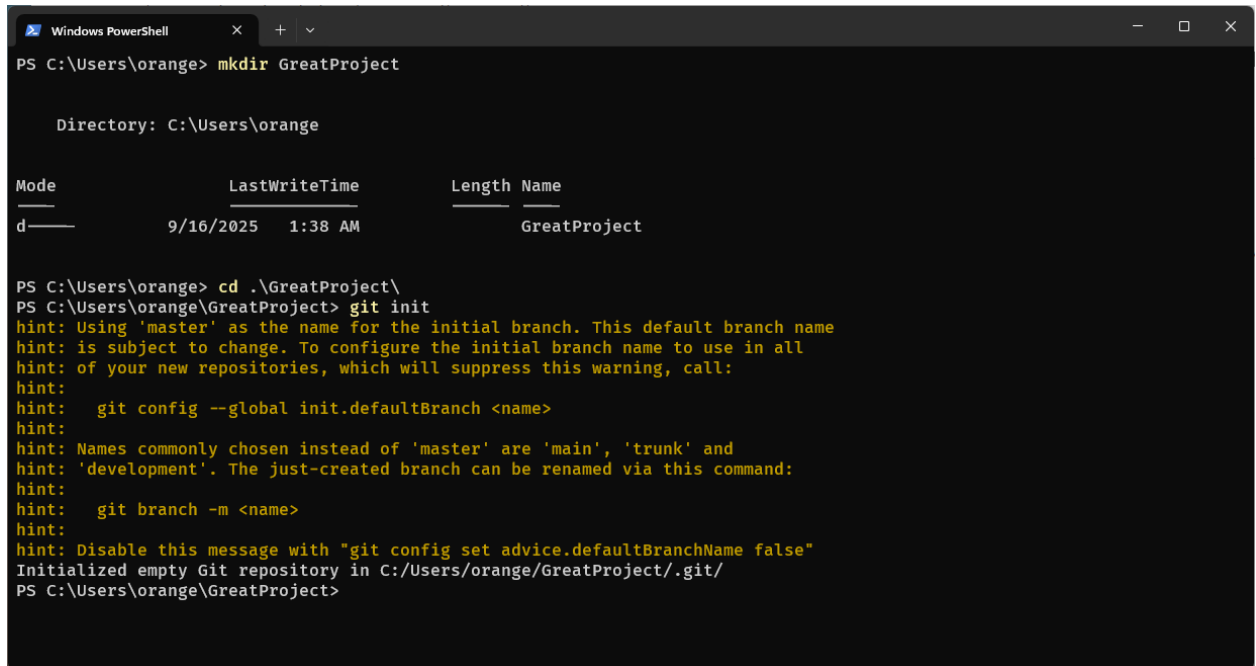


# Возможности утилиты командной строки git для работы с системой контроля версий Git

## 1. Создание репозитория

Чтобы создать Git репозиторий с помощью утилиты командной строки git, нужно перейти в директорию, в которой нужно создать репозиторий и выполнить команду **“git init”**.



```
Windows PowerShell
PS C:\Users\orange> mkdir GreatProject

Directory: C:\Users\orange

Mode                LastWriteTime         Length Name
----                -
d-----          9/16/2025   1:38 AM             GreatProject

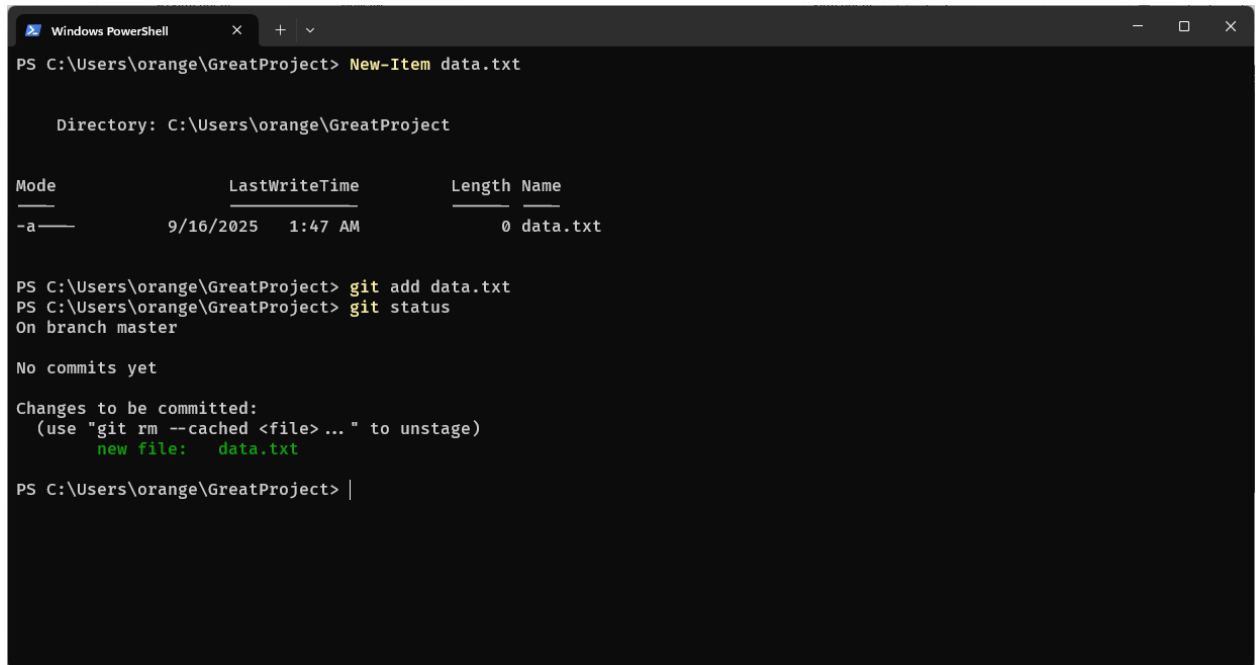
PS C:\Users\orange> cd .\GreatProject\
PS C:\Users\orange\GreatProject> git init
hint: Using 'master' as the name for the initial branch. This default branch name
hint: is subject to change. To configure the initial branch name to use in all
hint: of your new repositories, which will suppress this warning, call:
hint:
hint:   git config --global init.defaultBranch <name>
hint:
hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and
hint: 'development'. The just-created branch can be renamed via this command:
hint:
hint:   git branch -m <name>
hint:
hint: Disable this message with "git config set advice.defaultBranchName false"
Initialized empty Git repository in C:/Users/orange/GreatProject/.git/
PS C:\Users\orange\GreatProject>
```

