

# Uppsetning á Python vinnuumhverfi

## Python Virtual Environment

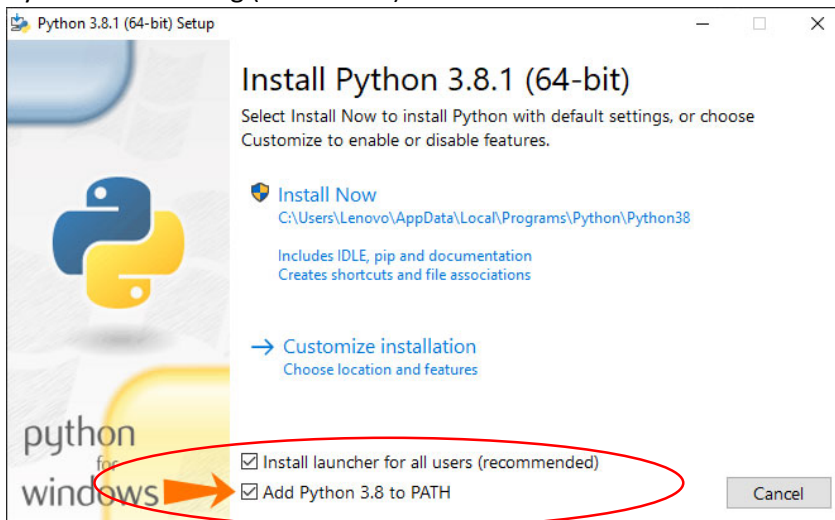
1. Python verður að keyra á kerfisskipanaforriti tölvunnar, *Command Prompt - Windows / Terminal – Mac/Linux*
2. Python sýndarvinnusvæði (*virtual environment - env*)
3. Flask og fleiri viðbætur settar í env möppuna með PIP

## Python Windows

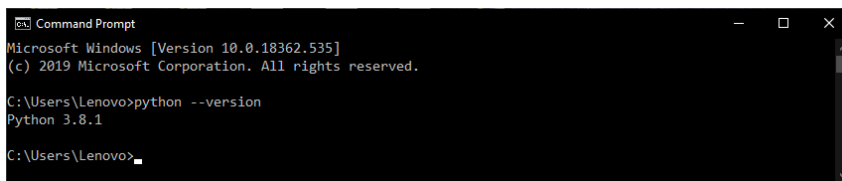
Flestir nemendur eru þegar búnir að stilla Python á stýrikerfið.

Það er nauðsynlegt til að búa til **sýndarvinnusvæði** (*virtual environment*)

- Python – innsetning (*installation*)



Mynd 1. Í fyrsta glugga innsetningar, hakið við [v] *Add Python 3.8 to PATH*



Mynd 2. Endurræsið tölvuna og ræsið kerfisskipanaforitið (CMD/Terminal).

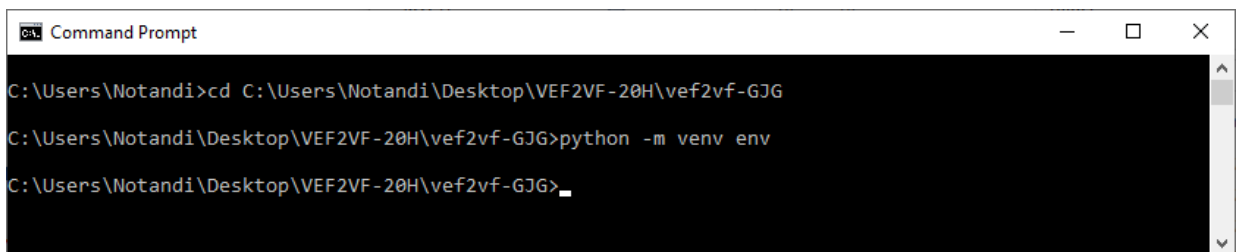
Skrifið „**python - -version**“ ef innsetning hefur tekist þá birtir CMD útgáfuna sem þú notar.

