



Python programmering

Föreläsning 5





Git & GitHub

ONE DOES NOT SIMPLY

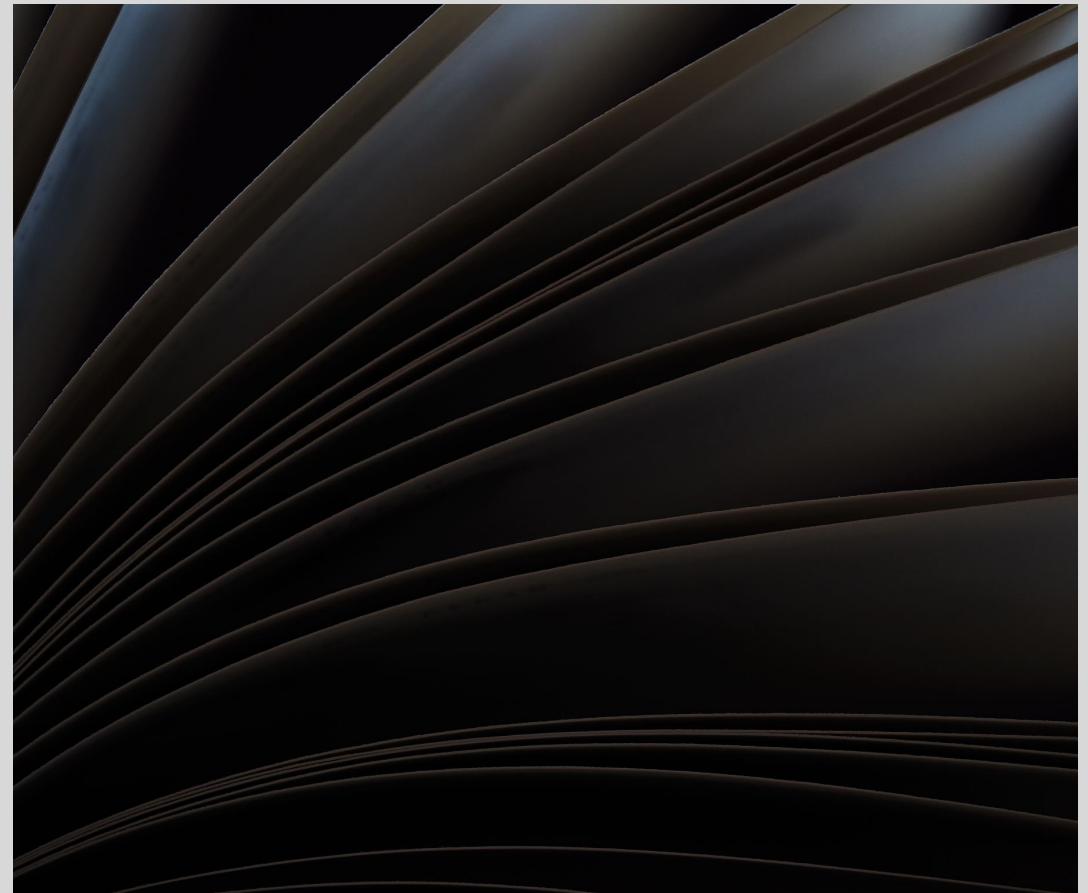
UNDERSTAND GIT

MemesHappen

Versionshantering

Vad innebär versionshantering?

- Spara data över hur filer/data har sett ut och förändrats över tid.
- Möjlighet att "spola tillbaka".
- "Loggbok" över *vem* som ändrat något, *när* det ändrades, *vad* som ändrades.
- Verktyg när man felsöker kod.



Förklarar Git väldigt mycket bättre än mig

- https://www.youtube.com/watch?v=2ReR1YJrNOM&ab_channel=ProgrammingwithMosh

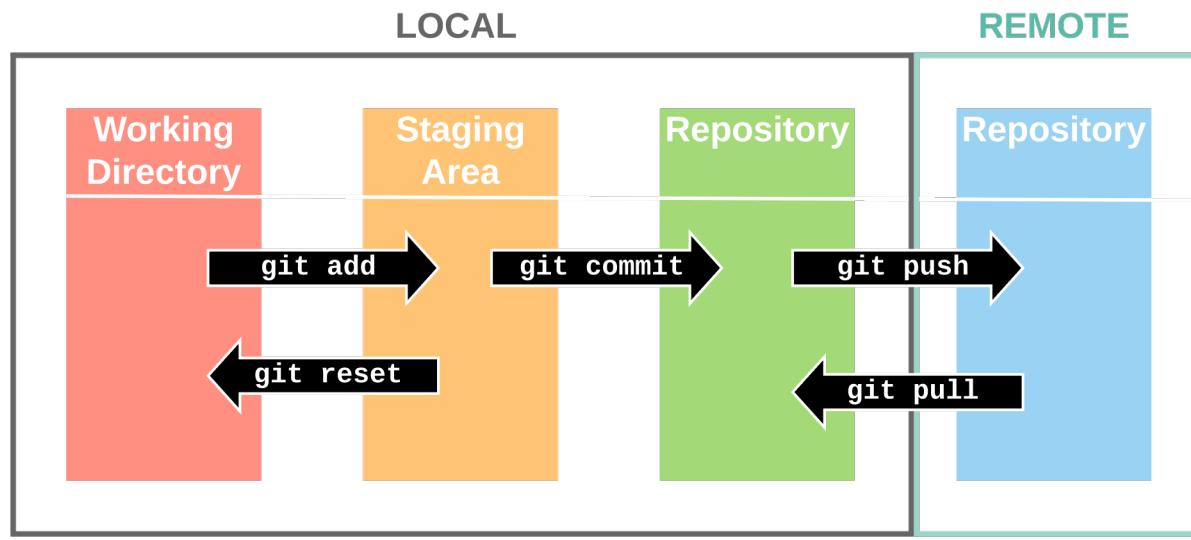


git

Git

- Git är ett verktyg som används för versionshantering .
- Används av nästan alla utvecklare, både i industrin, inom akademien och för privata projekt
- Skapades av Linus Torvalds
- Git är ett distribuerat versionshanteringssystem, vilket innebär att inget centralt arkiv krävs och att vem som helst kan skapa en egen kopia.
- Linux och MacOS kommer med git installerat som standard men kan vara gammal version
- Ladda ner och installera git på länken nedan för att installera senaste versionen.
- Länk: <https://git-scm.com/downloads>
- Dokumentation om git (eng): <https://git-scm.com/book/en/v2>
- Dokumentation om git (sve): <https://git-scm.com/book/sv/v2/>
- Git fungerar lokalt på din dator

Git struktur



- Git är ofta strukturerat i två delar: **local** och **remote**
 - **Local** = det som finns på din dator, det vill säga: ett lokalt repo, en staging area och din working directory (din kopia som du jobbar i).
 - **Remote** = ett repos som finns på en delad server och fungerar som master/main för projektet. Ofta används t.ex. Github för att hantera master/main-repot.

Git terminologi

Local branch

- Det är branchen på din lokala dator, den är inte samma som den som finns på servern (i molnet).

Remote branch

- En branch på servern/ i molnet.

Tracking branch

- En lokal branch kan (men måste inte) vara en tracking branch, om den har en motsvarande remote branch. När man säger "git push" i en tracking branch är det dess motsvarande remote branch som uppdateras.

HEAD

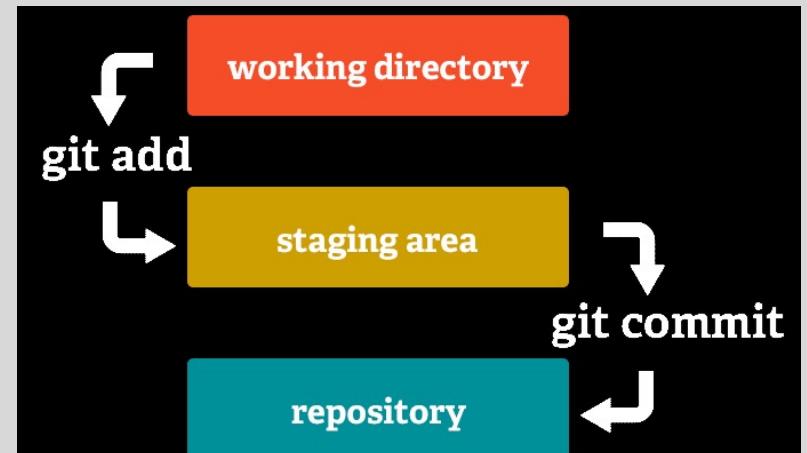
- Det är Gits sätt att referera till den nuvarande ögonblicksbilden av koden. När du använder kommandot "git checkout" så säger du åt Git att uppdatera så att Head pekar på den valda branchen eller commiten.

