Oppdrag 1

1. Nettverk og tilkobling

Koblet begge maskiner på Kuben.it internett

Satt en statisk IP på Raspberry Pi-en

Et bilde som inneholder tekst, skjermbilde, multimedia, programvare

KI-generert innhold kan være feil.

Jeg satt IP metoden til ‘Manual’ istedenfor ‘Automatic’ fordi manual metoden lar meg redigere IP-adressen.

Som man kan se på bildet satt jeg IP-en til å være 10.200.10.25

Nettmasken er 255.0.0.0, samme som nettmasken på andre maskinen

Også ruter Gateway er samme som på andre maskinen fordi begge maskiner er tilkoblet på samme ruter

Når jeg ble ferdig, så trykka jeg på ‘Apply’

Et bilde som inneholder tekst, Font, skjermbilde

KI-generert innhold kan være feil.

Etter at jeg lagra alle endringer så skrev jeg **‘hostname -I’** for å sjekke om IP-adressen endra seg. Man kan se oppe at IP-en bytta til en statisk IP jeg skrev.

Under det prøver jeg å pinge (prøver å få en melding tilbake av) en annen maskin, og her kan man se at jeg får tilbake svar fra IP-en 10.2.2.27 som tilhører PC-en min.

Et bilde som inneholder tekst, elektronikk, datamaskin, skjermbilde

KI-generert innhold kan være feil.

Her kan man se Windows sin perspektiv på pinging.

Her pinger jeg IP-adressen til Raspberry Pi, og får tilbake svar.

Hvorfor satt jeg opp statisk IP på Pi-en via innstillinger og ikke via terminalen?

* **Brukervennlighet**, altså det er mer oversiktlig når man setter den via innstillinger
* Når man setter IP-en via settings så unngår man skrivefeil i koden når mansetter den via terminalen. Skrivefeil kan føre til at nettverket funker ikke (skjedde med meg), og det kan bli veldig slitsomt
* Innstillinger viser all informasjon om nettverket så du har oversikt over feilsøking av ting

1. Server og tjenester (webserver)

Først, lagde jeg er html mappe på Linux der jeg skrev en kort kode

Et bilde som inneholder tekst, skjermbilde, display, Displayenhet

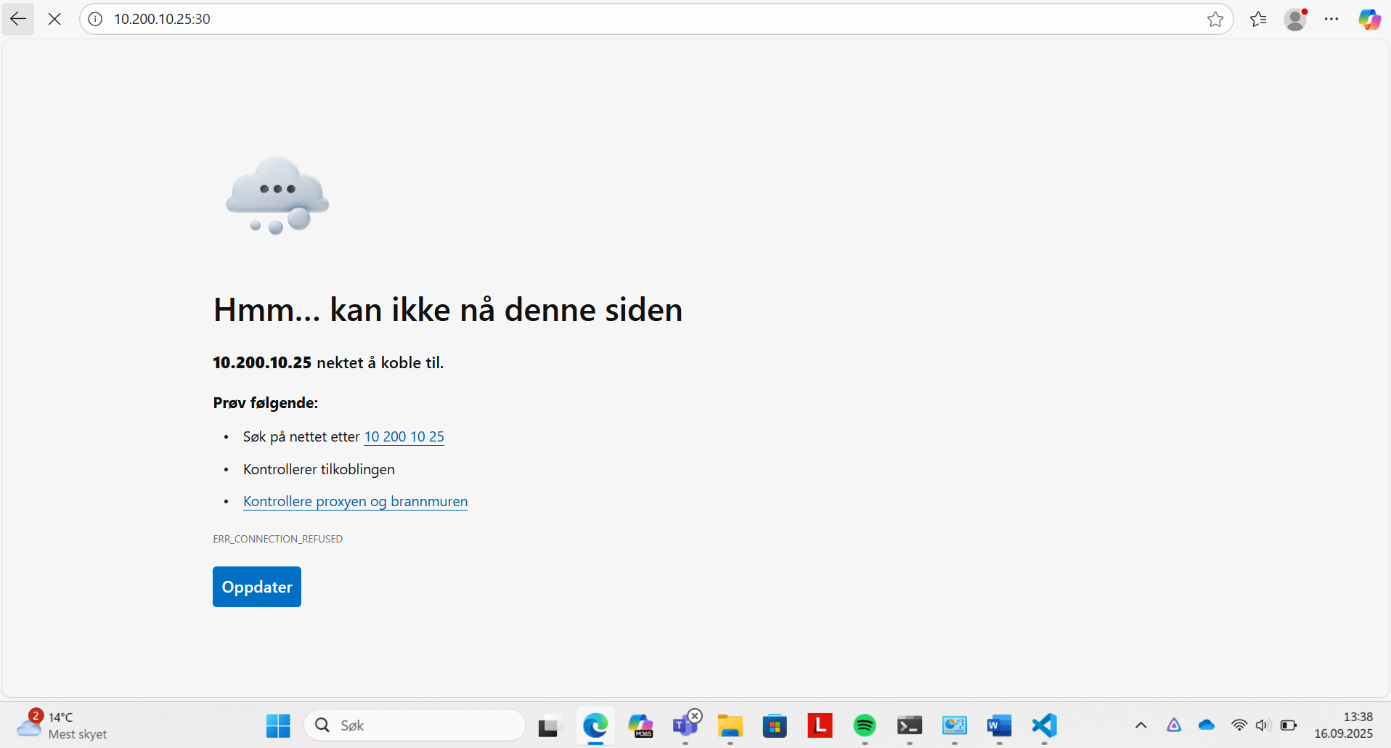
KI-generert innhold kan være feil.

