



Figure 1: Processing

Voorwoord



Figure 1: Het logo van De Jonge Onderzoekers



Figure 2: Het logo van Codestarter

Dit is het Processing boek van de Dojo. Processing is een programmeertaal. Dit boek leert je die programmeertaal.

Over dit boek

Dit boek heeft een CC-BY-NC-SA licentie.



Figure 3: De licentie van dit boek

(C) Dojo Groningen 2016

Het is nog een beetje een slordig boek. Zo staat bijvoorbeeld het plaatje dat eigenlijk op de kaft moet staan op pagina twee. Er zitten tiepvauten in en de opmaak is **niet altijd even mooi**.

Daarom staat dit boek op een GitHub. Om precies te zijn, op <https://github.com/richelbilderbeek/Dojo>. Hierdoor kan iedereen die dit boek te slordig vindt minder slordig maken.

Processing opstarten op cursuslaptop

Met een terminal kun je veel dingen doen, die soms niet met een normaal programma gedaan kunnen worden. Op de cursus start je Processing vanuit de terminal.

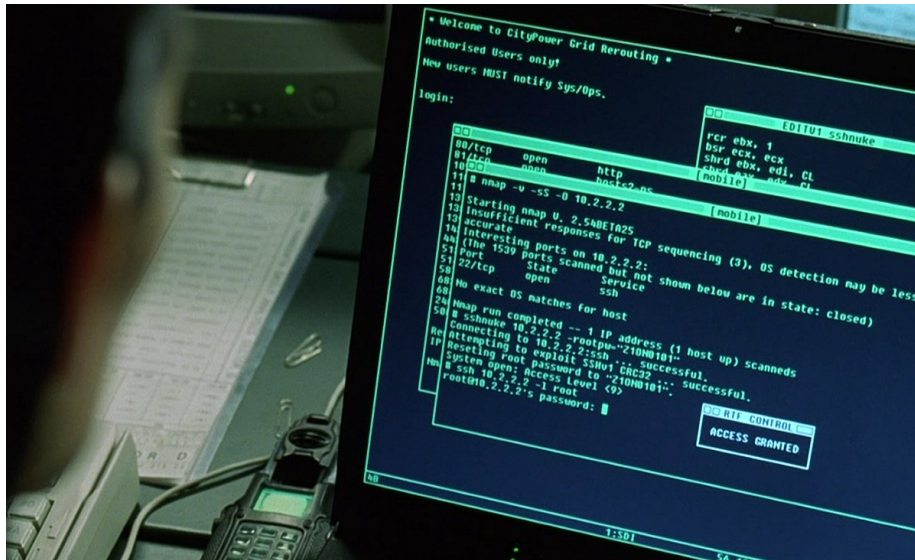


Figure 4: Hackers werken met een terminal

Processing starten

Start een Terminal. Dit kan soms met Win+T, CTRL+ALT+T, of deze te vinden in de menuutjes, of te zoeken op het woord Terminal

Ga naar de folder waar Processing in staat. Hiervoor is het commando cd. De afkorting cd staat voor 'Change Directory'. "Change Directory" is Engels voor 'Verander van folder'.

Zo ga je naar de folder waar Processing instaat:

```
cd Programs/processing-3.1.1
```

Dit hoeft je niet zo te typen! Een terminal kan je woorden aanvullen als je op Tab drukt. Vaak is het volgende typen voldoende:

```
cd Progr[TAB]/pr[TAB]
```

Dit werkt ook als een andere versie van Processing op de computer staat :-)

Nu je in de juiste folder bent, start Processing:

```
./processing
```

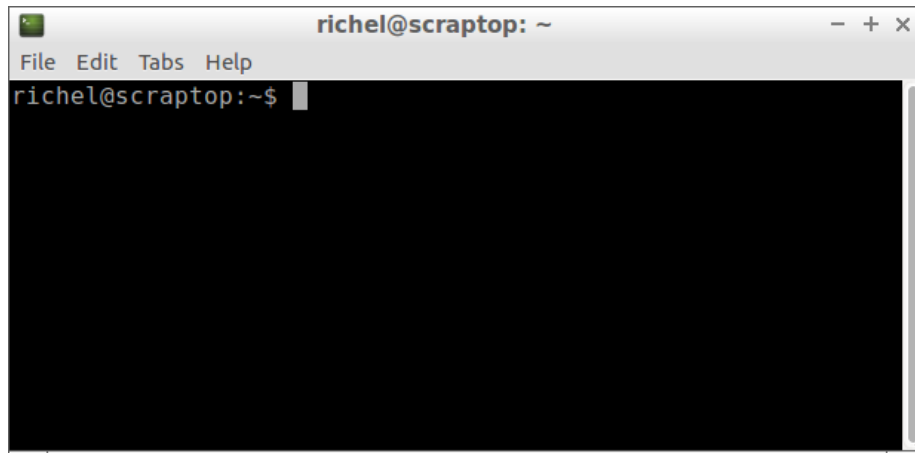


Figure 5: Een terminal

De `./` betekent 'Start hier'

Ook hier is Tab nuttig:

`./p[TAB]`

Je mag de terminal nu sluiten.

Opdrachten

- Start Processing

GitHub

GitHub is een site waar programmeurs samen aan programma's werken. Dit doen ze, omdat ze dan nooit meer hun code kwijtraken.

Wij gaan vandaag GitHub gebruiken om:

- Nooit meer code kwijt te raken
- Altijd bij onze code te kunnen

Wij gaan later GitHub gebruiken om:

- Samen te kunnen werken
- Onze games online to kunnen zetten

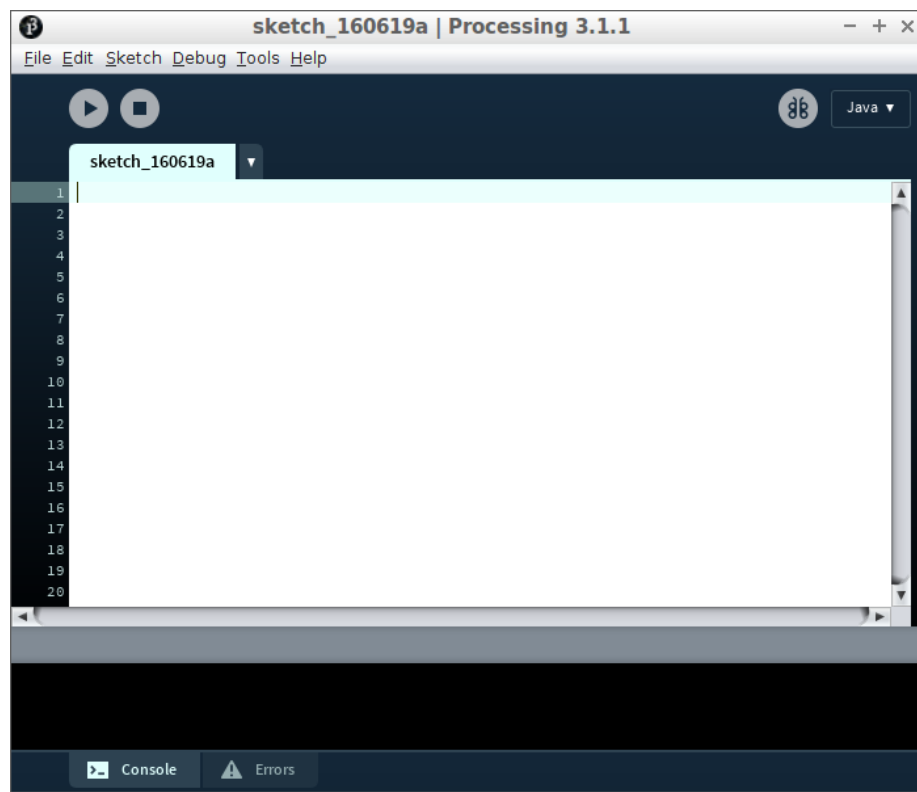


Figure 6: Processing zonder code

:exclamation: Waarschuwing :exclamation:

GitHub is openbaar. Iedereen kan zien wat je doet. Doe dus alleen slimme dingen op GitHub:

- Blijf altijd vriendelijk
- Blijf altijd netjes
- Grappen zijn prima, maar nooit als iemand zich beledigd voelt

GitHub aanmaken

In deze les maken we een GitHub account en gaan we onze code er op zetten.

Ga naar GitHub

- Start een webbrowser, zoals Firefox, Chromium, Chrome, Edge, Internet Explorer, Opera
- Ga naar www.github.com

Je komt nu op de GitHub homepage:

De kat heet Octocat en is de mascotte van GitHub.

Maak jezelf lid

- Klik op **Sign Up** (NL: **Inschrijven**)

Vul hier je gegevens in:

- **Username:** je gebruikersnaam. Kies geen gebruikersnaam waar je later spijt van gaat krijgen
- **Email address:** je emailadres. Dit moet je echte emailadres zijn
- **Password:** je wachtwoord. Deze moet een cijfer bevatten, bijvoorbeeld `iloverichel4ever` is een goede
- Klik **Create an account** om een account aan te maken.

Kies je account

In het volgende scherm moet je je soort account kiezen.

Wij gebruiken gewoon een gratis account

- Klik **Finish sign up** en je hebt een GitHub account!