Working tree

- Trädstrukturen för dem utcheckade filerna på din dator (innehållet i Head-commitens träd plus lokala ändringar som du inte har commitat än).

Staging area

- Det är där dina ändringar hamnar när du har kört kommandot "git add". Där ligger de tills du har gjort en commit.



"Imagine a box. You can put stuff into the box. You can take stuff out of the box. This box is the staging area of Git. You can craft commits here. Committing is like sealing that box and sticking a label on it. The contents of that box are your changes. So, why not have the label mean something? You wouldn't label a moving box with kitchen items as simply "stuff.""

Vanliga Git Kommandon

GIT KOMMANDON:

<https://confluence.atlassian.com/bitbucketserver/basic-git-commands-776639767.html>

GIT TERMINOLOGI:

<https://www.atlassian.com/git/glossary/terminology>

- **git init** – Skapa ett nytt lokalt repo
- **git clone** – Clona ett lokalt eller remote repo
- **git add** – Lägg till en flera filer till stagingmiljön
- **git commit** – Lägg till förändringar till head men inte till remote (än)
- **git push** – Skicka upp förändringarna till remota branchen
- **git status** – Lista filer som har ändrats
- **git pull** – Dra ned förändringar från din remota branch till din lokala branch
- **git branch** – Lista de branches som existerar lokalt på din maskin
- **git merge** - Slå ihop utvecklingsgrenar. Till exempel används detta när man vill slå ihop koden man har utvecklat i en "feature"-branch med koden i main-branchen.
- **git checkout** – används för att byta mellan olika branches, t.ex. **git checkout master** (byter från grenen du nyss stod i till master-branchen, från och med nu arbetar du i master-branchen, tills du har checkat ut någon annan branch istället).

Komma igång

- Din identitet:

```
:~$ git config --global user.name "Anna Larsson"  
:~$ git config --global user.email anna.larsson@example.com
```

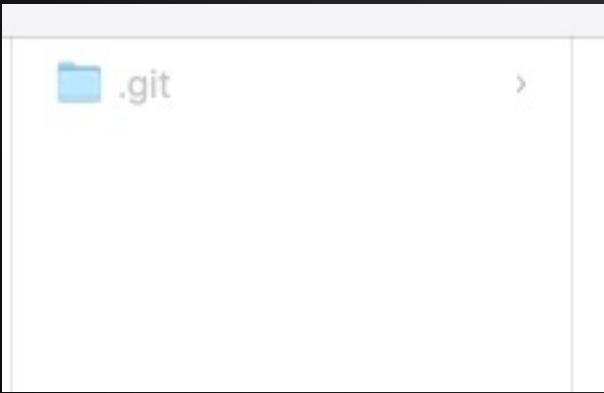
- Lista dina git inställningar:

```
:~$ git config --list  
  
user.name=Anna Larsson  
user.email=anna.larsson@example.com
```

- Skulle du behöva ändra manuellt i config filen kan du skriva:

- ”nano ~/.gitconfig” eller ”code ~/.gitconfig” – Linux & MacOS, ”code” öppnar filen i VSCode
- ”code .gitconfig” – skriv i din användares hem mapp, startar VS Code (om det är installerat)

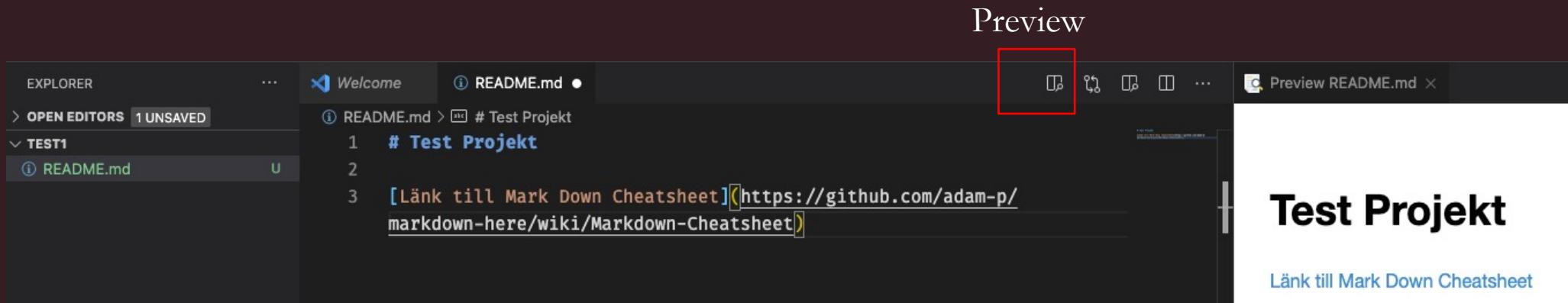
Skapa ett lokalt Git-repo



- Repo = Repository. Centralt fillagringsplats för att lagra flera versioner av filer
- ~ = hem-mapp
- cd = ändra mapp
- ls = lista innehåll i mapp
- Öppna kommandotolken/terminalen
- Skapa en Git mapp ”mkdir git” i:
 - Windows ”c:\git”
 - Mac & Linux ”~/git”
- Vi skapar foldern ovan för att enkelt spara alla repon i den
- Skapa en ny projekt mapp ”python” i git-mappen
 - ”mkdir python”
 - ”cd python”
- OBS! Har ni redan en mapp ni tycker om att jobba i där ni har virtuella miljön skapar ni INTE en ny mapp, men hoppar ovan steg
- För att börja versionshantera mappen skriver vi
 - ”git init”
 - Nu är mappen redo för versionshantering och en ny dold mapp (”.git”) skapades i din projekt mapp
- Skriv nu ”code .” för att öppna foldern i VSCode

Skapa en fil

- Skapa en ny fil README.md (md står för Mark Down och är ett simpelt text format syntax)
- Skriv in ”# Test Projekt”
- Samt ” [Länk till Mark Down Cheatsheet](<https://github.com/adam-p/markdown-here/wiki/Markdown-Cheatsheet>)”
- Spara filen



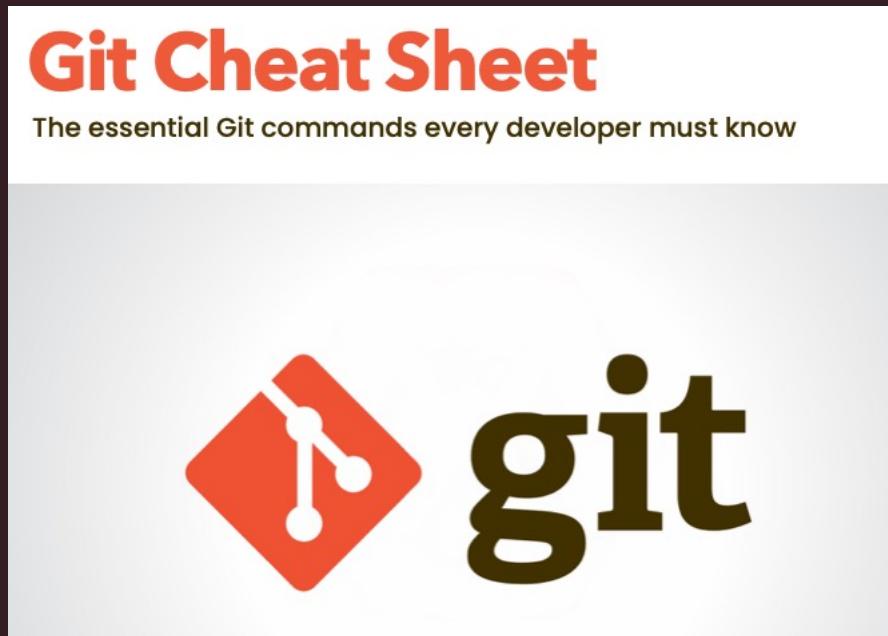
Öppna terminalen i VSCode

- Klicka på fliken ”terminal -> New Terminal” för att öppna en terminal i VSCode
- Gör filen Versionshanteringsbar:
 - ”git status” – Visar status för mappen. Röd betyder det är ändringar som inte är committed
 - ”git add filnamn.filändelse” – förbereder (staging) filen till nästa ”commit”
 - ”git add .” - förbereder all filer i mappen
 - ”git commit –m ’meddelande’ ” - Verkställer alla förberedda (staged files) filer
 - Allt som lades till med ”git add” är nu sparat i versionshanteringssystemet git

