Software Engineering Setup

Bas Terwijn

February 2, 2026

1 Introductie

Dit document helpt u bij het opzetten van de werkomgeving voor dit vak op uw computer.

1.1 Typewerk beparen

Om in de terminal veel typewerk te besparen kunt u "Tab Completion" gebruiken, uw typt bijvoorbeeld dit en drukt op de <Tab> knop:

```
cd s<Tab>
```

de filename wordt dan automatisch aangevuld:

```
cd softeng
```

Soms moet u meerdere keren op <Tab> drukken om de verschillende opties te zien te krijgen, type dan de eerstvolgende letter en weer <Tab> voor verdere aanvulling. Druk in veel verschillende situaties op <Tab> want vaker dan niet geeft het aanvullingen of opties. Een andere manier om typewerk te besparen is om de cursortoetsen te gebruiken:

- cursor↑, ga naar het vorige commando
- cursor↓, ga terug het volgende commando
- cursor←, loop cursor naar links
- cursor→, loop cursor naar rechts
- en type of backspace om een commando aan te passen voor u op Enter drukt

Oefen met "Tab Completion" en cursor toetsen zodat het een gewoonte wordt, u niet onnodig tijd versplit, en u dus snel en vooral prettig kunt werken. Na enige oefening is dit sneller dan met een muis op icoontjes klikken.

2 Python Installatie

Om te zien of en welke versie van Python u al geïnstalleerd heeft gebruikt u in de terminal het shell commando:

```
python --version
```

In dit vak gebruiken we een moderne Python versie van `3.12` of hoger. Installeer deze indien nodig met:

2.1 voor Ubuntu

Run in de terminal:

src/python_ubuntu.sh

```
sudo add-apt-repository ppa:deadsnakes/ppa
sudo apt update
```

```
sudo apt install python3.12 python3.12-venv python3-pip
```

2.2 voor MacOS

Installeer package python-3.13.2.pkg.

3 Python Venv

We maken een Python virtual environment voor dit vak aan waarin we kiezen voor een bepaalde versie van Python en waarin we gemakkelijk Python packages kunnen installeren.

3.1 voor Ubuntu

Run in de terminal:

src/venv_ubuntu.sh
```
python3.12 -m venv ~/py3.12venv                        # create environment
source ~/py3.12venv/bin/activate                       # activate environment
cp ~/.bashrc ~/.bashrc$(date +"%Y%m%d_%H%M%S")         # backup .bashrc
echo "source ~/py3.12venv/bin/activate" >> ~/.bashrc # auto activate environment
```

Open de configuratie file `~/.bashrc` met:

```
gedit ~/.bashrc
```

en zorg dat er **onderaan maar één keer** de regel `source ~/py3.12venv/bin/activate` in staat om de virtual environment te activeren, verwijder eventueel dubbele regels. Deze file wordt automatisch uitgevoerd bij het openen van een nieuwe terminal, goed om te weten voor eventueel latere eigen configuraties.

3.2 voor MacOS

Run in de terminal:

src/venv_macos.sh
```
python3.13 -m venv ~/py3.13venv                        # create environment
source ~/py3.13venv/bin/activate                       # activate environment
cp ~/.zshrc ~/.zshrc$(date +"%Y%m%d_%H%M%S")           # backup .zshrc
echo "source ~/py3.13venv/bin/activate" >> ~/.zshrc    # auto activate environment
```

Open de configuratie file `~/.zshrc` met:

```
open -a TextEdit ~/.zshrc
```

en zorg dat er **onderaan maar één keer** de regel `source ~/py3.13venv/bin/activate` in staat om de virtual environment te activeren, verwijder eventueel dubbele regels. Deze file wordt automatisch uitgevoerd bij het openen van een nieuwe terminal, goed om te weten voor eventueel latere eigen configuraties.

3.3 Test de Python installatie

Upgrade nu eerst pip met:

src/upgrade_pip.sh
```
python -m pip install --upgrade pip
```

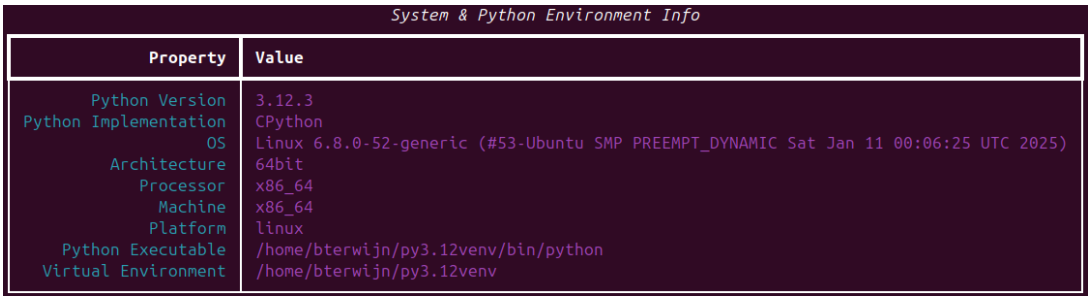
en installeer dan Python package `rich` in de terminal met:

src/pip_install_rich.sh
```
pip install rich
```

en ga vervolgens naar de `softeng/week1/1_setup/src` directory en run daar:

```
python print_python_venv.py
```

