

ITBC BI1 Python 1 en Basiskennis voor Bioinformatici

Afvinkopdracht 2

Afvinkopdracht 2

Inhoudsopgave

- Achtergrond
- Opdracht 1: Oefenen met Git
 - Deel 1
 - Deel 2
 - Deel 3
 - Optioneel: Merge Conflict
- Opdracht 2: Oefenen met Python
- Opdracht 3: Omgaan met Git

Achtergrond

Bij het ontwikkelen van programmeercode ga je fouten maken, dat doet iedere programmeur. Als er meerdere programmeurs aan een gezamenlijk project werken stapelen de fouten zich soms dus op. Uiteindelijk kan het handiger zijn om een deel van de code geheel overnieuw te schrijven, of een oudere versie van de code als nieuw uitgangspunt te gebruiken. Versiebeheer met behulp van Git is dus geen overbodige luxe.

GitHub heeft een pakket voor studentent waarin je een aantal handige stukken software kan vinden voor noppes zolang je student bent. (<https://education.github.com/pack>) Een van de voordelen van dit pakket is dat je gratis onbeperkt private repositories krijgt. Dit kan je ook krijgen bij GitLab en BitBucket, twee andere organisaties die online remote repo's aanbieden. Omdat wij met GitHub classroom werken is het in ieder geval de moeite waard om even naar het studentenaanbod te kijken.

Voor deze opdracht (en een groot aantal andere opdrachten) maken we gebruik van GitHub Classroom. GitHub Classroom is een omgeving waarin wij, de docenten, opdrachten kunnen maken waar jullie, de studenten, je op in kunnen schrijven. Bij iedere opdracht zal een link staan naar de opdracht waar je je op in kunt schrijven. Trek bij problemen even aan de bel bij je praktijkdocent.

Opdracht 1: Oefenen met Git

In deze opdracht ga je oefenen met Git.

Werken met Git kan niemand vanzelf. Voordat je aan het inlever gedeelte van de afvinkopdracht gaat beginnen mag je eerst oefenen. Je antwoorden kan je invullen op het antwoordmodel (BIN-OWE1_Git_antwoordmodel.docx).

Deel 1

Je gaat om te beginnen de workflow met een lokale Git een keer geheel doorlopen. Vul op het antwoordvel de commando's in die je invoert.

1. Ga naar de map waarin je een nieuwe lokale repo wilt maken
2. Initialiseer een nieuwe git repo
3. Maak een nieuw bestand (readme.txt) en zet hier iets zinvols in, zoals "Dit is de readme van mijn eerste repo."
4. Haal even de status van je repo op
5. Voeg readme.txt aan de staging area toe
6. Check nog een keer de status van de staging area om te controleren of dit gelukt is
7. Commit de readme naar de repo
8. Check nog een keer de status om te controleren of dit gelukt is
9. Wijzig het bestand readme.txt
10. Bekijk de nieuwe status van de staging area
11. Bekijk de verschillen tussen de oude versie van readme.txt en de nieuwe versie
12. Voeg readme.txt weer toe aan de staging area
13. Controleer of dit gelukt is
14. Commit de verandering weer

Deel 2

Omdat je straks een je opdracht moet inleveren op GitHub ga je nu alvast wat voorbereidingen treffen om hiermee te oefenen door een nieuw python bestand toe te voegen aan de lokale repo. Deze zal je in Deel 3 naar een remote repo gaan pushen.

1. Maak een python bestand en noem deze test.py. Als inhoud van dit bestand kan je "# Dit is een test bestand" nemen
2. Check de status
3. Voeg het bestand toe aan de staging area
4. Commit de verandering

Deel 3

Ga nu naar je GitHub account en maak een nieuwe remote repo aan. Klik rechtsbovenin op "+" en selecteer "New repository".