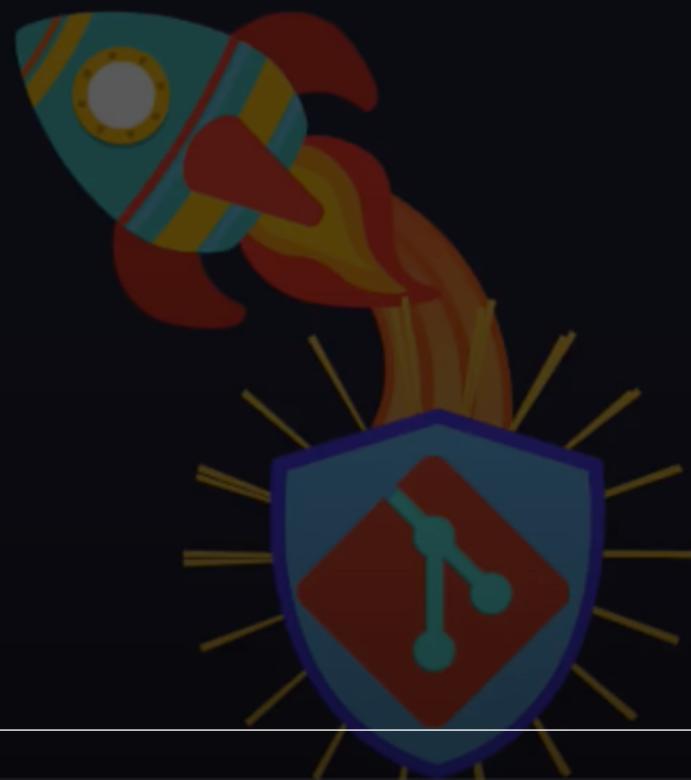
```
git base ~/G/test1 ➜ ... git status  
On branch master  
  
No commits yet  
  
Untracked files:  
  (use "git add <file>..." to include in what will be committed)  
    README.md  
  
nothing added to commit but untracked files present (use "git add" to track)  
git base ~/G/test1 ➜ ... git add README.md  
git base ~/G/test1 ➜ ... git status  
On branch master  
  
No commits yet  
  
Changes to be committed:  
  (use "git rm --cached <file>..." to unstage)  
    new file:   README.md  
  
git base ~/G/test1 ➜ ... git commit -m "Created a readme"  
[master (root-commit) 4a880be] Created a readme  
  1 file changed, 0 insertions(+), 0 deletions(-)  
  create mode 100644 README.md  
git base ~/G/test1 ➜ ... git status  
On branch master  
nothing to commit, working tree clean  
git base ~/G/test1 ➜ ...
```

Git Cheat Sheet av Mosh

- <https://programmingwithmosh.com/wp-content/uploads/2020/09/Git-Cheat-Sheet.pdf>



LEARN GIT IN 15 MINUTES



◦ <https://youtu.be/USjZcfj8yxE>

GitHub

- Github är en webbaserad och centraliserad lagring av versionshistorik för programvaruutvecklingsprojekt som använder versionshanteringssystemet Git.
- Github erbjuder gratis lagring men även abonnemang för extra funktionalitet.
- Man måste skapa ett konto: <https://github.com>





- När du är inloggad letar du fram knappen ”New” eller ”+” tecknet, fyll i enigt bild och välj ”Create repository”

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository](#).

Owner * **Repository name ***

deskavaenkelt / test ✓

Great repository names are short and memorable. Need inspiration? How about [ideal-adventure](#)?

Description (optional)

Test projekt för att demonstrera Git & Github

Public
Anyone on the internet can see this repository. You choose who can commit.

Private
You choose who can see and commit to this repository.

Initialize this repository with:
Skip this step if you’re importing an existing repository.

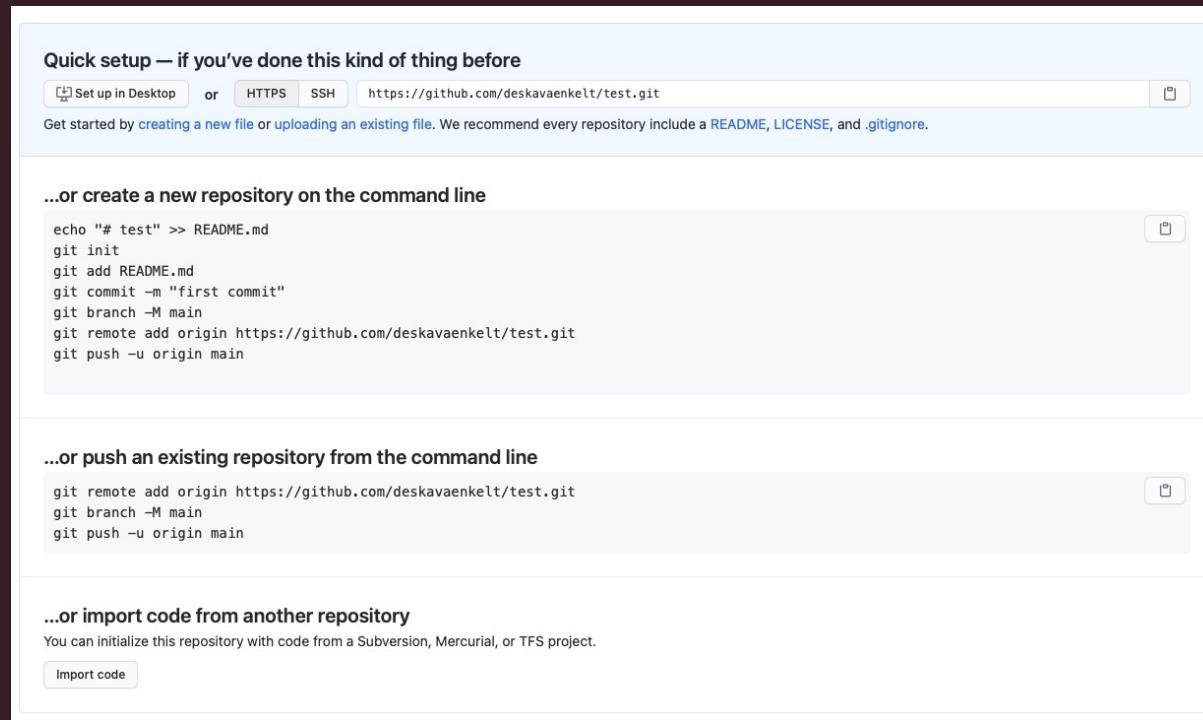
Add a README file
This is where you can write a long description for your project. [Learn more](#).

Add .gitignore
Choose which files not to track from a list of templates. [Learn more](#).

Choose a license
A license tells others what they can and can’t do with your code. [Learn more](#).

Create repository

Ladda upp vårt lokala repo till GitHub



- Då vi redan gjort fram till ”git commit” steget kan vi börja på ”..or push an existing repository from the command line”
- Du kommer behöva ditt användarnamn och lösenord för att kunna köra ”git push”
- Du kan även skapa SSH nycklar vilket man kan göra senare.
- SSH nycklar är ett alternativt sätt att identifiera sig utan att varje gång ange användarnamn och lösenord

Darwin iMac.local 20.1.0 x86_64
 15:36 up 7 days, 3:02, 3 users, load averages: 2.27 2.31 2.27
 * base ~/G/test1 ➜ git status

On branch master

No commits yet

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

nothing added to commit but untracked files present (use "git add" to track)
 * base ~/G/test1 ➜ git add README.md
 * base ~/G/test1 ➜ git status

On branch master

No commits yet

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

* base ~/G/test1 ➜ git commit -m "Created a readme"
 [master (root-commit) 4a880be] Created a readme
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 README.md

* base ~/G/test1 ➜ git status

On branch master

nothing to commit, working tree clean

* base ~/G/test1 ➜ git branch -M main

* base ~/G/test1 ➜ git remote add origin https://github.com/deskavaenkelt/test.git

* base ~/G/test1 ➜ git push -u origin main

win iMac.local 20.1.0 x86_64

36 up 7 days, 3:02, 3 users, load averages: 2.27 2.31 2.27
 * base ~/G/test1 ➜ git status

branch master

commits yet

tracked files:
 (use "git add <file>..." to include in what will be committed)
 README.md

nothing added to commit but untracked files present (use "git add" to track)
 * base ~/G/test1 ➜ git add README.md

* base ~/G/test1 ➜ git status

branch master

commits yet

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

* base ~/G/test1 ➜ git commit -m "Created a readme"
 [master (root-commit) 4a880be] Created a readme
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 README.md

* base ~/G/test1 ➜ git status

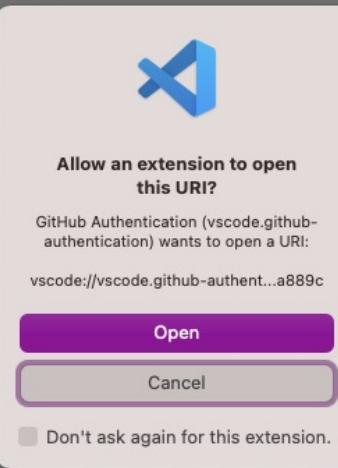
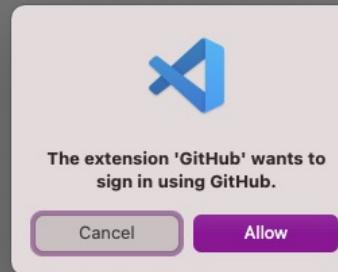
branch master

nothing to commit, working tree clean

* base ~/G/test1 ➜ git branch -M main

* base ~/G/test1 ➜ git remote add origin https://github.com/deskavaenkelt/test.git