Ved hjelp av koden **‘nano filnavn’** så skrev jeg en liten kode for at den skal være på nettsiden og lagra den ved å trykke **CTRL + S**.

Et bilde som inneholder tekst, skjermbilde, Font, informasjon

KI-generert innhold kan være feil.

Etter det, prøvde jeg å starte en http-server gjennom python ved å skrive ‘**sudo python3 -m http.server 30’**



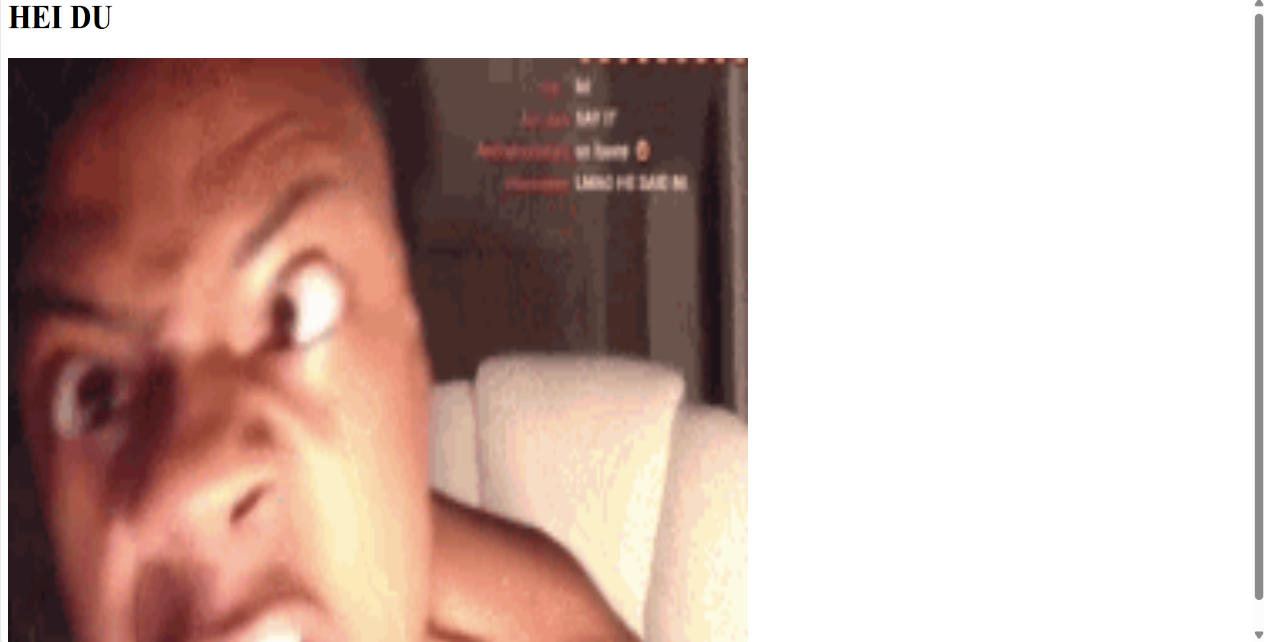
Som man kan se her, det er umulig for nettsiden til å laste opp, og det er en grunn til det

Et bilde som inneholder tekst, skjermbilde, meny, display

KI-generert innhold kan være feil.