В данном примере была создана директория “GreatProject”, в которой был инициализирован Git репозиторий.

Примечание: сообщение, которое напечаталось жёлтыми буквами при создании репозитория говорит о том, что имя главной ветки репозитория можно изменить с помощью команды **“git branch -m <имя>”**. А также можно установить имя главной ветки по умолчанию для будущих репозиторииев с помощью команды **“git config – global init.defaultBranch <имя>”**.

## 2. Добавление файлов

Все файлы, которые вы хотите сохранить нужно сперва добавить в индекс, таким образом Git будет о них знать. Сделать это можно с помощью команды “**git add <имя файла/файлов>**”. Если нужно добавить все файлы в данной директории, то можно передать точку в качестве аргумента команды “**git add**”.



```
Windows PowerShell
PS C:\Users\orange\GreatProject> New-Item data.txt

Directory: C:\Users\orange\GreatProject

Mode                LastWriteTime         Length Name
----                -
-a                9/16/2025   1:47 AM             0 data.txt

PS C:\Users\orange\GreatProject> git add data.txt
PS C:\Users\orange\GreatProject> git status
On branch master

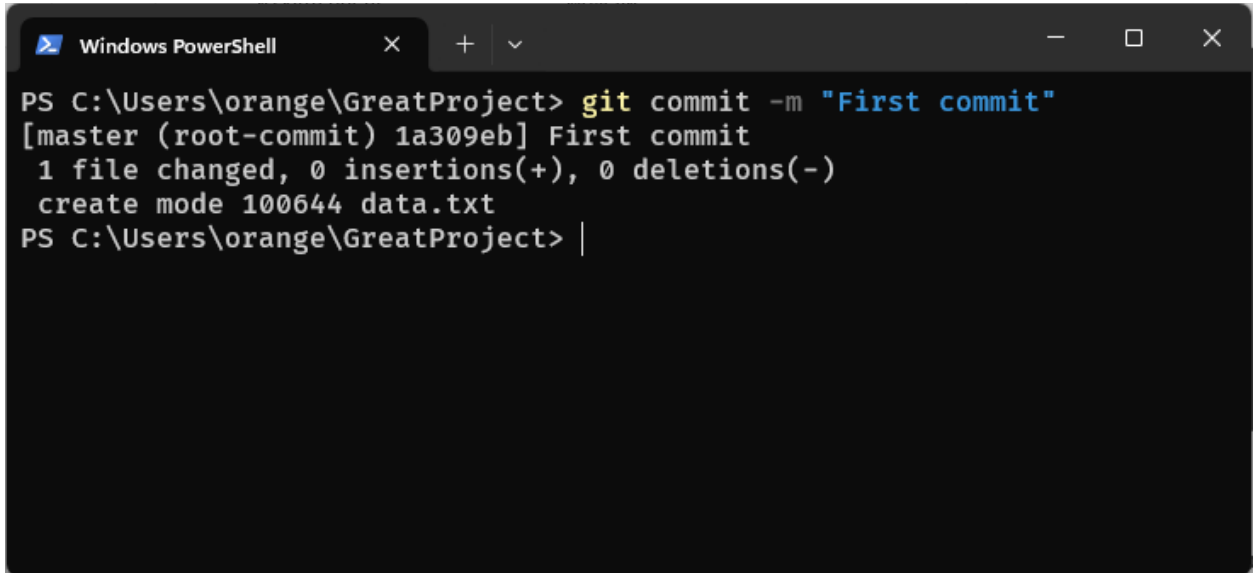
No commits yet

Changes to be committed:
  (use "git rm --cached <file> ..." to unstage)
        new file:   data.txt

PS C:\Users\orange\GreatProject> |
```