* base ~/G/test1 ➜ git push -u origin main



I terminalen i VSCode

- git remote add origin git@...
- git branch -M main
- git push -u origin main
- När du gör push kommer dialogrutan upp om att ansluta till GitHub om den inte redan är anslutit
- Följ instruktionerna efter du klickat på Allow.
- Innan du klickar på Open kan du klicka i rutan “Don’t ask again for this extension”

```
* base ~/G/test1 ➜ git status
On branch master
nothing to commit, working tree clean
* base ~/G/test1 ➜ git branch -M main
* base ~/G/test1 ➜ git remote add origin https://github.com/deskavaenkelt/test.git
* base ~/G/test1 ➜ git push -u origin main
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 226 bytes | 226.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/deskavaenkelt/test.git
 * [new branch]      main -> main
Branch 'main' set up to track remote branch 'main' from 'origin'.
* base ~/G/test1 ➜ main
```

Resultat

- Uppdatera GitHub sidan med ditt repo

Quick setup — if you've done this kind of thing before

Set up in Desktop or HTTPS SSH https://github.com/deskavaenkelt/test.git

Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

...or create a new repository on the command line

```
echo "# test" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/deskavaenkelt/test.git
git push -u origin main
```

...or push an existing repository from the command line

```
git remote add origin https://github.com/deskavaenkelt/test.git
git branch -M main
git push -u origin main
```

...or import code from another repository

You can initialize this repository with code from a Subversion, Mercurial, or TFS project.

[Import code](#)

- Uppdatera GitHub sidan med ditt repo

main 1 branch 0 tags

Go to file Add file Code

deskavaenkelt Created a readme 0b218d9 18 seconds ago 2 commits

README.md Created a readme 18 seconds ago

README.md

Test Projekt

Länk till Mark Down Cheatsheet

Varje gång man vill versionshantera en ändring

A screenshot of a terminal window with two tabs: "README.md" and "Preview README.md". The "README.md" tab shows the following content:

```
You, seconds ago | 1 author (You)
1 # Test Projekt
2
3 [Länk till Mark Down Cheatsheet]
  (https://github.com/adam-p/markdown-here/wiki/Markdown-Cheatsheet)
4
5 En ändring
```

The "Preview README.md" tab shows a preview of the README content:

Test Projekt

Länk till Mark Down Cheatsheet

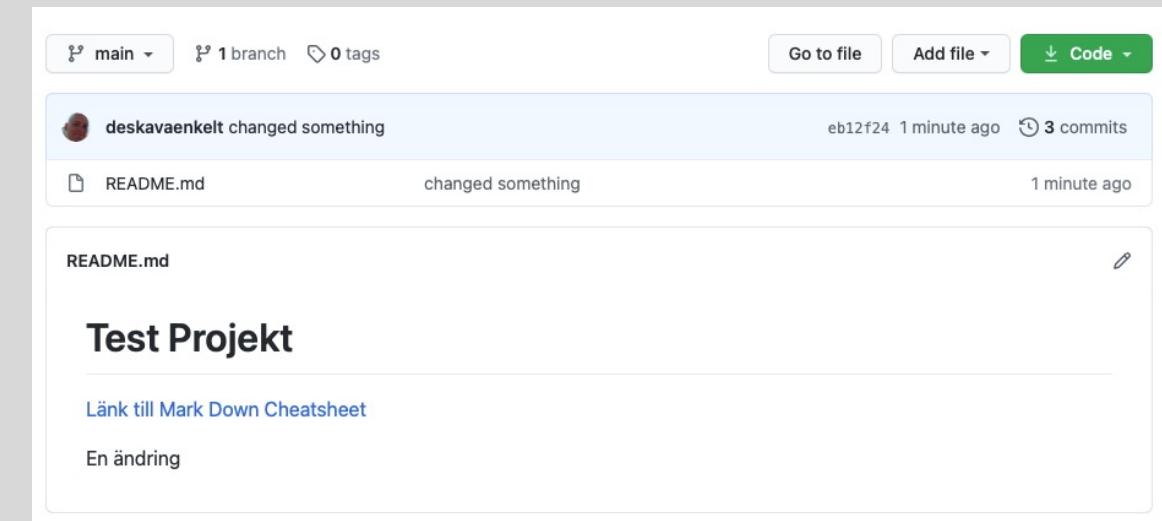
En ändring

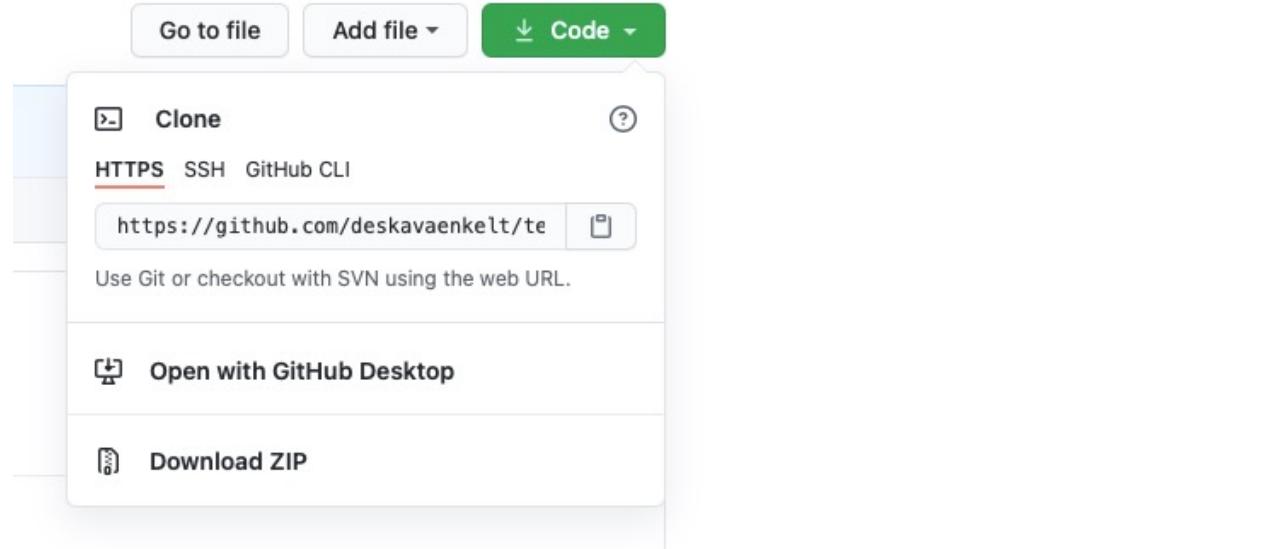
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

```
git add .
git commit -m "changed something"
git push
```

Terminal output:

```
[main eb12f24] changed something
  1 file changed, 3 insertions(+), 1 deletion(-)
git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 12 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 298 bytes | 298.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/deskavaenkelt/test.git
  0b218d9..eb12f24  main -> main
git push
```





```
git clone https://github.com/deskavaenkelt/test.git
Cloning into 'test'...
remote: Enumerating objects: 9, done.
remote: Counting objects: 100% (9/9), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 9 (delta 1), reused 9 (delta 1), pack-reused 0
Receiving objects: 100% (9/9), done.
Resolving deltas: 100% (1/1), done.
```

```
cd test/
total 8
-rw-r--r-- 1 lars staff 128B Jan  2 16:41 README.md
cat README.md
# Test Projekt

[Länk till Mark Down Cheatsheet](https://github.com/adam-p/markdown-here/wiki/Markdown-Cheatsheet)

En ändring
base ~/G/o/test main
```

Klona ett repo (kopiera till din dator)

- Klicka på ”Code-knappen”
- Kopiera länken med ”HTTPS” ifyllt
- Har du aktiverat SSH så kan du använda den länken

