



دانشگاه حکیم سبزواری
دانشکده فنی برق و کامپیوتر
مستند آموزشی

موضوع : Git

نویسنده :

محمد داود وهاب رجائی

تیر 1402

فهرست :

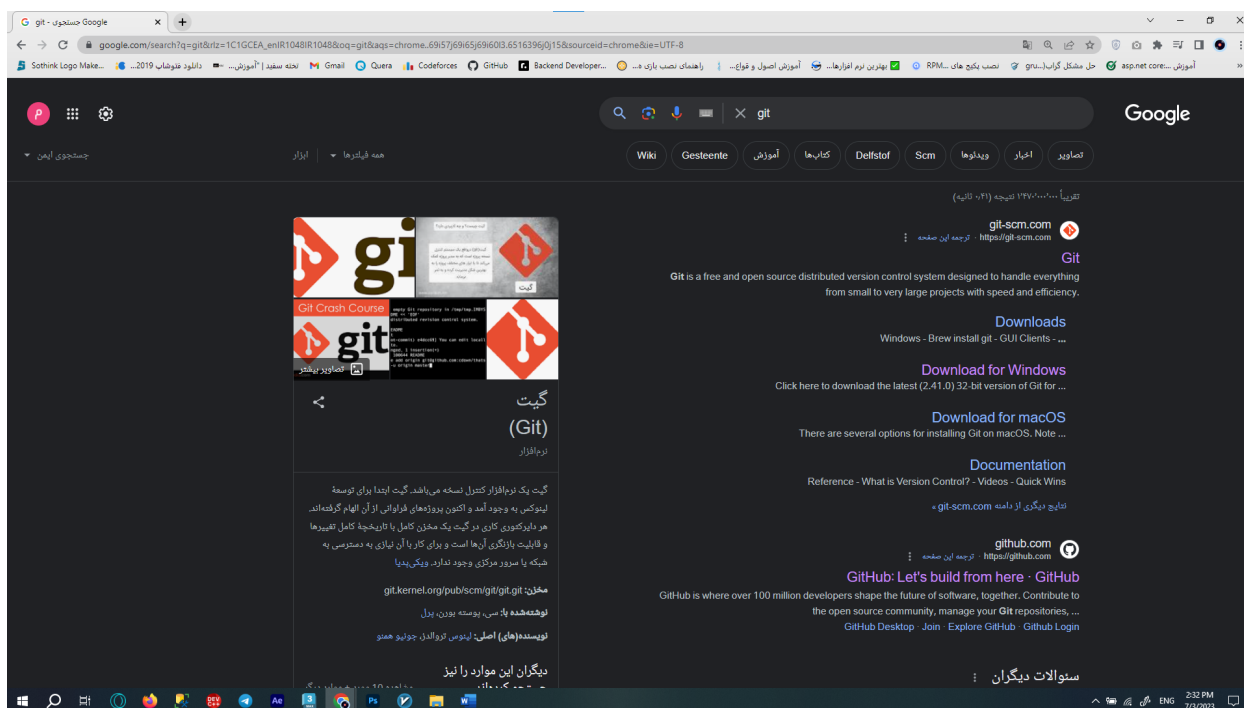
3.....	نصب <i>Git</i>
5.....	دستور <i>init</i>
6.....	دستور <i>status</i>
9.....	دستور <i>git add</i>
10.....	دستور <i>git commit</i>
11.....	دستور <i>git branch</i>
13.....	دستور <i>git checkout</i>
14.....	دستور <i>git reset</i>
15.....	دستور <i>git clone</i>
16.....	دستور <i>git remote</i>
18.....	دستور <i>git push</i>
19.....	دستور <i>git pull</i>

نصب Git

ابتدا در مرورگر **Git** را جستجو می کنیم، وارد سایت

<https://git-scm.com/>

می شویم.

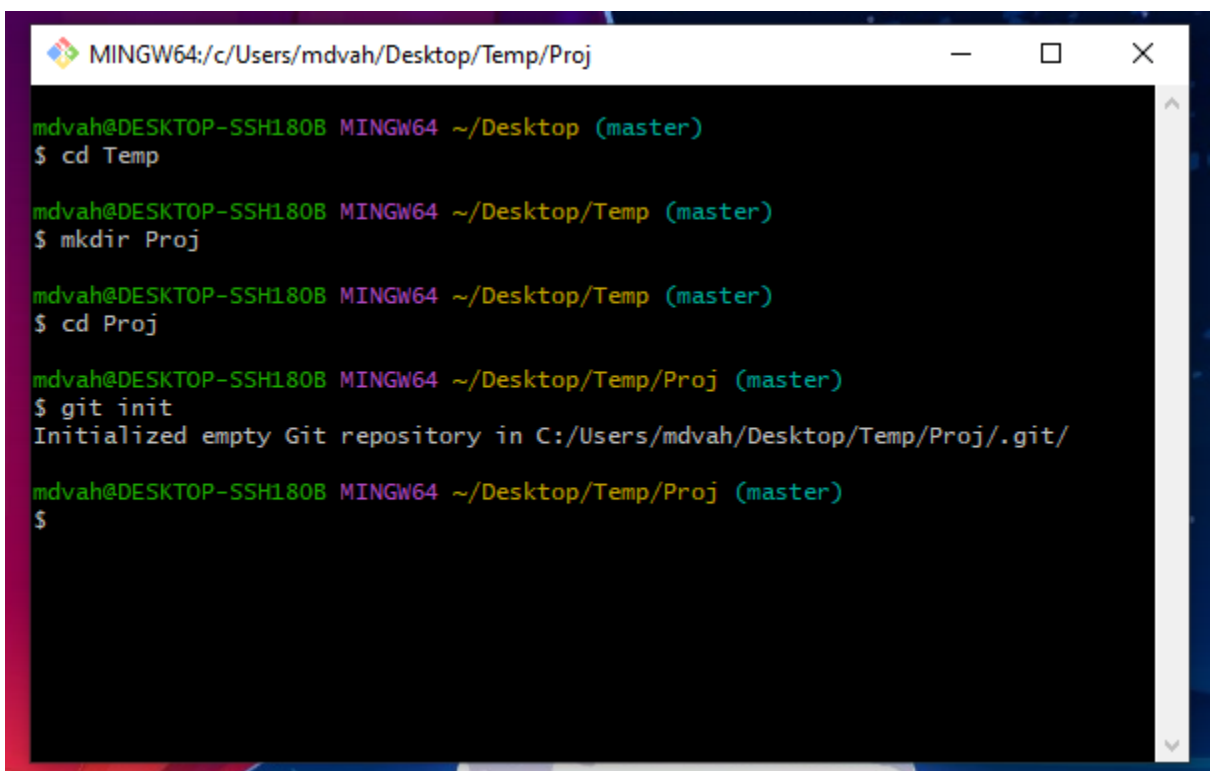


آخرین نسخه Git را برای سیستم عاملی که داریم، دانلود و نصب می کنیم.