The screenshot shows the GitHub interface. At the top, there's a dark header with the GitHub logo, a search bar, and navigation links: Pull requests, Issues, Marketplace, and Explore. On the right side of the header, a dropdown menu is open, showing options: New repository, Import repository, New gist, and New organization. The 'New repository' option is highlighted with a red box. Below the header, the main content area shows a repository page for 'SignMyPublicKey'. It displays a list of recent activity, including branch creations and comments. On the right side, there are two sections: 'Repositories you contribute to' and 'Your repositories'. The 'Your repositories' section has a 'New repository' button and a search bar.

Search GitHub

Pull requests Issues Marketplace Explore

SignMyPublicKey

GonnyHV created branch `master` at iabc-bin/owe1a-afvinkopdracht1-GonnyHV 22 hours ago

6 days ago
MartijnLiebrand commented on commit iabc-bin/owe1a-afvinkopdracht1-MartijnLiebrand@2eaae94ff6
Naamgeving / conventies Commentaar Recursie Flar

6 days ago
MartijnLiebrand commented on commit iabc-bin/owe1a-afvinkopdracht1-MartijnLiebrand@2eaae94ff6
The comment has since been removed.

MartijnLiebrand created branch `master` at iabc-bin/owe1a-afvinkopdracht1-MartijnLiebrand 6 days ago

iabc-bin added MartijnLiebrand to iabc-bin/owe1a-afvinkopdracht1-MartijnLiebrand 8 days ago

GonnyHV created repository iabc-bin/owe1a-afvinkopdracht1-MartijnLiebrand 8 days ago

Repositories you contribute to 5

| | |
|------------------------------------|-------|
| rosstuck/domconf | 0 ★ |
| Rdelangen/ProstateProjectISMI | 0 ★ |
| rubigdata/bigdata-blog-2017-Sig... | 0 ★ |
| jspunda/prostatex | 0 ★ |
| oniony/TMSU | 826 ★ |

Your repositories 9

New repository

Find a repository...

All Public Private Sources Forks

Rdelangen/ProstateProjectISMI

iabc-bin/owe1a-afvinkopdracht1-SignMy...

Hierna kom je in het volgende scherm terecht, waarbij je een naam voor je repo mag verzinnen. Gebruik nu even iets in de trand van “test”, gezien je deze repo straks ook makkelijk weer wilt kunnen vinden en verwijderen. Vergeet niet de repo prive te maken en te initialiseren (aan te maken) zonder readme. Die hebben we tenslotte al gemaakt in de lokale repo. Klik dan op “Create repository”.

Create a new repository

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

Owner

 SignMyPublicKey ▾

Repository name

pythontest ✓

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

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

Nadat de repo is aangemaakt ziet het scherm er ongeveer zo uit. Je ziet ook meteen instructies over hoe je een remote repo toe kan voegen aan je lokale repo en de inhoud van je lokale repo kunt pushen naar de remote.

 This repository Search Pull requests Issues Marketplace Explore

SignMyPublicKey / pythontest Private Unwatch 1 Star 0 Fork 0

[Code](#) Issues 0 Pull requests 0 Projects 0 Wiki Settings Insights

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

or **HTTPS** **SSH** `https://github.com/SignMyPublicKey/pythontest.git`

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

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

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

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

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

1. Voeg de remote toe als origin voor je lokale repo
2. Push je lokale repo naar de remote

Je remote repo ziet er via de website nu zo uit. Breng een wijziging aan in de readme via de website zelf.

SignMyPublicKey / pythontest Private Unwatch 1 Star 0 Fork 0

[Code](#) Issues 0 Pull requests 0 Projects 0 Wiki Settings Insights

No description, website, or topics provided. [Add topics](#) [Edit](#)

1 commit 1 branch 0 releases 0 contributors

Branch: master New pull request Create new file Upload files Find file Clone or download

ekok Initial commit Latest commit da9dd55 an hour ago

[readme.txt](#) Initial commit an hour ago

readme.txt

Dit is de readme voor mijn eerste repo.

3. Nu heb je een readme die niet hetzelfde is op de remote als lokaal. Hier kunnen we verandering in brengen door de bestanden te pullen vanaf de remote. Doe dit.