Nemlig, grunnen til at nettsiden ville ikke laste opp var fordi port 30 var ikke lagt inn som en regel i brannmuren

Som man kan se opp i koden så måtte jeg skrive **‘sudo ufw allow 30’** til at brannmuren skal legge inn en regel for at port 30 kan brukes.

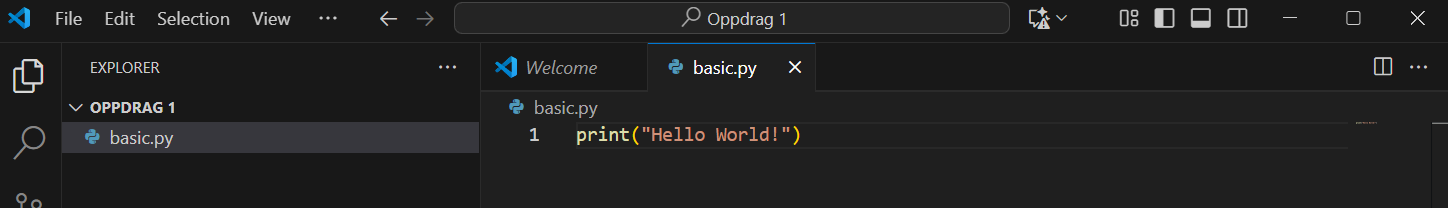


Etter å ha lagt ut regelen for port 30, så nettsiden riktig ut for brukere på nettverket

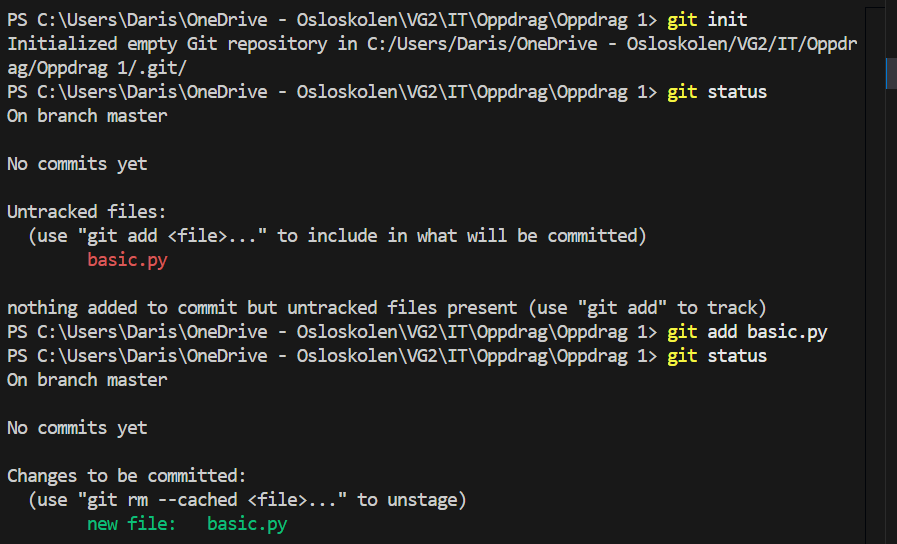
Hvorfor valgte jeg å bruke Python http-server istedenfor Apache for webserveren?

* Mer erfaring fra tidligere uker innen Python-http-server set up enn Apache. Har alt for lite erfaring i Apache.
* 2Det er veldig enkelt å sette opp en Python-http-server, og det er ingen tjenester eller andre ting som må startes bortsett fra brannmuren for en bestemt port man bruker
* Python er universell, og fungerer likt på operativsystemer som Linux og Windows. På Apache så er det forskjellig set up per operativsystem.

1. Python og GitHub



Først lagde jeg en Python-fil som heter **‘basic.py’** hvor jeg skrev en lett kode som printer ut ‘Hello World!’