دستور *init*

در دایرکتوری Desktop با دستور **cd Temp** وارد دایرکتوری پوشه **Temp** می شویم و سپس با دستور **mkdir Proj** در آن پوشه به نام **Proj** ایجاد می کنیم و سپس با دستور **cd Proj** وارد دایرکتوری اش می شویم و با دستور **git init** دایرکتوری را تحت کنترل **Git** قرار می دهیم.



```
MINGW64:/c/Users/mdvah/Desktop/Temp/Proj
mdvah@DESKTOP-SSH180B MINGW64 ~/Desktop (master)
$ cd Temp

mdvah@DESKTOP-SSH180B MINGW64 ~/Desktop/Temp (master)
$ mkdir Proj

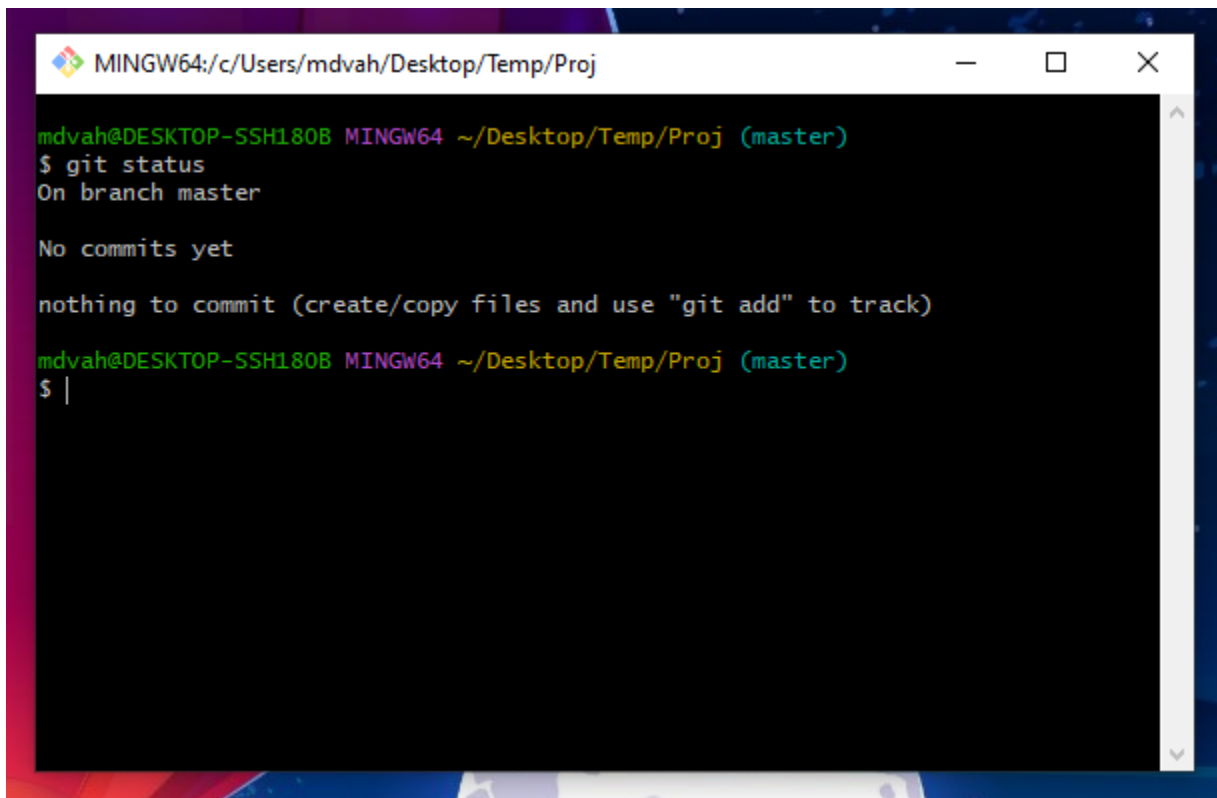
mdvah@DESKTOP-SSH180B MINGW64 ~/Desktop/Temp (master)
$ cd Proj

mdvah@DESKTOP-SSH180B MINGW64 ~/Desktop/Temp/Proj (master)
$ git init
Initialized empty Git repository in C:/Users/mdvah/Desktop/Temp/Proj/.git/

mdvah@DESKTOP-SSH180B MINGW64 ~/Desktop/Temp/Proj (master)
$
```

دستور *status*

وضعیت فایل‌های داخل دایرکتوری را مشخص می‌کند.



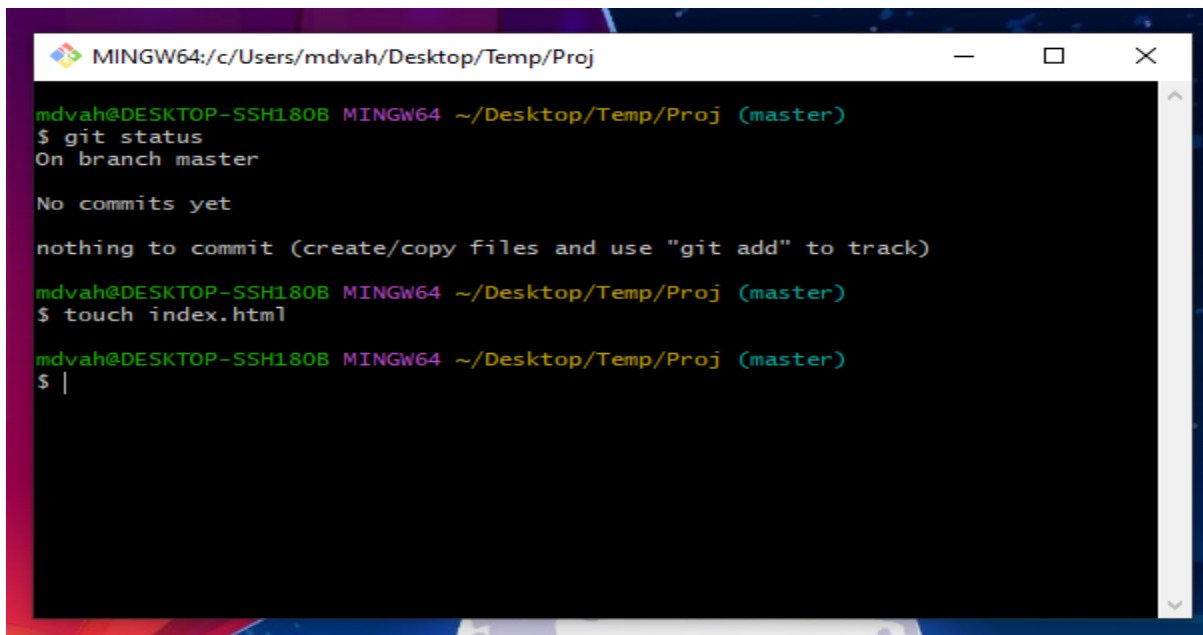
```
MINGW64:/c/Users/mdvah/Desktop/Temp/Proj
mdvah@DESKTOP-SSH180B MINGW64 ~/Desktop/Temp/Proj (master)
$ git status
On branch master

No commits yet

nothing to commit (create/copy files and use "git add" to track)

mdvah@DESKTOP-SSH180B MINGW64 ~/Desktop/Temp/Proj (master)
$ |
```