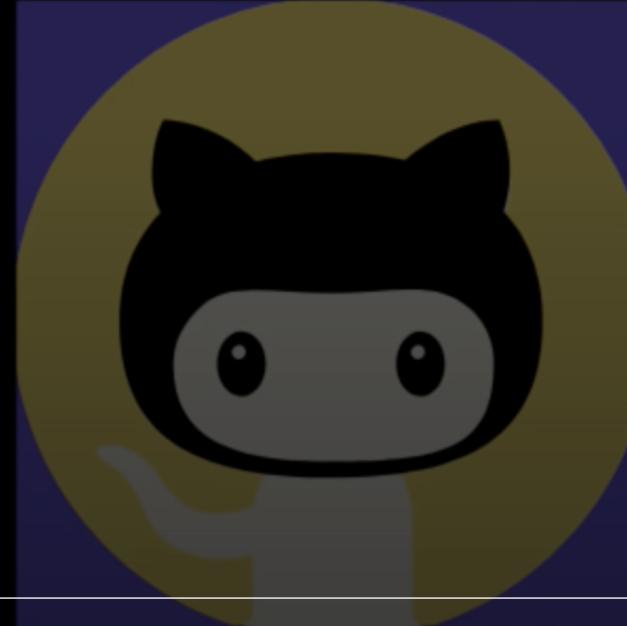
- Öppna terminalen
- Gå till din git mapp om du inte är i den
- I din ”git-mapp” skapa en mapp som heter ”other”
 - ”mkdir other && cd other”
- Skriv:
 - ”git clone https://github.com/ditt_github_namn/test.git”



GITHUB

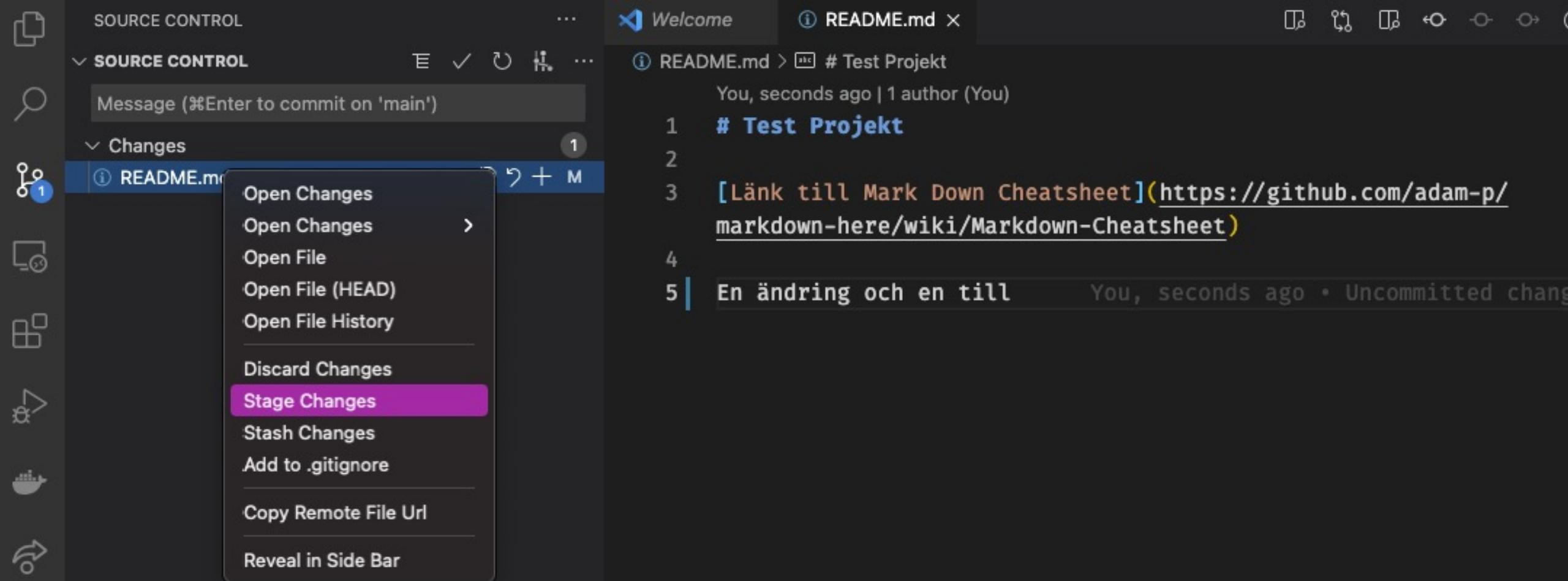
In 20 Minutes

- <https://www.youtube.com/watch?v=nhNq2kIvi9s&feature=youtu.be>



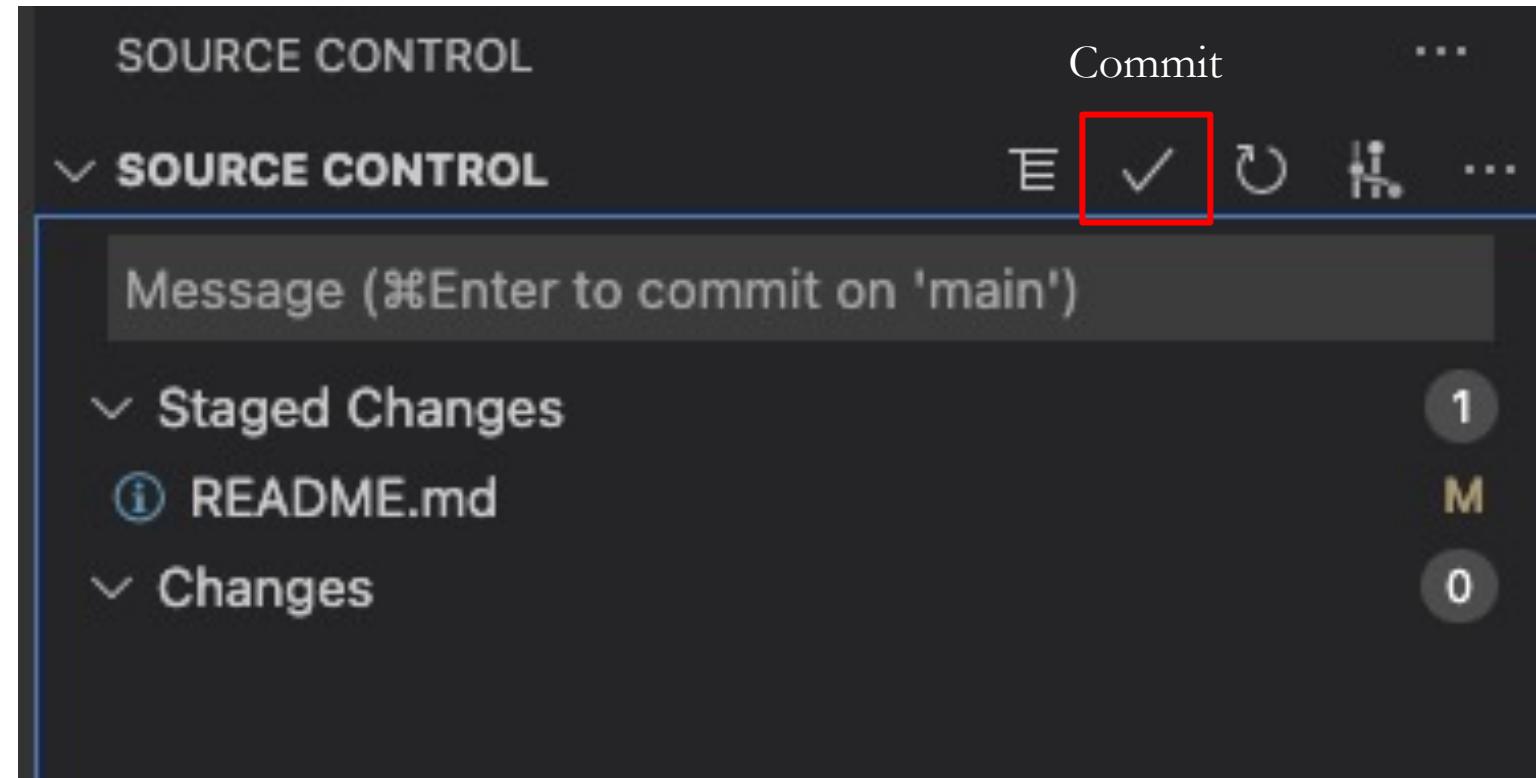
Övning

- Skapa ett par egna repor och radera dem
- <https://docs.github.com/en/free-pro-team@latest/github/administering-a-repository/deleting-a-repository>
- Om att skapa SSH keys (frivilligt)
- <https://docs.github.com/en/free-pro-team@latest/github/authenticating-to-github/adding-a-new-ssh-key-to-your-github-account>

