



Aan de slag met Git en GitHub

Wees nooit bang om werk te verliezen met dit professionele versiesysteem

Deze tutorial verwijst herhaaldelijk naar Git en GitHub. *Git* is een veelgebruikt versiebeheersysteem dat wordt gebruikt om code te beheren. Met Git kun je concepten van je code opslaan, zodat je terug kunt kijken naar eerdere versies en mogelijk gecompliceerde fouten ongedaan kunt maken. Een project beheerd met Git wordt een *Git-repository* genoemd.

GitHub is een populaire hostingservice voor Git-opslagplaatsen. Met GitHub kunt u uw lokale Git-opslagplaatsen opslaan in de cloud. Met GitHub kunt u een back-up maken van uw persoonlijke bestanden, uw code delen en samenwerken met anderen.

Kortom, GitHub is een hulpmiddel voor het werken met Git. Er zijn andere services om Git-repositories te hosten, maar GitHub is een vertrouwde, gratis service die wordt gebruikt door organisaties over de hele wereld, groot en klein.

Maak een GitHub-account

Om GitHub te gebruiken, hebt u een GitHub-account nodig.

In uw eigen browser:

1. Open een nieuw browsertabblad
2. Navigeer naar <https://github.com/>
3. Account aanmaken

Gebruikersnaam: leerlingnummer-scalda
(voorbeeld: 210425-scalda)

Als je al een GitHub-account hebt, ga je verder met de volgende oefening.

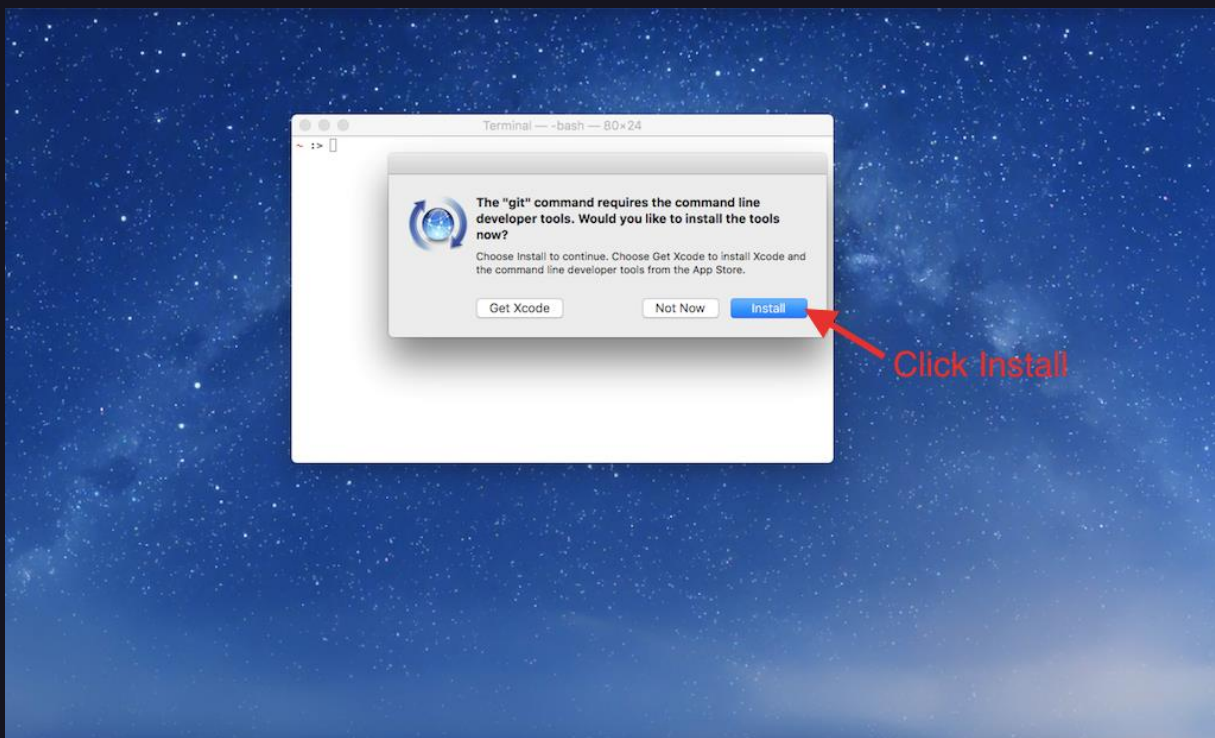
Nadat u zich heeft aangemeld, ontvangt u een verificatie-e-mail. Zorg ervoor dat u uw e-mailadres aan GitHub verifieert door de instructies in die e-mail te volgen.