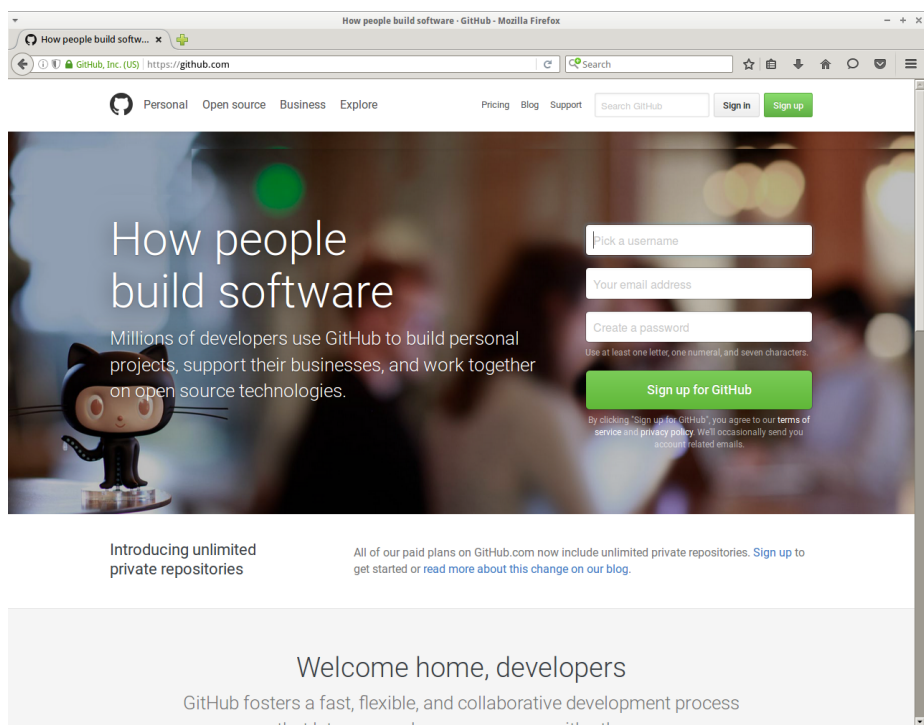


Figure 7: GitHub homepage

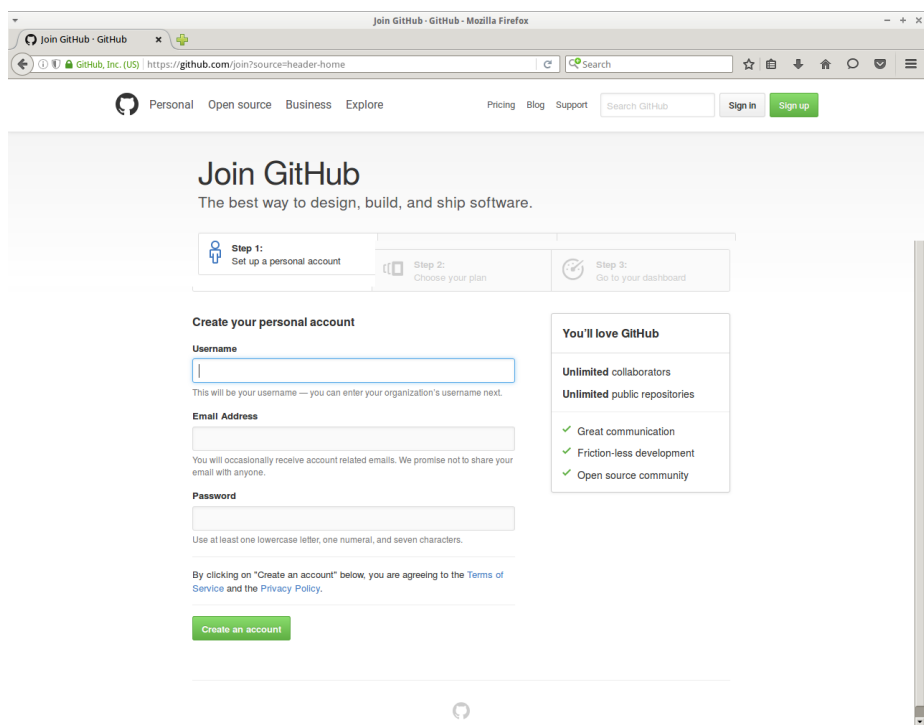


Figure 8: GitHub inschrijven

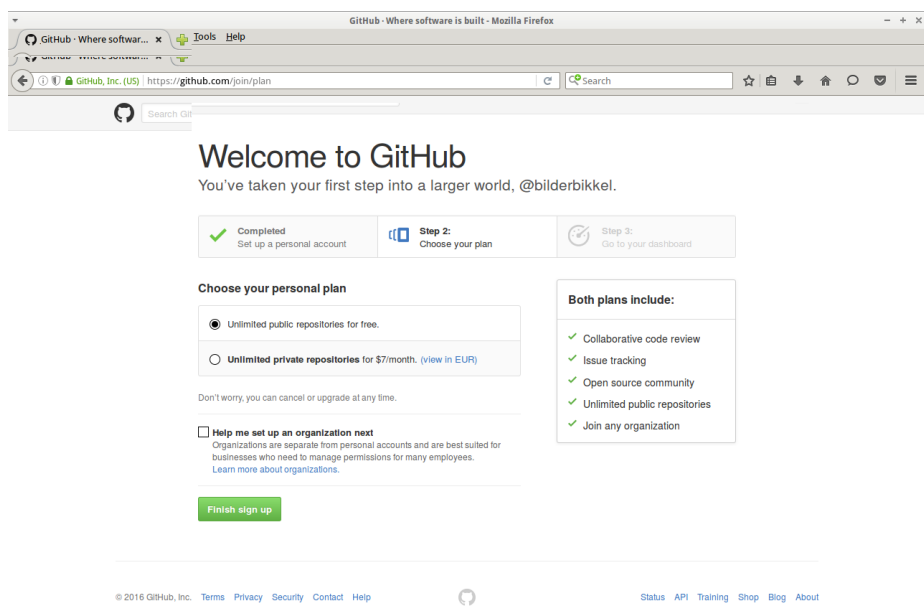


Figure 9: GitHub account kiezen

Vrienden maken

Nu je een account hebt, kun je vrienden maken

- Ga naar <https://github.com/richelbilderbeek> en klik op Follow om Richel te volgen
- Ga naar <https://github.com/thijsvb> en klik op Follow om Thijs te volgen
- Ga naar <https://github.com/LJK1991> en klik op Follow om Lucas te volgen
- Ga naar <https://github.com/Modanung> en klik op Follow om Frode te volgen

Bevestig je emailadres

- Ga naar je email toe
- Open het mailtje van GitHub
- Klik op Confirm om te bevestigen dat je een GitHub account hebt

Een eigen GitHub website aanmaken

:exclamation: Dit kan alleen als je je emailadres bevestigd hebt

Maak een nieuwe GitHub website

In bijna elk scherm, is er een kruis rechts bovenaan:

- Klik op Create New Repository om een nieuwe GitHub webpagina aan te maken

Stel de nieuwe GitHub website in

Vul hier in:

- Repository name: de naam van je GitHub website, bijvoorbeeld Minecraft
- Description: omschrijving van je GitHub website, bijvoorbeeld Minecraft-achtig spel, gemaakt in Processing
- Kies Public
- Vink aan: Initialize this repository with a README
- Kies bij Add .gitignore: Processing

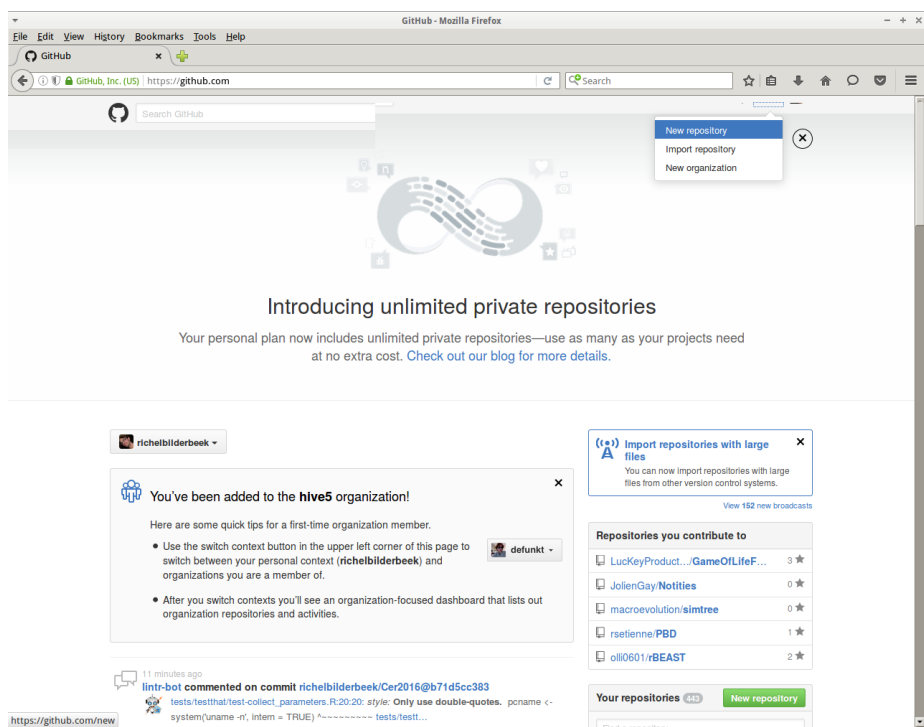


Figure 10: Maak een GitHub

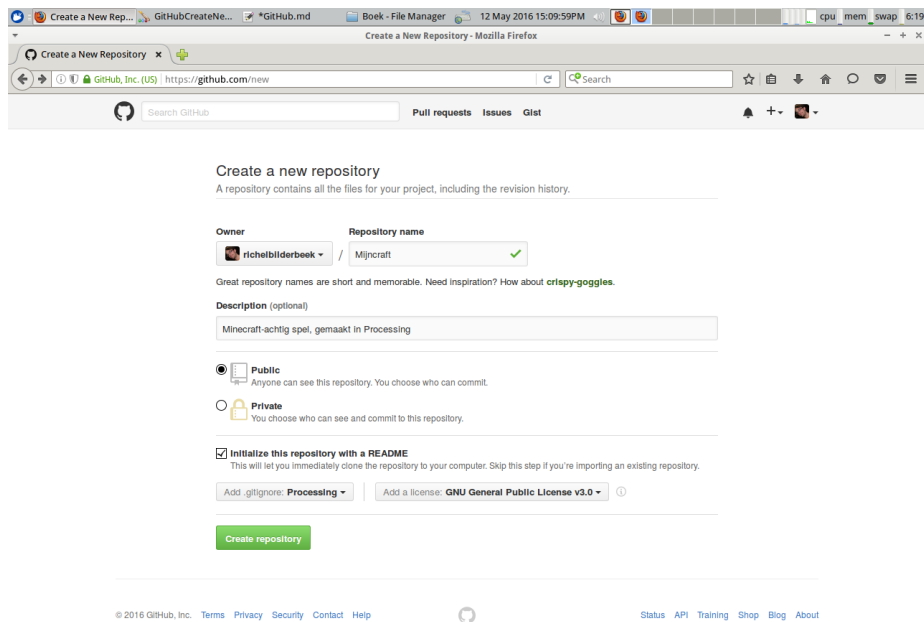


Figure 11: Maak een GitHub, eerste scherm

- Kies bij Add a license: GNU General Public License v3.0

Nu heb je je eigen GitHub website!

