

***دانشگاه حکیم سبزواری***

***دانشکده فنی برق و کامپیوتر***

*مستند**آموزشی*

*موضوع**: Git*

*نویسنده :‌*

*محمدداود وهاب رجائی*

***تیر 1402***

***فهرست :***

*نصب* ***3****…………………………………………………………… Git*

*دستور 5****……………………………………****…****…………………*** *init*

*دستور 6………………………………………………………status*

*دستور 9…………………………………………………… git add*

*دستور 10……………………………………………… git commit*

*دستور* ***11………………………………………………*** *git**branch*

*دستور* ***13……………………………………………****git**checkout*

*دستور 14………………………………………………… git reset*

*دستور 15………………………………………………… git clone*

*دستور 16……………………………………………… git remote*

*دستور 18………………………………………………… git push*

*دستور 19…………………………………………………… git pull*

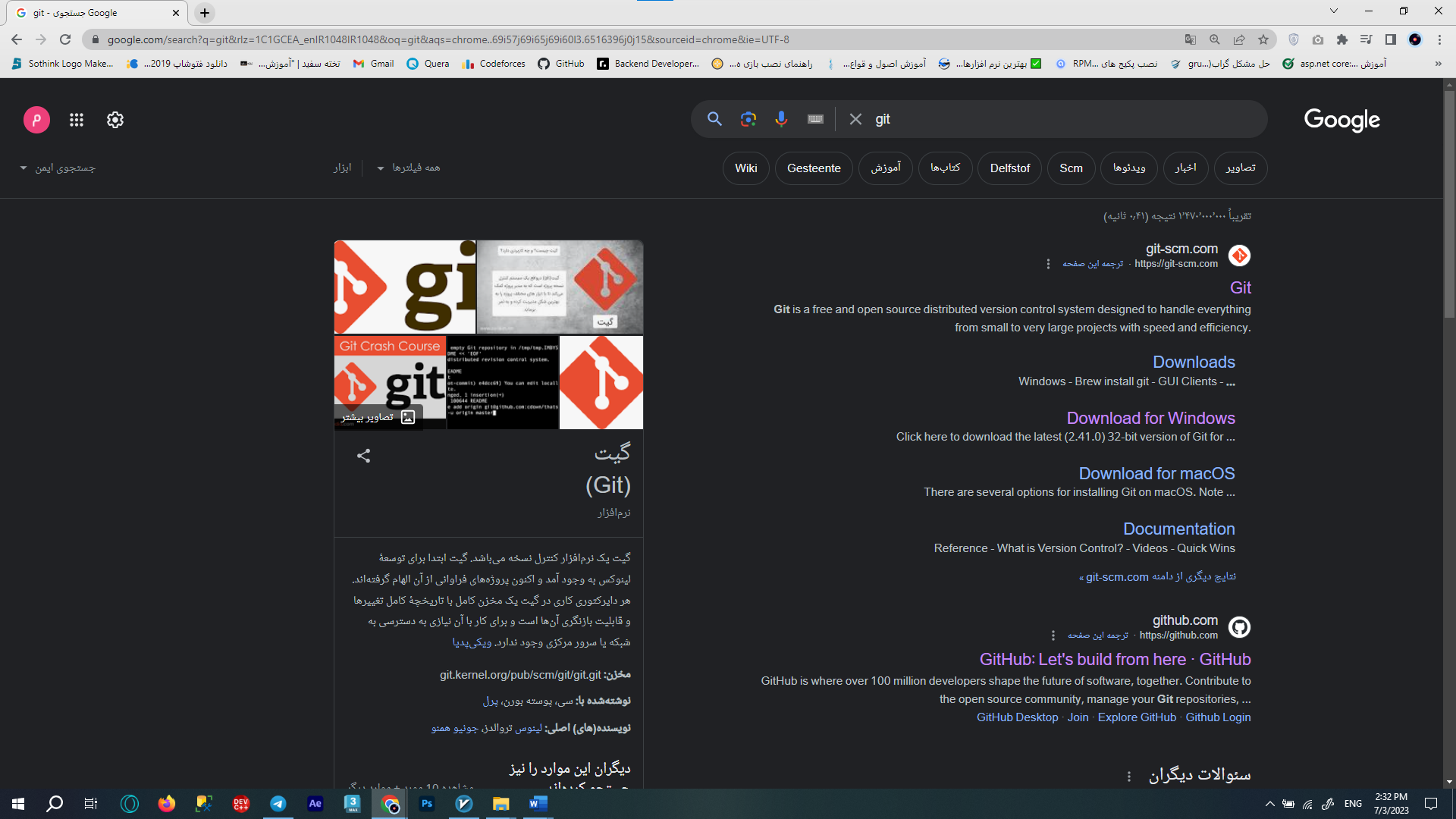
***دریافت و ارسال اطلاعات از گیت‌هاب................21.......................***