Optioneel: Merge Conflict

Het kan zijn dat je een merge conflict krijgt als je het bestand tussentijds ook lokaal veranderd hebt. Een merge conflict ziet er zo uit:

```
esther@mercurius:~/Documents$ git pull origin
Username for 'https://github.com': signmypublickey
Password for 'https://signmypublickey@github.com':
Auto-merging readme.txt
CONFLICT (content): Merge conflict in readme.txt
Automatic merge failed; fix conflicts and then commit the result.
esther@mercurius:~/Documents$
```

Dit kan je oplossen, zoals de command line al aangeeft, door de conflicten handmatig te repareren. Je ziet dat het betrokken bestand genoemd wordt. Je kan er ook achter komen welk bestand het merge conflict heeft door de status weer op te vragen. Dan zie je iets wat hier op lijkt:

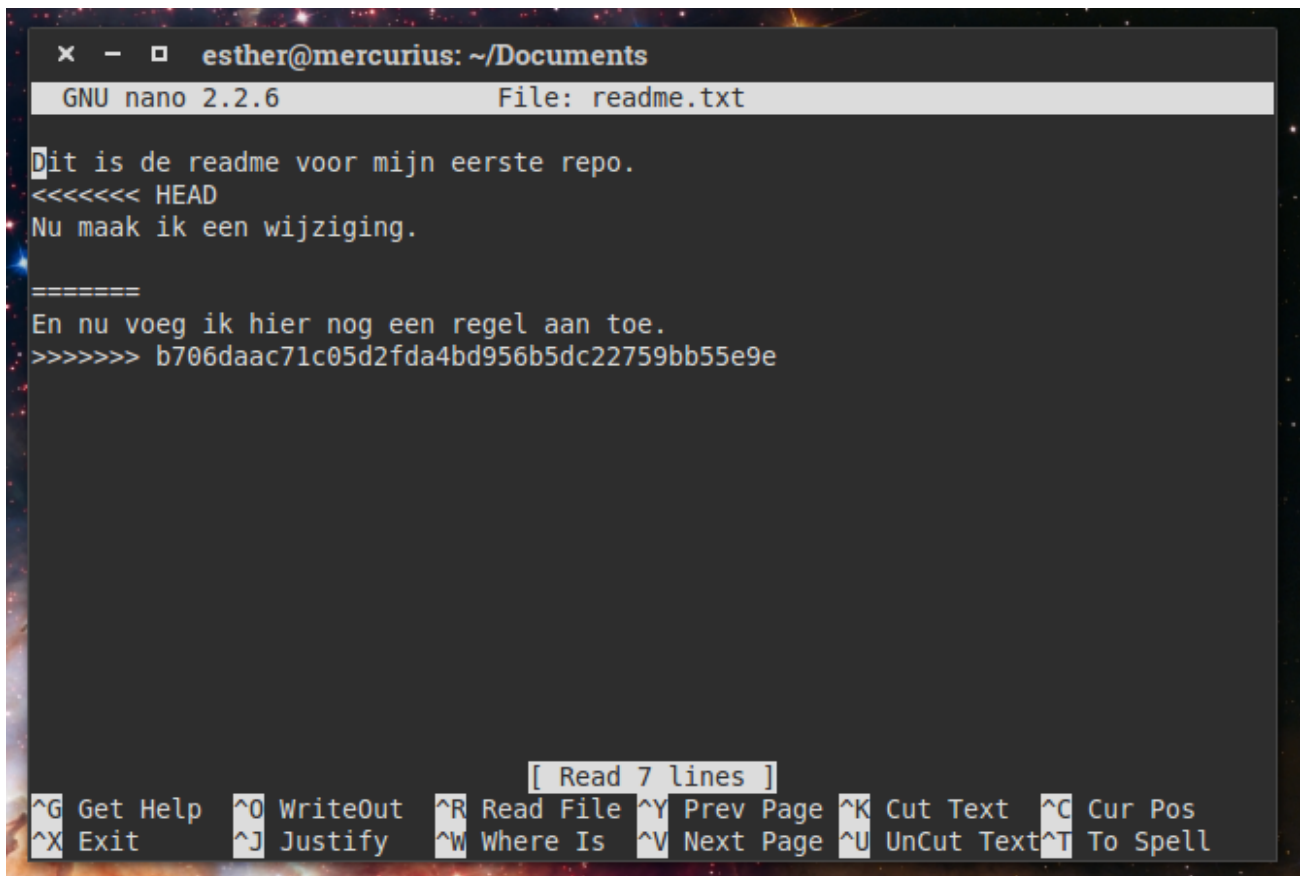
```
x - □ esther@mercurius: ~/Documents
esther@mercurius:~/Documents$ git status
On branch master
Your branch and 'origin/master' have diverged,
and have 1 and 1 different commit each, respectively.
(use "git pull" to merge the remote branch into yours)

You have unmerged paths.
(fix conflicts and run "git commit")

Unmerged paths:
(use "git add <file>..." to mark resolution)

      both modified:   readme.txt
```