Git Setup voor Mac en Windows

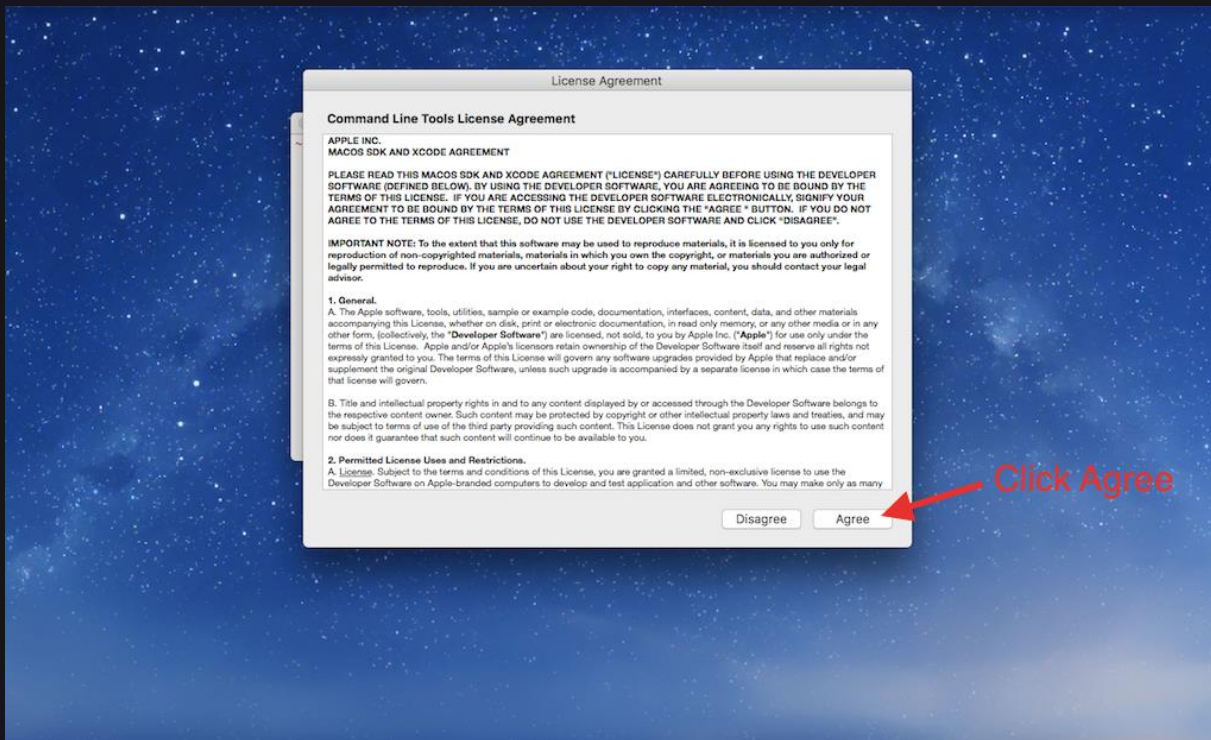
Vervolgens zullen we Git op uw personal computer instellen. Volg de instructies voor uw besturingssysteem.