***نصب Git***

**ابتدا در مرورگر Git را جستجو می‌کنیم، وارد سایت**

[**https://git-scm.com/**](https://git-scm.com/)

**می‌شویم.**

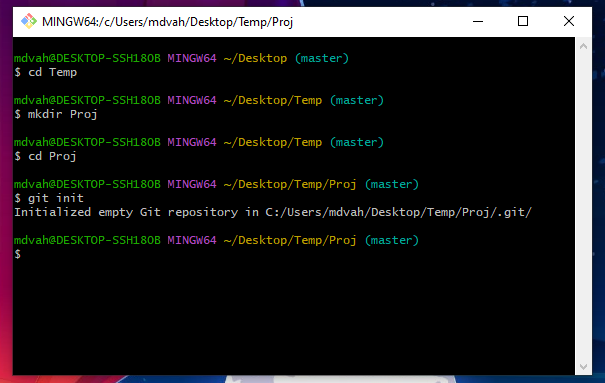
****

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



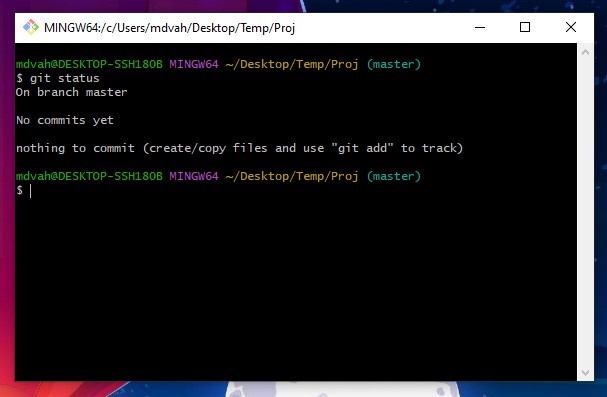
***دستور init***

**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.**

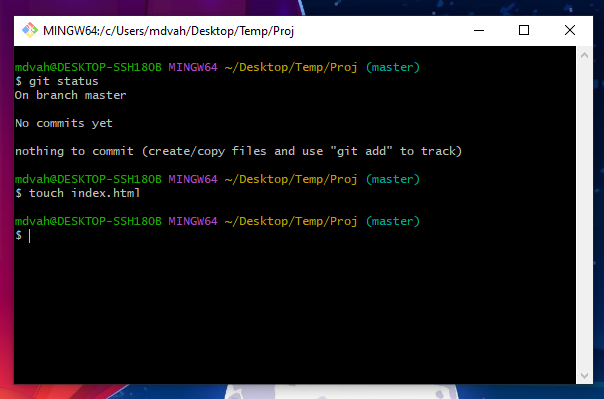
****

***دستور status***

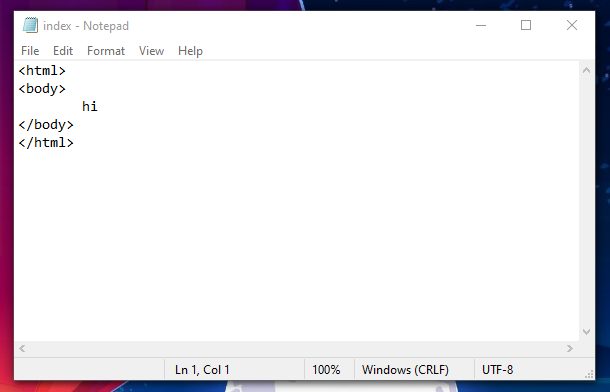
**وضعیت فایل‌های داخل دایرکتوری را مشخص می‌کند.**