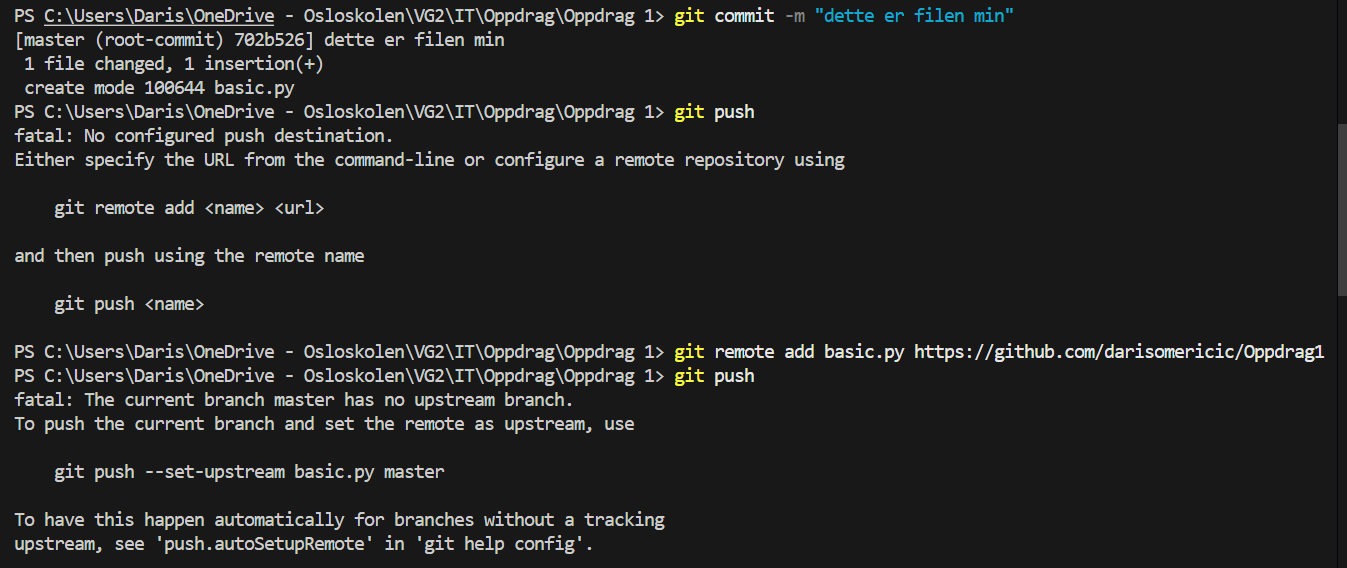


Da jeg skrev koden åpnet jeg terminalen for å overføre filen min til GitHub

Først startet jeg Git-en inni mappen min ved å skrive inn **‘git init’**

Da sjekker jeg status til Git-en for å sjekke om filen min er der (det er ikke så viktig å gjøre det, men fornuftig). Som man ser på bilde, filen er der, men er rødt, og det er fordi jeg legga den ikke til **mellomlagringen** som er en midlertidig lagring før jeg legger filen i GitHub

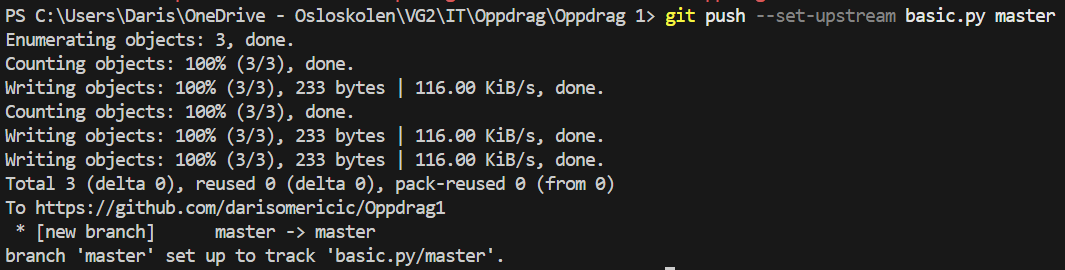
Når jeg skrev koden **‘git add basic.py’** så ser man at ny fil er legget til i mellomlagring, og er klar for overføring



Når jeg satt filen i mellomlagring så er det tid til å lagre permanent alt som ligger i mellomlagringen som en ny versjon i Git, og det gjør jeg ved å skrive koden **‘git commit’**, men jeg valgte å legge en melding så jeg skrev **‘git commit -m ‘dette er filen min’’**

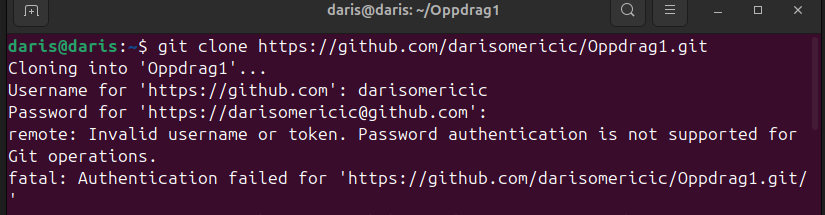
Når det ble ferdig så prøvde jeg å pushe filen, og det gikk selvfølgelig ikke fordi jeg glemte å velge en repository hvor filen skal bli overført til.