Mac-gebruikers:

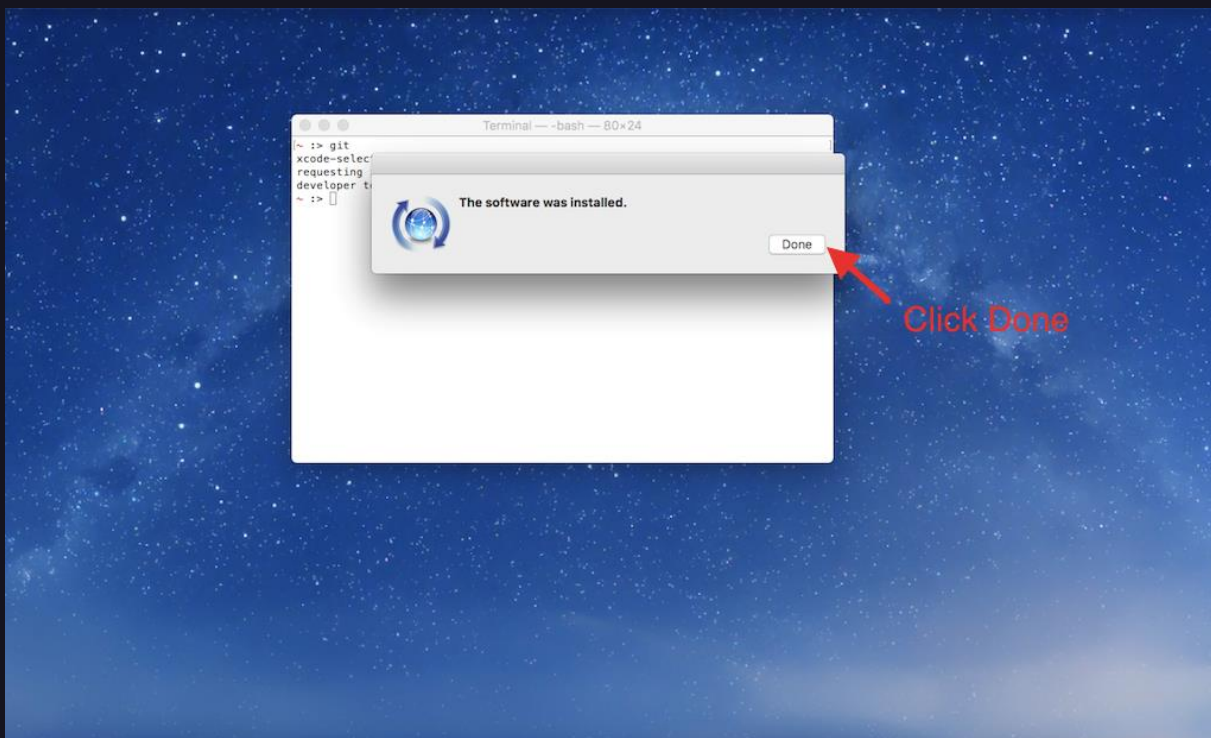
1. Start de **Terminal**-toepassing. Je kunt het vinden in **/ Toepassingen / Hulpprogramma's /** . U kunt ook de **Spotlight**-zoekfunctie (het kleine vergrootglas rechtsboven in uw scherm) gebruiken om naar **Terminal** te zoeken . Zodra **Spotlight** het heeft gevonden, klikt u op het resultaat met de vermelding **Terminal** .
2. Typ wanneer **Terminal** **wordt** geopend giten druk op Enter.
3. Als je Git nog niet hebt geïnstalleerd, verschijnt er een dialoogvenster met de tekst: 'De 'git'-opdracht vereist de opdrachtregelontwikkeltools. Wilt u de tools nu installeren? "Klik op " Installeren " .



Klik vervolgens op Agree tot he Terms of Service wanneer daarom wordt gevraagd.



4. Wanneer het downloaden is voltooid, zal het installatieprogramma vanzelf verdwijnen om aan te geven dat Git nu is geïnstalleerd! Klik op "Done" om het installatieproces te voltooien.



5. Navigeer naar GitHub's artikelen over het instellen van uw Git-gebruikersnaam en e-mail en volg de instructies voor elk gebruik van Terminal.