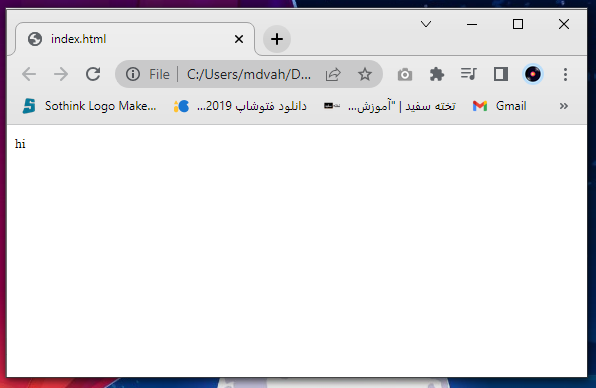
****

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

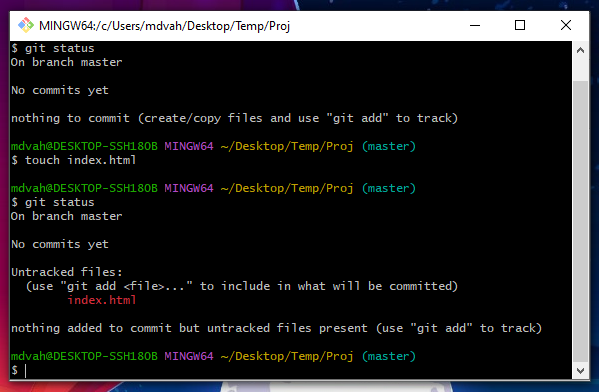


**وب پیج رو با NotePad باز کرده و با کد html ویرایشش می‌کنیم مثلاً چاپ hi**



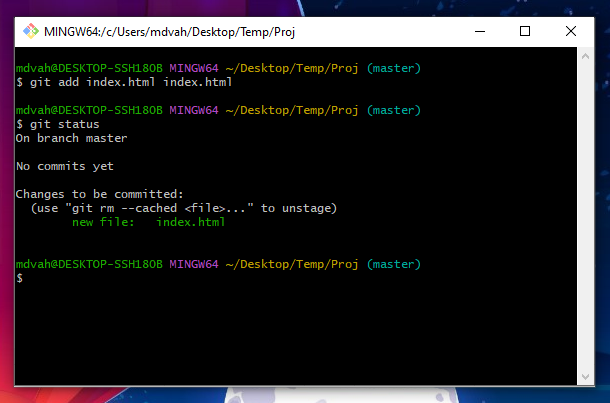


**با زدن دستور git status خواهیم دید که فایل index.html به عنوان فایل شناخته نشده در گزارش می‌شود.**



*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، می شود.**



**و دستورات دیگر در add کردن :**

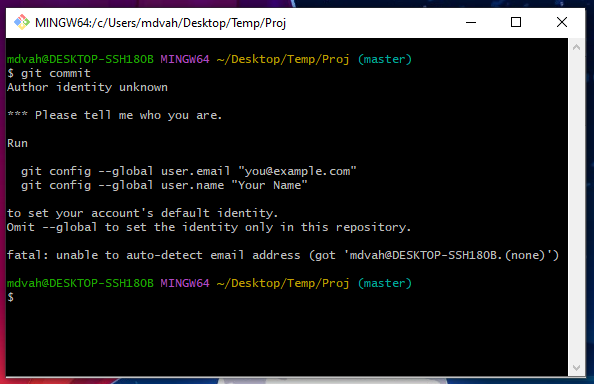
**git add “\*.html”**

**git add -A**

*دستور git commit*

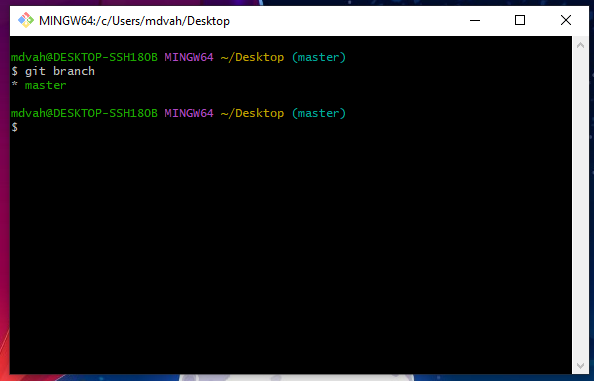
**با دستور -m ‘comment’ git commit فایل‌هایی که در وضعیت stage قرار گرفته‌اند با دستور :**