Zet dingen op je eigen GitHub website

Op je eigen GitHub website kun je veel dingen doen.

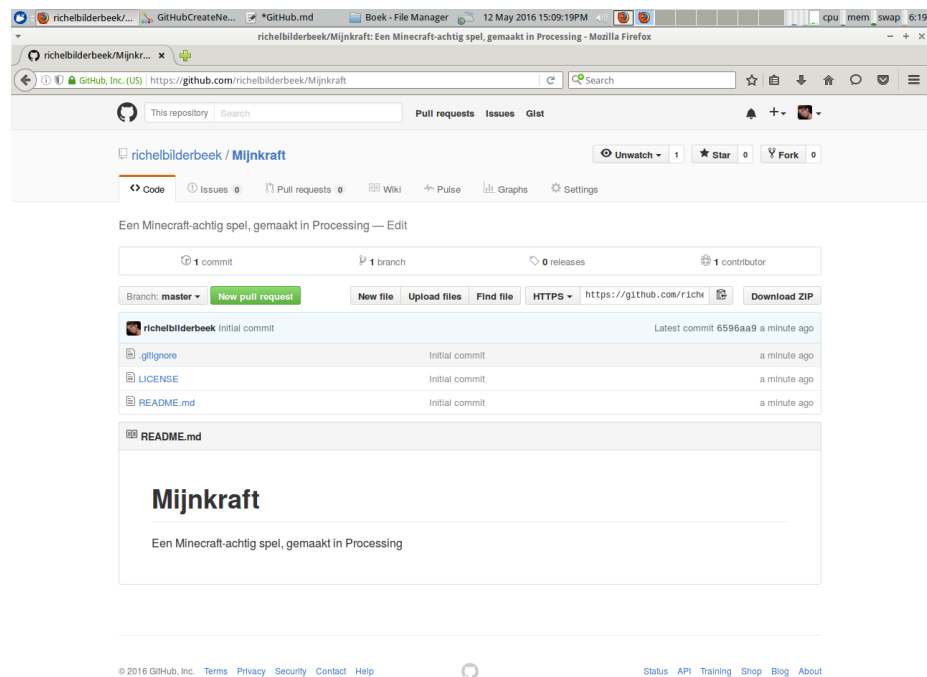


Figure 12: De Mijnkraft GitHub

- Klik op Upload file om een bestand op de website te zetten. Zet een mooi plaatje op de website.

Het bestand README.md bevat de welkomsttekst van je website. Dit gaan we aanpassen.

- Klik op het woord README.md
- Klik op het potloodje met de tekst Edit this file

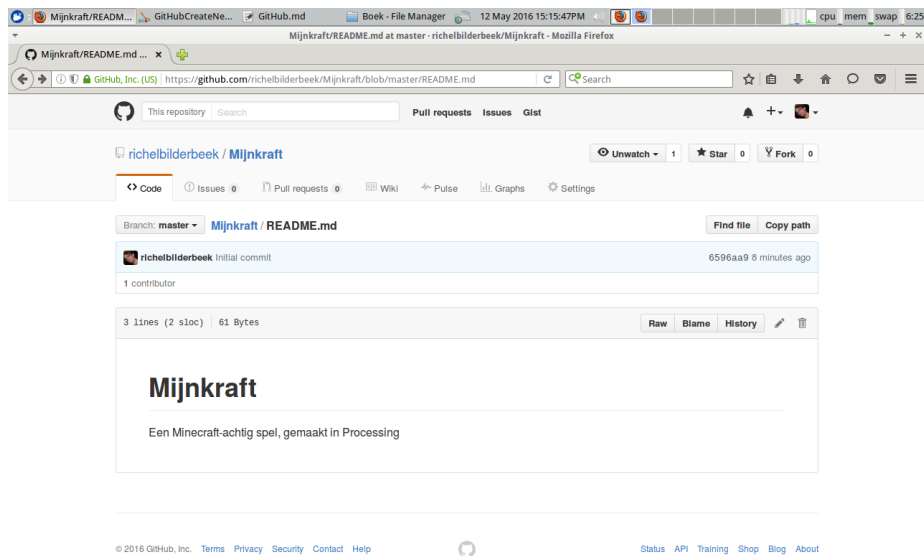


Figure 13: README.md wijzigen

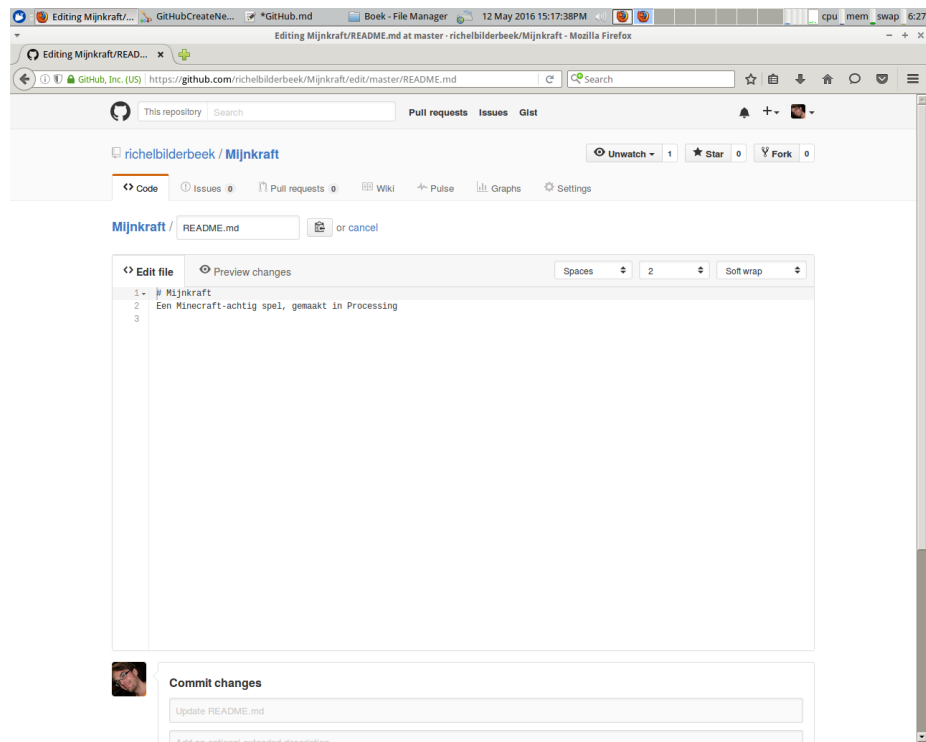


Figure 14: README.md wijzigen

- Zet hier iets leuks. Je kunt ook je plaatje laten zien, met bijvoorbeeld `![Dit vind ik leuk](plaatje.jpg)`
- Zet je code op GitHub

Als je nog geen GitHub account hebt, doe dan eerst [deze les](#)

GitHub Pages

GitHub Pages is een service van GitHub waarmee elke gebruiker zijn eigen website kan maken. Deze les gaan we een GitHub Page met homepage maken.

Het verschil met een repository is dat die op `www.github.com/gebruiker` staat, terwijl een GitHub Page op `www.gebruiker.github.io` staat.

Een repository gebruik je om: * Je code te bewaren * Je code te delen met de wereld * Samen aan code te werken

Een GitHub Page gebruik je om: * Mensen jouw spellen te laten spelen * De wereld te laten zien wat je kan

GitHub Page aanmaken

- Open een webbrowser
- Ga naar `www.github.com`

Je komt nu op de GitHub homepage:

- Klik op **Sign in** (NL: Aanmelden)
- Log in met je GitHub account (Heb je die niet? Doe dan eerst [deze les](#)!)

GitHub Page repository maken

- Klik op het plusje bovenin
- Klik op **New repository**

Vul hier in:

- **Repository name:** de naam van je GitHub Page moet zijn `gebruiker.github.io`, dus bijvoorbeeld `thijsvb.github.io`
- **Description:** omschrijving van je GitHub Page, bijvoorbeeld `Mijn geweldige website!`