6. GitHub biedt twee authenticatie-opties, HTTPS en SSH, om uw werk veilig te houden. Dit is een beveiligingsmaatregel die voorkomt dat niet-geautoriseerde personen wijzigingen in uw GitHub-repository

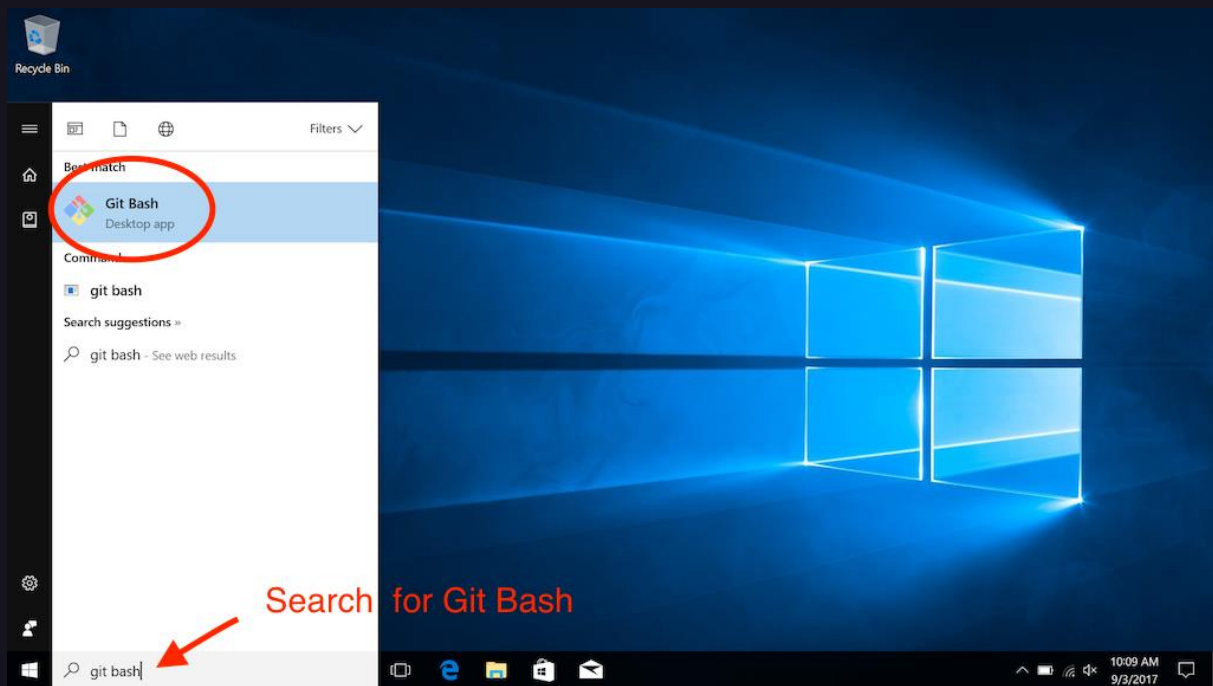
aanbrengen. In dit artikel zullen we HTTPS gebruiken. Navigeer naar het artikel van GitHub over het [opslaan van je wachtwoord](#) en volg de instructies om je computer te configureren om HTTPS te kunnen gebruiken.

Ga nu naar het gedeelte "Uitproberen!" Hieronder.