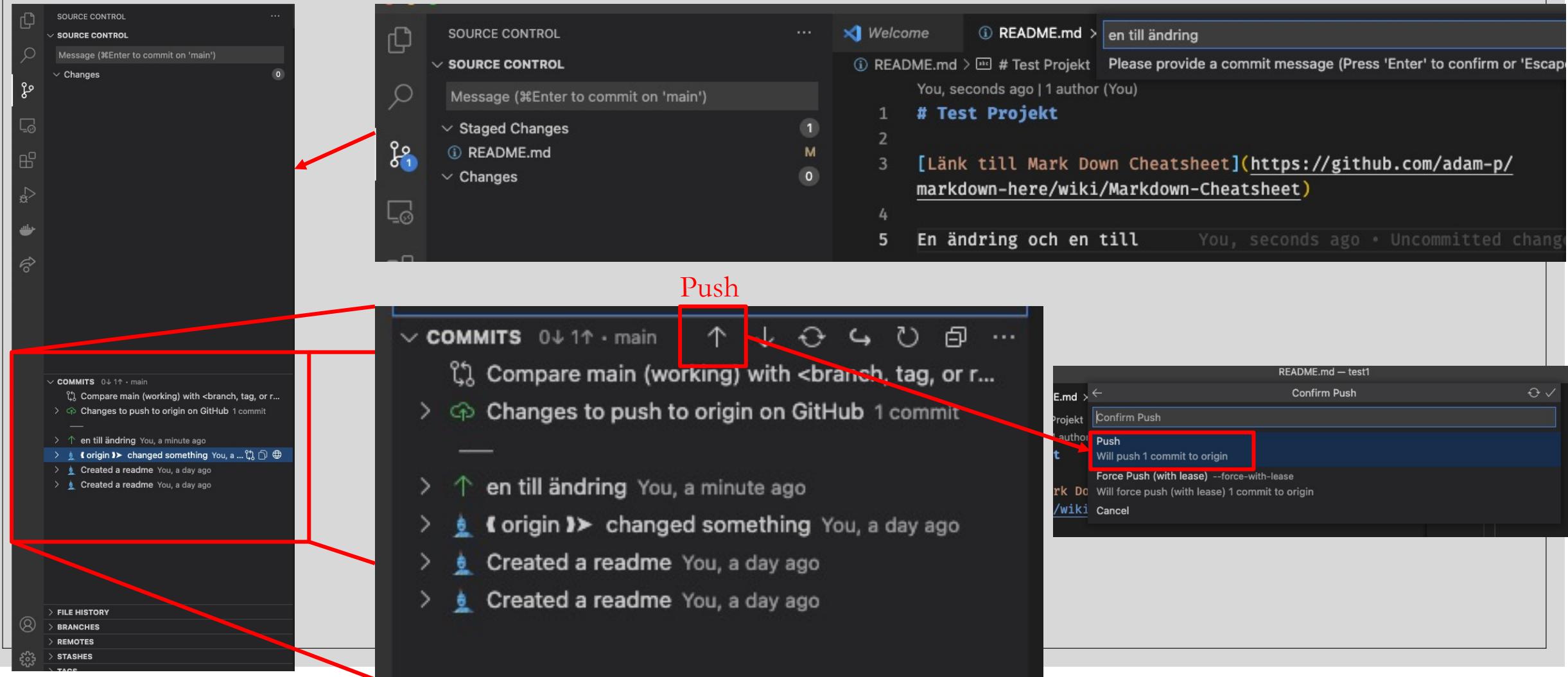


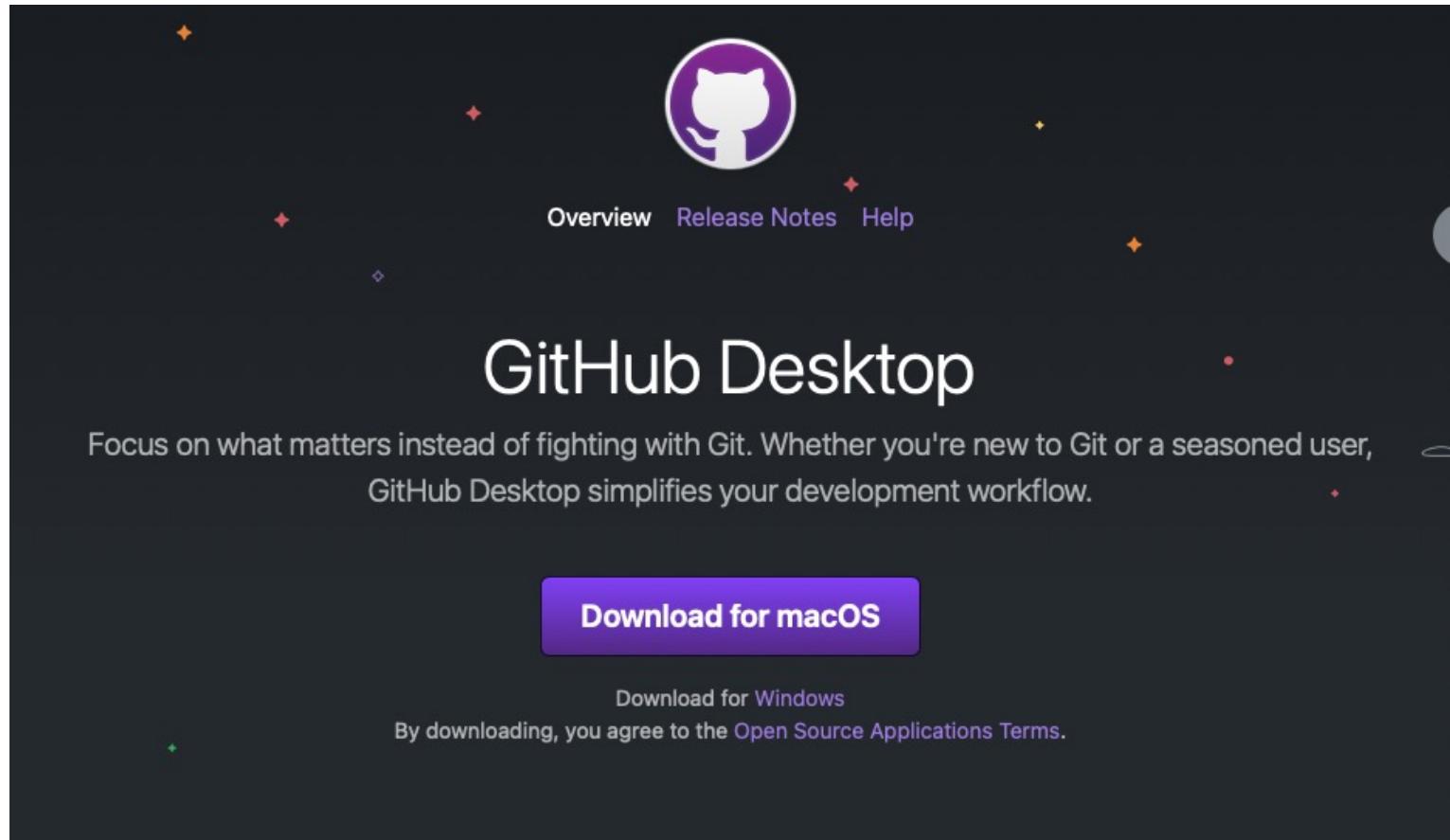
GIT INTEGRERING I VSCODE

Stageing



Skriv meddelande och enter





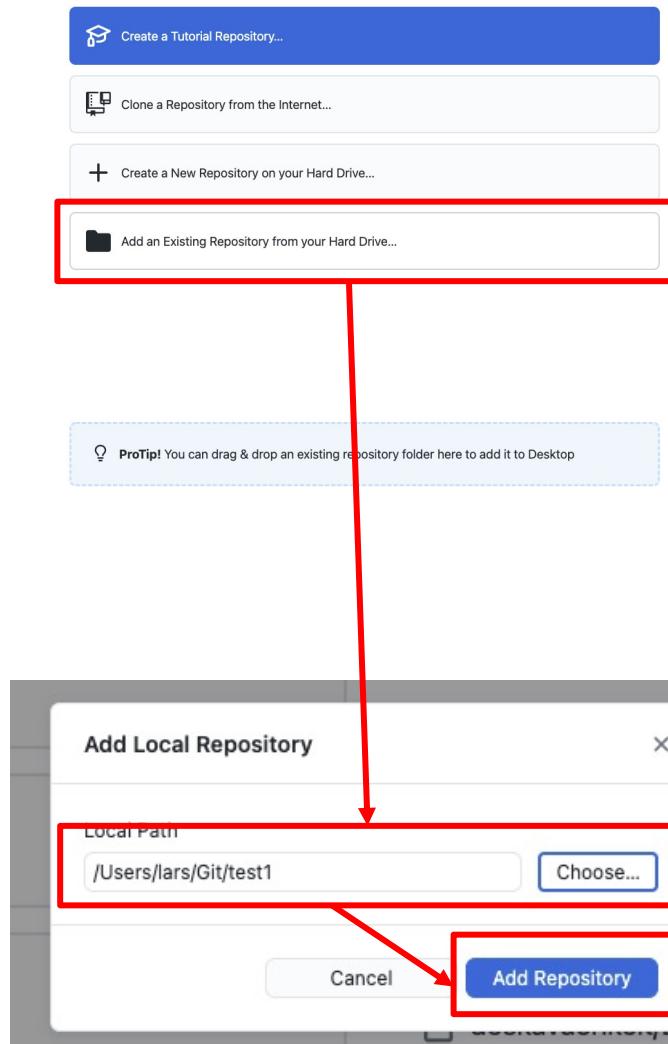
Ladda ner och installera för ditt OS

Github Desktop

- <https://desktop.github.com>
- Mac användare behöver flytta den nerladdade filen till "Applications folder" på datorn