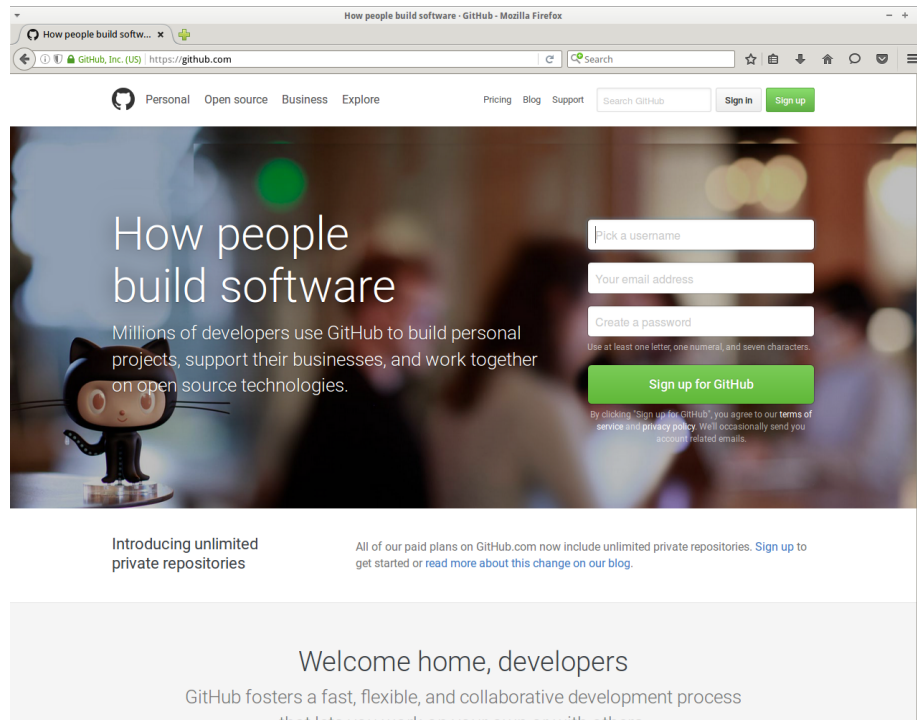


Figure 15: GitHub homepage

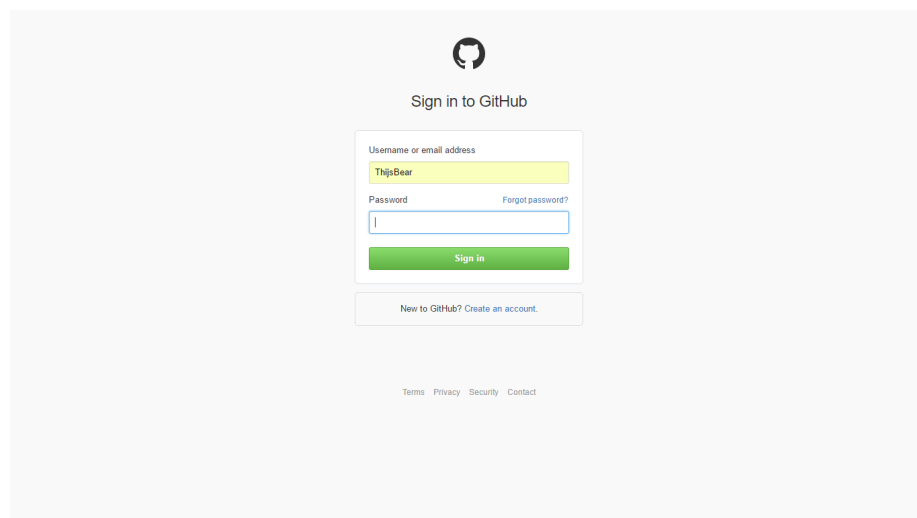


Figure 16: GitHub aanmelden

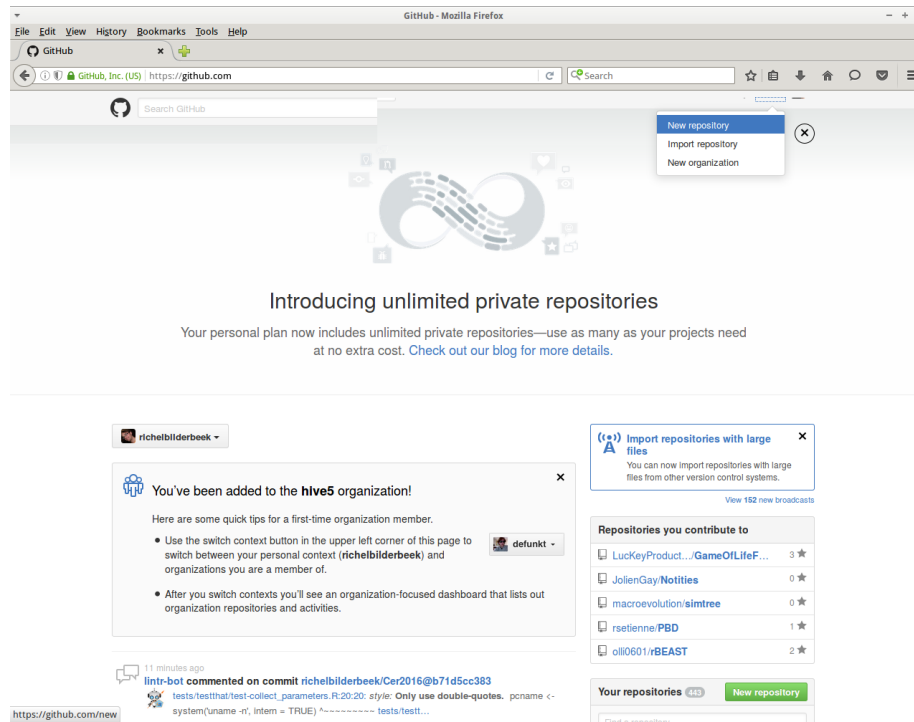


Figure 17: Maak een GitHub

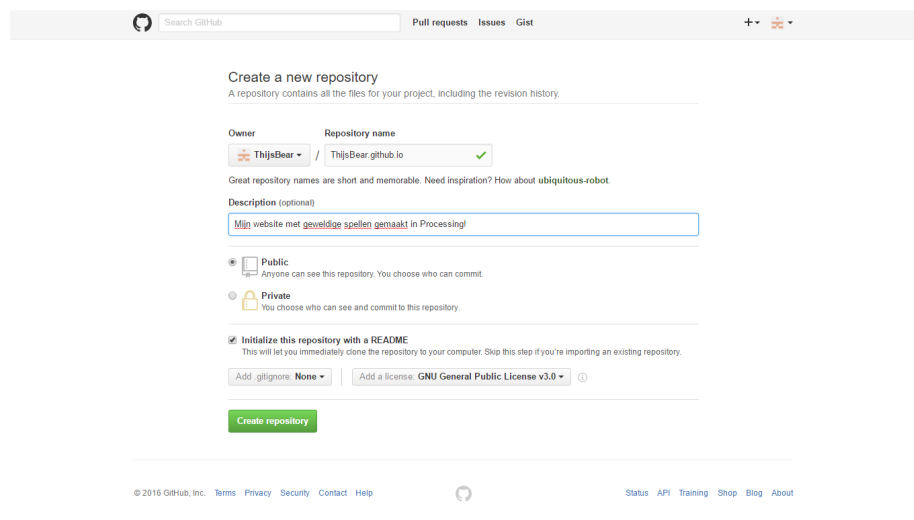


Figure 18: Maak GitHub Page repository

- Kies **Public**
- Vink aan: **Initialize this repository with a README**
- Kies bij **Add a license**: **GNU General Public License v3.0**

Je hebt nu een GitHub Page!

HTML

Als je nu in een browser naar je GitHub Page gaat, is er nog niks. Je moet een homepage maken, en om dat te doen moet je HTML-bestanden kunnen schrijven.

HTML staat voor HyperText Markup Language en is een opmaaktaal. Met een opmaaktaal kun je aangeven hoe tekst eruit moet komen te zien, een andere veel gebruikte opmaaktaal is Markdown (deze pagina is in Markdown geschreven).

Om tekst opmaak te doen in een programma zoals *Microsoft Word* heb je allerlei knoppen; dikgedrukte tekst, schuine tekst, plaatje, etc. In HTML heb je daar *tags* voor; ``, `<i></i>`, ``.

Een stukje HTML met een tag gaat altijd op deze manier:

```
<tag>(n)iets</tag>
```

- `<tag>` opent de tag, in plaats van `tag` staat hier het soort tag wat je wil gebruiken (bijvoorbeeld `h1` of `p`)
- `(n)iets` kan een heleboel zijn, zoals teks, nog meer tags, of helemaal niks
- `</tag>` sluit de tag, hier moet in plaats van `tag` het zelfde als in de openende tag staan

Een HTML-bestand begint altijd met `<!doctype html>`, dit verteld een programma wat het bestand opent dat het in HTML geschreven is. De rest van het bestand staat in één `html` tag, die weer is opgedeeld in één `head` tag en één `body` tag.

In de `head` tag staat informatie *over* de webpagina, in de `body` tag staat wat er *op* de webpagina komt.

Een stukje HTML kan er bijvoorbeeld zo uit zien:

```
<body>
  <h1>Hello World</h1>
  <p>This is <b>my</b> website!</p>
</body>
```

De homepage maken

Elke website is een map met bestanden, alleen staat de map niet op jouw computer. Een repository is ook een map, en de GitHub Page repository is ook jouw website. De homepage van elke website is het bestand `index.html`, die gaan we dus maken!

- Ga naar je GitHub Page repository

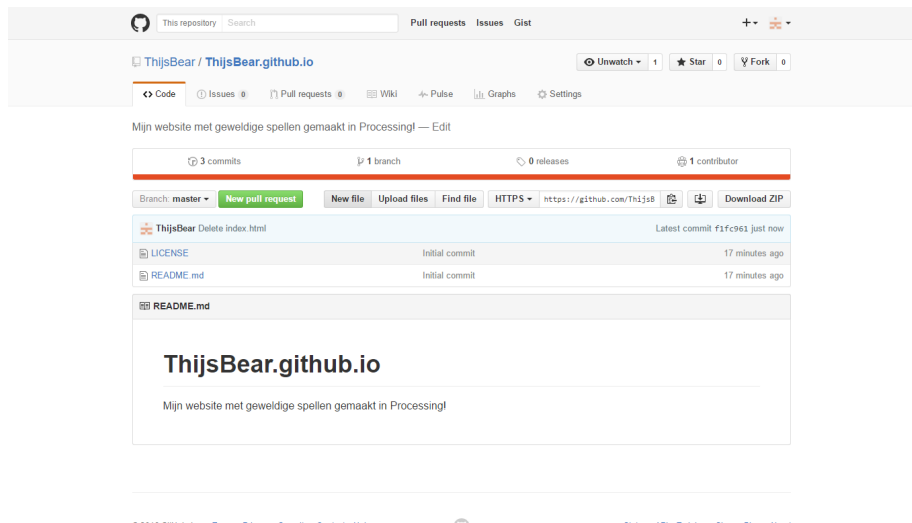


Figure 19: De GitHub Page repository

- Klik op **New file**
- Noem het nieuwe bestand `index.html`
- Begin het bestand met `<!doctype html>`
- Open op een nieuwe regel een `html` tag met `<html>`
- Open op een nieuwe regel een `head` tag met `<head>`
- Open op een nieuwe regel een `title` tag met `<title>`
- Typ hier de titel van je webpagina, bijvoorbeeld `Hello World`
- Sluit de `title` tag met `</title>`
- Sluit op een nieuwe regel de `head` tag met `</head>`
- Open op een nieuwe regel een `body` tag met `<body>`
- Open op een nieuwe regel een `h1` tag met `<h1>`
- Typ hier een kop voor je webpagina, bijvoorbeeld `Hallo`
- Sluit de `h1` tag met `</h1>`