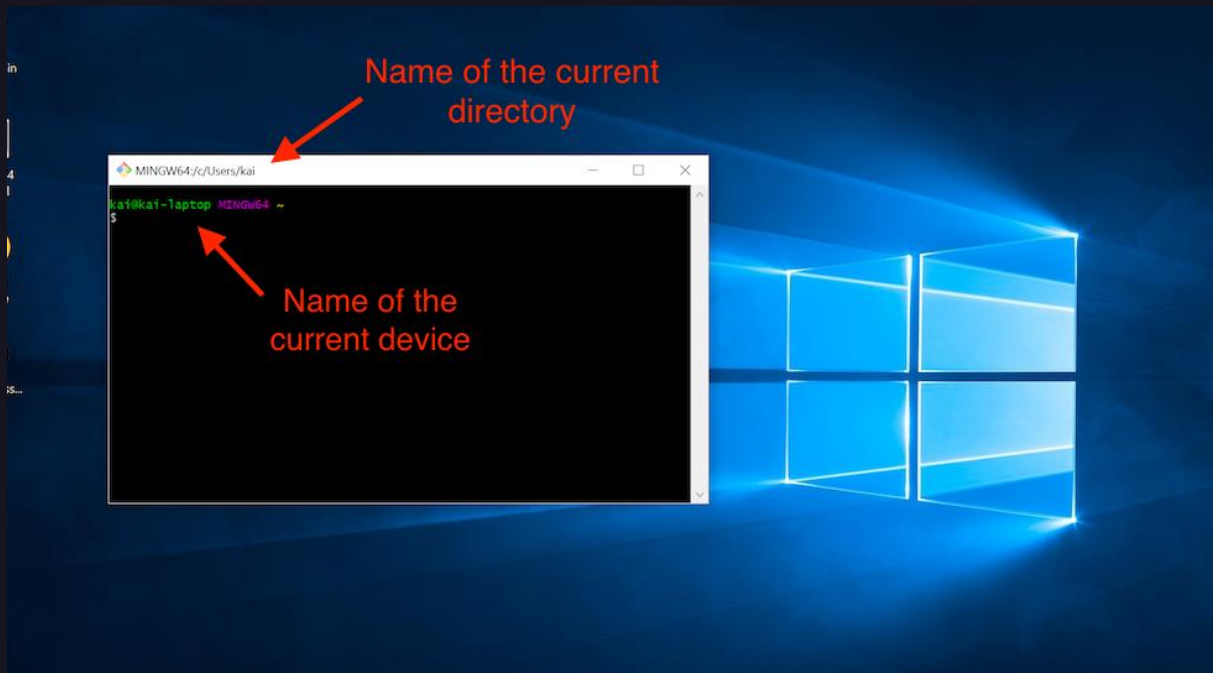
Windows-gebruikers:

In dit gedeelte van de gids wordt ervan uitgegaan dat je al een programma hebt geïnstalleerd met de naam Git Bash waarmee we toegang hebben tot Git op Windows. Als je Git Bash niet hebt geïnstalleerd, raadpleeg dan de vorige tutorial over de opdrachtregelinterface (CLI) Setup en volg de instructies voor het installeren van Git Bash op Windows. Nadat u dit hebt voltooid, kunt u doorgaan met deze handleiding.

1. Open het menu Start en zoek naar de app, git bash. Je zou 'Git Bash Desktop-app' moeten zien verschijnen. Druk op Enter of klik op het Git Bash-pictogram om de app te openen.

