



دانشگاه حکیم سبزواری
دانشکده فنی برق و کامپیوتر
مستند آموزشی

موضوع : Git

نویسنده :

محمد داود وهاب رجائی

تیر 1402

فهرست :

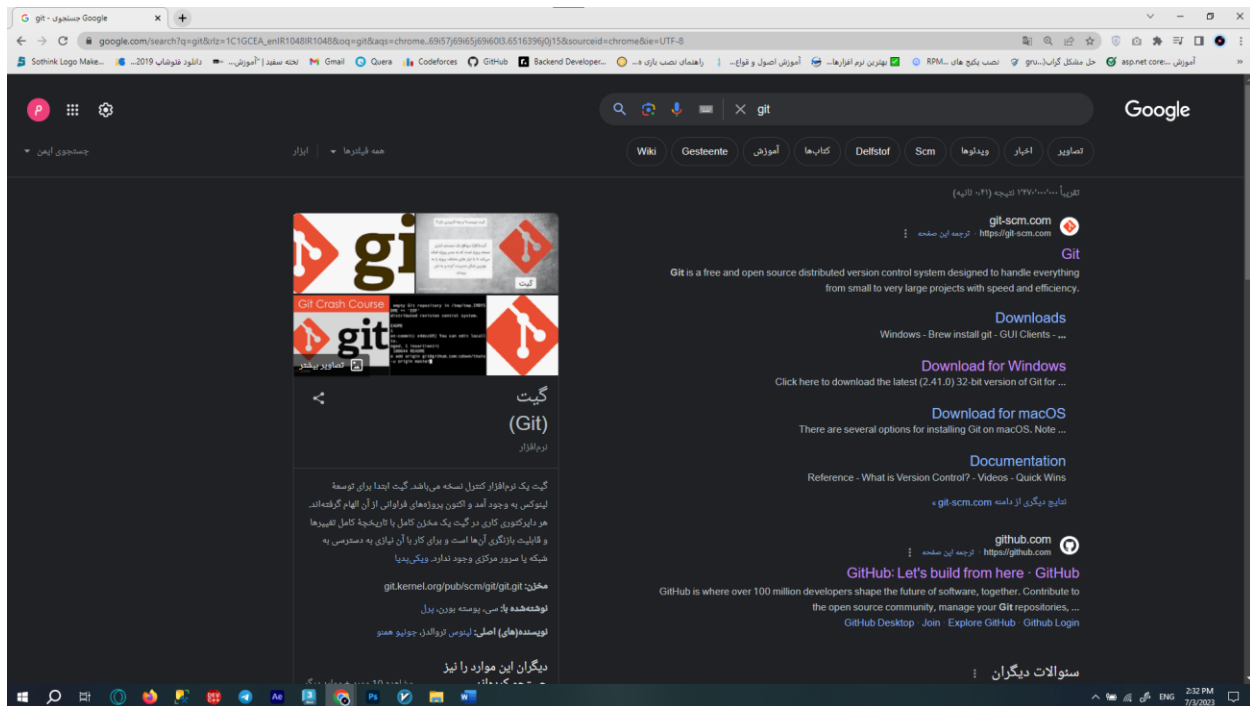
3.....	نصب <i>Git</i>
5.....	دستور <i>init</i>
6.....	دستور <i>status</i>
9.....	دستور <i>git add</i>
10.....	دستور <i>git commit</i>
11.....	دستور <i>git branch</i>
13.....	دستور <i>git checkout</i>
14.....	دستور <i>git reset</i>
15.....	دستور <i>git clone</i>
16.....	دستور <i>git remote</i>
18.....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

نصب Git

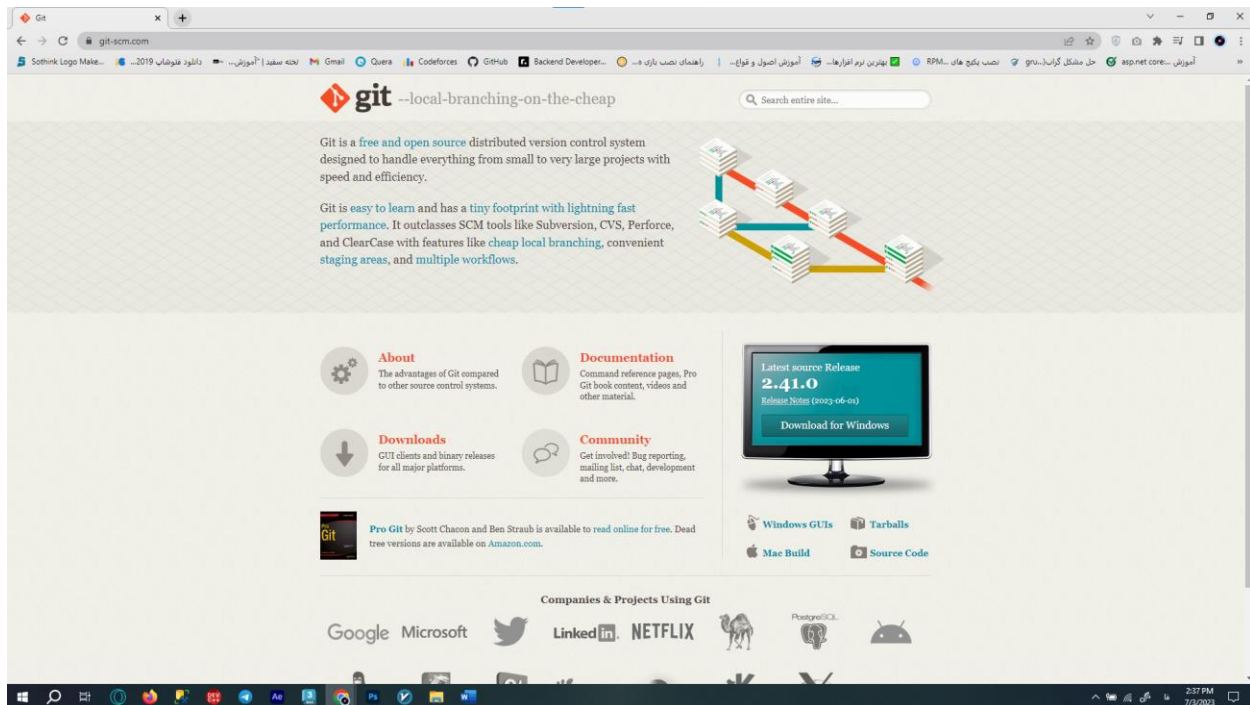
ابتدا در مرورگر **Git** را جستجو می کنیم، وارد سایت

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

می شویم.