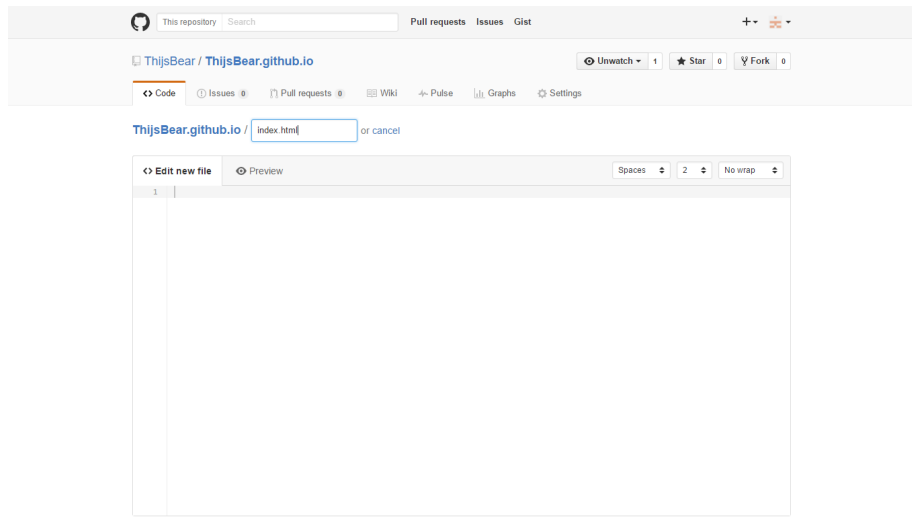


Figure 20: Maak index.html

- Open op een nieuwe regel een `p` tag met `<p>`
- Type hier een stukje tekst voor je webpagina, bijvoorbeeld `Welkom op mijn website`
- Sluit de `p` tag met `</p>`
- Sluit op een nieuwe regel de `body` tag met `</body>`
- Sluit op een nieuwe regel de `html` tag met `</html>`

Als het goed is ziet je bestand er nu zo uit:

- Klik onderaan op de groene **Commit new file** knop
- Ga in je browser naar je GitHub Page en bewonder je eigen website!
- Ga terug naar je GitHub repository
- Klik bij `index.html` op het podloodje om het bestand te bewerken
- Maak een paar veranderingen en klik op **Commit changes**
- Bewonder nu je eigen gepersonaliseerde website en bekijk ook eens die van je klasgenoten!

Als je nog geen GitHub account hebt, doe dan eerst [deze les](#)

Als je nog geen GitHub Page hebt, doe dan eerst [deze les](#)

Processing.js

Processing.js is een JavaScript wat een Processing sketch op een webpagina kan runnen. Deze les gaan we een Processing sketch op onze GitHub Page zetten.

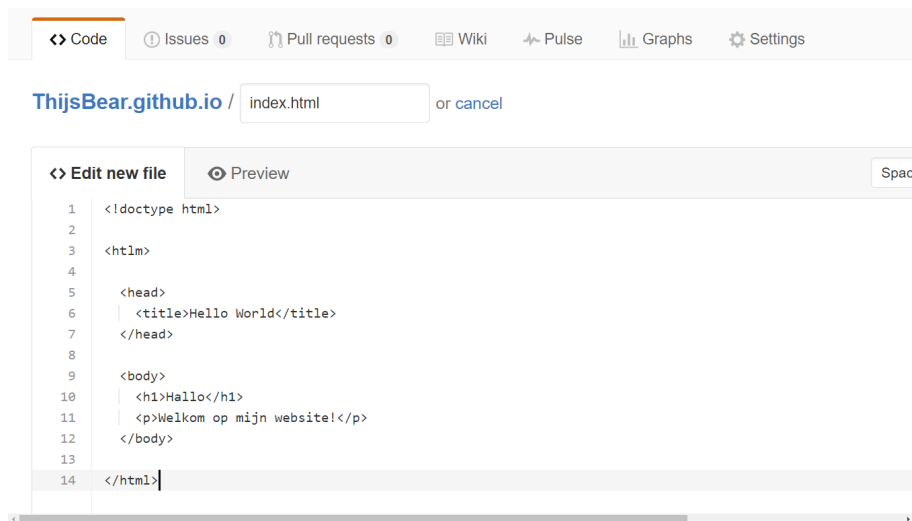


Figure 21: index.html

Met Processing.js kunnen mensen jou spel spelen, kunstwerk bewonderen of programma runnen op jouw GitHub Page. De code voor het spel, kunstwerk of programma staat op een GitHub repository.

Processing.js downloaden

Processing.js is een JavaScript bestand. Om het te gebruiken moeten we het eerst downloaden. * Ga in een webbrowser naar www.processingjs.org

Je komt nu op de homepage

- Klik bovenin op **download**

Je komt nu op de download pagina

- Klik met de rechter muisknop op **processing.js**
- Klik op **Link opslaan als...** (in het Engels: **Save link as...**)
- Kies een plek op je computer waar je het bestand zo terug kan vinden (Tip: **Bureaublad** of **Desktop** is vaak makkelijk te vinden!)
- Ga naar je GitHub Page repository
- Klik op **Upload file**