با دستور **touch index.html** در دایرکتوری یک وب پیج می‌سازیم.



```
MINGW64:/c/Users/mdvah/Desktop/Temp/Proj
mdvah@DESKTOP-SSH180B MINGW64 ~/Desktop/Temp/Proj (master)
$ git status
On branch master

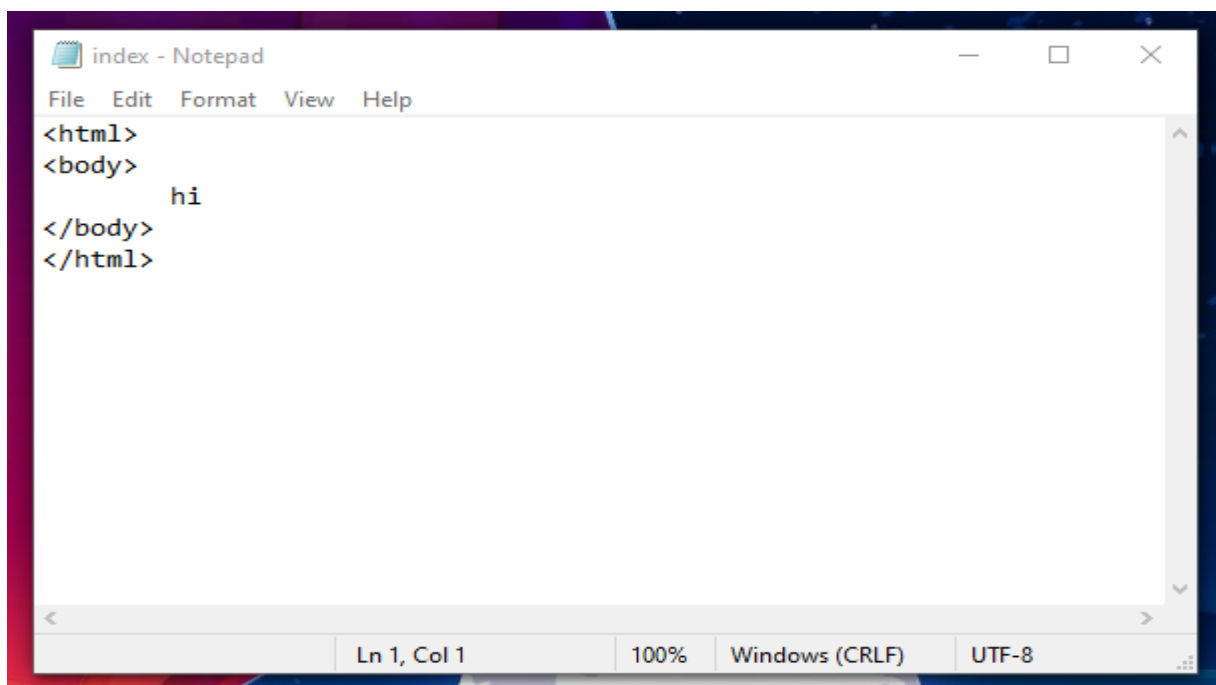
No commits yet

nothing to commit (create/copy files and use "git add" to track)

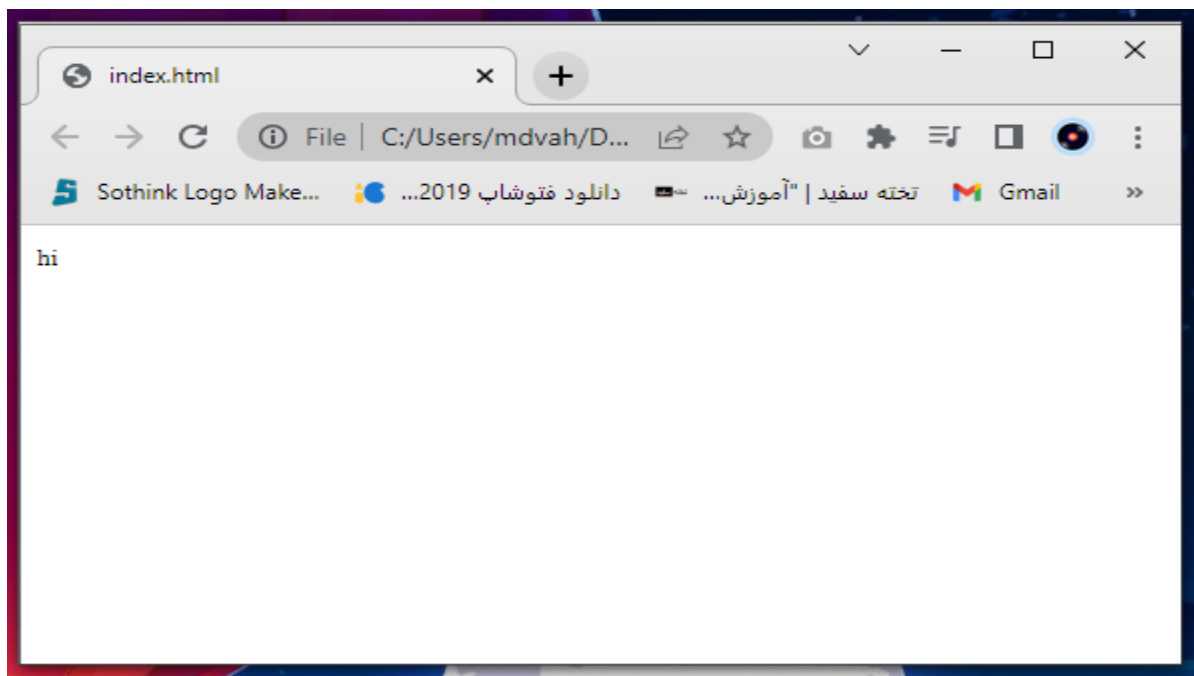
mdvah@DESKTOP-SSH180B MINGW64 ~/Desktop/Temp/Proj (master)
$ touch index.html

mdvah@DESKTOP-SSH180B MINGW64 ~/Desktop/Temp/Proj (master)
$ |
```

وب پیج رو با **NotePad** باز کرده و با کد **html** ویرایشش می‌کنیم مثلاً چاپ
hi



```
index - Notepad
File Edit Format View Help
<html>
<body>
    hi
</body>
</html>
```