Het resultaat zou een tabel moeten zijn zoals in figuur Figure 1 met de juiste Python versie en op de laatste twee regels uw python environment “py3.12venv” of “py3.13venv”. Als dit werkt maak dan een screenshot van deze tabel en sla deze op als `softeng/week1/1_setup/assignments/python_venv.png` (telt mee voor uw eindcijfer). Als dit nog niet werkt loop dan bovenstaande stappen nog eens door en vraag uw TA om hulp als dit aanhoudt. Blijft u tegen laptopproblemen aanlopen, neem dan contact op met Laptop Support op Science Park.



System & Python Environment Info

Property	Value
Python Version	3.12.3
Python Implementation	CPython
OS	Linux 6.8.0-52-generic (#53-Ubuntu SMP PREEMPT_DYNAMIC Sat Jan 11 00:06:25 UTC 2025)
Architecture	64bit
Processor	x86_64
Machine	x86_64
Platform	linux
Python Executable	/home/bterwijn/py3.12venv/bin/python
Virtual Environment	/home/bterwijn/py3.12venv

Figure 1: Colored table with system and environment information.

4 Visual Studio Code

Om Python code te schrijven gebruiken we binnen dit vak Visual Studio Code als Integrated Development Environment (IDE), een editor met behulpzame tools. Ga naar Download Visual Studio Code en:

4.1 voor Windows met WSL

Download de Windows executable en voer deze uit, vink bij de installatie de **“Add to PATH (requires shell restart)”** optie aan.

4.2 voor Ubuntu

Download de “.deb” file naar uw Downloads directory en run in uw terminal:

src/vscode_ubuntu.sh

```
cd ~/Downloads
latest_code=$(ls -t code_*.deb | head -n 1) # get latest version
echo "installing file: $latest_code"
sudo dpkg -i $latest_code                   # install using Debian package manager
```

4.3 voor MacOS

Download de MacOS zip file naar uw Downloads directory en run in uw terminal:

```
unzip ~/Downloads/VSCode-darwin-universal.zip -d ~/Downloads/
mv "Visual Studio Code.app" /Applications/
```

Start daarna "Visual Studio Code.app" vanuit de Applications directory (of zoek het na `Cmd Space` in Spotlight Search) en druk op `Cmd + Shift + P` om het Command Palette te openen. Type daar:

```
Shell Command: Install 'code' command in PATH
```

zodat u daarna met `code .` Visual Studio Code in de terminal kunt starten.

5 Visual Studio Code, Python setup

Open een nieuwe terminal, ga naar de `softeng/week1/1_setup/src` directory en run daar:

```
code .
```

om Visual Studio Code (vscode) te starten.

5.1 Python plugin

Installeer vervolgende de Python Plugin zoals aangegeven in figuur Figure 2:

1. Klik de Extensions Icon in de Activity Bar.
2. Type “python” in de search box.
3. Selecteer de “Python” plugin van “Microsoft”.
4. Installeer deze plugin.

Selecteer binnen dit vak geen andere plugins die voor verwarring zouden kunnen gaan zorgen.

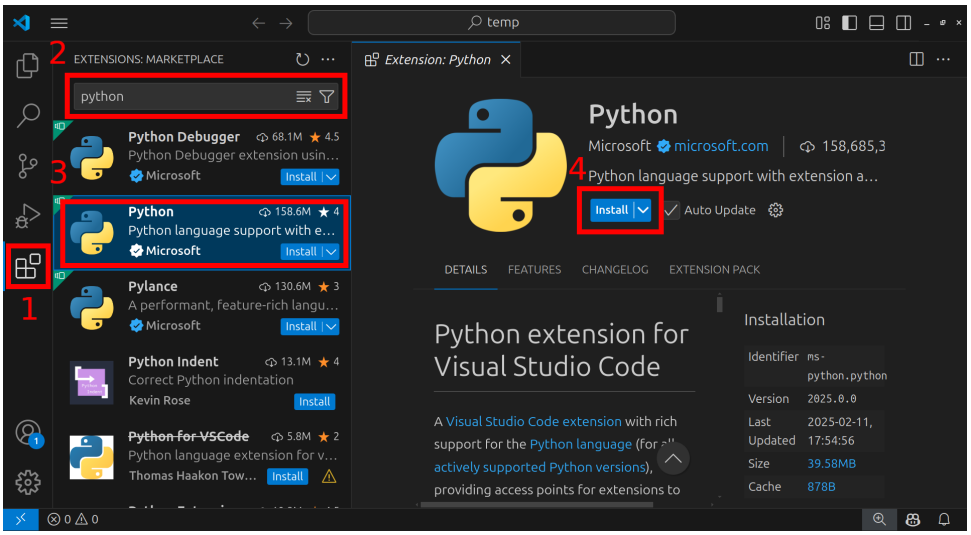


Figure 2: Installeer de Python Plugin.