Jeg måtte skrive koden ‘**git remote ‘filnavn’ ‘url’’** for at jeg kunne dytte filen til GitHub



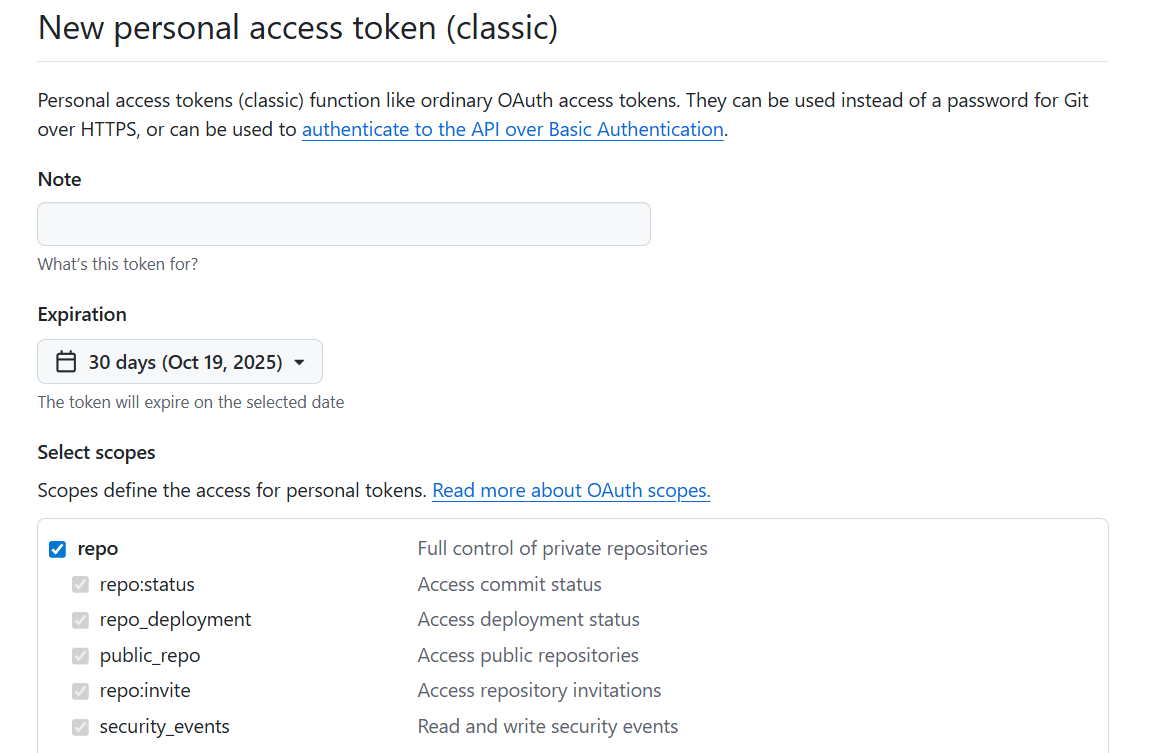
Når jeg endelig fikk fikse hvilken repository jeg skulle putte filen i, så var det tid til å dytte filen, men ikke med klassisk **‘git push’** men den koden man ser over på bildet

Grunnen til at jeg dytta filene sånn som på bildet er for å sette en kobling mellom filen min og GitHub sånn at jeg kan bruke kommandoen pull og push når jeg vil endre filen