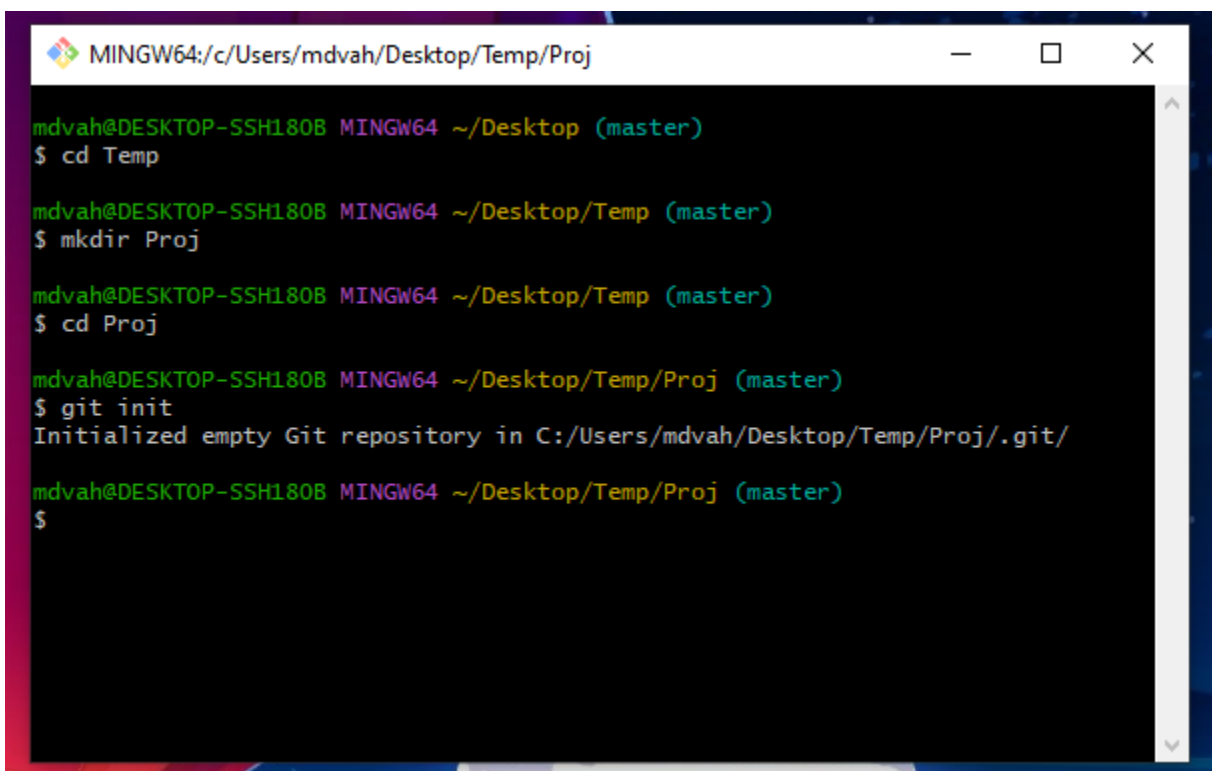


آخرین نسخه **Git** را برای سیستم عاملی که داریم، دانلود و نصب می‌کنیم.



دستور *init*

در دایرکتوری Desktop با دستور **cd Temp** وارد دایرکتوری پوشه **Temp** می‌شویم و سپس با دستور **mkdir Proj** در آن پوشه به نام **Proj** ایجاد می‌کنیم و سپس با دستور **cd Proj** وارد دایرکتوری اش می‌شویم و با دستور **git init** دایرکتوری را تحت کنترل **Git** قرار می‌دهیم.



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

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

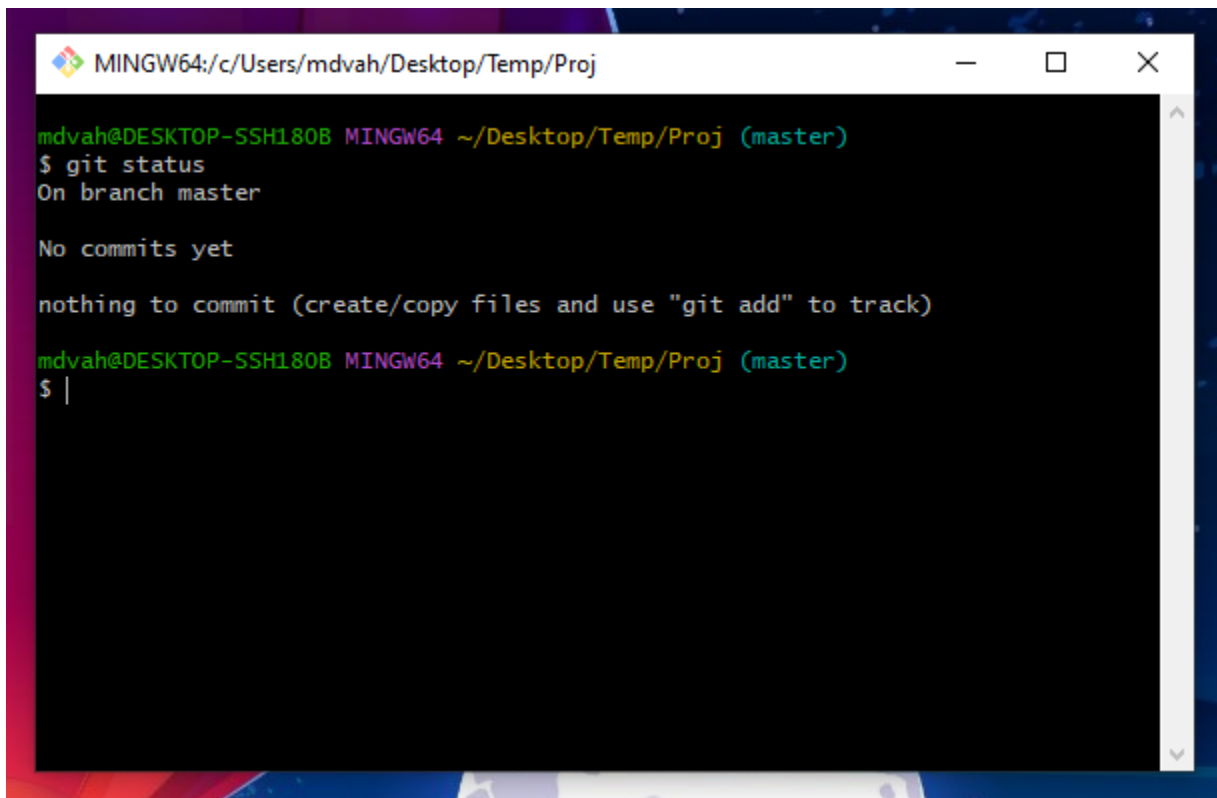
mdvah@DESKTOP-SSH180B MINGW64 ~/Desktop/Temp (master)
$ cd Proj

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.