### 3. Сохранение файлов

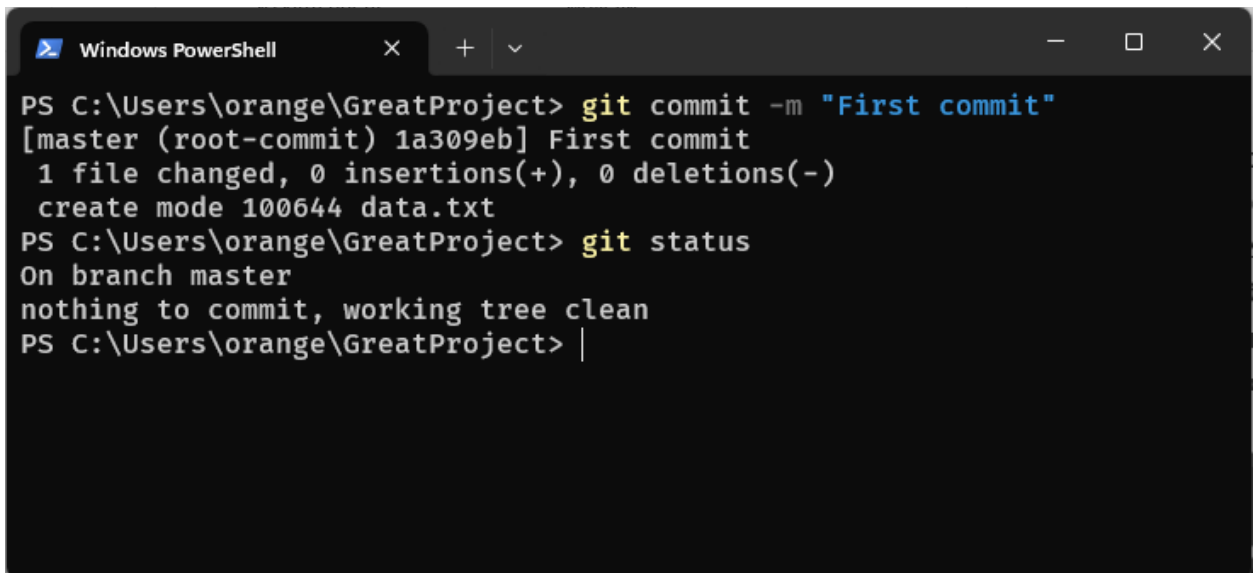
Для сохранения текущего состояния проекта существует команда **"git commit"**. Данная команда создаёт своего рода слепок текущего состояния всех индексированных файлов. У каждого такого слепка обязательно должно быть название, чтобы их можно было различать. При выполнении данной команды откроется консольный текстовый редактор, куда нужно будет ввести название слепка. Для удобства можно сразу передать имя слепка с помощью параметра **"-m"**.



```
Windows PowerShell
PS C:\Users\orange\GreatProject> git commit -m "First commit"
[master (root-commit) 1a309eb] First commit
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 data.txt
PS C:\Users\orange\GreatProject> |
```

### 4. Проверка состояния репозитория

Для получения текущего состояния репозитория существует команда **"git status"**. Она сообщит какие файлы были добавлены в индекс, изменены или удалены. Также она сообщит на какой ветке вы сейчас находитесь.

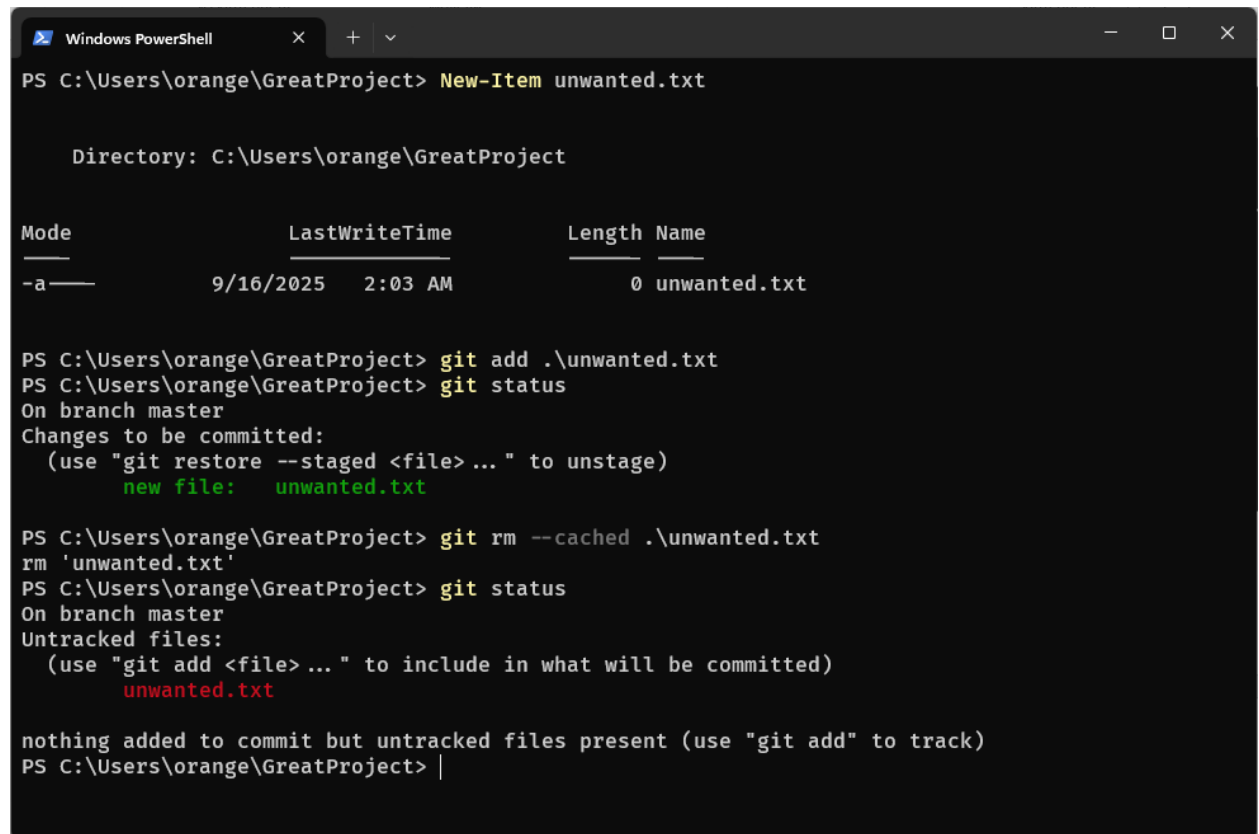


```
Windows PowerShell
PS C:\Users\orange\GreatProject> git commit -m "First commit"
[master (root-commit) 1a309eb] First commit
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 data.txt
PS C:\Users\orange\GreatProject> git status
On branch master
nothing to commit, working tree clean
PS C:\Users\orange\GreatProject> |
```