**git add <file-name>**



***دستور git branch***

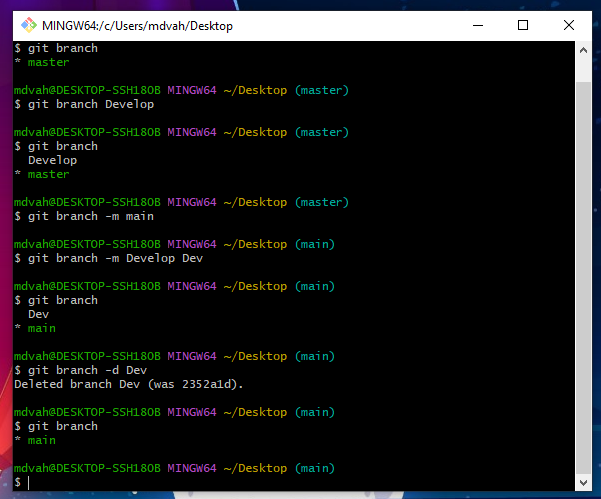
**دستور git branch تمامی شاخه‌هایی که داریم نشان می‌دهد، شاخه فعلی که در آن هستیم را با \* نشان می‌دهد و رنگ سبز به آن اختصاص می‌دهد.**

****

**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 می‌خوایم حذف کنیم باشیم.  
با دستور git branch -m <branch-name> می‌توانیم نام شاخه اصلی را عوض کنیم و با دستور**