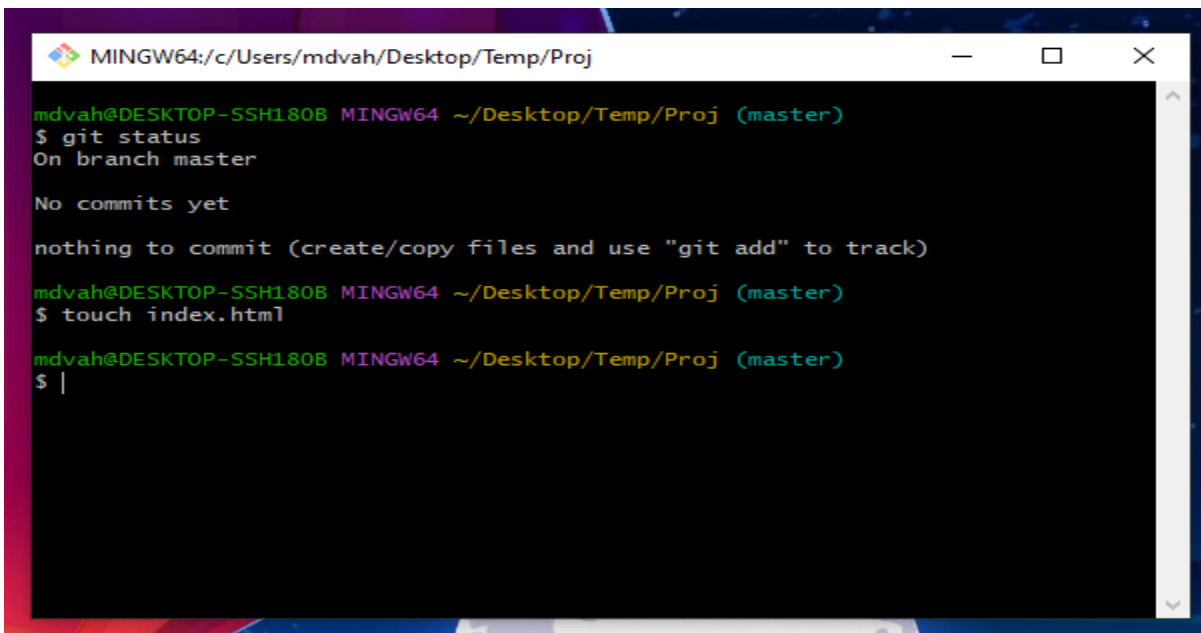
```
MINGW64:/c/Users/mdvah/Desktop/Temp/Proj
mdvah@DESKTOP-SSH180B MINGW64 ~/Desktop/Temp/Proj (master)
$ git status
On branch master

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

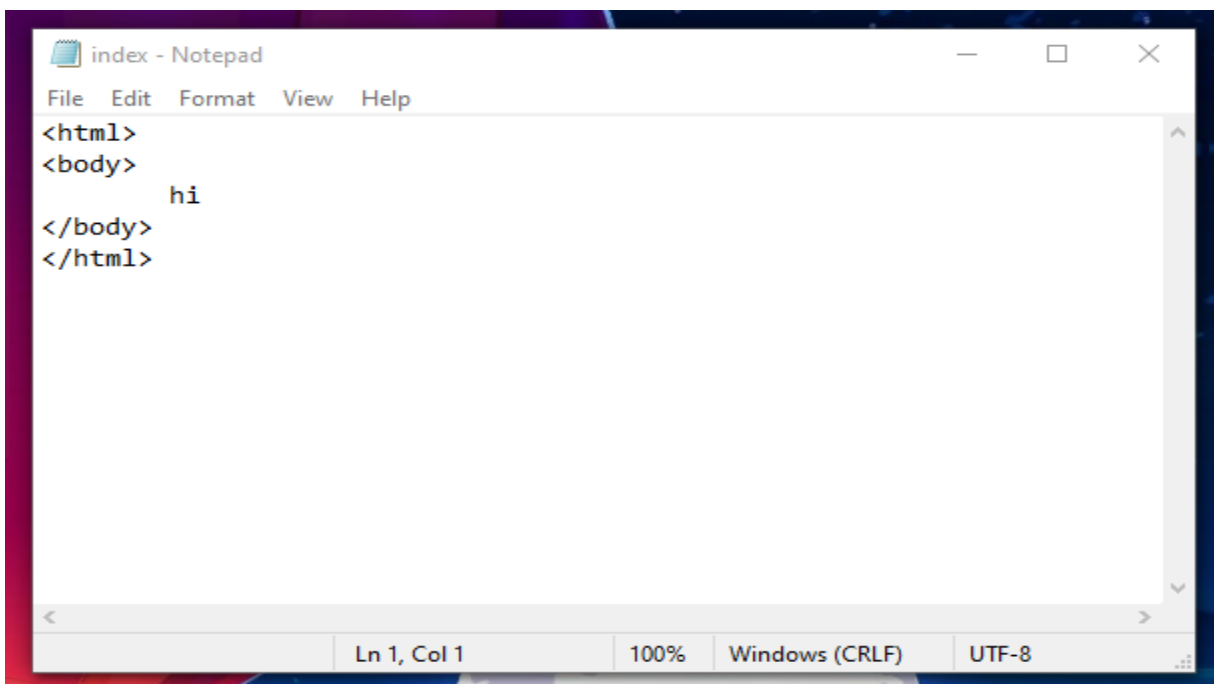
No commits yet

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

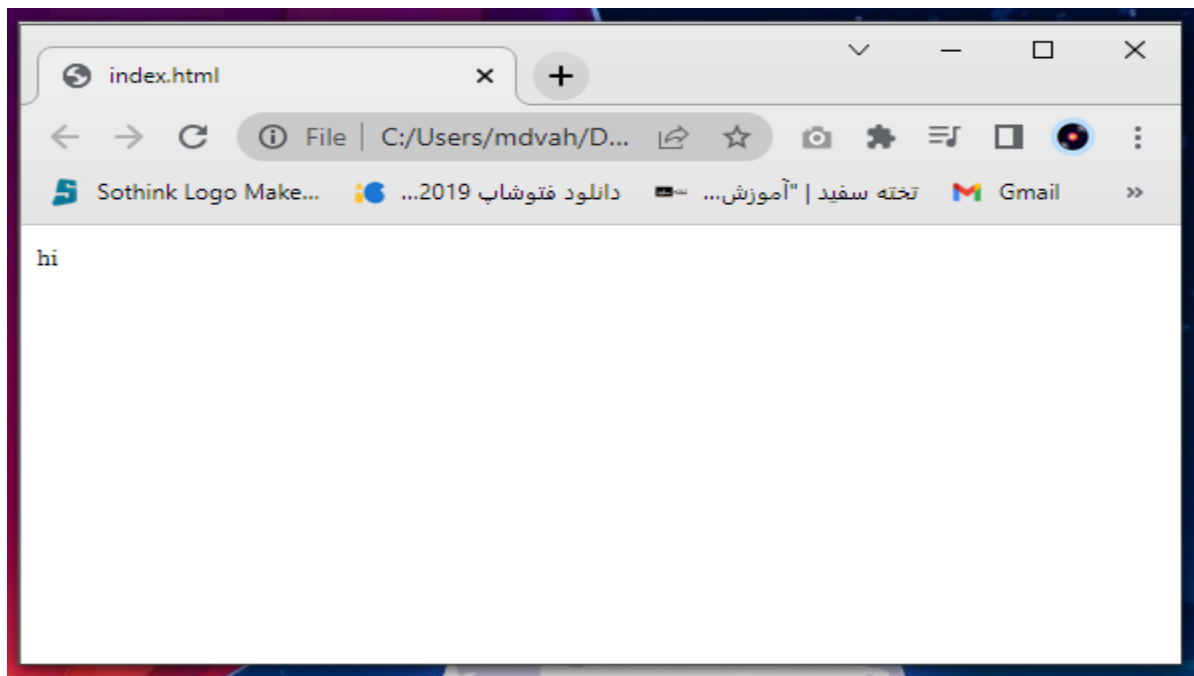
mdvah@DESKTOP-SSH180B MINGW64 ~/Desktop/Temp/Proj (master)
$ touch index.html

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>
Ln 1, Col 1 100% Windows (CRLF) UTF-8
```