با زدن دستور **git status** خواهیم دید که فایل **index.html** به عنوان فایل شناخته نشده در گزارش می شود.

```
MINGW64:/c:/Users/mdvah/Desktop/Temp/Proj
$ git status
On branch master

No commits yet

nothing to commit (create/copy files and use "git add" to track)

mdvah@DESKTOP-SSH180B MINGW64 ~/Desktop/Temp/Proj (master)
$ touch index.html

mdvah@DESKTOP-SSH180B MINGW64 ~/Desktop/Temp/Proj (master)
$ git status
On branch master

No commits yet

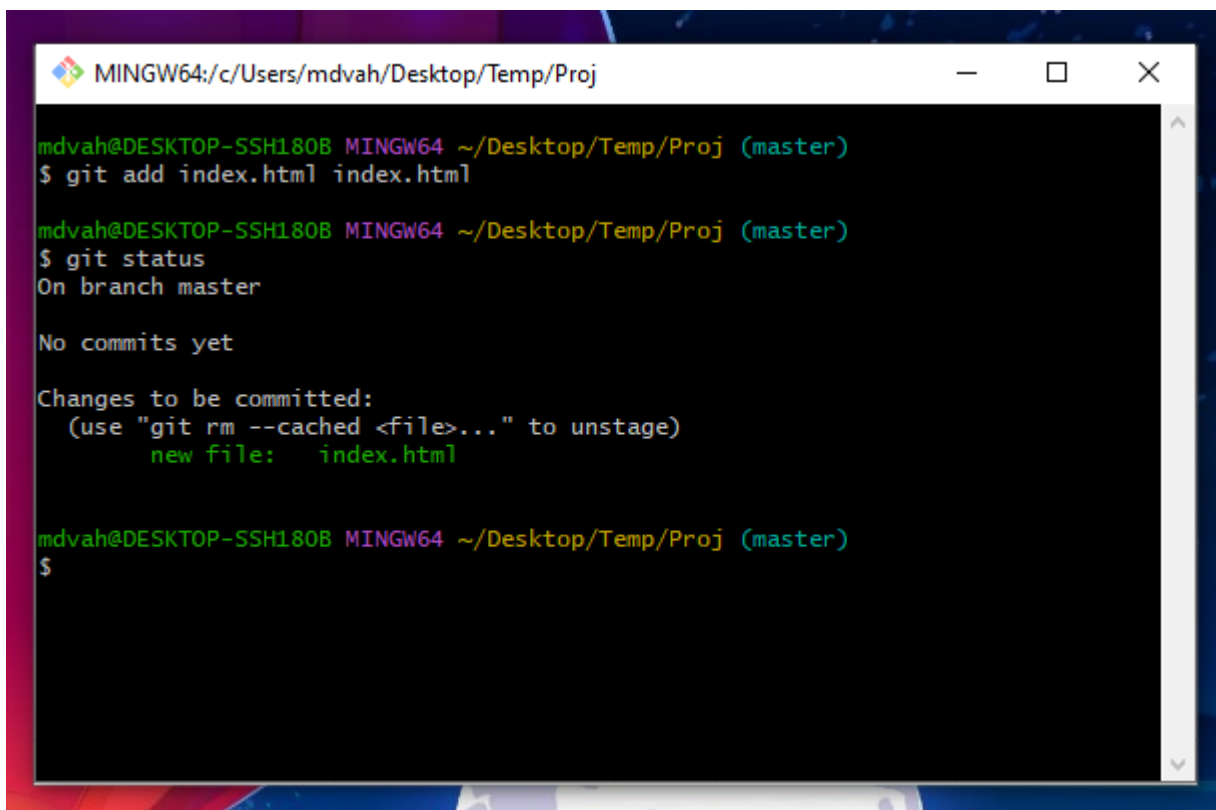
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        index.html

nothing added to commit but untracked files present (use "git add" to track)

mdvah@DESKTOP-SSH180B MINGW64 ~/Desktop/Temp/Proj (master)
$ |
```


دستور *git add*

با دستور `git add index.html` گیت فایل `index.html` را شناسایی می کند و به وضعیت `stage` می برد و آماده `commit` شدن، می شود.



```
MINGW64:/c:/Users/mdvah/Desktop/Temp/Proj
mdvah@DESKTOP-SSH180B MINGW64 ~/Desktop/Temp/Proj (master)
$ git add index.html index.html

mdvah@DESKTOP-SSH180B MINGW64 ~/Desktop/Temp/Proj (master)
$ git status
On branch master

No commits yet

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

mdvah@DESKTOP-SSH180B MINGW64 ~/Desktop/Temp/Proj (master)
$
```

و دستورات دیگر در `add` کردن :