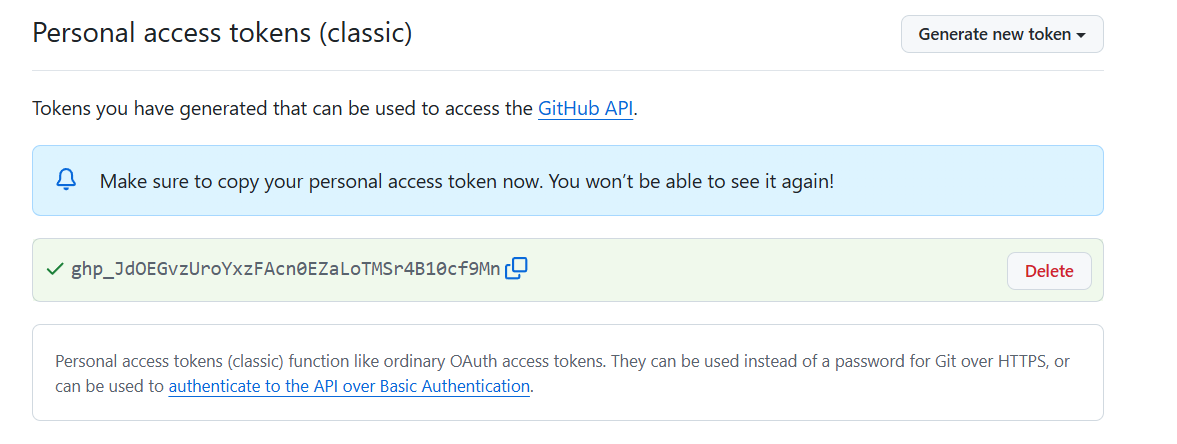


Første gang jeg prøvde å klone filen til **Raspberry Pi** slet jeg med å forstå hva var feil med passordet koblet til GitHub-en min, og prøvde også til å resette den

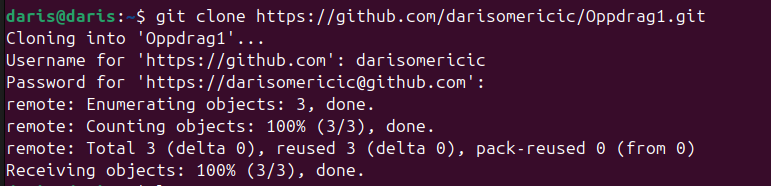
Raskt skjønte jeg at jeg må bruke noe som heter **‘GitHub token’** for å kunne sette en kobling mellom Pi-en og GitHub-en



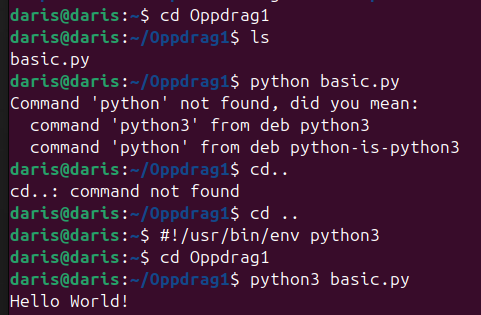
Her er vinduet, hvor jeg kan lage **‘access token’** for å få tilgang til filene mine på **Raspberry Pi-en,** og her kan man se ett vindu som heter **‘Select scopes’** og under den er det en kategori som heter **‘repo’** eller ‘**repository’.** Uten å slå denne på så kan man ikke klone filen til Raspberry Pi fordi alle filene ligger i et repository i GitHub.