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

```
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

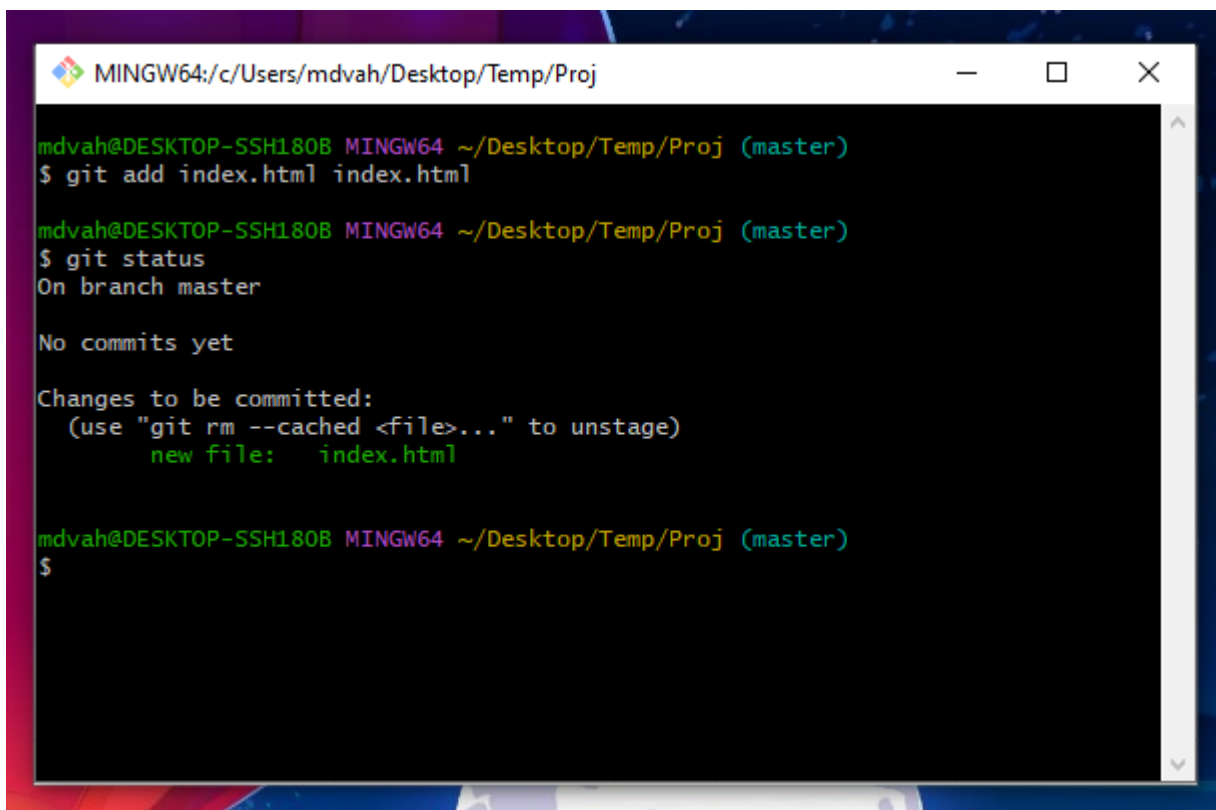
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        index.html

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 می‌کند و به وضعیت `stage` می‌برد و آماده `commit` شدن، می‌شود.



```
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 :

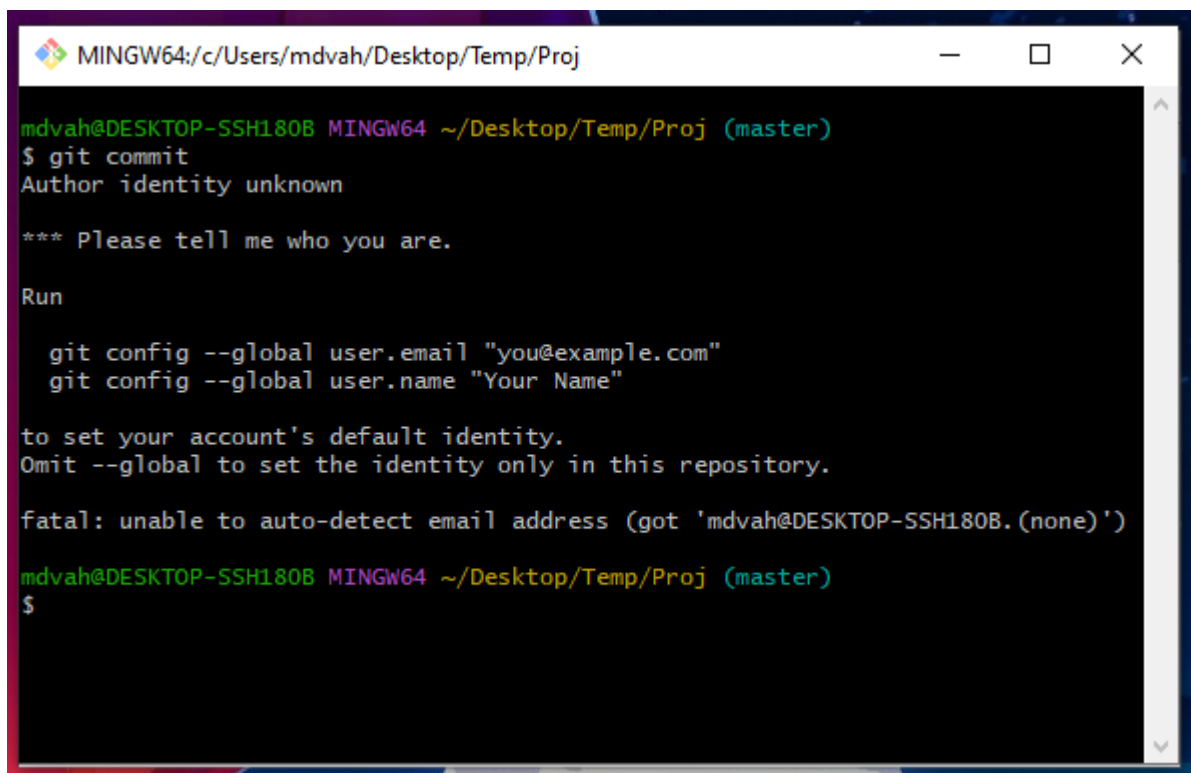
`git add "*.html"`

`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>`



```
MINGW64:/c:/Users/mdvah/Desktop/Temp/Proj
mdvah@DESKTOP-SSH180B MINGW64 ~/Desktop/Temp/Proj (master)
$ git commit
Author identity unknown

*** Please tell me who you are.

Run

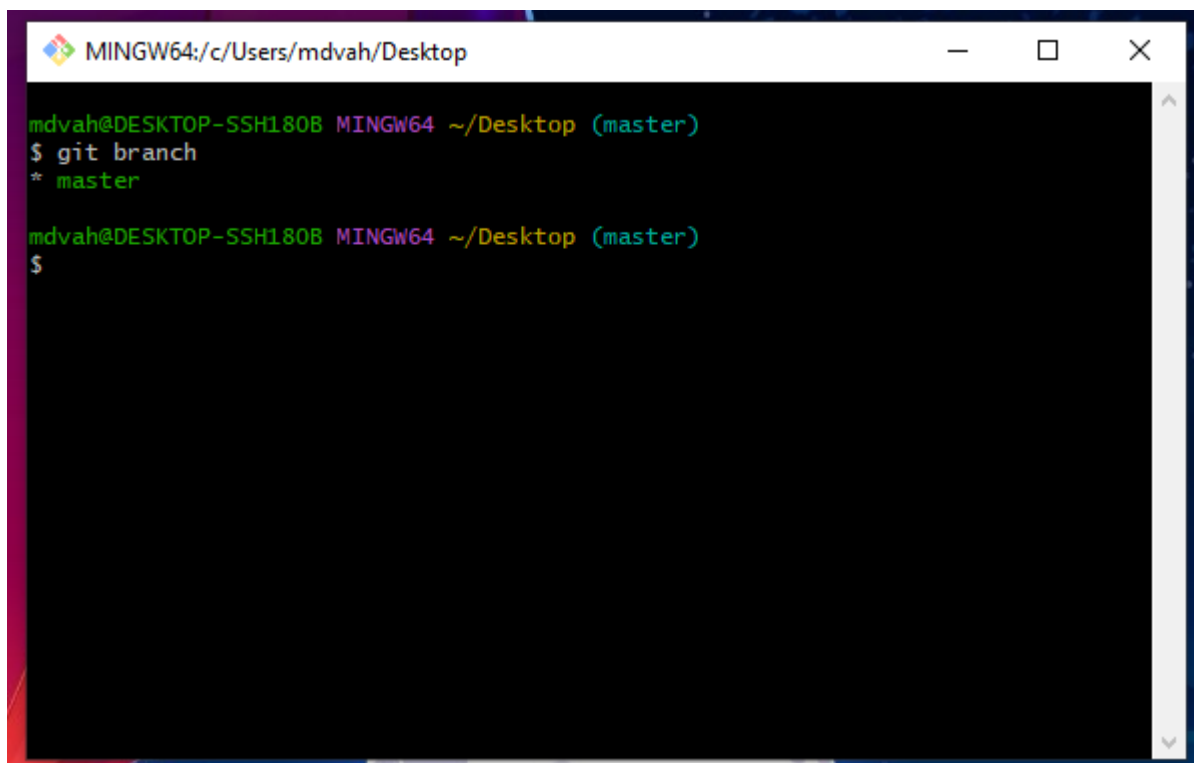
  git config --global user.email "you@example.com"
  git config --global user.name "Your Name"

to set your account's default identity.
Omit --global to set the identity only in this repository.

fatal: unable to auto-detect email address (got 'mdvah@DESKTOP-SSH180B.(none)')
mdvah@DESKTOP-SSH180B MINGW64 ~/Desktop/Temp/Proj (master)
$
```

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.