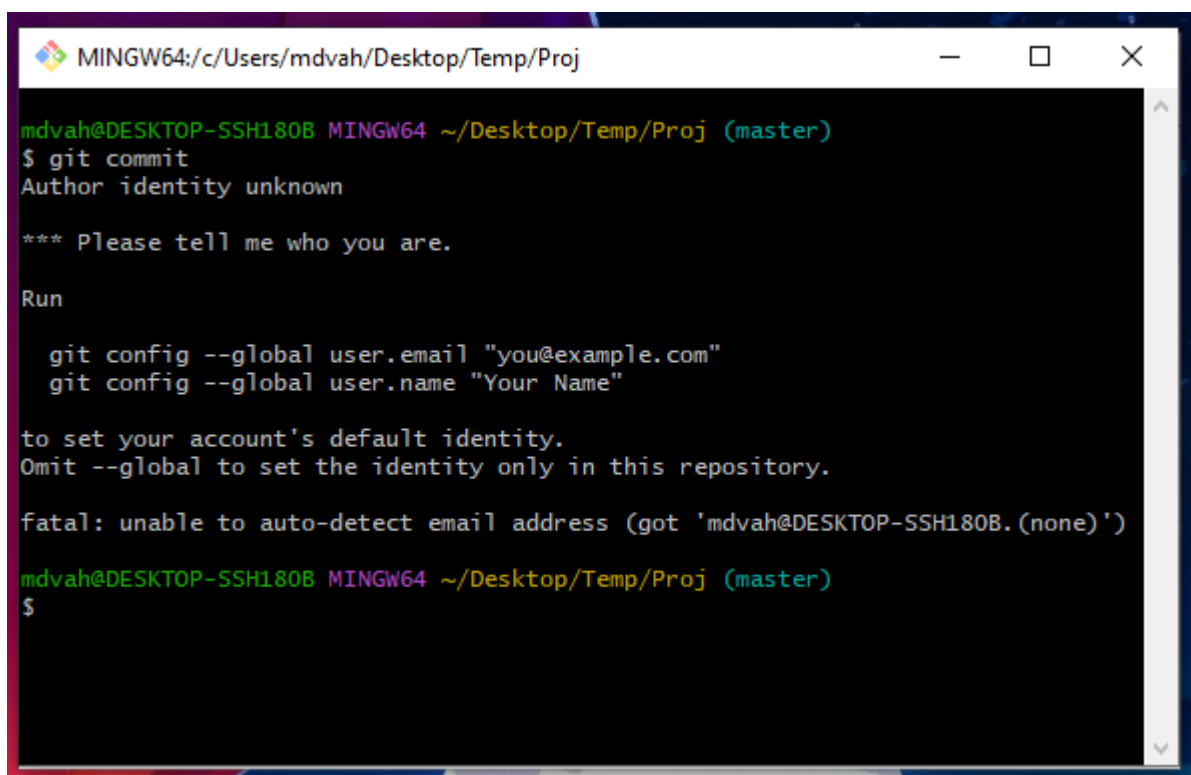
`git add "*.html"`

`git add -A`

دستور *git commit*

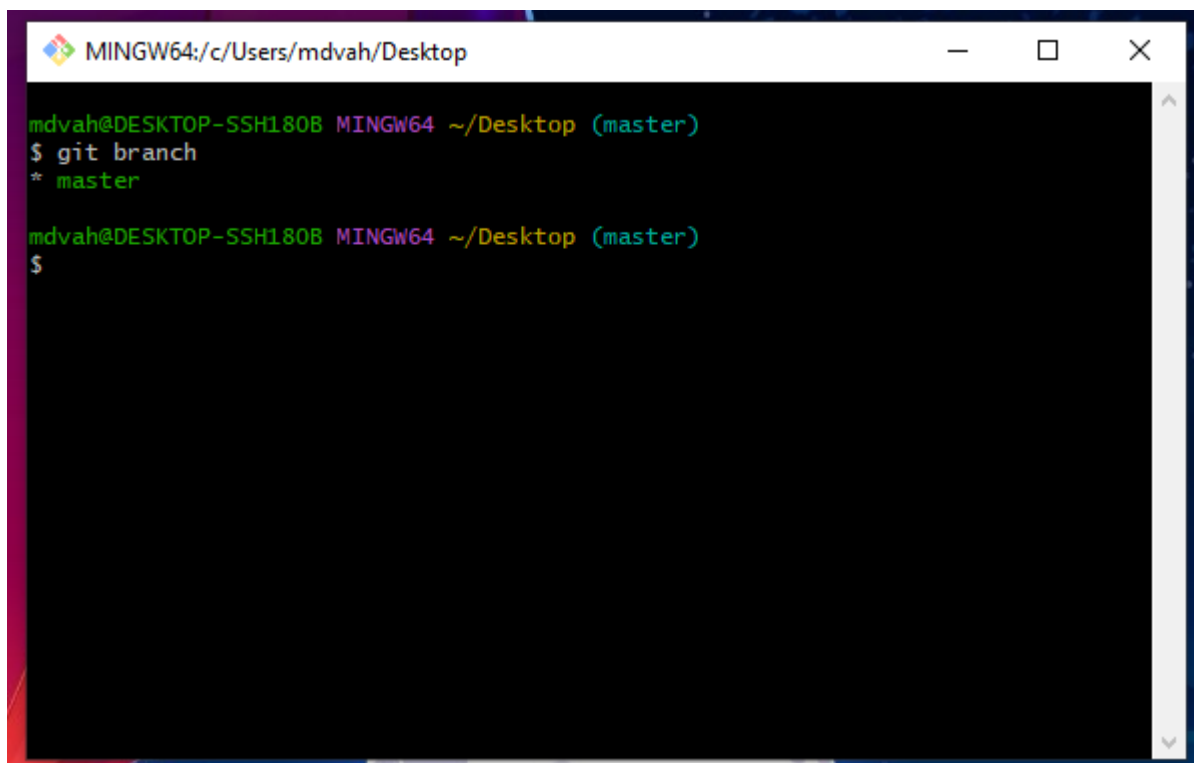
با دستور `git commit -m 'comment'` فایل‌هایی که در وضعیت **stage** قرار گرفته‌اند با دستور :

`git add <file-name>`

A screenshot of a Windows terminal window titled 'MINGW64: c:/Users/mdvah/Desktop/Temp/Proj'. The prompt is 'mdvah@DESKTOP-SSH180B MINGW64 ~/Desktop/Temp/Proj (master)'. The user enters '\$ git commit', and the terminal displays the error 'Author identity unknown' followed by '*** Please tell me who you are.' and instructions to run 'git config --global user.email "you@example.com"' and 'git config --global user.name "Your Name"'. It then shows the fatal error 'fatal: unable to auto-detect email address (got 'mdvah@DESKTOP-SSH180B.(none)')'. The prompt returns to '\$'.

دستور *git branch*

دستور *git branch* تمامی شاخه‌هایی که داریم نشان می‌دهد، شاخه فعلی که در آن هستیم را با * نشان می‌دهد و رنگ سبز به آن اختصاص می‌دهد.



```
MINGW64:/c/Users/mdvah/Desktop
mdvah@DESKTOP-SSH180B MINGW64 ~/Desktop (master)
$ git branch
* master

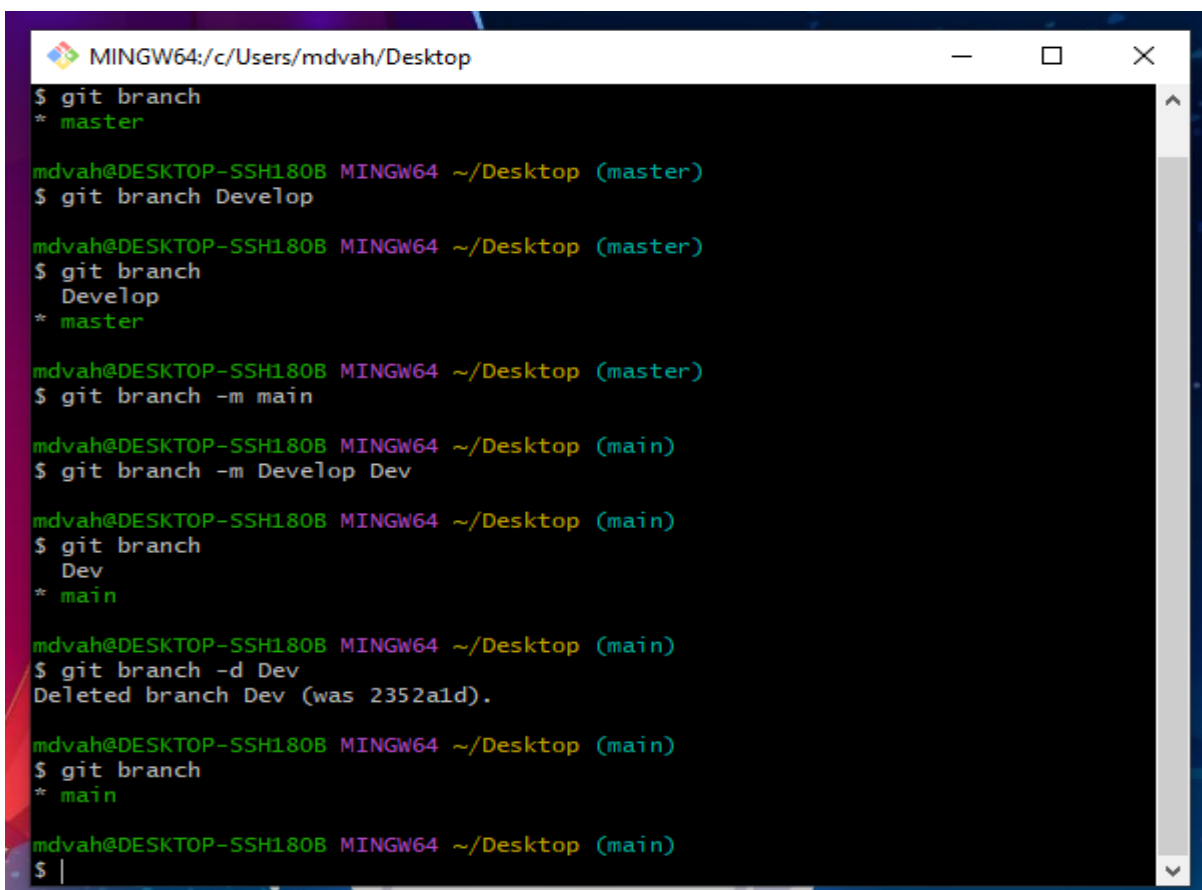
mdvah@DESKTOP-SSH180B MINGW64 ~/Desktop (master)
$
```

با دستور `git branch <branch-name>` می‌توانیم شاخه جدیدی ایجاد کنیم و با دستور `git branch -d <branch-name>` شاخه‌ای را حذف کنیم ولی به این نکته باید توجه کرد که موقع حذف نباید در آن شاخه ای که می‌خواهیم حذف کنیم باشیم.