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

**نام هر شاخه‌را می‌توانیم عوض کنیم.**

****

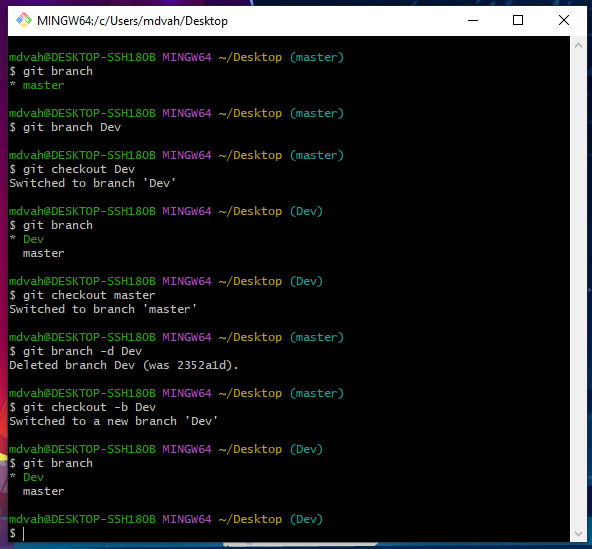
***دستور git checkout***

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

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

**شاخه‌جدید می‌سازیم و به آن شاخه منتقل می‌شویم و با دستور :**

**git checkout <file-name>**

**فایل مورد نظر را به حالت قبلی برمی‌گردانیم.**

***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.**

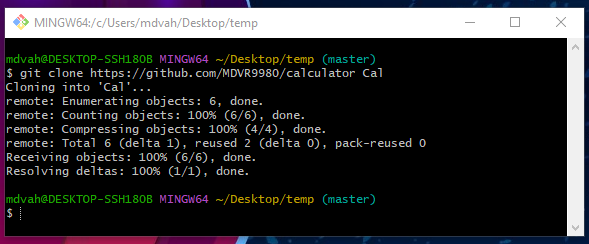
***دستور git clone***

**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 پوشه ای که با نام مخزن ایجاد می‌شود، استفاده می‌شود.**



***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 :**

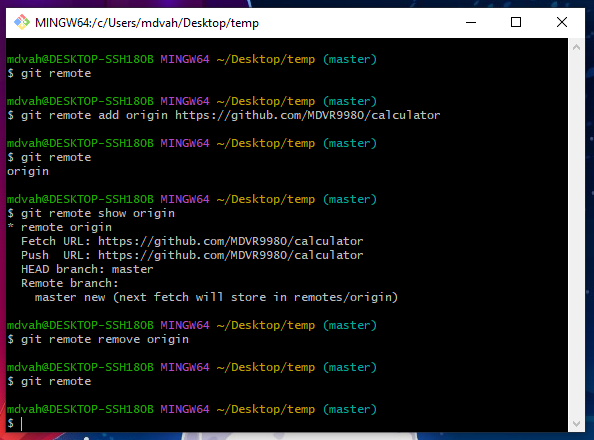
**git remote rename <remote-name> <remote-new-name>**

**نمایش جزئیات نشانگر :**

**git remote show <name>**

**نمایش تمامی نام‌های نشانگر‌ها و آدرس هایشان‌ :**

**git remote -v**

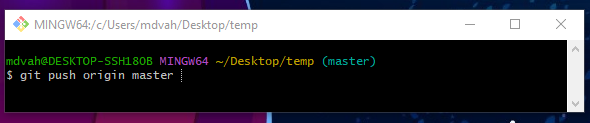
****

***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.**

****

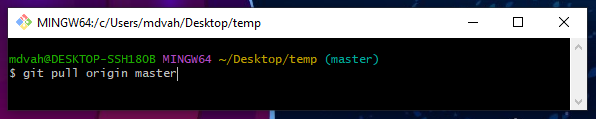
***دستور git pull***

**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.**

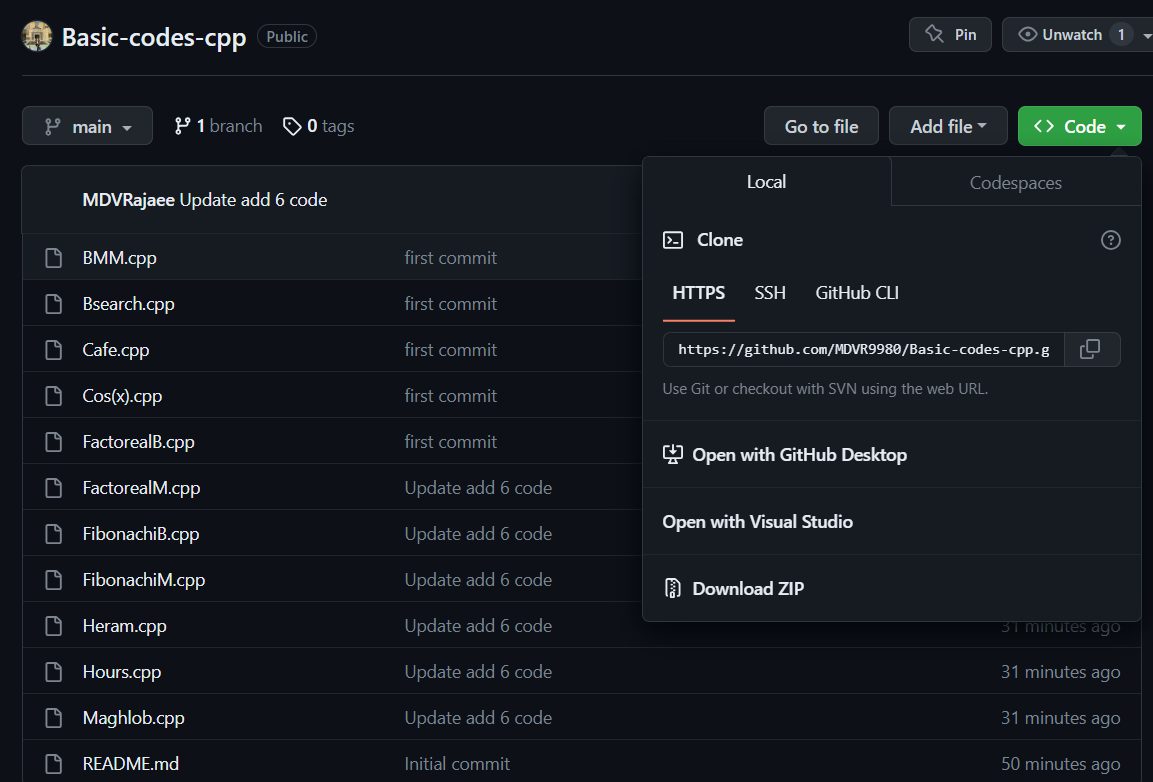
**branch-name‌ :  
نام شاخه محلی است که می‌خواهید تغییرات را در آن ادغام کنید.**