5.2 WSL Plugin voor Windows

Alleen als u WSL in Windows gebruikt, installeer dan ook de “WSL” plugin van Microsoft, type vervolgens Shift-Ctrl-P voor de “Command Palette”, en type/selecteer daar “Connect to WSL” zodat Visual Studio Code de WSL omgeving gebruikt.

5.3 Python Versie

Open vervolgens de file “compute_distance.py” en configureer vscode zodat het de Python interpreter gebruikt binnen uw virtual environment (venv) zoals aangegeven in figuur Figure 3.

1. Klik de Explorer Icon in de Activity Bar.
2. Open file “compute_distance.py”
3. Open de interpreter selector.
4. Selecteer de Python versie binnen uw virtual environment (Venv).

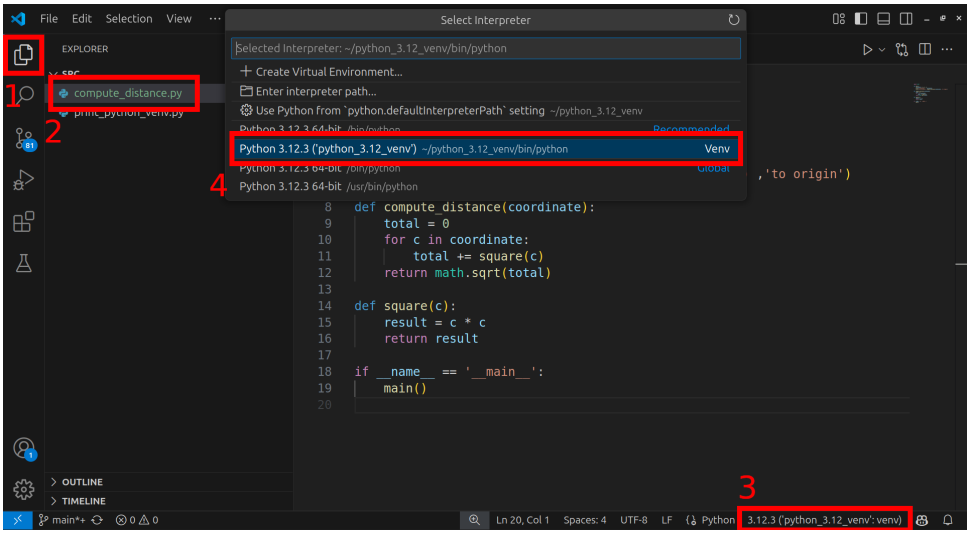


Figure 3: Selecteer de Python interpreter binnen uw virtual environment.

5.4 Code Schrijven

Visual Studio Code kan u op veel manieren helpen bij het schrijven van code. De twee belangrijkste manieren voor nu vind u in paragraaf:

- Autocomplete and IntelliSense
- Navigation

6 Visual Studio Code, Debugger

Deze video geeft een kort introductie over het gebruik van Visual Studio Code en met name de debugger tool:

YouTube: Visual Studio Code Debugger

Het `compute_distance.py` programma in deze video berekent de afstand van het punt (3,4,2) tot de oorsprong (0,0,0) zoals weergegeven in figuur Figure 4 met:

$$\text{distance} = \sqrt{x^2 + y^2 + z^2}$$

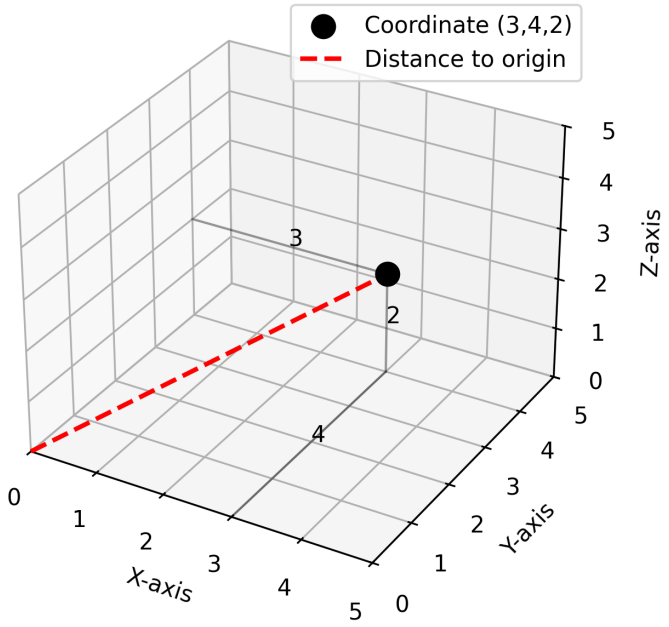


Figure 4: De afstand van de oorsprong tot het punt (3,4,2).