## 5. Удаление файлов из индекса

Если вы случайно добавили какие-то файлы в индекс и хотите убрать их оттуда, то можно воспользоваться командой **"git rm --cached <имя файла/файлов>".**

Параметр **"--cached"** здесь очень важен, потому что если его не передать, то Git удалит файл как из индекса, так и с диска.



```
Windows PowerShell
PS C:\Users\orange\GreatProject> New-Item unwanted.txt

Directory: C:\Users\orange\GreatProject

Mode                LastWriteTime         Length Name
----                -
-a-----          9/16/2025   2:03 AM             0 unwanted.txt

PS C:\Users\orange\GreatProject> git add .\unwanted.txt
PS C:\Users\orange\GreatProject> git status
On branch master
Changes to be committed:
  (use "git restore --staged <file> ..." to unstage)
        new file:   unwanted.txt

PS C:\Users\orange\GreatProject> git rm --cached .\unwanted.txt
rm 'unwanted.txt'
PS C:\Users\orange\GreatProject> git status
On branch master
Untracked files:
  (use "git add <file> ..." to include in what will be committed)
        unwanted.txt

nothing added to commit but untracked files present (use "git add" to track)
PS C:\Users\orange\GreatProject> |
```

Также существует отдельная команда “**git reset**”, которая отменяет все изменения, добавленные в индекс.

В примере ниже, я как будто бы случайно добавил три файла в индекс, затем я выполнил команду “**git reset**” и в результате, все добавленные файлы убрались из индекса.

```
Windows PowerShell
PS C:\Users\orange\GreatProject> git status
On branch master
Untracked files:
  (use "git add <file> ..." to include in what will be committed)
        unwanted1.txt
        unwanted2.txt
        unwanted3.txt

nothing added to commit but untracked files present (use "git add" to track)
PS C:\Users\orange\GreatProject> git add .
PS C:\Users\orange\GreatProject> git status
On branch master
Changes to be committed:
  (use "git restore --staged <file> ..." to unstage)
        new file:   unwanted1.txt
        new file:   unwanted2.txt
        new file:   unwanted3.txt

PS C:\Users\orange\GreatProject> git reset
PS C:\Users\orange\GreatProject> git status
On branch master
Untracked files:
  (use "git add <file> ..." to include in what will be committed)
        unwanted1.txt
        unwanted2.txt
        unwanted3.txt

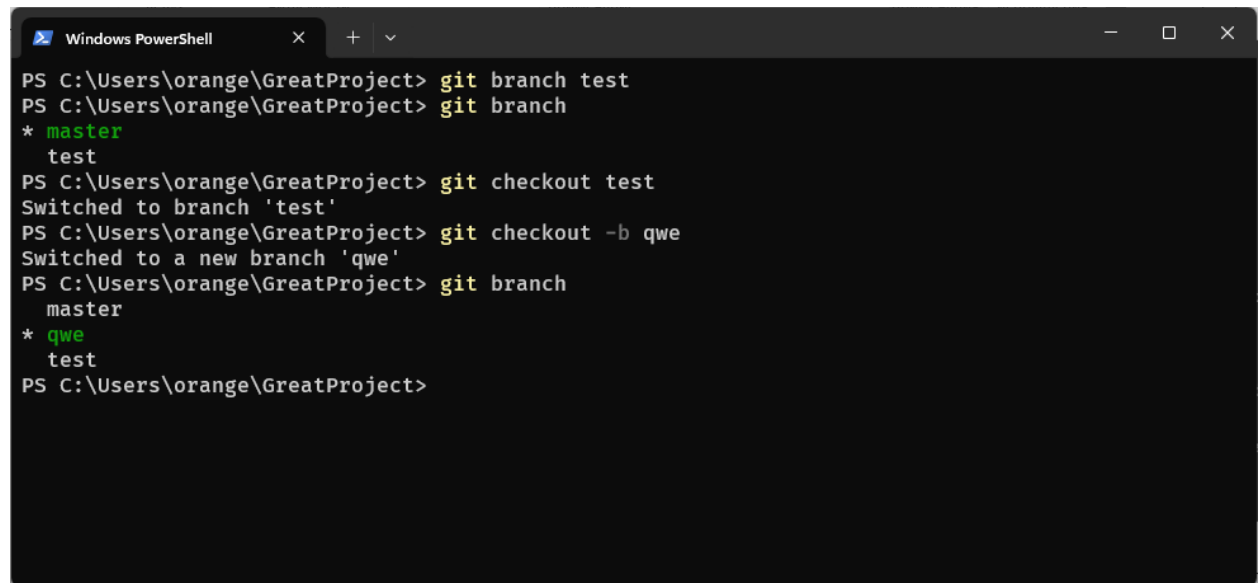
nothing added to commit but untracked files present (use "git add" to track)
PS C:\Users\orange\GreatProject> |
```

## 6. Создание веток

Чтобы создать ответвление от текущей ветки, можно использовать команду **"git branch <имя ветки>"**. Для смены ветки существует команда **"git checkout <имя ветки>"**.

Также, можно выполнить эти два действия сразу с помощью команды **"git checkout -b <имя новой ветки>"**. Данная команда создаст новую ветку и тут же на неё переключится.