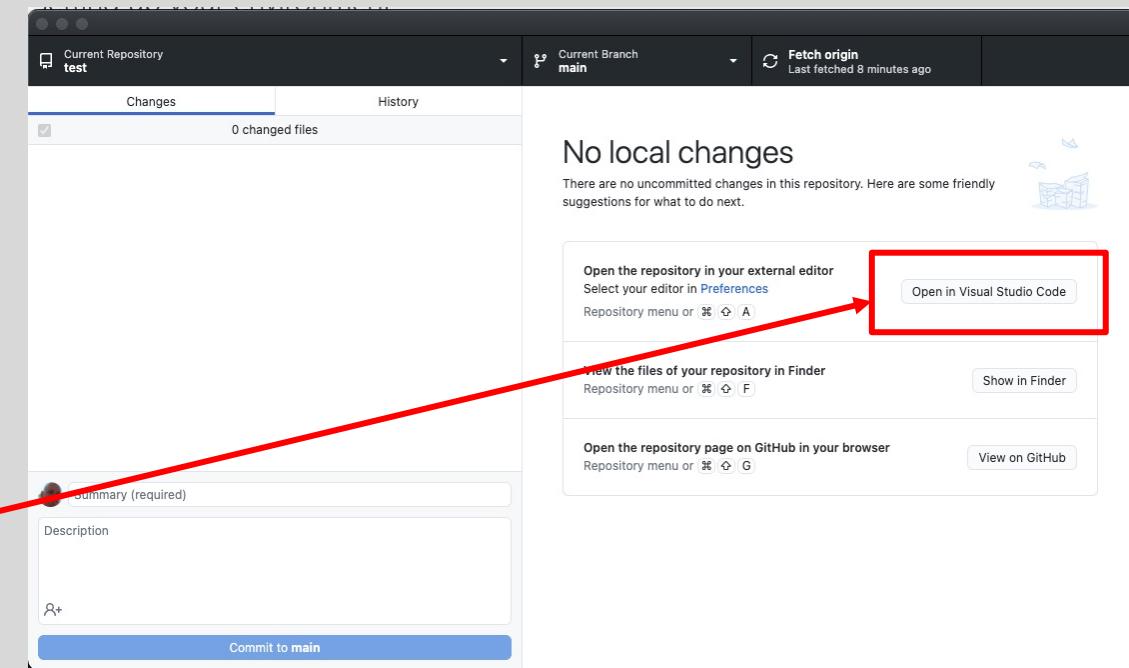
Let's get started!

Add a repository to GitHub Desktop to start collaborating



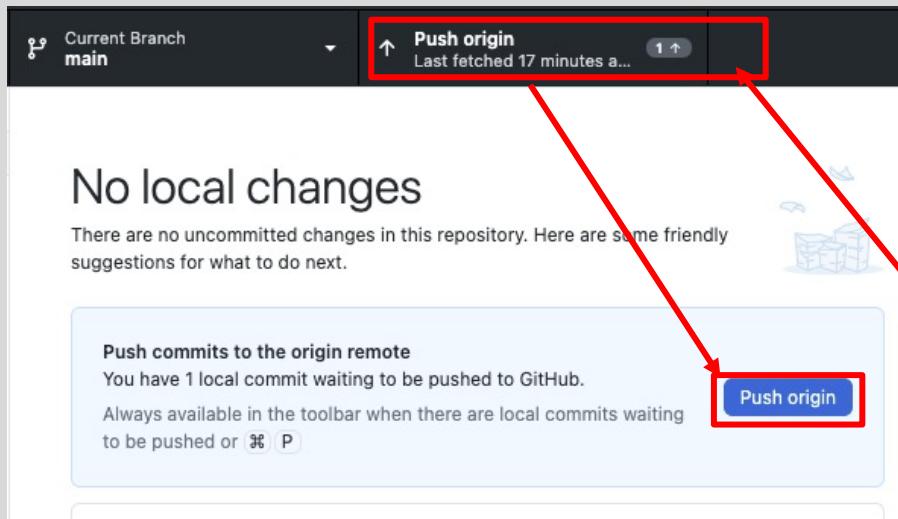
Starta GitHub desktop och logga in på ditt konto

Öppna mappen vi skapade tidigare



Ändra i readme filen

```
You, seconds ago | 1 author (You)
1 # Test Projekt
2
3 [Länk till Mark Down Cheatsheet](https://github.com/adam-p/markdown-here/wiki/Markdown-Cheatsheet)
4
5 En ändring och en till
6
7 Github Desktop ändring You, seconds ago • Uncommitted changes
```



Spara och öppna GitHub Desktop igen

The screenshot shows the GitHub Desktop application interface after saving and reopening it. The left pane shows the "Changes" tab with a single file named "README.md" selected. The right pane shows the content of "README.md" with the following text:

```
@@ -2,4 +2,6 @@
3 [Länk till Mark Down Cheatsheet](https://github.com/adam-p/markdown-here/wiki/Markdown-Cheatsheet)
4
5 -En ändring och en till
6 +En ändring och en till
7 +Github Desktop ändring
```

A red box highlights the file list in the left pane, and another red box highlights the file content in the right pane.

In the bottom right corner, a commit dialog is open:

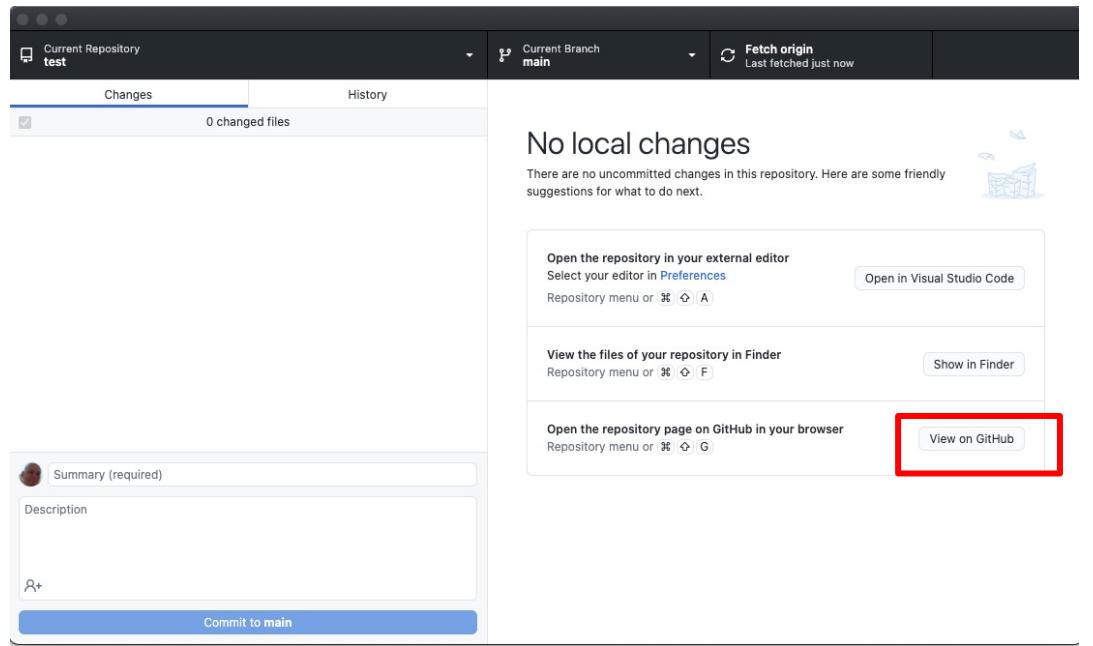
- "Update README.md" is entered in the "Update" field.
- "Description" is empty.
- "Commit to main" is highlighted with a red box.

Red arrows point from the "Push origin" button in the first screenshot to the "Push origin" message in the second, and from the "Push origin" message in the second to the "Commit to main" button in the third.