Ef CMD segist ekki þekkja python skipunina verður þú að taka Python forritið út af tölvunni (*uninstall*) og setja nýju útgáfuna inn aftur og muna að haka við „PATH“ eða finna út úr því hvar í stýrikerfinu hægt er að haka við PATH (*það verður ekki sýnt hér*). **PyCharm** forritið er með innbyggt kerfi sem hægt er að nota til að búa til sýndarvinnusvæði, það má nota það en *það er á ykkar ábyrgð*.

## Uppsetning vinnusvæðis (virtual environment -env)

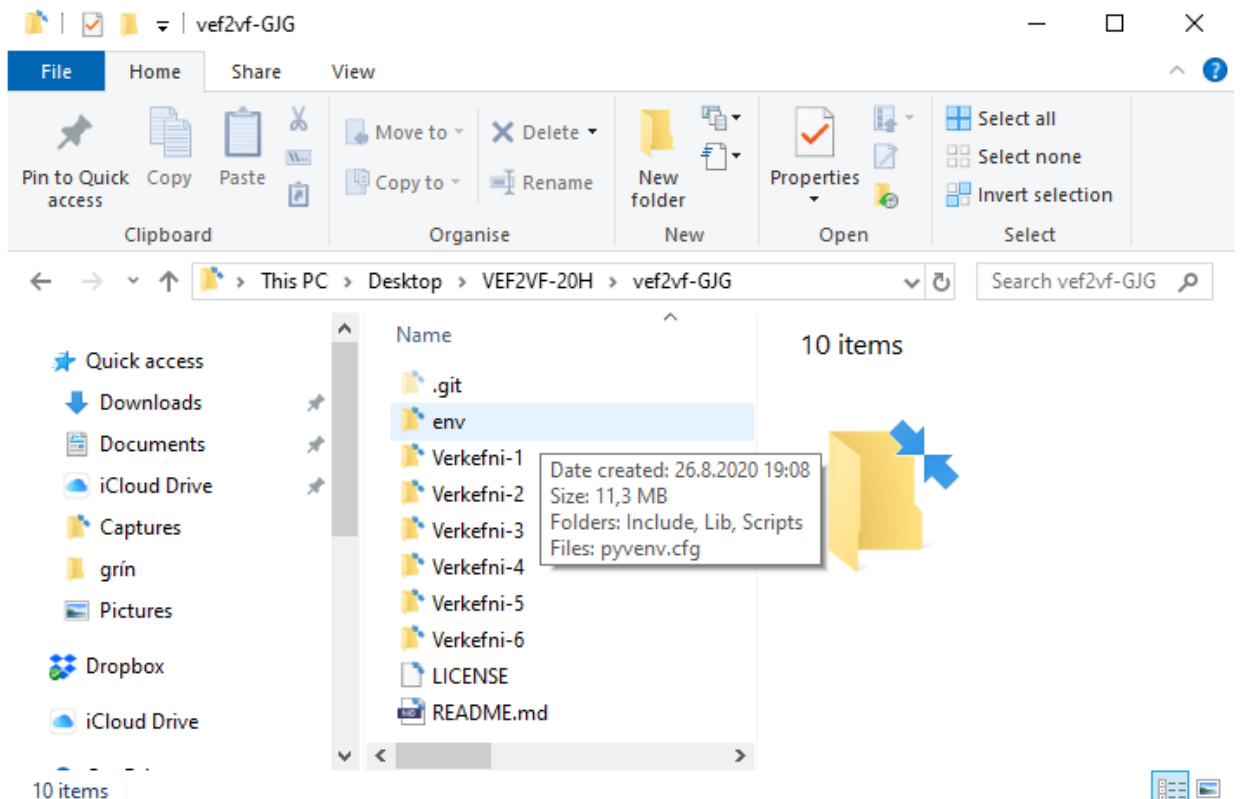
- **Git-Bash.** Sækið áfangageymsluna af Github `$ Git clone`
- Opnið *Command Prompt* (CMD) eða *Terminal*.
- Vísið CMD í áfangageymsluna ykkar `$ cd vef2vf-user/`
- Skrifið síðan