Oefen zelf even met de debugger tool in Visual Studio Code met het `compute_distance.py` programma door de stappen in de video te volgen. Wil je meer weten over de debugger tool, bekijk dan ook deze video:

YouTube: Everything You Need to Know about Debugging in VSCode, ArjanCodes

7 Installatie van `softeng` Package

Update en installeer de "softeng" package in directory `softeng` met commando's:

```
git pull                 # update softeng, get latest commits
pip install --upgrade .  # install the softeng package
```

De output hiervan zou bij succes moeten eindigen met de regel "Successfully installed softeng-0.0.1". Voer deze twee commando's opnieuw uit als er een nieuwe versie van `softeng` beschikbaar komt waarin eventuele fouten zijn verholpen.

8 Opdracht `addition.py`

In directory `softeng/week1/1_setup/assignments` vindt u bestand `addition.py` met daarin de `add(a, b)` functie waar een duidelijke bug in zit:

assignments/addition.py

```
def add(a, b):
    """ Returns the result of adding 'a' and 'b'."""
    return a   # bug!!! should ofcourse be:   return a + b
```

Om automatische tests uit te voeren op dit bestand kunt u in deze directory het commando:

```
pytest test_addition.py
```

of gebruik `-v` of `-vv` voor meer verbosity (informatie):

```
pytest -vv test_addition.py
```

uitvoeren, wat door de bug resulteert in enkele `AssertionErrors` met waar mogelijk een aanwijzing voor het oplossen hiervan:

```
FAILED test_addition.py::test_add_positive - AssertionError: add(10, 7) should return 17
FAILED test_addition.py::test_add_negative - AssertionError: add(10, -7) should return 3
```

Debug nu de `add(a, b)` functie in `softeng/week1/1_setup/assignments` en run de tests opnieuw totdat het programma correct werkt en de tests slagen. Om alle tests in een directory uit te voeren gebruikt u het commando:

```
pytest
```

Dit voert dan ook de `test_file_python_venv_png.py` test uit die controleert of u de eerdere screenshot `python_venv.png` correct heeft opgeslagen.

9 Beoordeling Opdrachten

De beoordeling van op tijd ingeleverde opdrachten is afhankelijk van:

- resultaat van bijgeleverde tests
- resultaat van niet-vrijgegeven tests die sterk lijken op de bijgeleverde tests, maar net iets anders zijn
- eventuele beoordeling door nakijker van structuur en leesbaarheid van code
- het op verzoek kunnen uitleggen van uw werk

9.1 Uitleggen van uw werk

U mag alleen eigen werk inleveren. Daarom kan u worden uitgnodigd om uw werk uit te leggen. Als u dat onvoldoende kunt dan zien we dit als geen eigen werk, en moeten wij dit melden bij de examencommisie. Dit kan dan worden gezien als Fraude en Plagiaat wat ernstige gevolgen kan hebben voor het voltooien van uw studie. Lever dus alleen eigen werk in en copy-paste geen werk afkomstig van GenAI (UvA AI Chat, ChatGPT, Github Copilot, Gemini, DeepSeek, ...), andere studenten, of programmeerfora en tutorial-websites (Stack Overflow, GeeksForGeeks, Real Python, ...). Werk kopieëren zal daarnaast ook onvoldoende oefening blijken als voorbereiding op het tentamen. U mag de genoende bronnen wel gebruiken, maar alleen om zelf van te leren, op zo een manier dat u eigen werk inlevert waarin u zelf keuzes heeft gemaakt en wat u dus kunt uitleggen.

10 Opdrachten Inleveren

Opdrachten moeten per week op Canvas worden ingeleverd als zip-file. Deze zip-file kunt u maken door in directory `softeng` het volgende script uit te voeren:

- Ubuntu: `./zip_linux.sh week1`
- MacOS: `./zip_mac.sh week1`

En dan natuurlijk met week2, week3, ... voor de latere weken. Het is ook een goed idee om uw werk regelmatig te backup-en door deze deze zip-file te maken en te kopieëren naar uw OneDrive of andere cloud storage, voor het geval uw laptop het spontaan begeeft of u deze kwijtraakt.