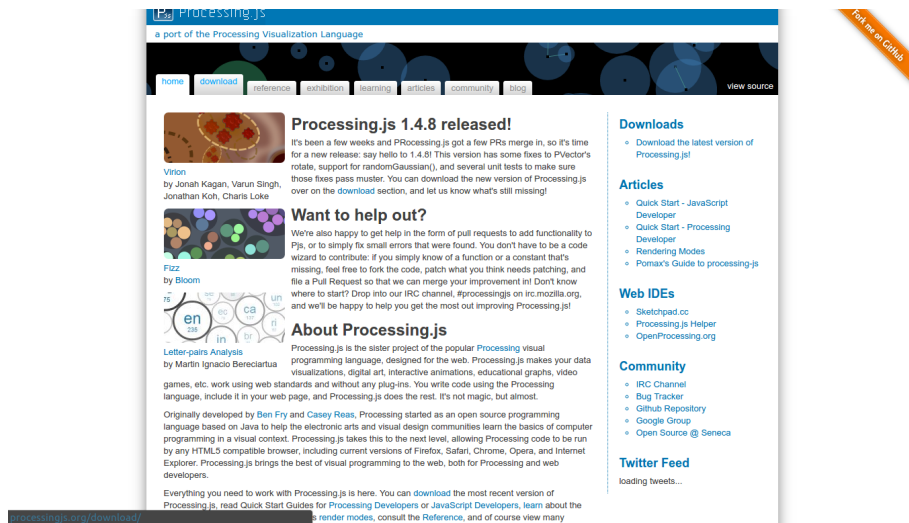


Figure 22: ProcessingJS homepage

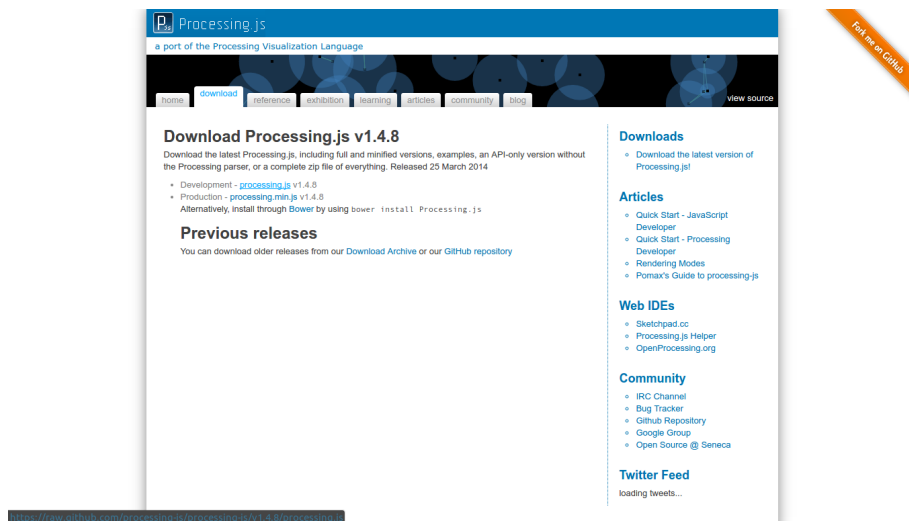


Figure 23: ProcessingJS download

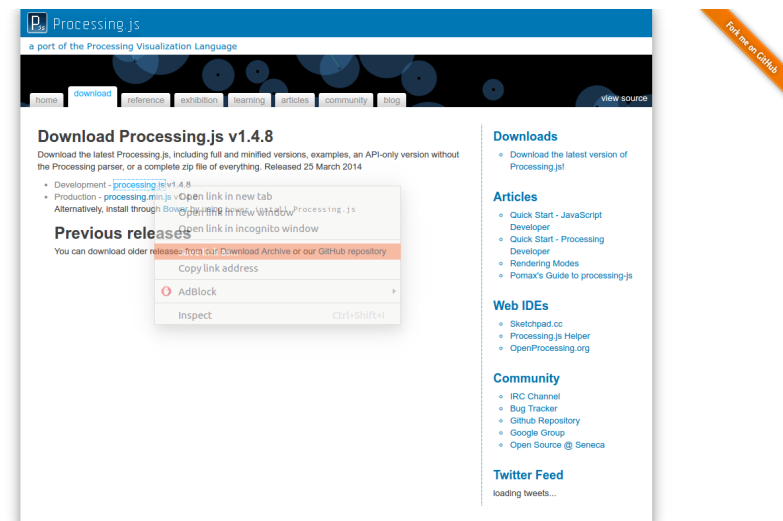


Figure 24: Link opslaan als

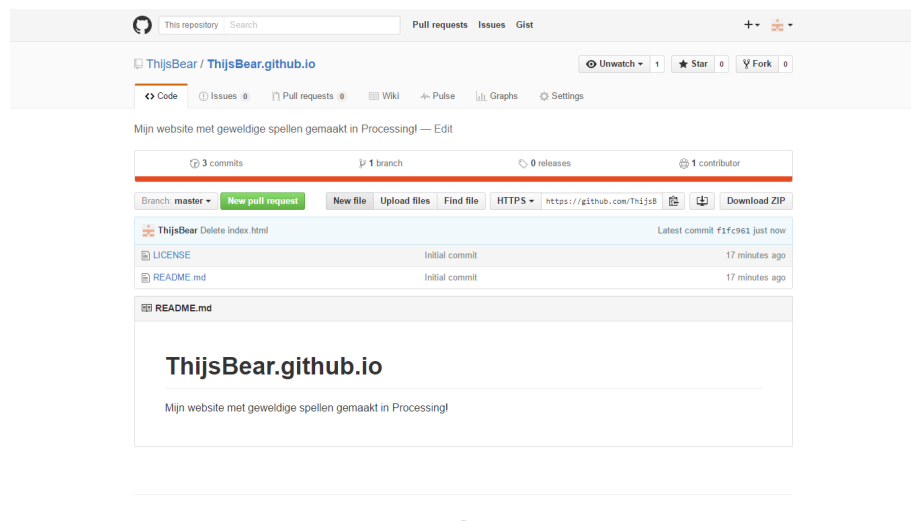


Figure 25: GitHub Page repository

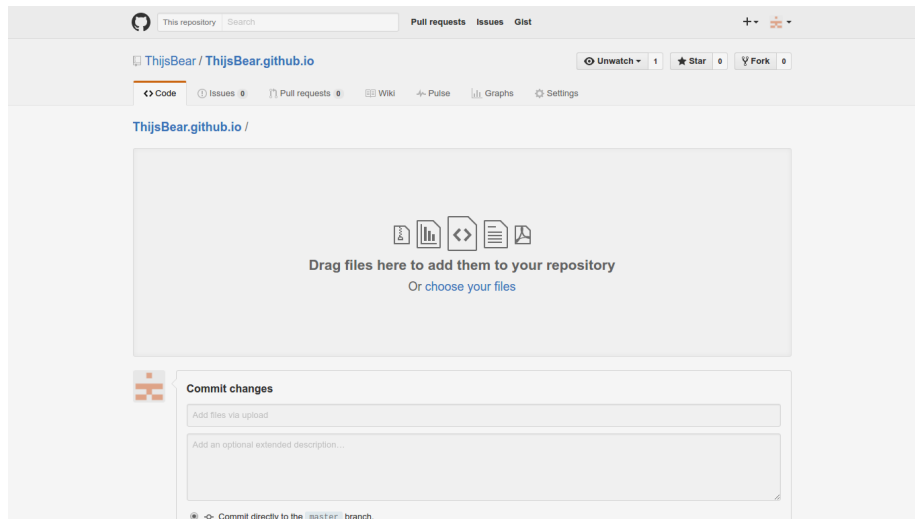


Figure 26: GitHub upload file

- Sleep `processing.js` dit scherm in of klik op `select your files` en dubbelklik op `processing.js`
- Klik onderaan op de groene knop `Commit changes`

Processing sketch uploaden

- Zoek een Processing sketch op je computer om op je website te zetten (Tip: kies een simpele sketch)
- Als je in de sketch map kijkt staat daar een `.pde` bestand, deze hebben we straks nodig
- Ga naar je GitHub Page repository
- Klik op `Upload file`
- Sleep het `.pde` bestand dit scherm in of klik op `select your files` en dubbelklik op het `.pde` bestand
- Klik onderaan op de groene knop `Commit changes`

Nu ga je de sketch op je homepage zetten

- Ga naar je GitHub Page repository

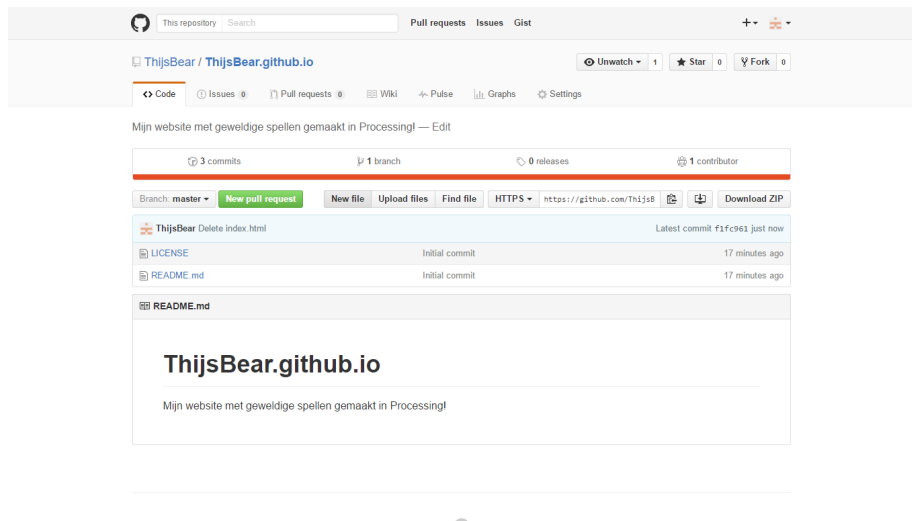


Figure 27: GitHub Page repository

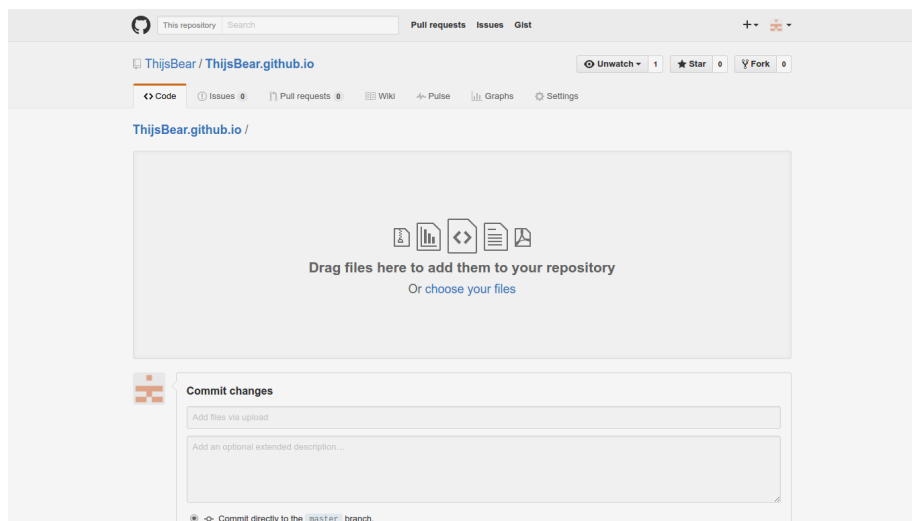


Figure 28: GitHub upload file

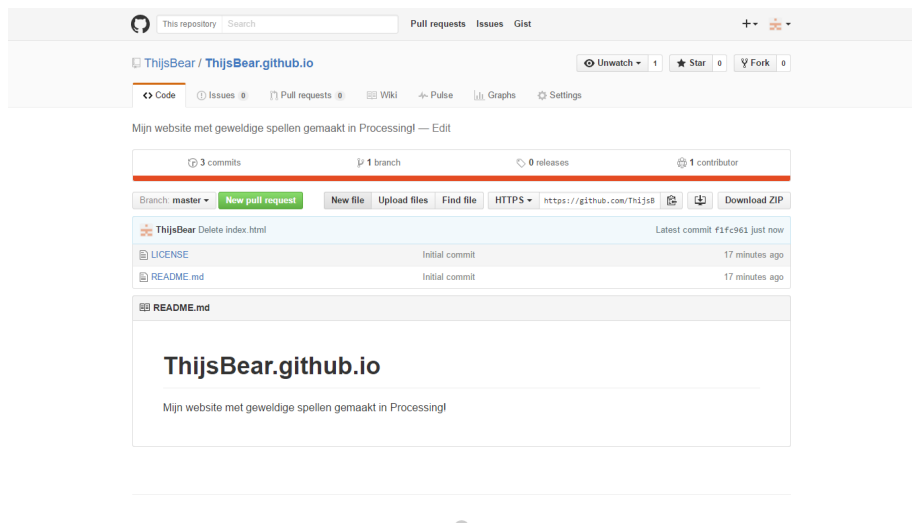


Figure 29: GitHub Page repository

- Klik op `index.html`
- Klik op het podloodje rechts boven
- Open een nieuwe regel in head (dus tussen `<head>` en `</head>`)
- Typ hier deze regel over: `<script src="processing.js"></script>`
- Open een nieuwe regel in body (dus tussen `<body>` en `</body>`)
- Typ hier deze regel over: `<canvas data-processing-sources="voorbeeld.pde"></canvas>`
- Vervang nu `voorbeeld` voor de naam van je `.pde` bestand

Het bestand moet er nu ongeveer zo uit zien:

- Klik onderaan op de groene knop **Commit changes**
- Ga naar je website en bewonder je werk!

GitHub chat

In deze les gaan we de chatroom van de cursus in.

Wat is een chatroom?