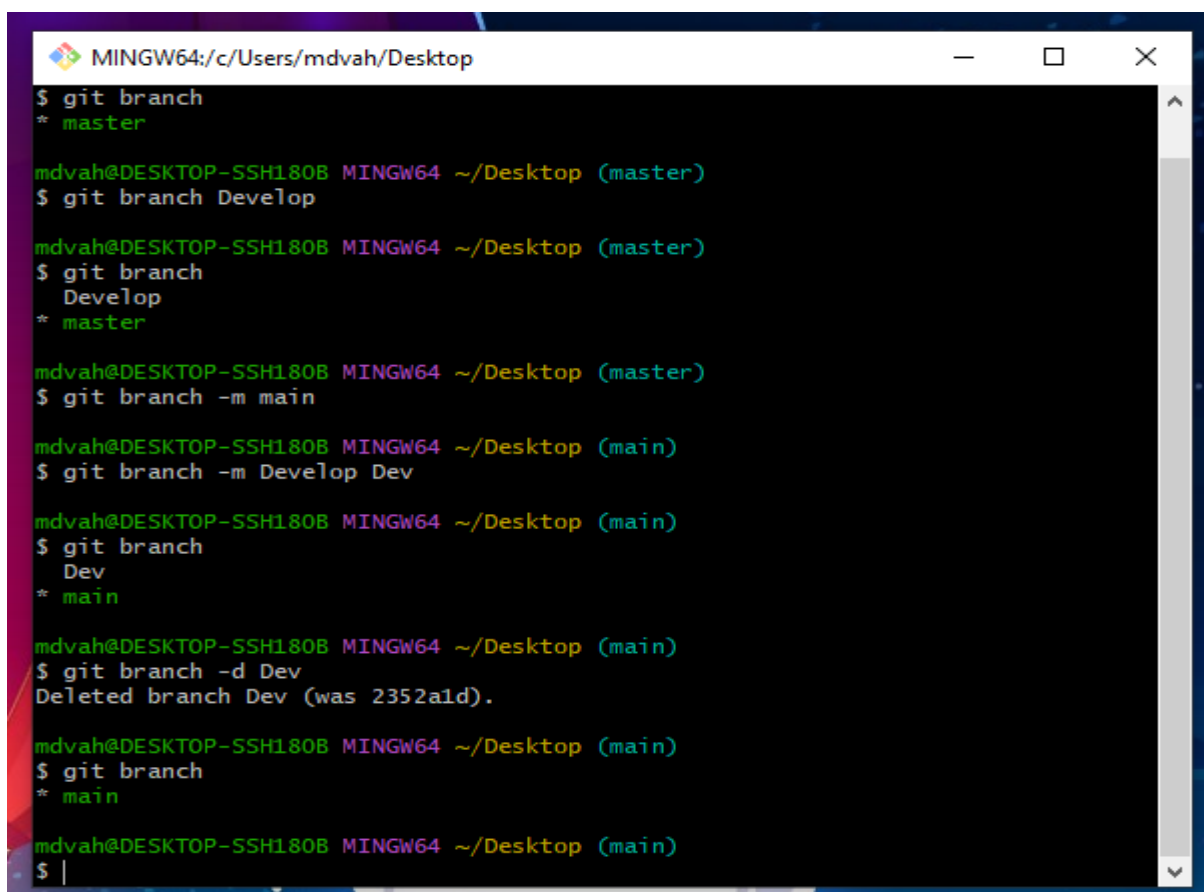
```
MINGW64:/c/Users/mdvah/Desktop
mdvah@DESKTOP-SSH180B MINGW64 ~/Desktop (master)
$ git branch
* master

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 شاخه را 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*

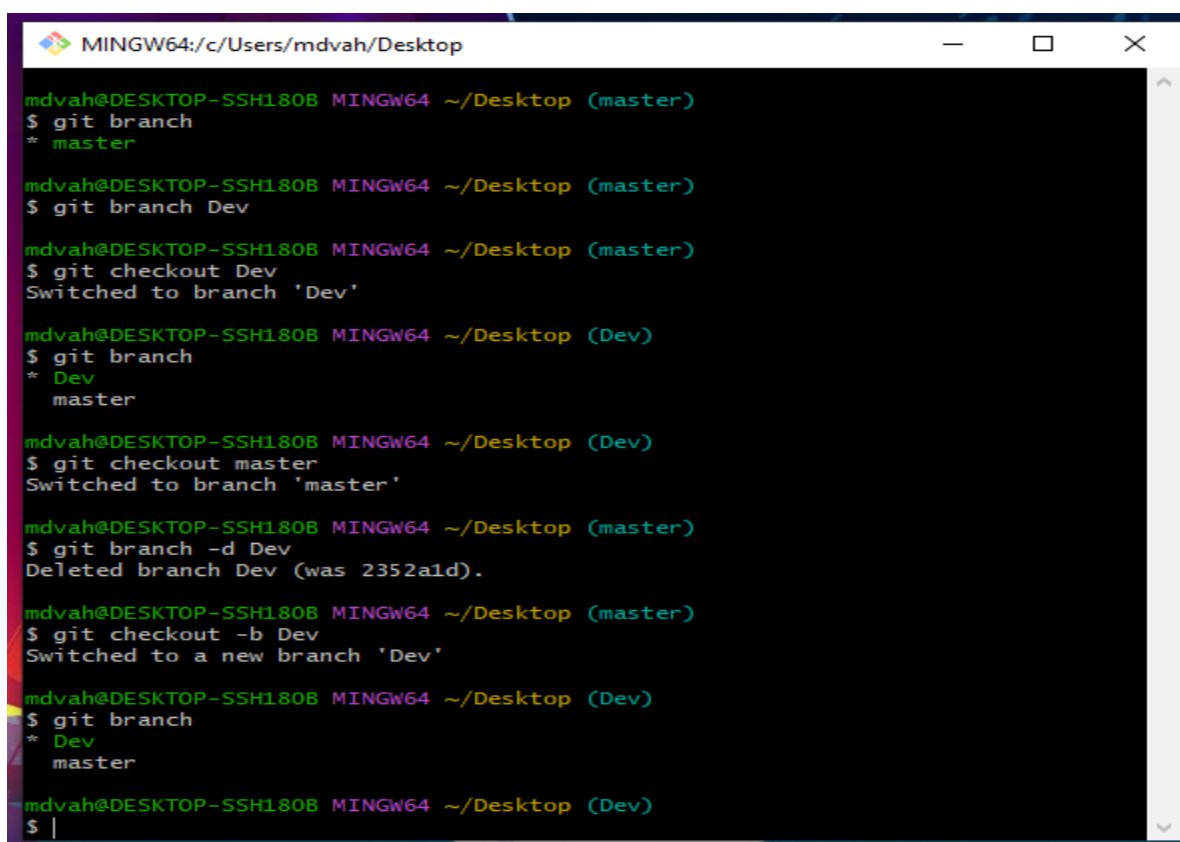
با دستور `git checkout <branch-name>` از شاخه فعلی به شاخه دیگر می‌توانیم منتقل شویم و با دستور :

`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.

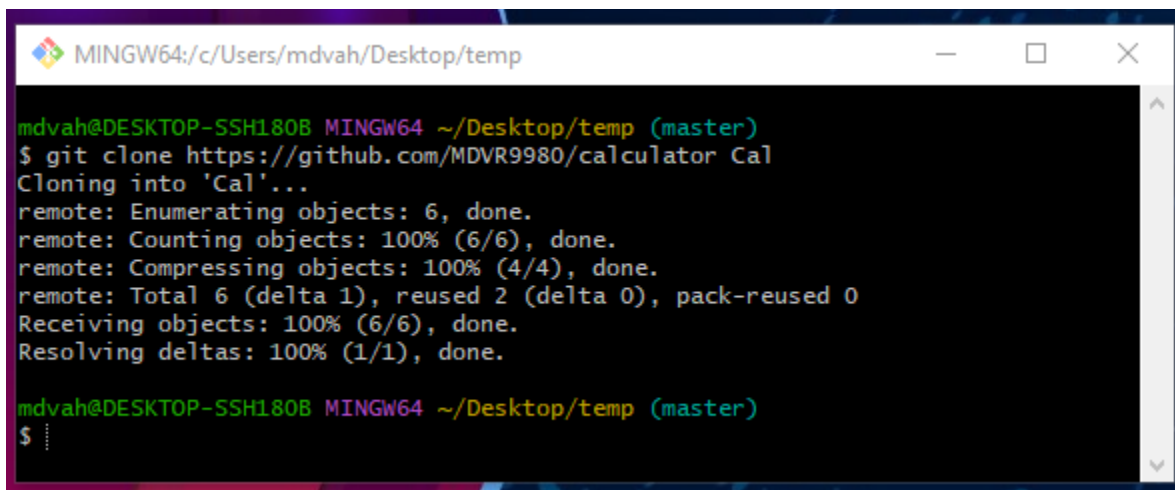
git clone <repository-url> [*<directory-name>*]

: 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 :

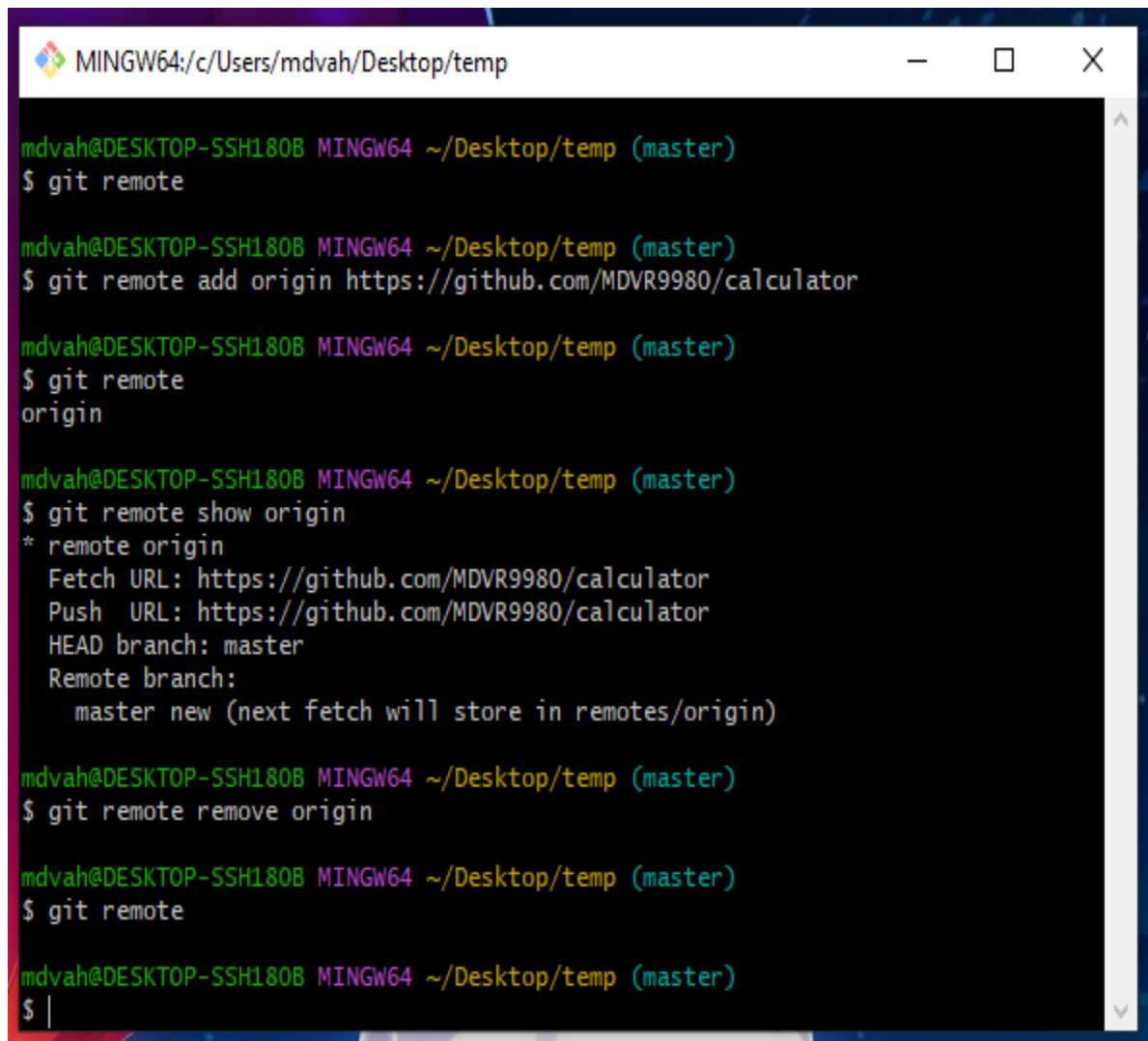
git remote rename <remote-name> <remote-new-name>

نمایش جزئیات نشانگر :

git remote show <name>

نمایش تمامی نام‌های نشانگرها و آدرس هایشان :

git remote -v

A screenshot of a Windows command prompt window titled "MINGW64:/c/Users/mdvah/Desktop/temp". The window has standard Windows window controls (minimize, maximize, close) in the top right corner. The command prompt shows a series of git commands and their outputs. The user is in the directory ~/Desktop/temp on the master branch. 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 shows "origin". 4. "git remote show origin" which displays details for the 'origin' remote: Fetch URL, Push URL, HEAD branch (master), and Remote branch (master new). 5. "git remote remove origin" which removes the 'origin' remote. 6. "git remote" which shows no output. 7. The prompt ends with a cursor on a new line.