با دستور `git branch -m <branch-name>` می‌توانیم نام شاخه اصلی را عوض کنیم و با دستور

`git branch -m <old-branch-name> <new-branch-name>`

نام هر شاخه‌را می‌توانیم عوض کنیم.



```
MINGW64:/c:/Users/mdvah/Desktop
$ git branch
* master

mdvah@DESKTOP-SSH180B MINGW64 ~/Desktop (master)
$ git branch Develop

mdvah@DESKTOP-SSH180B MINGW64 ~/Desktop (master)
$ git branch
Develop
* master

mdvah@DESKTOP-SSH180B MINGW64 ~/Desktop (master)
$ git branch -m main

mdvah@DESKTOP-SSH180B MINGW64 ~/Desktop (main)
$ git branch -m Develop Dev

mdvah@DESKTOP-SSH180B MINGW64 ~/Desktop (main)
$ git branch
Dev
* main

mdvah@DESKTOP-SSH180B MINGW64 ~/Desktop (main)
$ git branch -d Dev
Deleted branch Dev (was 2352a1d).

mdvah@DESKTOP-SSH180B MINGW64 ~/Desktop (main)
$ git branch
* main

mdvah@DESKTOP-SSH180B MINGW64 ~/Desktop (main)
$
```

دستور *git checkout*

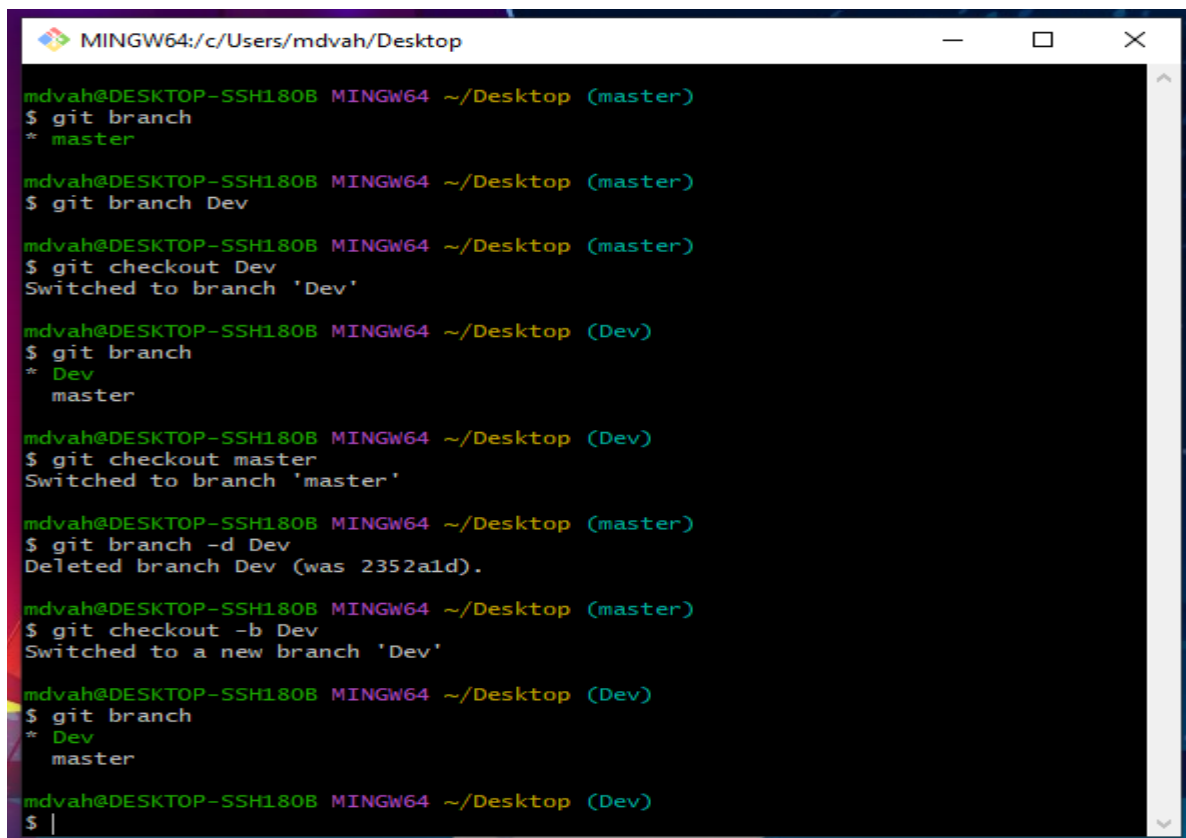
با دستور `git checkout <branch-name>` از شاخه فعلی به شاخه دیگر می‌توانیم منتقل شویم و با دستور :

`git checkout -b <feature-branch-name>`

شاخه جدید می‌سازیم و به آن شاخه منتقل می‌شویم و با دستور :

`git checkout <file-name>`

فایل مورد نظر را به حالت قبلی برمی‌گردانیم.



```
MINGW64: c:/Users/mdvah/Desktop
mdvah@DESKTOP-SSH180B MINGW64 ~/Desktop (master)
$ git branch
* master

mdvah@DESKTOP-SSH180B MINGW64 ~/Desktop (master)
$ git branch Dev

mdvah@DESKTOP-SSH180B MINGW64 ~/Desktop (master)
$ git checkout Dev
Switched to branch 'Dev'

mdvah@DESKTOP-SSH180B MINGW64 ~/Desktop (Dev)
$ git branch
* Dev
  master

mdvah@DESKTOP-SSH180B MINGW64 ~/Desktop (Dev)
$ git checkout master
Switched to branch 'master'

mdvah@DESKTOP-SSH180B MINGW64 ~/Desktop (master)
$ git branch -d Dev
Deleted branch Dev (was 2352a1d).

mdvah@DESKTOP-SSH180B MINGW64 ~/Desktop (master)
$ git checkout -b Dev
Switched to a new branch 'Dev'

mdvah@DESKTOP-SSH180B MINGW64 ~/Desktop (Dev)
$ git branch
* Dev
  master

mdvah@DESKTOP-SSH180B MINGW64 ~/Desktop (Dev)
$
```

دستور *git reset*

- با دستور **git reset --soft HEAD** لغو تغییرات (**unstage changes**) می‌توانیم از حالت stage فایل‌های را به مرحله قبل **add** کردن ببرید و فایل‌ی آماده **commit** شدن، نباشد.
- با دستور **git reset --hard HEAD** لغو تغییرات و حذف تغییرات (**Discard changes**) تمامی تغییراتی که در مرحله **stage** و **unstage** انجام شده را لغو و حذف می‌کند، این عمل باعث برگشتن به وضعیت **commit** قبلی می‌شود.
- با دستور **git reset --soft <commit-hash>** تاریخچه را به کامیتی که هَش شو وارد می‌کنیم برمیگرداند و تاریخچه را پاک می‌کند از این دستور باید با احتیاط استفاده شود.

دستور *git clone*

دستور *git clone* در **Git** برای کپی یک مخزن (**repository**) است، یک نسخه کپی از یک مخزن را می‌توانیم ایجاد کنیم و بر روی سیستم قرار دهیم.