Vilka filer har ändrats

Vad har ändrats i filen

Skriv ett meddelande



main 1 branch 0 tags

Go to file Add file Code

deskavaenkelt Update README.md 6fb5073 2 minutes ago 5 commits

README.md Update README.md 2 minutes ago

README.md

Test Projekt

Länk till Mark Down Cheatsheet

En ändring och en till

Github Desktop ändring

RESULTAT

.gitignore

- <https://www.toptal.com/developers/gitignore>
- För att slippa onödiga filer så skapar vi en ”.gitignore”-fil som vi lägger in filer och foldrar som ska ignoreras och inte versionshanteras av git
- Lägg till de ”tags” som ni vill exkludera på denna sidan
- Klicka på ”Create”

gitignore.io

Create useful .gitignore files for your project



[Source Code](#) | [Command Line Docs](#)

Kopiera texten och lägg in i en fil enl följande

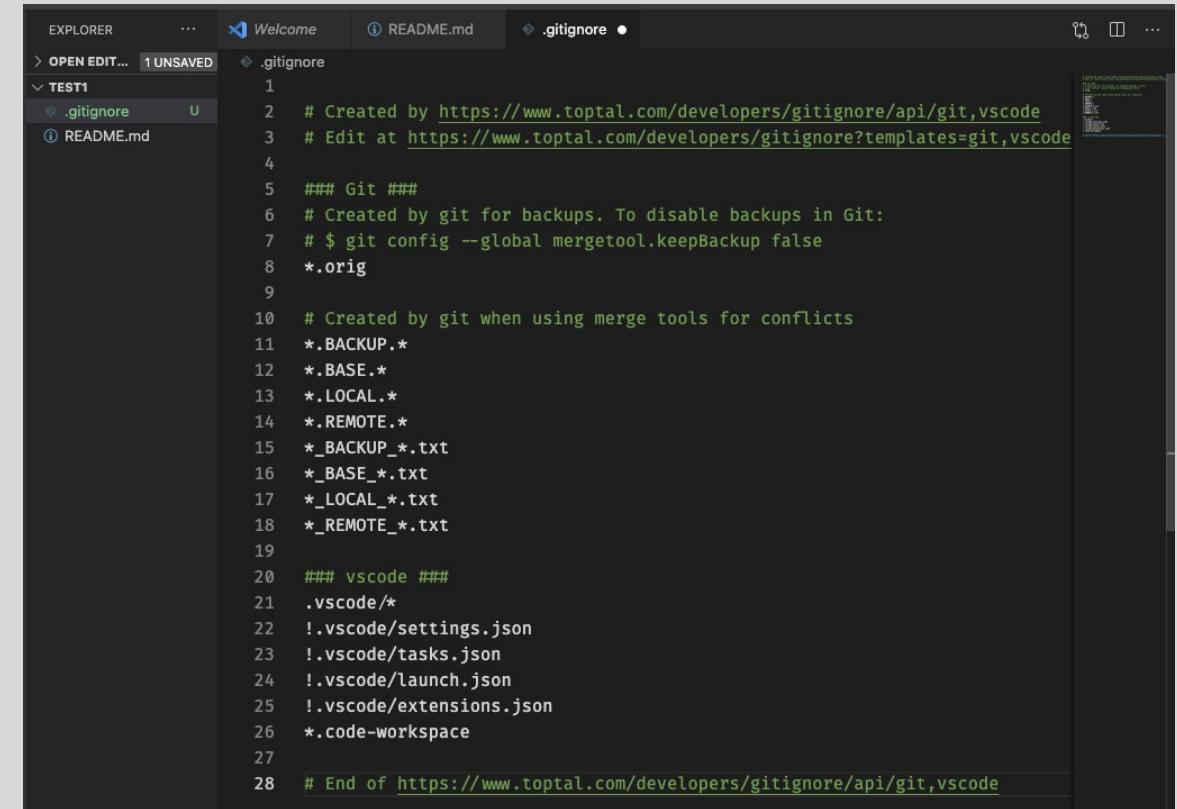
```
# Created by https://www.toptal.com/developers/gitignore/api/git,vscode
# Edit at https://www.toptal.com/developers/gitignore?templates=git,vscode

### Git ###
# Created by git for backups. To disable backups in Git:
# $ git config --global mergetool.keepBackup false
*.orig

# Created by git when using merge tools for conflicts
*.BACKUP.*
*.BASE.*
*.LOCAL.*
*.REMOTE.*
*_BACKUP_*.txt
*_BASE_*.txt
*_LOCAL_*.txt
*_REMOTE_*.txt

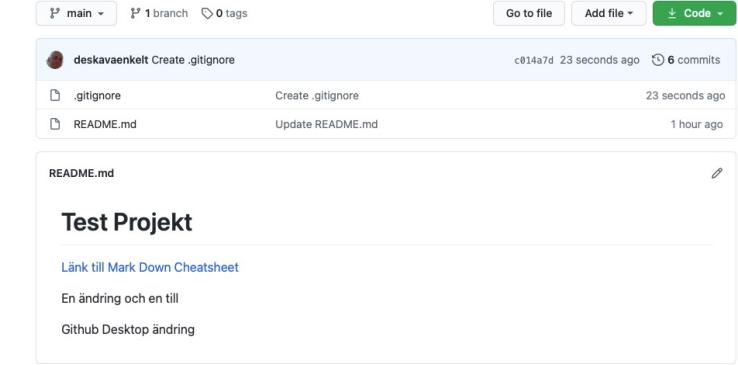
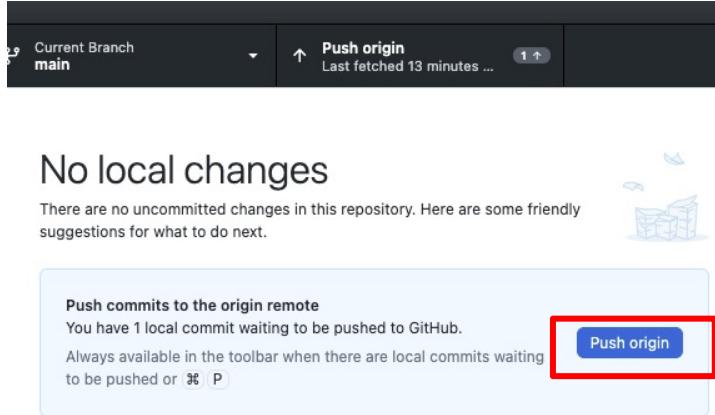
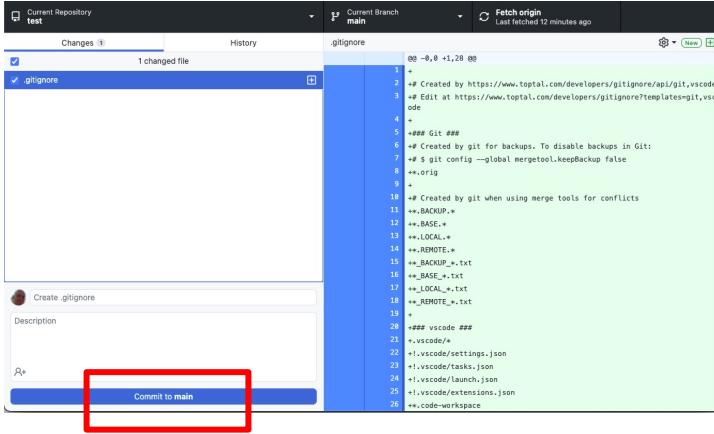
### vscode ###
.vscode/*
!.vscode/settings.json
!.vscode/tasks.json
!.vscode/launch.json
!.vscode/extensions.json
*.code-workspace

# End of https://www.toptal.com/developers/gitignore/api/git,vscode
```



A screenshot of the Visual Studio Code interface. The title bar shows 'Welcome' and '.gitignore'. The Explorer sidebar on the left has 'TEST1' expanded, showing '.gitignore' and 'README.md'. The main editor area displays the same text as the code block above, starting with '# Created by https://www.toptal.com/developers/gitignore/api/git,vscode' and ending with '# End of https://www.toptal.com/developers/gitignore/api/git,vscode'. The code is numbered from 1 to 28.

```
1 # Created by https://www.toptal.com/developers/gitignore/api/git,vscode
2 # Edit at https://www.toptal.com/developers/gitignore?templates=git,vscode
3
4
5 ### Git ###
6 # Created by git for backups. To disable backups in Git:
7 # $ git config --global mergetool.keepBackup false
8 *.orig
9
10 # Created by git when using merge tools for conflicts
11 *.BACKUP.*
12 *.BASE.*
13 *.LOCAL.*
14 *.REMOTE.*
15 *_BACKUP_*.txt
16 *_BASE_*.txt
17 *_LOCAL_*.txt
18 *_REMOTE_*.txt
19
20 ### vscode ###
21 .vscode/*
22 !.vscode/settings.json
23 !.vscode/tasks.json
24 !.vscode/launch.json
25 !.vscode/extensions.json
26 *.code-workspace
27
28 # End of https://www.toptal.com/developers/gitignore/api/git,vscode
```



Gör en ny commit med .gitignore-filen

Open the repository page on GitHub in your browser
Repository menu or ⌘ ⌄ G

View on GitHub