```
mdvah@DESKTOP-SSH1808 MINGW64 ~/Desktop/temp (master)
$ git remote

mdvah@DESKTOP-SSH1808 MINGW64 ~/Desktop/temp (master)
$ git remote add origin https://github.com/MDVR9980/calculator

mdvah@DESKTOP-SSH1808 MINGW64 ~/Desktop/temp (master)
$ git remote
origin

mdvah@DESKTOP-SSH1808 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-SSH1808 MINGW64 ~/Desktop/temp (master)
$ git remote remove origin

mdvah@DESKTOP-SSH1808 MINGW64 ~/Desktop/temp (master)
$ git remote

mdvah@DESKTOP-SSH1808 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.

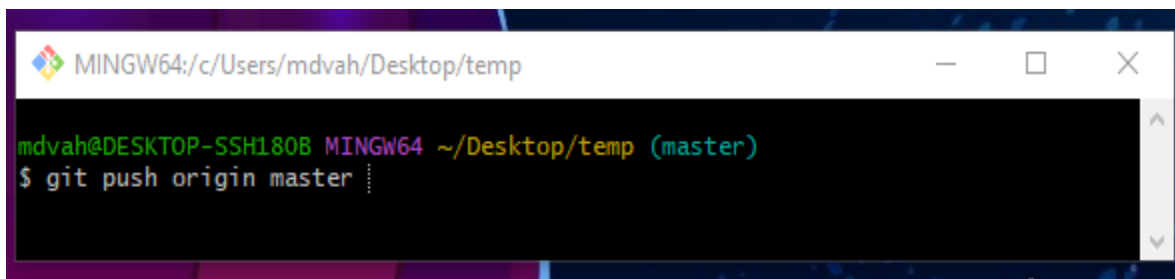
git push <remote-name> <branch-name>

: 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 می‌خواهید تغییرات آن را ارسال کنید.



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

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:

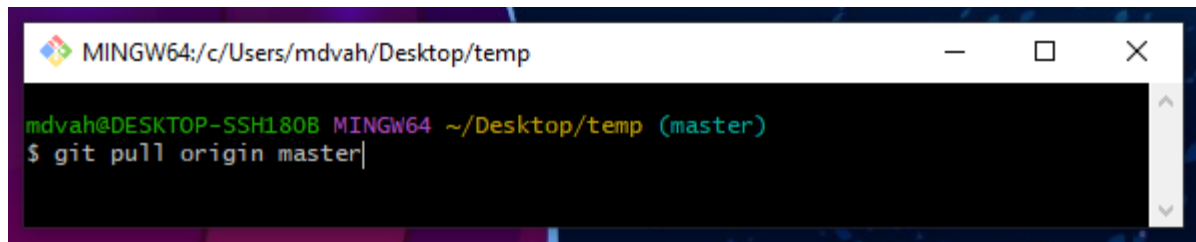
git pull <remote-name> <branch-name>

: 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|'.

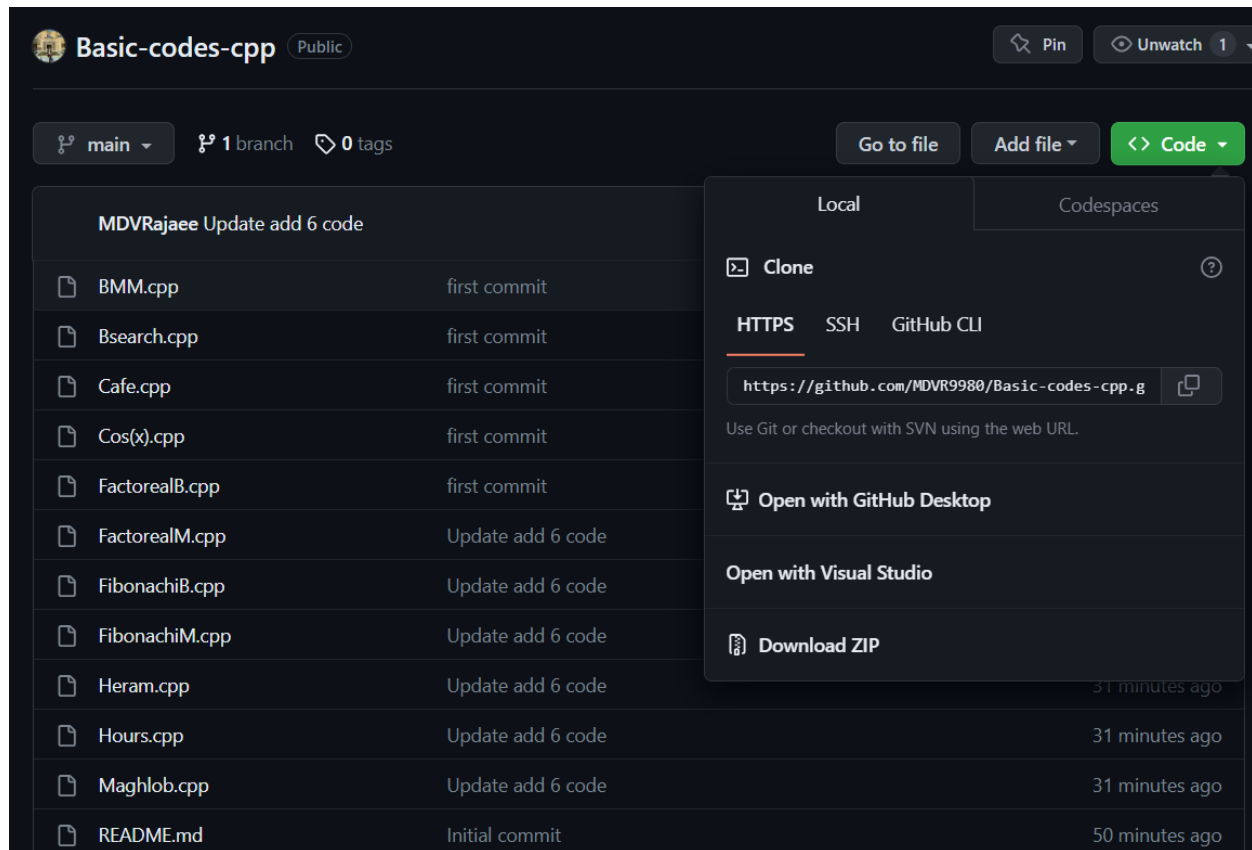
```
MINGW64:/c/Users/mdvah/Desktop/temp
mdvah@DESKTOP-SSH180B MINGW64 ~/Desktop/temp (master)
$ git pull origin master|
```

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.