Als je in het bestand readme.txt kijkt zie je het volgende:



```
esther@mercurius: ~/Documents
GNU nano 2.2.6 File: README.txt

Dit is de readme voor mijn eerste repo.
<<<<<<< HEAD
Nu maak ik een wijziging.

=====
En nu voeg ik hier nog een regel aan toe.
>>>>>>> b706daac71c05d2fda4bd956b5dc22759bb55e9e

[ Read 7 lines ]
^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell
```

Observeer het volgende:

- De bovenste regel is in beide bestanden hetzelfde, dus is hier ook gelijk
- Het begin van het aangegeven conflict is de regel met <<<<<<<
- Het einde van het conflict is de regel met >>>>>>>
- De twee verschillende versies worden aangegeven met het scheidingsteken =====

Nu moet je gaan kiezen welke van de regels het goede is en welke je dus wilt houden. Je kan er ook voor kiezen om een combinatie van de twee te houden. Je zou deze regels kunnen samenvoegen tot “Nu maak ik een wijziging door hier nog een regel aan toe te voegen.” Sla de verandering op, voeg het bestand toe aan de staging area, commit de verandering en voer nogmaals de pull uit. Je output moet er nu zo uitzien:

```
esther@mercurius:~/Documents$ git add README.txt
esther@mercurius:~/Documents$ git commit -m "fixed merge conflict"
[master 19b61c7] fixed merge conflict
esther@mercurius:~/Documents$ git pull origin
Username for 'https://github.com': signmypublickey
Password for 'https://signmypublickey@github.com':
Already up-to-date.
```

Proficiat, je hebt zojuist je eerste merge conflict opgelost! Vergeet niet ook even een push naar de remote te doen, zodat anderen die aan hetzelfde project werken ook kunnen profiteren van deze reparatie.

Opdracht 2: Oefenen met Python

Deze opdracht geeft je oefening met Input/Output in verwerken in Python. Maak de volgende opdrachten uit het boek, Hoofdstuk 2, Programming Exercises:

1. (1) Personal Information
2. (2) Sales Prediction
3. (4) Total Purchase
4. (5) Distance Traveled
5. (7) Miles per Gallon
6. (10) Ingredient Adjuster

Voor deze opdrachten hoef je dus nog geen gebruik te maken van Git (maar het mag wel!). Het combineren van de twee ga je in de volgende opdracht doen.

Opdracht 3: Omgaan met Git

Nu ga je het eerder geleerde combineren. Je gaat een Python programma schrijven die Input/Output verwerkt, volgens de opdracht van weektaak 2. Wijzigingen tijdens het programmeren hou je direct bij met behulp van Git. Maak individueel de opdracht die uitgegeven staat op GitHub Classroom ([link](#)) Hiervoor heb je je GitHub account nodig. Maak bij het maken van de opdracht ten minste 5 commits.

Hierbij is het dus de bedoeling dat je de opdracht inlevert via de remote repository op GitHub, zodat je docent deze kan aftekenen. De opdracht moet op GitHub staan, anders kan je geen vinkje verdienen.