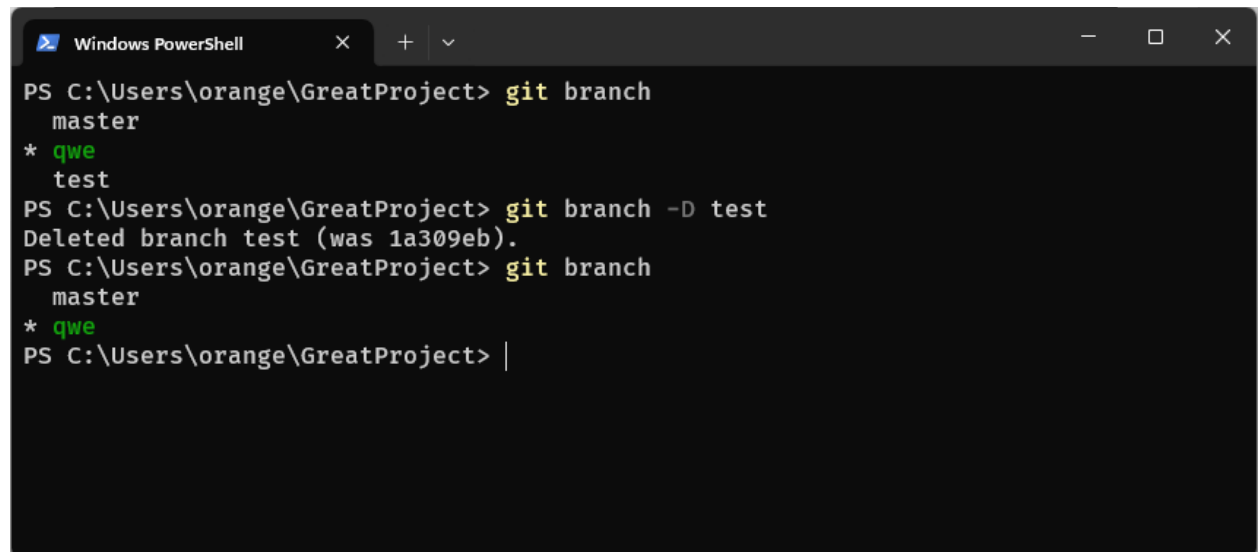
Чтобы посмотреть все существующие ветки, можно выполнить команду **"git branch"**, не передавая никаких аргументов.



```
Windows PowerShell
PS C:\Users\orange\GreatProject> git branch test
PS C:\Users\orange\GreatProject> git branch
* master
  test
PS C:\Users\orange\GreatProject> git checkout test
Switched to branch 'test'
PS C:\Users\orange\GreatProject> git checkout -b qwe
Switched to a new branch 'qwe'
PS C:\Users\orange\GreatProject> git branch
  master
* qwe
  test
PS C:\Users\orange\GreatProject>
```

## 7. Удаление веток

Для удаления ненужных веток существует команда **"git branch -D <имя ветки>"**.

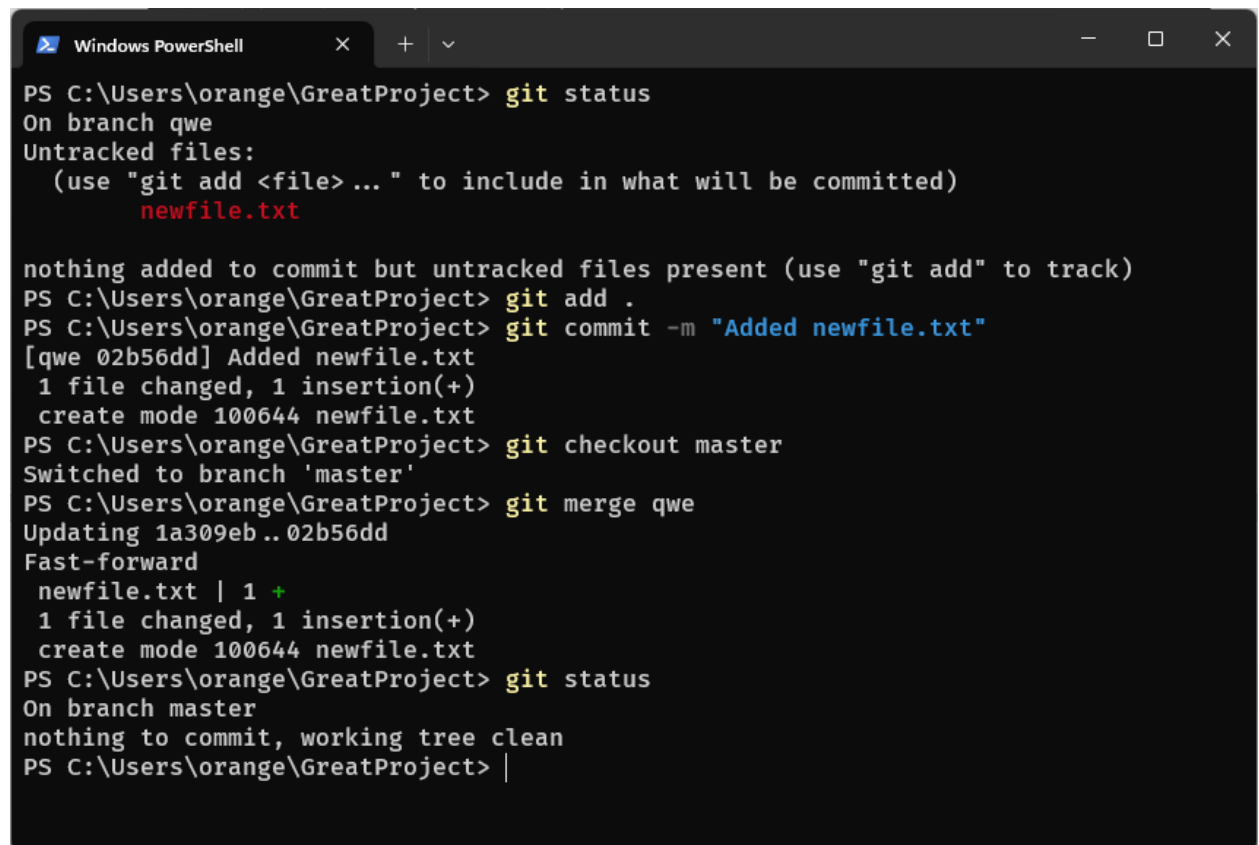


```
Windows PowerShell
PS C:\Users\orange\GreatProject> git branch
  master
* qwe
  test
PS C:\Users\orange\GreatProject> git branch -D test
Deleted branch test (was 1a309eb).
PS C:\Users\orange\GreatProject> git branch
  master
* qwe
PS C:\Users\orange\GreatProject> |
```

## 8. Слияние веток

После того, как работа на какой-то ветке закончена, можно выполнить слияние этой ветки с главной веткой, чтобы включить все изменения, сделанные на другой ветке в главную. Чтобы это сделать, нужно для начала, перейти в ту ветку, в которую вы хотите добавить изменения с помощью команды **"git checkout <имя ветки>"**, а затем выполнить команду **"git merge <имя ветки>"**.