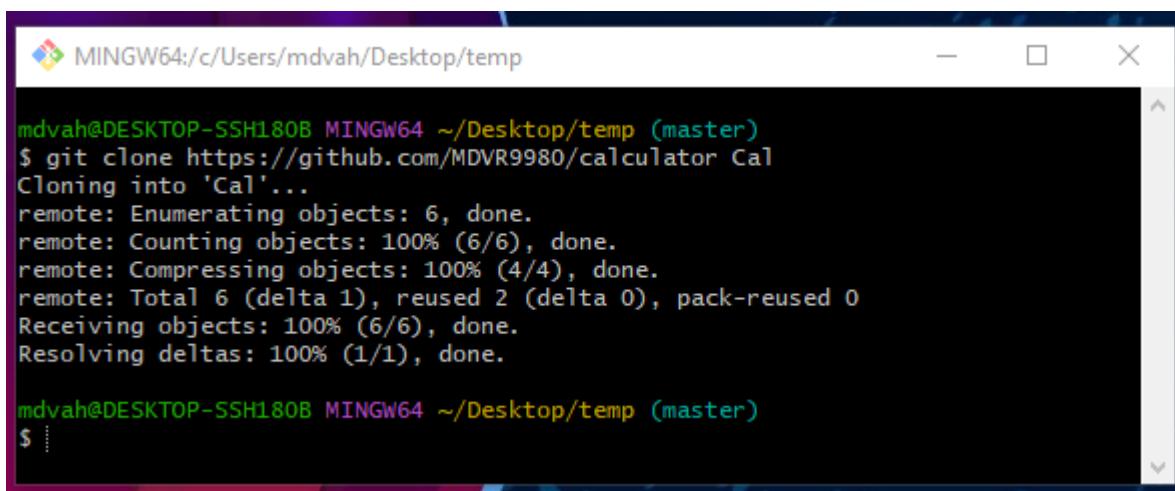
git clone <repository-url> [<directory-name>]

: repository-url

نشانی (**URL**) مخزن **Git** مورد نظر است که قصد دارید کپی کنید.

: Directory-name

نام پوشه‌ای است که شما می‌خواهید مخزن را در آن قرار دهید. این پارامتر اختیاری است و اگر وارد نشود، نام پوشه‌ای که با نام مخزن ایجاد می‌شود، استفاده می‌شود.



```
MINGW64:/c/Users/mdvah/Desktop/temp
mdvah@DESKTOP-SSH180B MINGW64 ~/Desktop/temp (master)
$ git clone https://github.com/MDVR9980/calculator Cal
Cloning into 'Cal'...
remote: Enumerating objects: 6, done.
remote: Counting objects: 100% (6/6), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 6 (delta 1), reused 2 (delta 0), pack-reused 0
Receiving objects: 100% (6/6), done.
Resolving deltas: 100% (1/1), done.

mdvah@DESKTOP-SSH180B MINGW64 ~/Desktop/temp (master)
$
```

دستور *git remote*

دستور *git remote* در **Git** برای مدیریت و نمایش نام‌ها و نشانگرهای ارتباطی با مخازن از راه دور (**remote repository**) استفاده می‌شود. این دستور امکاناتی را برای افزودن، حذف، نمایش و تنظیم نشانگرهای ارتباطی با مخازن از راه دور فراهم می‌کند.

نمایش لیست نشانگرها :

git remote

افزودن نشانگر جدید به یک مخزن از راه دور :

git remote add <name> <url>

برای حذف نشانگر از دستور :

git remote remove <name>

تغییر نام نشانگر :

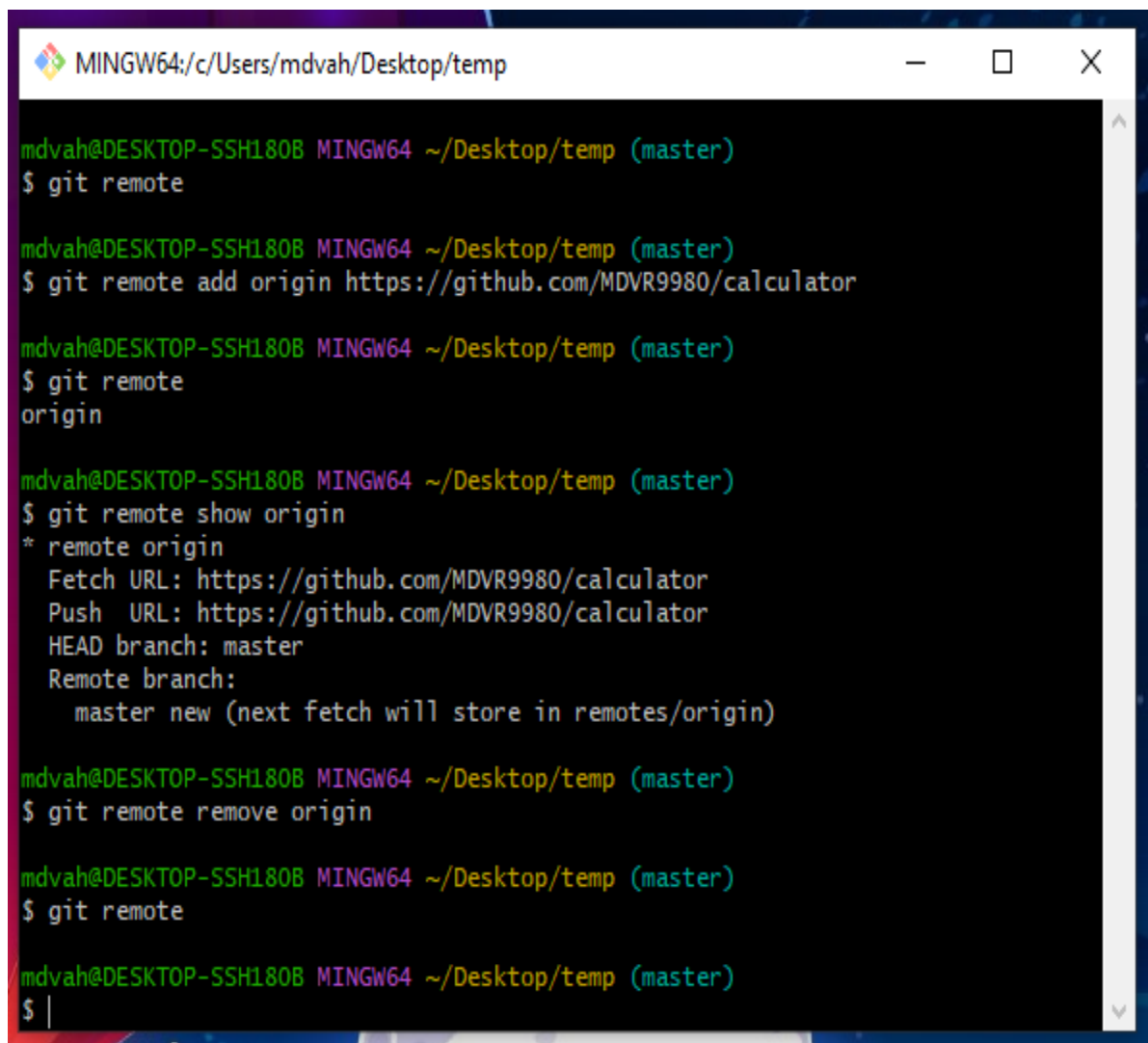
git remote rename <remote-name> <remote-new-name>