Er zijn meer soorten chatrooms. In deze les gebruiken we een IRC chatroom. IRC bestaat sinds 1986 en wordt nog veel gebruikt. Elke hackerspace heeft wel een eigen IRC chatroom (dit noemen we ook een 'IRC kanaal'). Een IRC

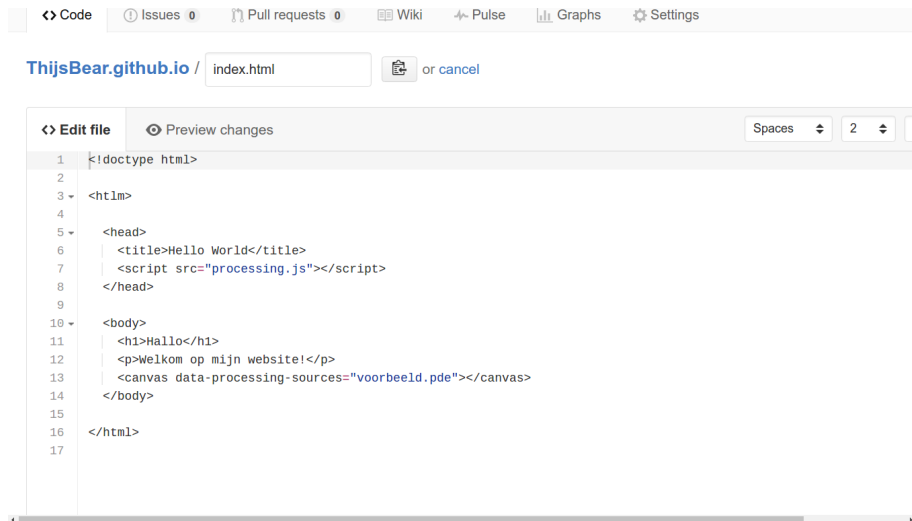


Figure 30: index

chatroom is net een echt gesprek met een groepje mensen: je weet niet wat er is gezegd voordat je met ze sprak, en je weet niet wat ze zeggen als je weer weggaat. Er zijn wel IRC chatrooms die wel de gesprekken bewaren.

Gedragsregels

Een paar van <https://wiki.debian.org/IRC/Netiquette>:

- Wees behulpzaam
- Praat over het thema, in ons geval programmeren
- Probeer mee te doen met het gesprek, inplaats van nieuwe gesprekken te starten
- Stel jezelf voor als je mee doet in een gesprek en nieuw bent
- ‘Don’t feed the trolls’. Help de mensen die de sfeer verpesten niet hierbij: blijf altijd vriendelijk
- Heb plezier

Website

Ga naar de GitHub van de cursus, <https://github.com/richelbilderbeek/Dojo>. Klik op irc: **#dojoGroningen**

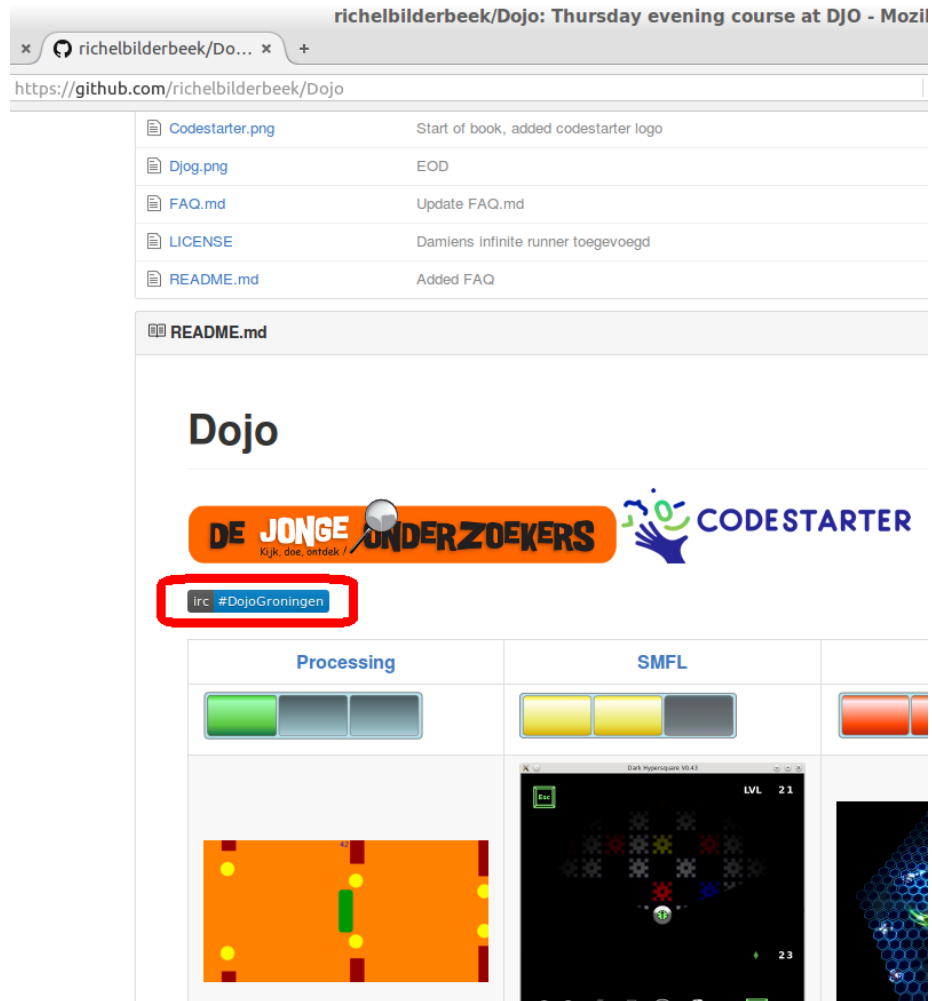


Figure 31: GitHub.png

IRC inloggen

- Vul bij Nickname je bijnaam in
- Klik op I am not a robot

Some krijg je een vraag, zoals hier:

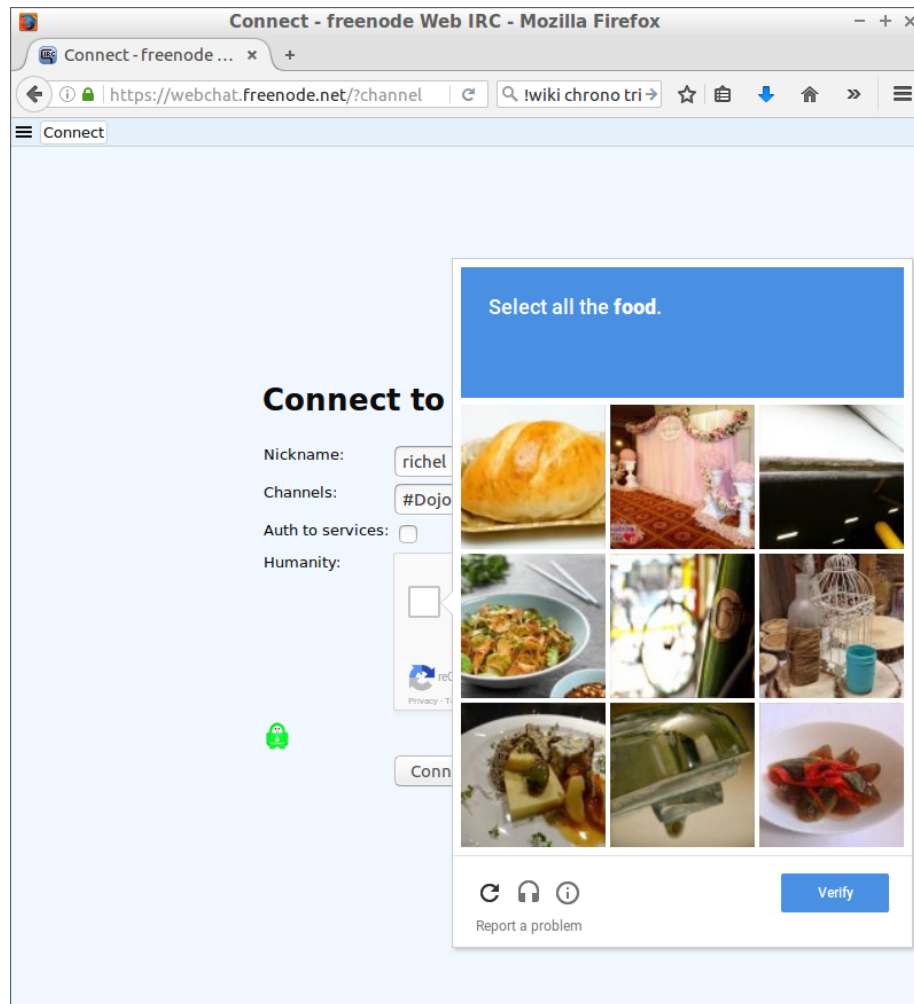


Figure 32: IrcInlog.png

Beantwoord de vraag.

Klik dan op Connect

IRC server login

Eerst komt er een scherm met veel tekst:

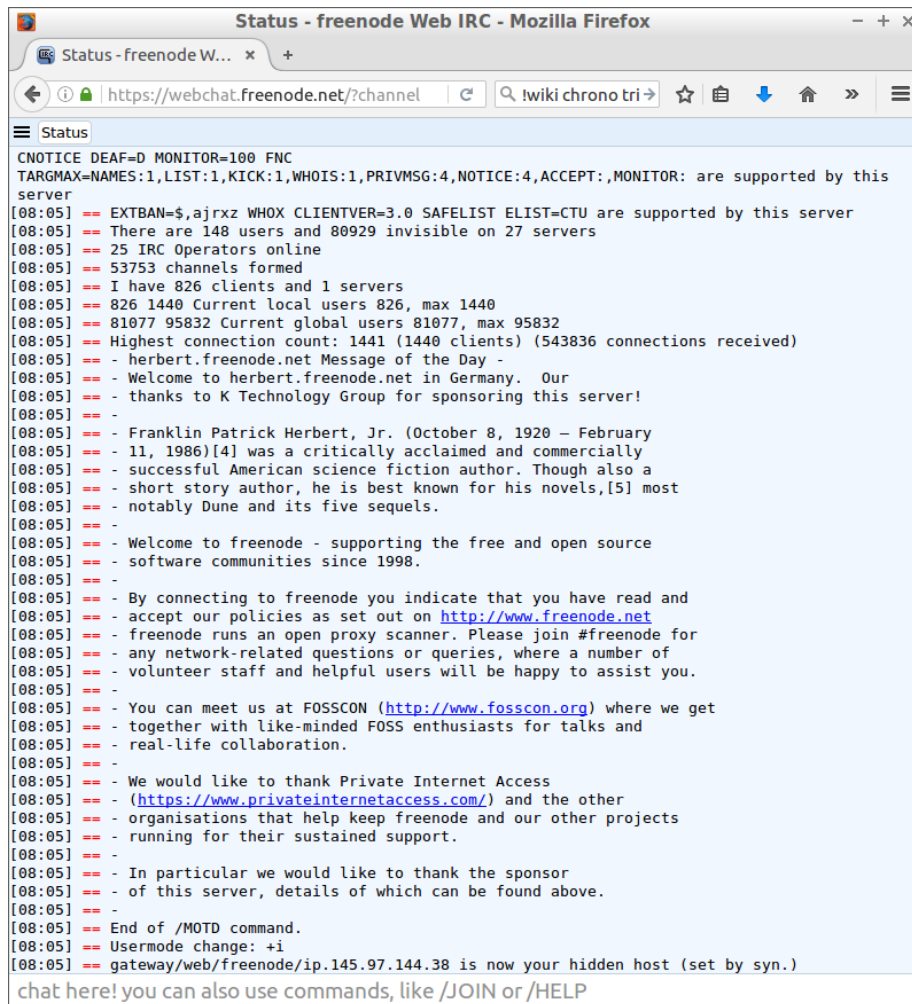


Figure 33: IrcServerMessages.png

Wacht tot dit scherm voorbij is

IRC chatroom

Je komt nu in de chatroom:

Rechts kun je een lijst zien van iedereen die in deze kamer aanwezig is.