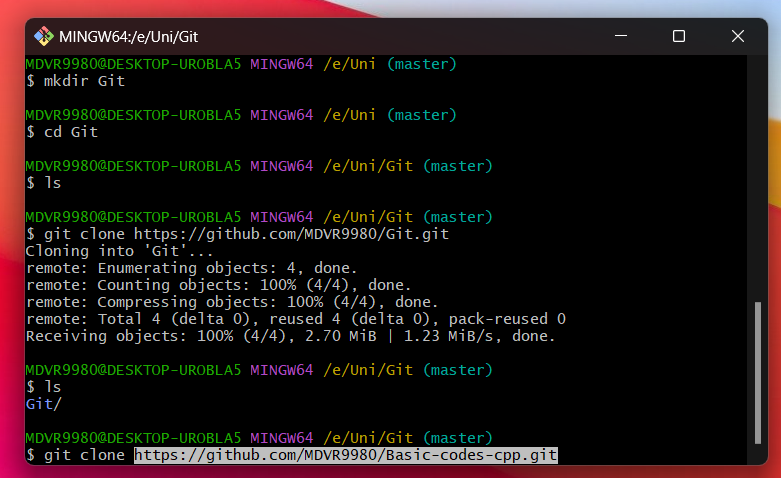
**در اینجا origin نام نشانگر مخزن اصلی است و master نام شاخه محلی است که میخواهید تغییرات را در آن ادغام کنید.**

**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.**

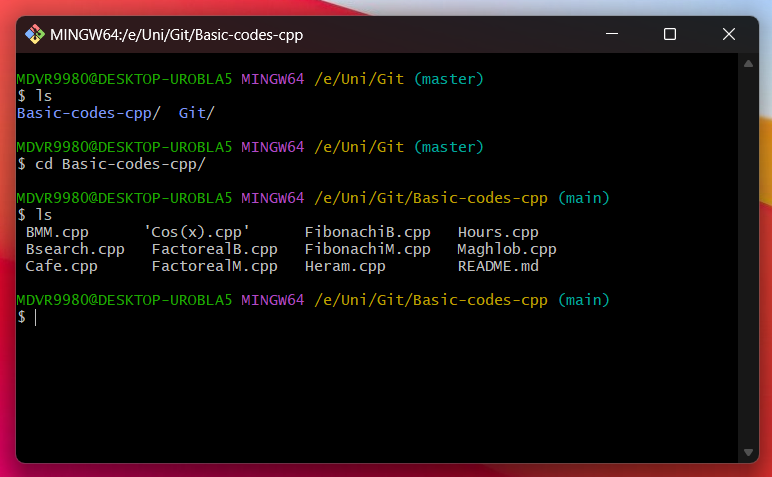
**توجه داشته باشید که قبل از اجرای دستور git pull، مناسب است تغییرات محلی خود را با دستور git commit ثبت کنید تا دارای تاریخچه قبل از ادغام تغییرات باشید.**