- **# macOS/Linux**  
`sudo apt-get install python3-venv`  
**# eða**  
`python3 -m venv env`
- **# Windows**  
`python -m venv env`

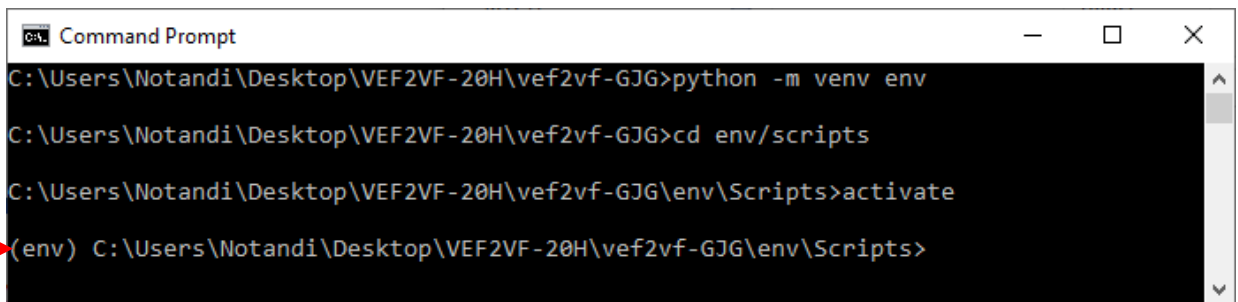


```
Command Prompt
C:\Users\Notandi>cd C:\Users\Notandi\Desktop\VEF2VF-20H\vef2vf-GJG
C:\Users\Notandi\Desktop\VEF2VF-20H\vef2vf-GJG>python -m venv env
C:\Users\Notandi\Desktop\VEF2VF-20H\vef2vf-GJG>
```

- **env** er heiti möppunnar sem heldur utanum Python forritaviðbæturnar
- Ef innsetning tókst þá er **env** komið í áfangageymsluna



- Nú þarf að virkja vinnusvæðið, farið í CMD og skrifið `cd env/scripts`
- Og skráið síðan `activate`

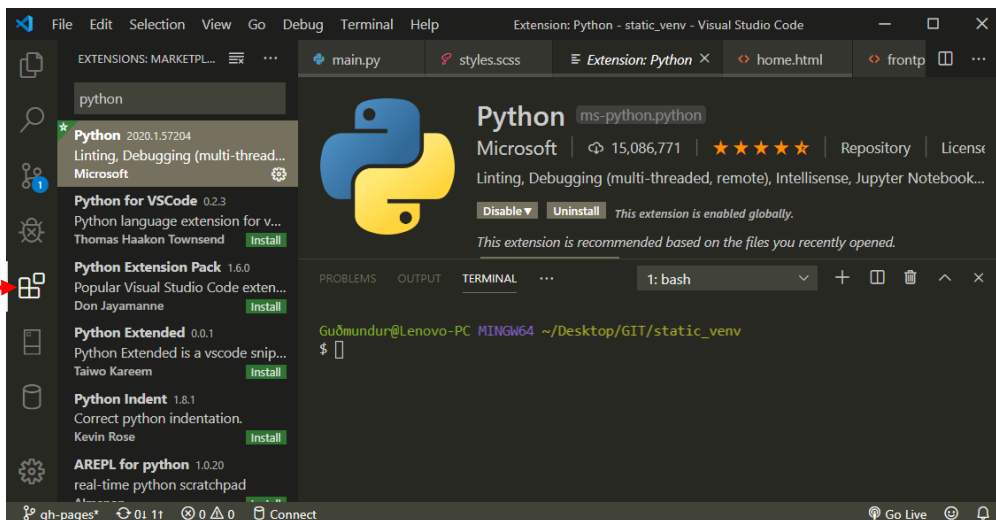


```
C:\Users\Notandi\Desktop\VEF2VF-20H\vef2vf-GJG>python -m venv env
C:\Users\Notandi\Desktop\VEF2VF-20H\vef2vf-GJG>cd env/scripts
C:\Users\Notandi\Desktop\VEF2VF-20H\vef2vf-GJG\env\Scripts>activate
(env) C:\Users\Notandi\Desktop\VEF2VF-20H\vef2vf-GJG\env\Scripts>
```