نمایش جزئیات نشانگر :

git remote show <name>

نمایش تمامی نام‌های نشانه‌گرها و آدرس هایشان :

git remote -v

A screenshot of a Windows command prompt window titled "MINGW64: c:/Users/mdvah/Desktop/temp". The window shows a series of Git commands and their outputs. The user is in the "master" branch of a repository located at "~/Desktop/temp". The commands executed are: 1. "git remote" which shows no output. 2. "git remote add origin https://github.com/MDVR9980/calculator" which adds a new remote named "origin". 3. "git remote" which now shows "origin". 4. "git remote show origin" which displays details for the "origin" remote, including the fetch and push URLs, the HEAD branch (master), and the remote branch (master new). 5. "git remote remove origin" which removes the "origin" remote. 6. "git remote" which shows no output. 7. The prompt "\$ |" is shown at the bottom, indicating the command is being typed.

```
mdvah@DESKTOP-SSH180B MINGW64 ~/Desktop/temp (master)
$ git remote

mdvah@DESKTOP-SSH180B MINGW64 ~/Desktop/temp (master)
$ git remote add origin https://github.com/MDVR9980/calculator

mdvah@DESKTOP-SSH180B MINGW64 ~/Desktop/temp (master)
$ git remote
origin

mdvah@DESKTOP-SSH180B MINGW64 ~/Desktop/temp (master)
$ git remote show origin
* remote origin
  Fetch URL: https://github.com/MDVR9980/calculator
  Push URL: https://github.com/MDVR9980/calculator
  HEAD branch: master
  Remote branch:
    master new (next fetch will store in remotes/origin)

mdvah@DESKTOP-SSH180B MINGW64 ~/Desktop/temp (master)
$ git remote remove origin

mdvah@DESKTOP-SSH180B MINGW64 ~/Desktop/temp (master)
$ git remote

mdvah@DESKTOP-SSH180B MINGW64 ~/Desktop/temp (master)
$ |
```

دستور *git push*

دستور *git push* در *git* برای ارسال (*push*) تغییرات محلی به یک مخزن از راه دور (*remote repository*) استفاده می‌شود. با استفاده از این دستور، شما می‌توانید تغییراتی که در نسخه محلی اعمال کرده‌اید را به مخزن از راه دور ارسال کنید تا دیگران بتوانند به آن دسترسی داشته باشند. توضیحات زیر نحوه استفاده از دستور *git push* را توضیح می‌دهد.

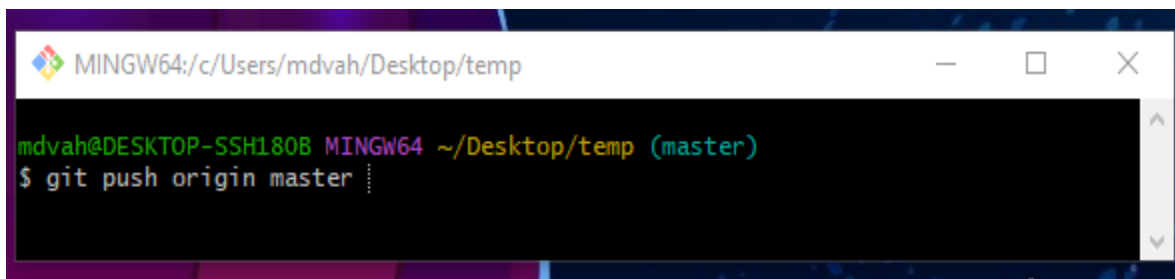
git push <remote-name> <branch-name>

: remote-name

نام نشانگر مخزن از راه دور است که می‌خواهید تغییرات را به آن ارسال کنید، معمولاً نشانگر *origin* برای مخزن اصلی استفاده می‌شود.

: branch-name

نام شاخه محلی است که می‌خواید تغییرات آن را ارسال کنید.

A screenshot of a terminal window titled 'MINGW64: c:/Users/mdvah/Desktop/temp'. The prompt is 'mdvah@DESKTOP-SSH180B MINGW64 ~/Desktop/temp (master)'. The command '\$ git push origin master' has been entered and is followed by three dots, indicating it is still running or the output is being truncated.

دستور *git pull*

دستور *git pull* در **Git** برای دریافت (**pull**) تغییرات از مخزن راه دور و ادغام آن با شاخه محلی استفاده می‌شود. با استفاده از این دستور، شما می‌توانید تغییرات اعمال شده توسط دیگران در مخزن از راه دور را دریافت و با شاخه محلی خود ادغام کنید. توضیحات زیر نحوه استفاده از دستور *git pull* توضیح می‌دهد:

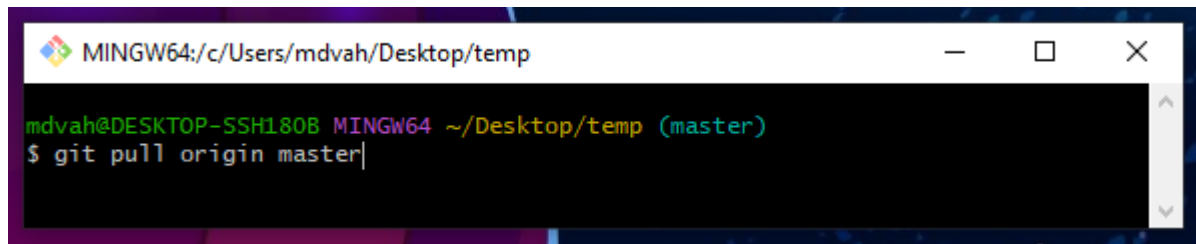
git pull <remote-name> <branch-name>

: remote-name

نام نشانگر مخزن از راه دور است که می‌خواهید تغییرات را از آن دریافت کنید، معمولاً نشانگر **origin** برای مخزن اصلی استفاده می‌شود.

: branch-name

نام شاخه محلی است که می‌خواهید تغییرات را در آن ادغام کنید.

A screenshot of a MINGW64 terminal window. The title bar shows the path 'MINGW64:/c/Users/mdvah/Desktop/temp'. The terminal content shows the prompt 'mdvah@DESKTOP-SSH180B MINGW64 ~/Desktop/temp (master)' followed by the command '\$ git pull origin master|'.

```
MINGW64:/c/Users/mdvah/Desktop/temp
mdvah@DESKTOP-SSH180B MINGW64 ~/Desktop/temp (master)
$ git pull origin master|
```

در اینجا **origin** نام نشانگر مخزن اصلی است و **master** نام شاخه محلی است که می‌خواهید تغییرات را در آن ادغام کنید.

دستور **git pull** در واقع دو دستور **git fetch** و **git merge** را ترکیب می‌کند. ابتدا با **git fetch** تغییرات را از مخزن از راه دور دریافت می‌کند و سپس با **git merge** آن را با شاخه محلی ادغام می‌کند.

توجه داشته باشید که قبل از اجرای دستور **git pull**، مناسب است تغییرات محلی خود را با دستور **git commit** ثبت کنید تا دارای تاریخچه قبل از ادغام تغییرات باشید.

The End