В примере ниже, находясь на ветке **"qwe"**, я создал файл **"newfile.txt"** и закоммитил его. Затем я перешел на главную ветку **"master"** и выполнил слияние с веткой **"qwe"**. Таким образом все коммиты, сделанные на ветке **"qwe"** скопировались в ветку **"master"**.

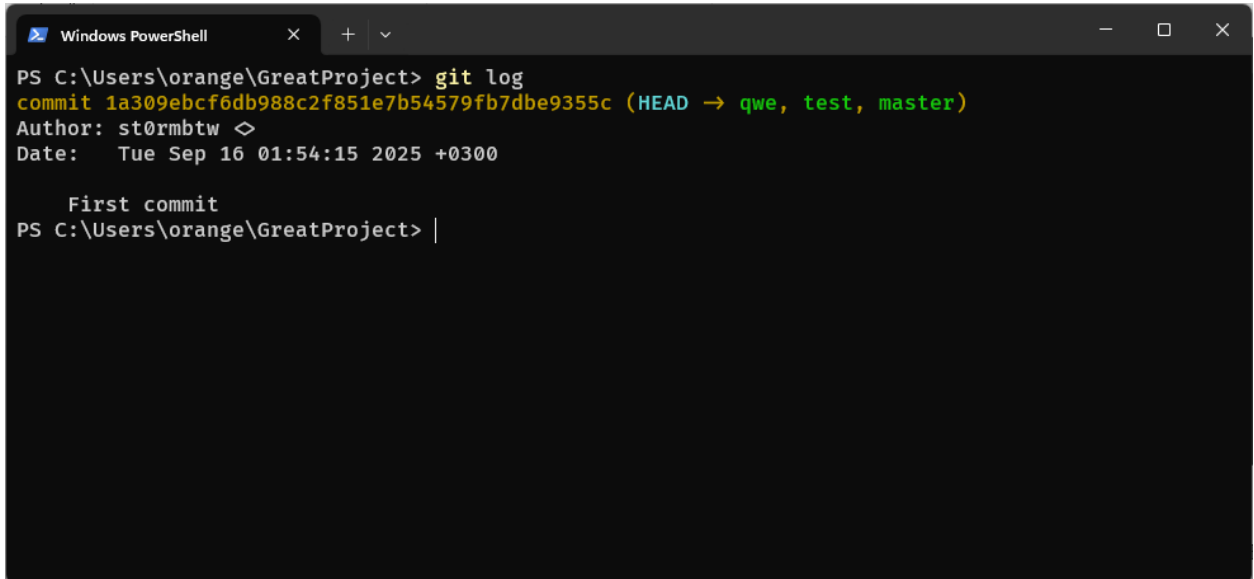


```
Windows PowerShell
PS C:\Users\orange\GreatProject> git status
On branch qwe
Untracked files:
  (use "git add <file> ..." to include in what will be committed)
    newfile.txt

nothing added to commit but untracked files present (use "git add" to track)
PS C:\Users\orange\GreatProject> git add .
PS C:\Users\orange\GreatProject> git commit -m "Added newfile.txt"
[qwe 02b56dd] Added newfile.txt
1 file changed, 1 insertion(+)
create mode 100644 newfile.txt
PS C:\Users\orange\GreatProject> git checkout master
Switched to branch 'master'
PS C:\Users\orange\GreatProject> git merge qwe
Updating 1a309eb..02b56dd
Fast-forward
 newfile.txt | 1 +
1 file changed, 1 insertion(+)
create mode 100644 newfile.txt
PS C:\Users\orange\GreatProject> git status
On branch master
nothing to commit, working tree clean
PS C:\Users\orange\GreatProject> |
```

## 9. Просмотр истории

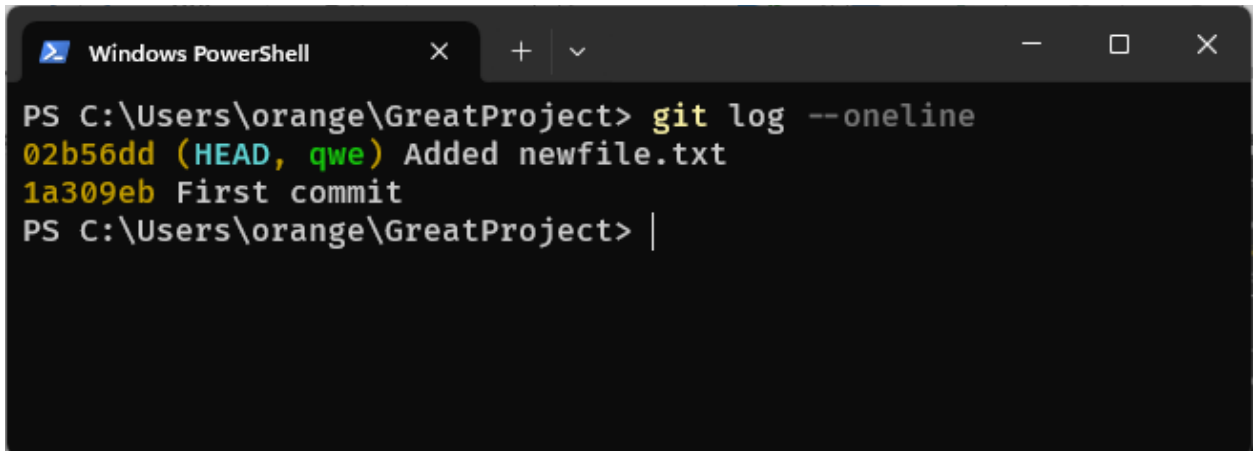
Команда “**git log**” выведет все коммиты на текущей ветке. Вывод данной команды содержит такую информацию как: хеш-номер коммита, его авторы, дата его создания и название.



```
Windows PowerShell
PS C:\Users\orange\GreatProject> git log
commit 1a309ebcf6db988c2f851e7b54579fb7dbe9355c (HEAD -> qwe, test, master)
Author: st0rmbtw <
Date: Tue Sep 16 01:54:15 2025 +0300

    First commit
PS C:\Users\orange\GreatProject> |
```