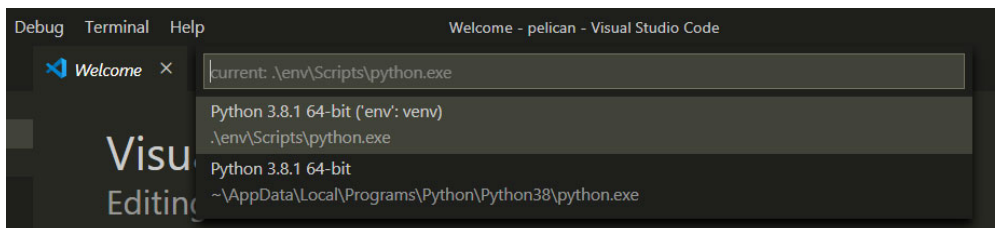
- (env) sýnir að áfangageymslan inniheldur Python sýndarvinnusvæði

## Visual Studio Code

- Opnið áfangageymsluna **Visual Studio Code**, (eða öðrum vefrita að eigin vali)
  - **file>open folder>** ~/áfangageymslan
- Til að *Python* virki í *Visual Studio Code* þarf að virkja Python stuðning í forritinu. Það er gert með því að ná í Python viðbót (*extension*) á VSC viðbótamarkaðnum.



- í valslá (efst) veljið **View > Command Palette**.  
Þar er valið **Python: Select Interpreter**  
og síðan Python 3.8.1('env':venv) útgáfuna



Þá er Python (sýndar)vinnusvæðið tilbúið til notkunar

## Python PyPi

Python Package Index (PyPi) er hugbúnaðargeymsla fyrir Python forritunarmálið. Þar eru hugbúnaðarpakkar þróaðir og samnýttir af Python samfélaginu. Sjá nánar á <https://pypi.org/>

### PyPi er innbyggð í Python 3.x

## FLASK

[Flask](#) er léttur [WSGI](#) vefforritarammi (*webframework*). Það er auðvelt er fyrir byrjendur að forrita öpp í Flask og síðan er hægt að stækkað þau og gera þau að flóknum forritum.

- náði í Flask með pip:

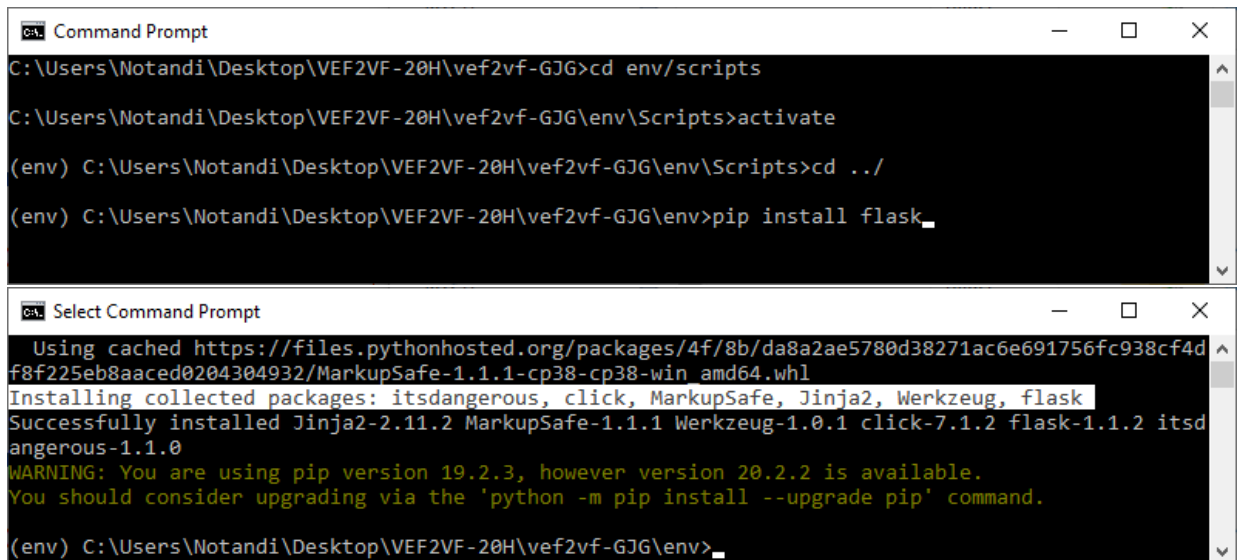
```
# macOS/Linux

pip3 install flask

# Windows

pip install flask
```