دریافت و ارسال اطلاعات از گیت هاب



در ابتدا وارد اکانت خود در **github.com** می شویم و ریپوزیتوی که قصد داریم دریافتش کنیم و تغییراتی را روی آن اعمال کنیم و مجدد آن را روی **github.com** بارگذاری کنیم را **HTTPS، URL** اش را کپی می کنیم.

```
MINGW64:/e/Uni/Git
MDVR9980@DESKTOP-UROBLA5 MINGW64 /e/Uni (master)
$ mkdir Git

MDVR9980@DESKTOP-UROBLA5 MINGW64 /e/Uni (master)
$ cd Git

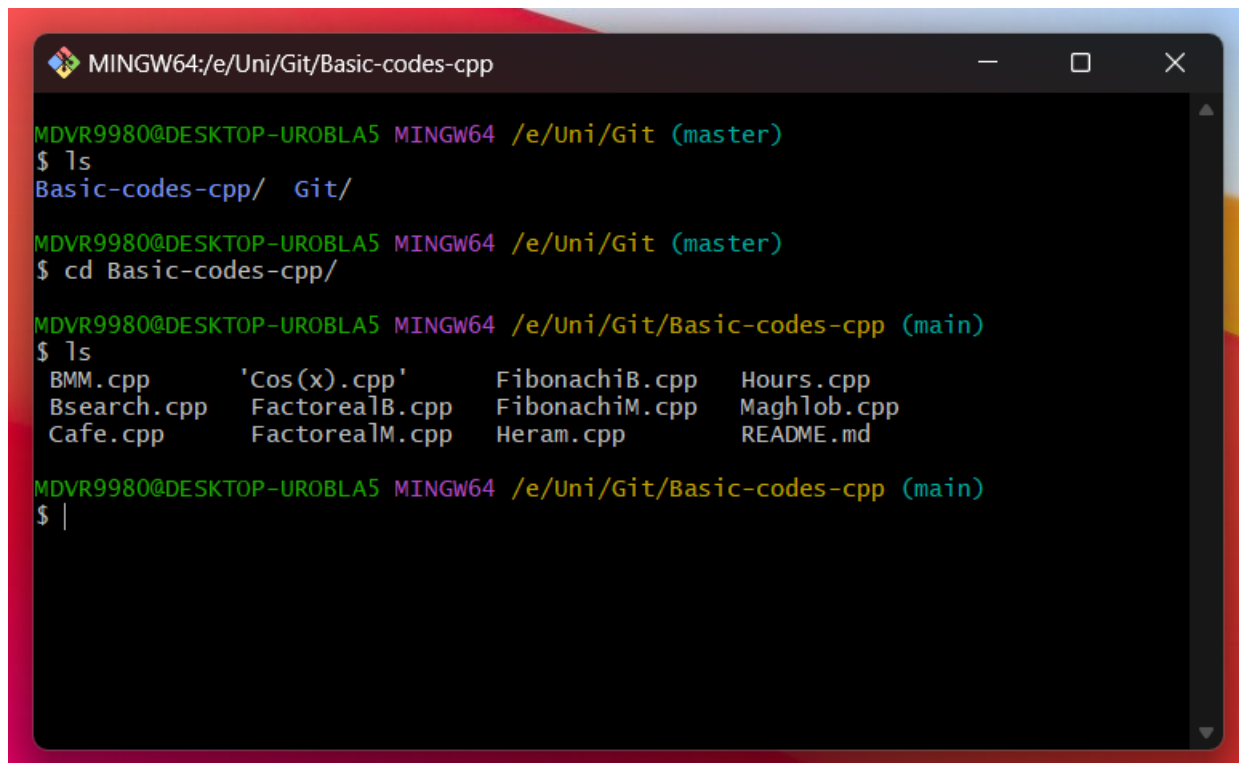
MDVR9980@DESKTOP-UROBLA5 MINGW64 /e/Uni/Git (master)
$ ls

MDVR9980@DESKTOP-UROBLA5 MINGW64 /e/Uni/Git (master)
$ git clone https://github.com/MDVR9980/Git.git
Cloning into 'Git'...
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 4 (delta 0), reused 4 (delta 0), pack-reused 0
Receiving objects: 100% (4/4), 2.70 MiB | 1.23 MiB/s, done.

MDVR9980@DESKTOP-UROBLA5 MINGW64 /e/Uni/Git (master)
$ ls
Git/

MDVR9980@DESKTOP-UROBLA5 MINGW64 /e/Uni/Git (master)
$ git clone https://github.com/MDVR9980/Basic-codes-cpp.git
```

با دستور **git clone** یک کپی روی سیستم لوکالمان از ریپوزیتوری میگیریم.



```
MINGW64:/e/Uni/Git/Basic-codes-cpp
MDVR9980@DESKTOP-UROBLA5 MINGW64 /e/Uni/Git (master)
$ ls
Basic-codes-cpp/  Git/
MDVR9980@DESKTOP-UROBLA5 MINGW64 /e/Uni/Git (master)
$ cd Basic-codes-cpp/
MDVR9980@DESKTOP-UROBLA5 MINGW64 /e/Uni/Git/Basic-codes-cpp (main)
$ ls
BMM.cpp      'Cos(x).cpp'  FibonacciB.cpp  Hours.cpp
Bsearch.cpp  FactorealB.cpp FibonacciM.cpp  Maghlob.cpp
Cafe.cpp     FactorealM.cpp Heram.cpp      README.md
MDVR9980@DESKTOP-UROBLA5 MINGW64 /e/Uni/Git/Basic-codes-cpp (main)
$ |
```

اطلاعات وارد سیستم‌مان می‌شود.

```
MINGW64:/e/Uni/Git/Basic-codes-cpp
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        Masahat zozenaghe.cpp
        Matrix.cpp

nothing added to commit but untracked files present (use "git add" to track)
MDVR9980@DESKTOP-UROBLA5 MINGW64 /e/Uni/Git/Basic-codes-cpp (main)
$ git add .

MDVR9980@DESKTOP-UROBLA5 MINGW64 /e/Uni/Git/Basic-codes-cpp (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   Masahat zozenaghe.cpp
        new file:   Matrix.cpp
```

دو تا کد جدیدی که می خواهیم به همراه دیگر کد ها آن ها را هم وارد **github.com** مون کنیم را در پوشه لوکال ریپوزیتوری قرار می دهیم و سپس با دستور **git add** آن ها را به وضعیت **stage** می بریم.


```
MINGW64:/e/Uni/Git/Basic-codes-cpp

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   Masahat zozenaghe.cpp
        new file:   Matrix.cpp

MDVR9980@DESKTOP-UROBLA5 MINGW64 /e/Uni/Git/Basic-codes-cpp (main)
$ git commit -m 'Update add 2 code'
[main 27c0f82] Update add 2 code
 2 files changed, 112 insertions(+)
 create mode 100644 Masahat zozenaghe.cpp
 create mode 100644 Matrix.cpp

MDVR9980@DESKTOP-UROBLA5 MINGW64 /e/Uni/Git/Basic-codes-cpp (main)
$ git status
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean

MDVR9980@DESKTOP-UROBLA5 MINGW64 /e/Uni/Git/Basic-codes-cpp (main)
$ |
```

با دستور `git commit -m 'Update add 2 code'` فایل هایی که در مرحله `stage` هستند رو `commit` می کنیم.

```
MINGW64:/e/Uni/Git/Basic-codes-cpp
Your branch is ahead of 'origin/main' by 1 commit.
(use "git push" to publish your local commits)

nothing to commit, working tree clean

MDVR9980@DESKTOP-UROBLA5 MINGW64 /e/Uni/Git/Basic-codes-cpp (main)
$ git remote -v
origin https://github.com/MDVR9980/Basic-codes-cpp.git (fetch)
origin https://github.com/MDVR9980/Basic-codes-cpp.git (push)

MDVR9980@DESKTOP-UROBLA5 MINGW64 /e/Uni/Git/Basic-codes-cpp (main)
$ git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 975 bytes | 975.00 KiB/s, done.
Total 4 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/MDVR9980/Basic-codes-cpp.git
746cde9..27c0f82 main -> main

MDVR9980@DESKTOP-UROBLA5 MINGW64 /e/Uni/Git/Basic-codes-cpp (main)
$
```

در آخر با دستور **git push** اطلاعات را به همراه تغییرات در گیت قرار می‌دهیم.

 MDVR9980 Update add 2 code

27c0f82 3 minutes ago 4 commits

 BMM.cpp	first commit	51 minutes ago
 Bsearch.cpp	first commit	51 minutes ago
 Cafe.cpp	first commit	51 minutes ago
 Cos(x).cpp	first commit	51 minutes ago
 FactorealB.cpp	first commit	51 minutes ago
 FactorealM.cpp	Update add 6 code	44 minutes ago
 FibonacciB.cpp	Update add 6 code	44 minutes ago
 FibonacciM.cpp	Update add 6 code	44 minutes ago
 Heram.cpp	Update add 6 code	44 minutes ago
 Hours.cpp	Update add 6 code	44 minutes ago
 Maghlob.cpp	Update add 6 code	44 minutes ago
 Masahat zozenaghe.cpp	Update add 2 code	3 minutes ago
 Matrix.cpp	Update add 2 code	3 minutes ago
 README.md	Initial commit	1 hour ago

README.md



Basic-codes-cpp

و در ریپوزیتوری یمان قرار میگیرند.

The End