Если передать параметр “**--oneline**” в данную функцию, то она выведет историю коммитов в укороченном формате, что иногда может быть удобно.



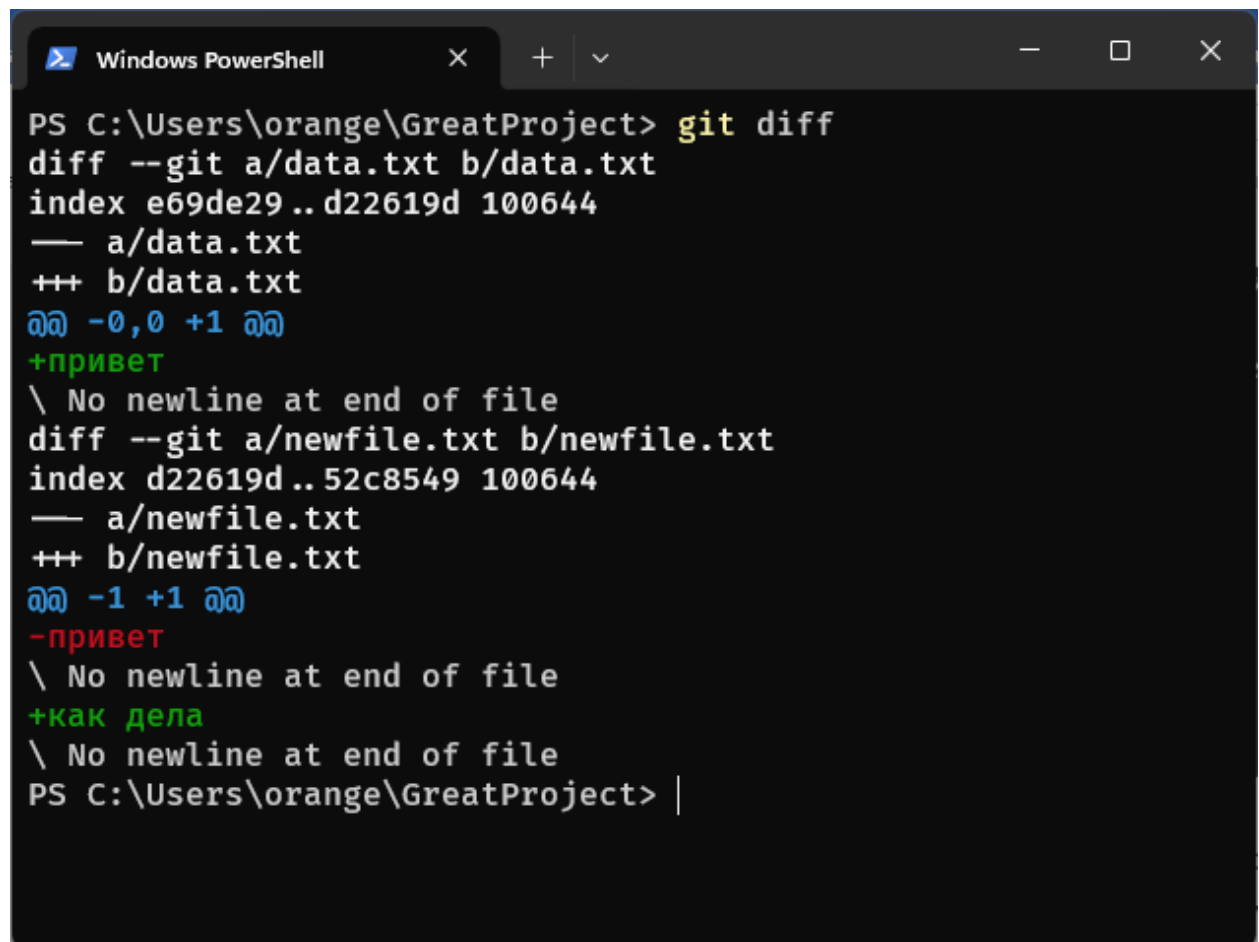
```
Windows PowerShell
PS C:\Users\orange\GreatProject> git log --oneline
02b56dd (HEAD, qwe) Added newfile.txt
1a309eb First commit
PS C:\Users\orange\GreatProject> |
```



## 10. Просмотр сделанных изменений

В утилите командой строки Git есть ещё одна полезная функция, которая позволяет просмотреть все сделанные изменения. Данную функцию позволяет выполнять команда **"git diff"**. Если выполнить эту команду, то в терминале выведется интерактивный список всех текущих незакомиченных изменений.