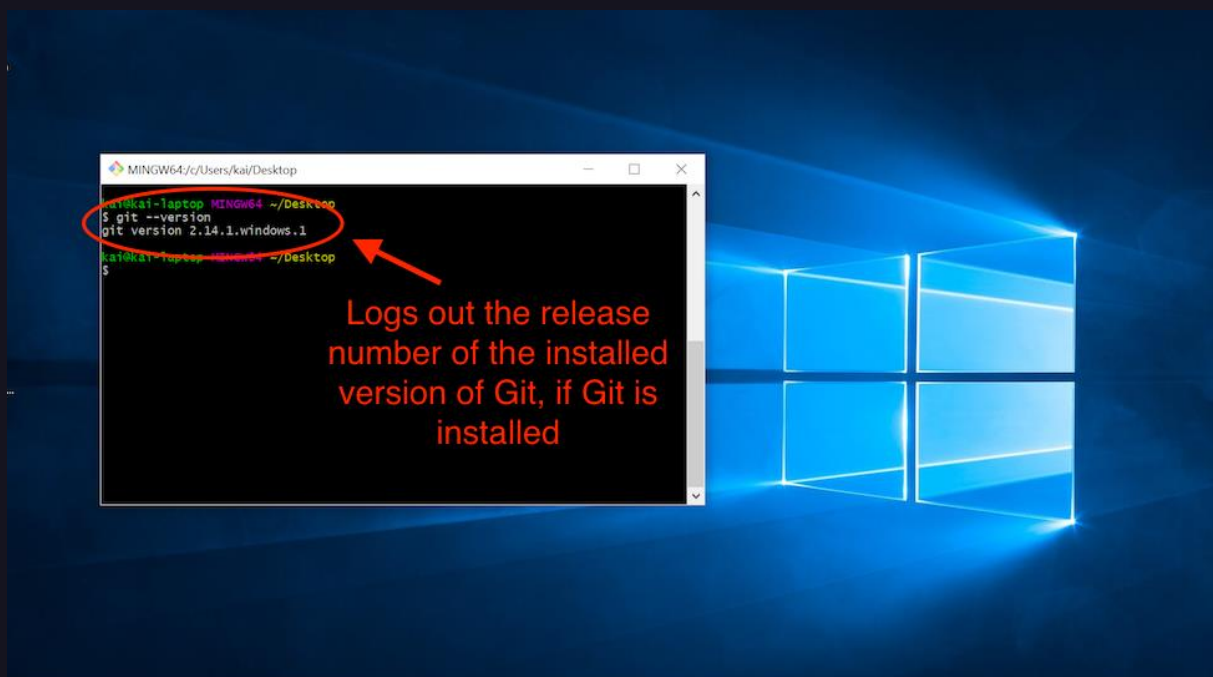


Er wordt een nieuw venster geopend dat er als volgt uit ziet:



Dit venster is onze CLI, waar we onze Git-opdrachten zullen gebruiken.

2. Als je er zeker van wilt zijn dat Git is geïnstalleerd, voer je het uit `git --version` in de CLI. Je zou een reactie moeten zien die je de geïnstalleerde versie van Git geeft. Het ziet er zo uit:



Git kan nu worden gebruikt in de Git Bash-app!

3. Navigeer naar GitHub's artikelen over het instellen van uw Git-gebruikersnaam en e-mail en volg de instructies voor elk gebruik van Git Bash.

4. GitHub biedt twee authenticatie-opties, HTTPS en SSH, om uw werk veilig te houden. Dit is een beveiligingsmaatregel die voorkomt dat niet-geautoriseerde personen wijzigingen in uw GitHub-repository aanbrengen. In dit artikel zullen we HTTPS gebruiken. Navigeer naar het artikel van GitHub over het opslaan van je wachtwoord en volg de instructies om je computer te configureren om HTTPS te kunnen gebruiken.

Probeer het!

Nu heb je alles wat je nodig hebt om je Git-vaardigheden op je lokale computer te oefenen. Neem even de tijd om de onderstaande opdrachten uit te voeren om een Git-repository te initialiseren. We zullen deze Git-repository later in deze tutorial opnieuw gebruiken, dus zorg ervoor dat je deze stappen precies uitvoert zoals beschreven.

1. `mkdir git_practice` om een nieuwe map te maken om te oefenen.
2. `cd git_practice` om van de nieuwe map uw werkmap te maken.
3. `git init` om de huidige, lege map in een nieuwe Git-repository te veranderen.
4. `echo "Hello Git and GitHub" >> README.txt` om een nieuw README-bestand te maken (waarover later meer) met wat voorbeeldtekst.
5. `git add README.txt` om het nieuwe bestand toe te voegen aan het Git staging gebied.
6. `git commit -m "First commit"` om je eerste commit te maken met het nieuwe README-bestand.

Je eerste externe repository op GitHub

Ten slotte maken we een repository op GitHub en koppelen we deze vervolgens aan een lokale repository op uw computer. Hiermee kunt u constant en veilig een back-up van uw werk maken, zodat u zich nooit meer zorgen hoeft te maken dat u uw werk kwijtraakt!