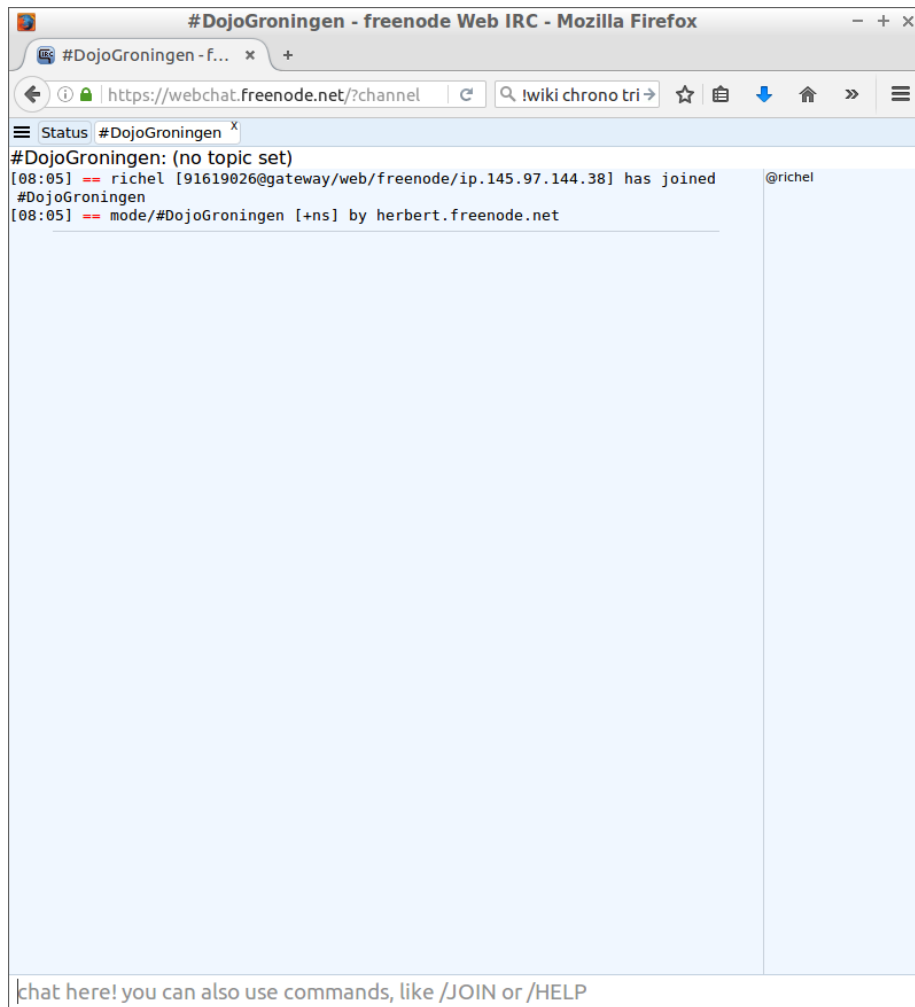


Figure 34: IrcChatroom1.png

Onderin kun je je tekst typen:

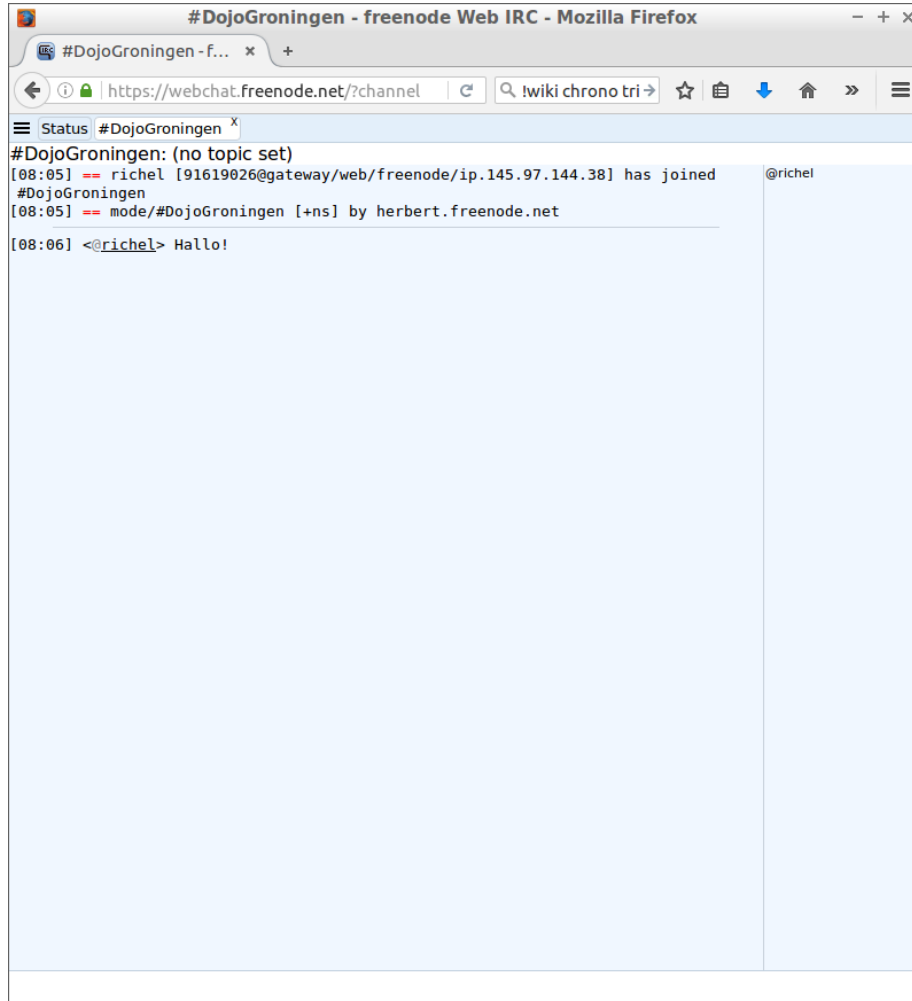


Figure 35: IrcChatroom2.png

De tekst verschijnt bij iedereen meteen op hun scherm.

git, GitHub samenwerken

Samenwerken is lastig, zeker met programmeren: de ander (of jij) maakt misschien dingen stuk en/of kwijt.

In deze les gaan we leren samenwerken met git en GitHub. Dit doen we (nog) niet met code, maar met een Nederlandse tekst.

git, GitHub

git houdt de geschiedenis van de tekst/code bij. Je kunt ook terug in de geschiedenis gaan. Als je per ongeluk een bestand hebt gewist, kun je deze weer terughalen.

Sprookje

In deze les gaan we samen een sprookje maken. Hier staat de begintekst:

```
# Sprookje
```

```
## 1
```

```
Heel erg lang geleden ...
```

```
## 2
```

```
## 3
```

```
## 4
```

```
## 5
```

```
## 6
```

```
## 7
```

```
... en ze leefden nog lang en gelukkig.
```

Het sprookje is nog erg saai. Samen kunnen we er een leuk verhaal van maken.

De GitHub op je computer zetten

Om het sprookje op je computer te zetten, start je (onder Windows) Git Bash, onder Linux een terminal.

Type dan het volgende:

```
git clone https://github.com/richelbilderbeek/Sprookje
```

Er is nu een map gemaakt, genaamd **Sprookje** met daarin het bestand **README.md**. **README.md** bevat de tekst van het sprookje en kun je openen in Kladblok/Wordpad/leafpad of een andere teksteditor.

Samenwerken

Verander de tekst in `README.md` en save.

Doe dan in Git Bash:

```
git pull
git add --all :/
git commit -m "Verbetering"
git push
```

- `git pull`: update de GitHub
- `git add --all :/`: bekijk alles wat veranderd is
- `git commit -m "Verbetering"`: benoem de veranderingen `Verbetering`
- `git push`: publiceer je veranderingen

Merge conflicts

Soms krijg je ‘Merge conflicts’. Dit gebeurt als `git` niet weet, waar welke veranderingen horen.

Stel, je hebt dit:

```
## 1
```

```
## 2
```

Nu maakt Jantje er dit van:

```
## 1
```

```
Er woonde een gevaarlijke draak in de berg ten oosten van het dorpje.
```

```
## 2
```

En maakt Truus er tegelijkertijd dit van:

```
## 1
```

```
De prinses was op zoek naar de knapste ridder.
```

```
## 2
```

`git` kan dit niet samenvoegen, omdat de volgorde tussen de zinnen onduidelijk is.

Degene die het merge conflict krijgt, krijgt een `README.md` zoals dit:



Figure 36: Voorblad



Figure 37: Het logo van De Jonge Onderzoekers



Figure 38: Het logo van Codestarter



Figure 39: De licentie van dit boek