- 



The first screenshot shows a Command Prompt window with the following commands and output:

```
C:\Users\Notandi\Desktop\VEF2VF-20H\vef2vf-GJG>cd env/scripts
C:\Users\Notandi\Desktop\VEF2VF-20H\vef2vf-GJG\env\Scripts>activate
(env) C:\Users\Notandi\Desktop\VEF2VF-20H\vef2vf-GJG\env\Scripts>cd ../
(env) C:\Users\Notandi\Desktop\VEF2VF-20H\vef2vf-GJG\env>pip install flask_
```

The second screenshot shows the output of the pip install command:

```
Using cached https://files.pythonhosted.org/packages/4f/8b/da8a2ae5780d38271ac6e691756fc938cf4d
f8f225eb8aacd0204304932/MarkupSafe-1.1.1-cp38-cp38-win_amd64.whl
Installing collected packages: itsdangerous, click, MarkupSafe, Jinja2, Werkzeug, flask
Successfully installed Jinja2-2.11.2 MarkupSafe-1.1.1 Werkzeug-1.0.1 click-7.1.2 flask-1.1.2 itsd
angerous-1.1.0
WARNING: You are using pip version 19.2.3, however version 20.2.2 is available.
You should consider upgrading via the 'python -m pip install --upgrade pip' command.
(env) C:\Users\Notandi\Desktop\VEF2VF-20H\vef2vf-GJG\env>
```

- Nú er hægt að búa til mörg aðskild vefforrit í áfangamöppunni sem nota Flask ofl viðbætur - þessar viðbætur tilheyra eingöngu áfangageymslunni

## [.gitignore](#)

Nú er **env** mappan full af forritum sem gera ekkert gagn í Github áfangageymslunni (*remote*). Til að koma í veg fyrir að Git fari að grautast í **env** möppunni þá er hægt að nota Git kerfis skjalið **.gitignore**

- Búið til skjalið **.gitignore** í áfangageymslunni sem er staðbundin (*local*). Passið upp á að skjalið hafi *engra skráarendingu* td .txt eða .md
- Skráið í .gitignore skjalið
  - **env/** (Ef env mappan hefur annað heiti í þinni geymslu þá skráir þú það í staðinn)

Nú er vinnuumhverfið klárt og við getum byrjað á vefforritun í verkefni 1, í áfangageymslunni.

Farið eftir verklýsingunni í INNU ....



[HEROKU](#) vefmiðlari (*webserver*)

1. Skráðu þig sem notanda (*user*) á Heroku
2. Stofnaðu vefhýsingu ([Heroku Dyno](#))  
Það er hægt að vera með 5 *Dyno* hýsingar í einu án gjalds. Ókosturinn er að þjónustan er aðeins virk þegar kallað er á appið, sem sagt afar hægvirkt þjónusta
3. Tengja *Github geymslu*\* við hýsinguna
4. Birta (*deploy*) Python vefforrit á internetinu

**\* Það er ekki hægt að tengja áfangageymsluna við Heroku því Tækniskólinn er eigandi geymslunnar.**

1. Stofnið geymslu (*Repository*) á eigin Github reikningi
2. Afritið öll vinnugögnin úr áfangageymslunni í nýju geymsluna
3. Tengjið nýju geymsluna við Heroku  
Til að það sé hægt verðið þið að gefa Heroku *Admin* réttindi á Github reikningin