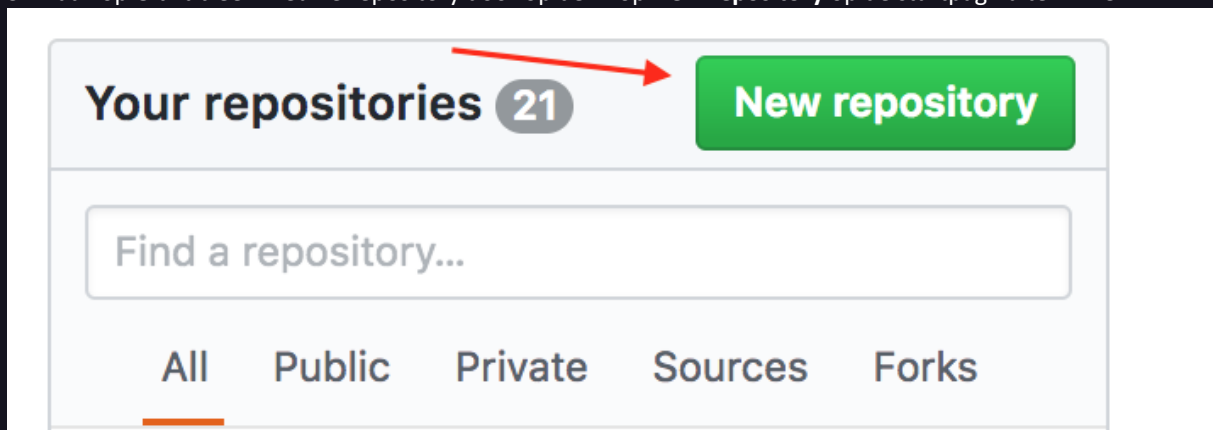
Laten we nu onze lokale Git-repository verbinden met GitHub.

Instructions

1. Controleer in uw opdrachtregelinterface of uw huidige werkmap uw nieuwe Git-repository is. Navigeer daarheen als dat niet het geval is.
2. Controleer de status van welke bestanden en mappen nieuw zijn of zijn bewerkt. Er mogen geen bestanden worden gewijzigd.

\$ git status

3. Maak op GitHub een nieuwe repository door op de knop **New repository** op de startpagina te klikken .



4. Geef uw repository een naam op de nieuwe repository-pagina. Het is niet nodig, maar het zou handig zijn om het dezelfde naam te geven als de map, **git_practice** . Nadat u de repository een naam hebt gegeven, klikt u op Create repository.

Create a new repository

A repository contains all the files for your project, including the revision history.


Owner


Repository name

/ git_practice ✓

Great repository names are short and memorable. Need inspiration? How about **automatic-parakeet**.

Description (optional)

☒  **Public**
Anyone can see this repository. You choose who can commit.

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

☐ **Initialize this repository with a README**
This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: **None** ▾ | Add a license: **None** ▾ ⓘ

Create repository

5. Na het aanmaken van een repository geeft GitHub de repository-pagina weer. Zorg ervoor dat bovenaan de pagina "HTTPS" is geselecteerd.

<> Code | 0 Issues | 0 Pull requests | 0 Projects | Wiki | Pulse | Graphs | Settings