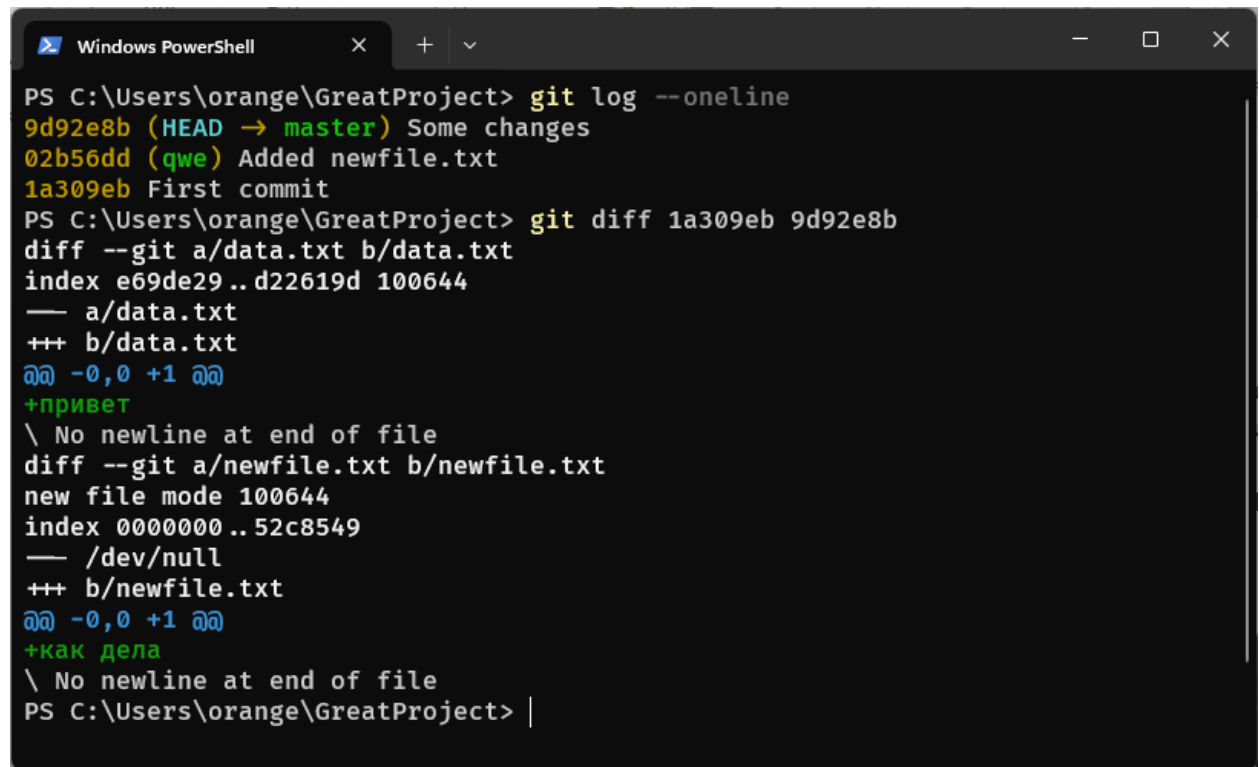
В примере ниже, я добавил в файл **"data.txt"** слово «привет», а файл **"newfile.txt"** я изменил, заменив слово «привет», на фразу «как дела». Команда **"git diff"** вывела эти изменения.



```
PS C:\Users\orange\GreatProject> git diff
diff --git a/data.txt b/data.txt
index e69de29..d22619d 100644
--- a/data.txt
+++ b/data.txt
@@ -0,0 +1 @@
+привет
\ No newline at end of file
diff --git a/newfile.txt b/newfile.txt
index d22619d..52c8549 100644
--- a/newfile.txt
+++ b/newfile.txt
@@ -1,1 @@
-привет
+как дела
\ No newline at end of file
PS C:\Users\orange\GreatProject> |
```

Также, с помощью этой функции можно посмотреть все изменения, сделанные в промежутке между двумя какими-то коммитами, передав первым аргументом хеш-номер начального коммита, а вторым аргументом хеш-номер конечного коммита. Например: “**git diff 1a309eb 9d92e8b**”.

В примере ниже, я посмотрел все изменения между первым и последним коммитом:



```
Windows PowerShell
PS C:\Users\orange\GreatProject> git log --oneline
9d92e8b (HEAD -> master) Some changes
02b56dd (qwe) Added newfile.txt
1a309eb First commit
PS C:\Users\orange\GreatProject> git diff 1a309eb 9d92e8b
diff --git a/data.txt b/data.txt
index e69de29..d22619d 100644
--- a/data.txt
+++ b/data.txt
@@ -0,0 +1 @@
+привет
\ No newline at end of file
diff --git a/newfile.txt b/newfile.txt
new file mode 100644
index 0000000..52c8549
--- /dev/null
+++ b/newfile.txt
@@ -0,0 +1 @@
+как дела
\ No newline at end of file
PS C:\Users\orange\GreatProject> |
```

В Git существует указатель “HEAD”, который указывает на текущий коммит. Для удобства, существует способ обращения к коммитам через относительные указатели. Например, команда для получения разницы между текущим коммитом и пред предыдущем коммитом будет выглядеть так: “**git diff HEAD~2 HEAD**”.

```
Windows PowerShell
PS C:\Users\orange\GreatProject> git diff HEAD~2 HEAD
diff --git a/data.txt b/data.txt
index e69de29..d22619d 100644
--- a/data.txt
+++ b/data.txt
@@ -0,0 +1 @@
+привет
\ No newline at end of file
diff --git a/newfile.txt b/newfile.txt
new file mode 100644
index 0000000..52c8549
--- /dev/null
+++ b/newfile.txt
@@ -0,0 +1 @@
+как дела
\ No newline at end of file
PS C:\Users\orange\GreatProject> |
```