git config --global init.defaultBranch main
```

1.3

```
git config --global core.autocrlf true
```

```
git config --global core.safecrlf warn
```

2.1

```
mkdir work
```

```
cd work
```

```
New-Item -Path . -Name "hello.html" -ItemType "file"
```

2.2

```
git init
```

2.3

```
git add hello.html
```

```
git commit -m "Initial Commit"
```

3.1

```
git status
```

5

```
git add
```

```
git checkout
```

6

```
git add a.html
```

```
git add b.html
```

```
git commit -m "Changes for a and b"
```

```
git add c.html
```

```
git commit -m "Unrelated change to c"
```

7

```
git commit
```

```
Added h1 tag
```

```
Esc :wq enter
```

8

```
git add .
```

9

`git log`

`git log --pretty=oneline`

`git log --all --pretty=format:"%h %cd %s (%an)" --since="7 days ago"`

`git log --pretty=format:"%h %ad | %s%d [%an]" --date=short`

- `--pretty="..."` — определяет формат вывода.
- `%h` — укороченный хеш коммита.
- `%ad` — дата коммита.
- `|` — просто визуальный разделитель.
- `%s` — комментарий.
- `%d` — дополнения коммита («головы» веток или теги).
- `%an` — имя автора.
- `--date=short` — сохраняет формат даты коротким и симпатичным.

Сохранение формата лога

`git config --global format.pretty '%h %ad | %s%d [%an]'`

`git config --global log.date short`

10

`git checkout <hash>` меняет текущую точку

`cat hello.html` показывает содержимое

`git switch main` вернуться к последней точке в main

11

`git tag v1` тег для текущей версии

`git checkout v1^` Переключение на предшествующий коммит

`git tag` просмотр тегов

13

`git reset HEAD hello.html` сбрасывает текущие add

14

`git revert HEAD` отменяет последний коммит либо коммит по хешу

15

`git reset --hard v1` сброс к предшествующему коммиту

`git log --all` показывает все коммиты

16

`git tag -d oops` удаление тега

17

`git commit --amend -m "Added copyright statement with email"` изменение предыдущего

КОММИТА

18

git switch -c style создание новой ветки

19 git switch main переключение между ветками

20

git log hello.html просмотр изменений в конкретном файле

git log style.css

git show v1 просмотр изменений в коммите

mv hello.html index.html переименование файла

git add . добавление в коммит

mkdir css

git mv style.css css/style.css перемещение файла безопасным способом

git log --follow css/style.css лог файла с историей до переименования

21

git add README.txt добавление файла в ветку

22

git log --all --graph просмотр веток и их расхождений

23

git switch style

git merge main слияние

25

git merge --abort отмена слияния

27

git reset --hard HEAD~2 сброс до коммита который находится за два коммита до head

30

git clone work home клонирование репозитория

ls просмотр репозитория

31

git remote получение имени удаленного репозитория

git remote show origin более подробная информация

32

git branch просмотр веток доступных в клонированном репозитории

git branch -a список удаленных веток

35

git fetch подтягивание коммитов из удаленного репозитория

37

git branch --track style origin/style добавление локальной ветки, которая отслеживает локальную ветку

38

git clone --bare work work.git создание чистого репозитория

39

git remote add shared ../work.git добавление чистого репозитория как удаленного к оригинальному репозиторию

40

git push shared main отправить изменения в общий репозиторий