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

Set up in Desktop or **HTTPS** SSH https://github.com/ /git_practice.git

We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

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

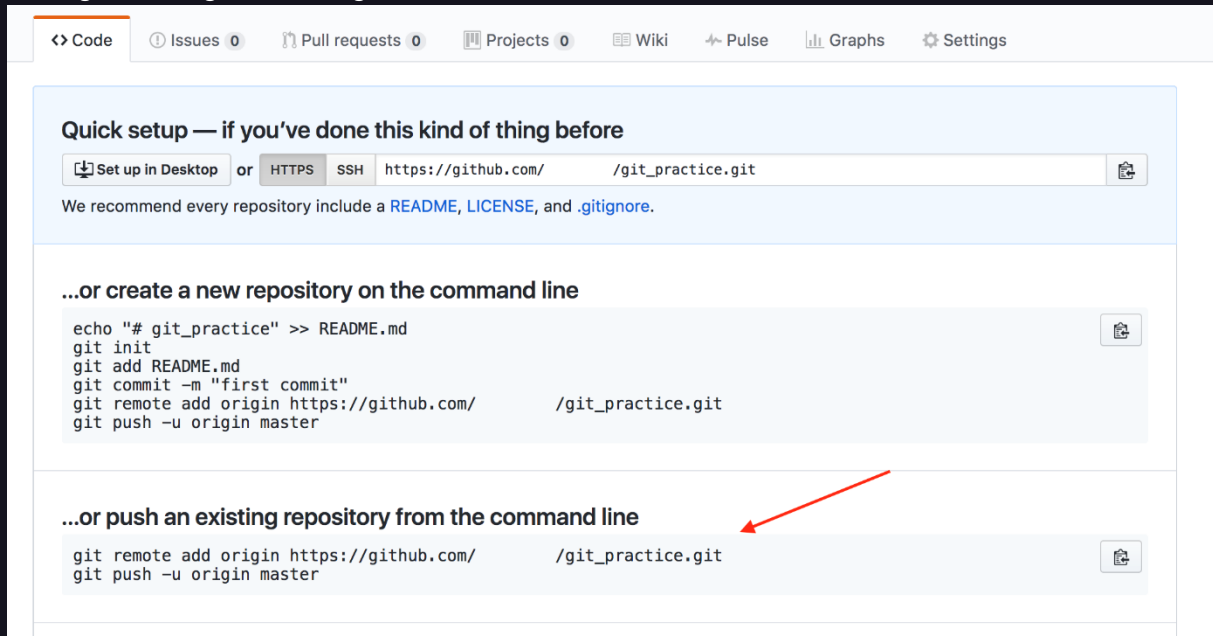
```
echo "# git_practice" >> README.md
git init
git add README.md
git commit -m "first commit"
git remote add origin https://github.com/ /git_practice.git
git push -u origin master
```

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

```
git remote add origin https://github.com/ /git_practice.git
git push -u origin master
```

6. De repository is leeg, dus het is tijd om het te verbinden met uw bestaande werk. Kopieer de Git-opdrachten op de GitHub-pagina, onder de titel "... of duw een bestaande repository vanaf de opdrachtregel", en plak ze in je opdrachtregelinterface. Als u deze opdrachten uitvoert, wordt een externe repository toegevoegd en wordt uw lokale repository naar de externe repository gepusht.

Wanneer u om een gebruikersnaam en wachtwoord wordt gevraagd, typt u uw GitHub-gebruikersnaam en -wachtwoord in en drukt u enter na elk op. Schrik niet als u de tekens die u typt niet kunt zien, ze zijn opzettelijk verborgen als veiligheidsmaatregel.



Opmerking: als je tweefactorauthenticatie instelt met GitHub (maak je geen zorgen als je dat niet hebt gedaan), volg dan [deze instructies](#) om een OAuth-token te genereren voor gebruik in plaats van je wachtwoord in bash. Standaard stelt GitHub geen tweefactorauthenticatie in. Als u niet bekend bent met tweefactorauthenticatie, hoeft u geen OAuth-token te genereren.

7. Zodra uw opdrachtregelinterface meldt dat de push is voltooid, vernieuwt u de pagina op GitHub. Je zou nu de tekst die je eerder schreef in het README-bestand moeten zien, "Hallo Git en GitHub."

GitHub geeft automatisch de inhoud van een bestand met de naam **README.txt** weer als deze in de repository bestaat. Het README-bestand is de perfecte plaats om een beschrijving van uw project te schrijven.

Daar heb je het! Je eerste GitHub-repository, gekoppeld aan je lokale Git-repository. Je hebt een aantal grote sprongen gemaakt, dus wees trots! Nu kunt u uw kennis van Git gebruiken om de voortgang op uw lokale computer te volgen en die voortgang naar GitHub te pushen wanneer u maar wilt. Je kunt erop vertrouwen dat elke stap van je voortgang veilig wordt opgeslagen in GitHub.