***دریافت و ارسال اطلاعات از گیت‌هاب***

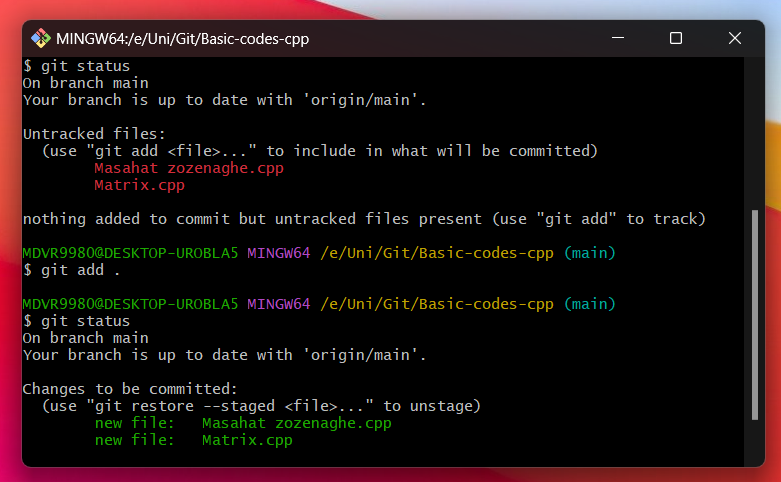
**در ابتدا وارد اکانت خود در github.com می‌شویم و ریپوزیتوی که قصد داریم دریافتش کنیم و تغییراتی را روی آن اعمال کنیم و مجدد آن را روی github.com بارگذاری کنیم را HTTPS، URL اش را کپی می‌کنیم.**



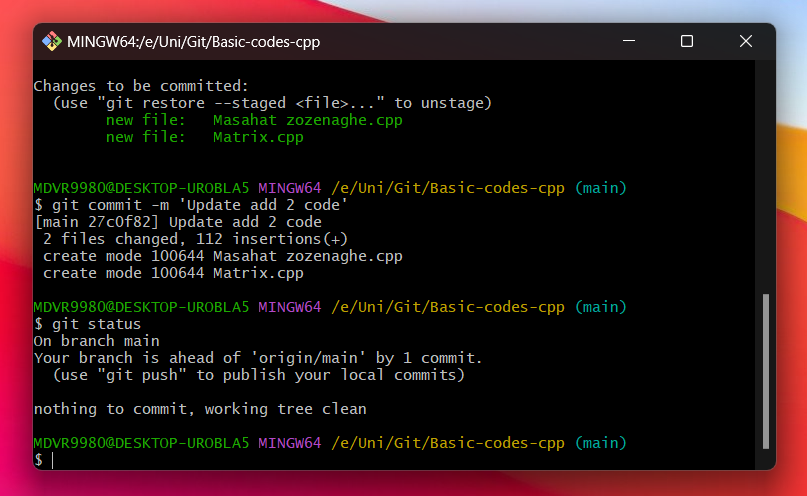
**با دستور git clone یک کپی روی سیستم لوکالمان از ریپوزیتوری میگیریم.**



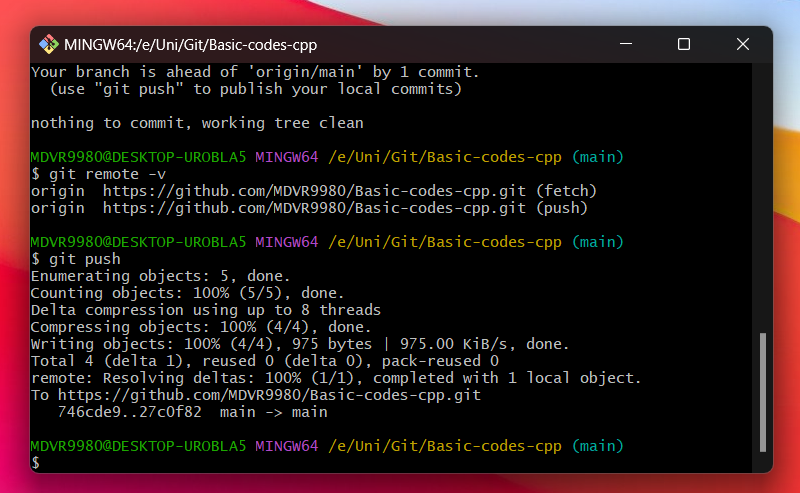
**اطلاعات وارد سیستم‌مان می شود.**



**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 می‌دهیم و سپس با دستور git add . آن ها را به وضعیت stage می‌بریم.**



**با دستور git commit -m ‘Update add 2 code’ فایل هایی که در مرحله stage هستند رو commit می‌کنیم.**



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

****

**و در ریپوزیتوری یمان قرار میگیرند.**

***The End***