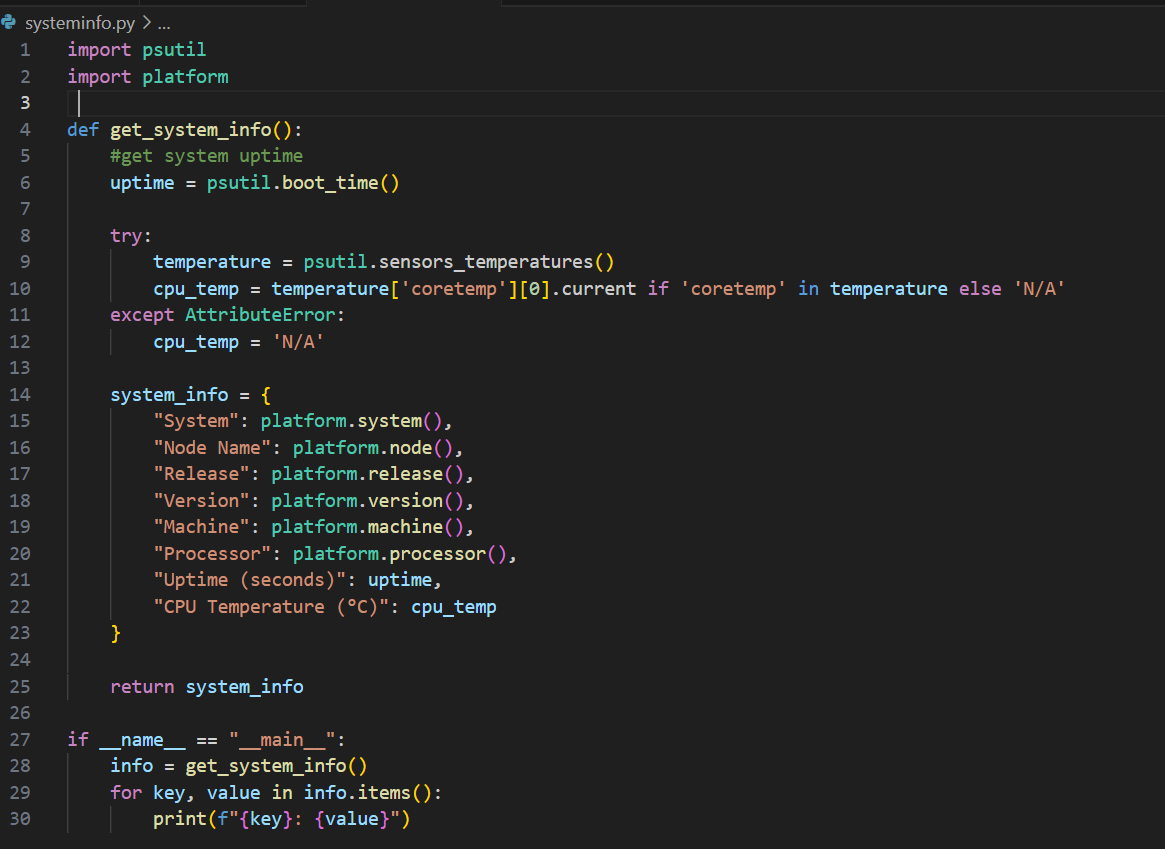
Etter å ha laget en **‘access token’** så får man denne koden som en tok man skal bruke istedenfor passordet sitt når man skal klone filen til Raspberry Pi-en



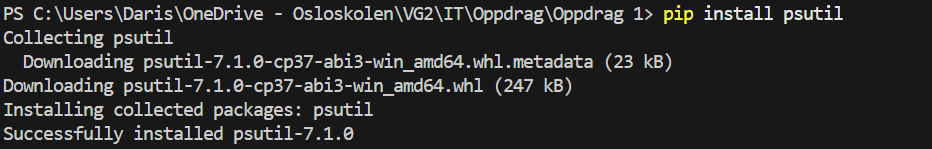
Etter at man har skrevet passord til tokenet riktig, så burde man få tilgang til filen sin i Raspberry Pi, og den burde se sånn ut



Da jeg fikk tilgang så flytta jeg meg til ‘Oppdrag’ filen og sjekka om Python filen var der, og det var den. Med en gang så prøvde jeg å kjøre koden, fordi jeg trodde det funker som på terminalen i Windows, men det gjør det ikke. Det jeg måtte gjøre er å skrive koden **‘#!/usr/bin/env python3’** som spesifiserer tolken som skal utføre scripten. Når jeg gjorde det så kjørte jeg koden ved å skrive **‘python3 basic.py’** og fikk **‘Hello World’** printet ut.



Da jeg fullførte oppgaven, så kom jeg på en ide på å lage systeminformasjon og kjøre den inn på Raspberry Pi-en



Koden inneholder **‘import psutil’** så jeg måtte laste den ned for at koden min skal fungere når jeg kjører den. **Psutil** er altså et bibliotek i Python som forteller om PC-informasjon.

Et bilde som inneholder tekst, skjermbilde, programvare, Multimedieprogramvare

KI-generert innhold kan være feil.

Når jeg overførte andre filen også til GitHub, det skjedde ikke noe på Pi-en, så jeg måtte slette filen og klone den igjen sånn at jeg får den nye filen inn.

Et bilde som inneholder tekst, skjermbilde, Font

KI-generert innhold kan være feil.

Her kloner jeg igjen, og sjekker at begge filer er der. Da prøver jeg å kjøre ‘**systeminfo.py’** filen, og jeg får systeminformasjon til Pi-en

Hvorfor brukte jeg HTTPS tilkloning istedenfor SSH

Enklere å sette opp, trenger ikke nøkkeloppsetting

Veldig lett å klone GitHub repository. Klikke på Code -> kopiere url