

Software Engineering Setup

Bas Terwijn

March 13, 2025

1 Introductie

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

1.1 Doelen

- Bekendheid met basis shell commando's
- Python installatie
- Visual Studio Code Setup
- 'softeng' Installatie
- Pytest en Beoordeling
- Inleveren en Backups

2 Basis Shell Commando's

In dit vak werken we vaak in de terminal met shell commando's. Een terminal start u afhankelijk van uw besturingssysteem:

- terminal in [Ubuntu](#)
- terminal in [MacOS](#)
- PowerShell in [Windows 11](#)
- of websearch hoe u een terminal (of PowerShell voor Windows) opent op uw systeem

In the terminal kunt u deze shell commado's gebruiken:

- **pwd** - print working directory
- **ls** - list files in working directory
- **cd** - change directory
- **cp** - copy a file
- **mv** - move/rename a file
- **rm** - remove a file

Een voorbeeld hiervan ziet u in figuur ??:

```

$ pwd      # where am I?
/home/bterwijn
$ ls      # what files are here?
Desktop  Documents  Downloads temp
$ cd temp  # change to 'temp' directory
$ pwd      # where am I?
/home/bterwijn/temp
$ ls      # what files are here?
music other_things.txt test.txt
$ cp test.txt test2.txt # copy a file
$ ls      # what files are here?
music other_things.txt test2.txt test.txt
$ mv test2.txt test_new.txt # move/rename a file
$ ls      # what files are here?
music other_things.txt test_new.txt test.txt
$ rm test_new.txt # remove a file
$ ls      # what files are here?
music other_things.txt test.txt
$ cd ..    # change up/back one directory
$ pwd      # where am I?
/home/bterwijn
$

```

Figure 1: Voorbeeld van gebruik van shell commando's in de terminal.

2.1 Voorbeeldgebruik van de shell commando's

- [Ubuntu](#)
- [MacOS](#)
- [Windows](#), de video gebruikt “Command Prompt”, maar gebruik zelf dus de “PowerShell”

2.2 Typewerk besparen

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

```
cd t<Tab>
```

de filename wordt dan automatisch aangevuld:

```
cd temp
```

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 situaties op <Tab> want vaker dan niet geeft het aanvullingen/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 naar links
- cursor→, loop naar rechts
- en backspace of type om het commando aan te passen

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.

3 Python Installatie

Installeer een moderne Python versie.

3.1 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
python3.12 -m pip install --upgrade pip
```

3.2 voor MacOS

Installeer package [python-3.13.2.pkg](#).

3.3 voor Windows

Download en run [python-3.13.2-amd64.exe](#) en selecteer in het installatie-proces “Add python to PATH”.

4 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.

4.1 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.

4.2 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.

4.3 voor Windows

Run in de PowerShell:

```
src/venv_windows.bat
python3.13 -m venv %USERPROFILE%\py3.13venv      # create environment
& %USERPROFILE%\py3.13venv\Scripts\Activate.ps1  # activate environment
$backupFile = "$PROFILE.$(Get-Date -Format 'yyyyMMdd_HH:mm:ss').bak"
cp $PROFILE $backupFile                          # backup profile
echo "& $env:USERPROFILE\py3.13venv\Scripts\Activate.ps1" >> $PROFILE # automate
```

Open de configuratie file `$PROFILE` met:

```
notepad $PROFILE
```

en zorg dat er **onderaan maar één keer** de regel `& $env:USERPROFILE\py3.13venv\Scripts\Activate.ps1` in staat om de virtual environment te activeren, verwijder eventueel dubbele regels. Deze file wordt automatisch uitgevoerd bij het openen van een nieuwe PowerShell, goed om te weten voor eventueel latere eigen configuraties.

4.4 Test de Python installatie

Installeer Python package `rich` in de terminal met:

```
pip install rich
```

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

```
python print_python_venv.py
```

Het resultaat zou een tabel moeten zijn zoals in figuur ?? 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 `softwareengineering/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 2: Colored table with system and environment information.

5 Visual Studio Code

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

5.1 voor Ubuntu

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

```
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
```

5.2 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 Visual Studio Code in de terminal kan starten met `code .`

5.3 voor Windows

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

6 Visual Studio Code, Python setup

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

```
code .
```

om Visual Studio Code (vscode) te starten.

6.1 Python plugin

Installeer vervolgens de Python Plugin zoals aangegeven in figuur ??:

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.

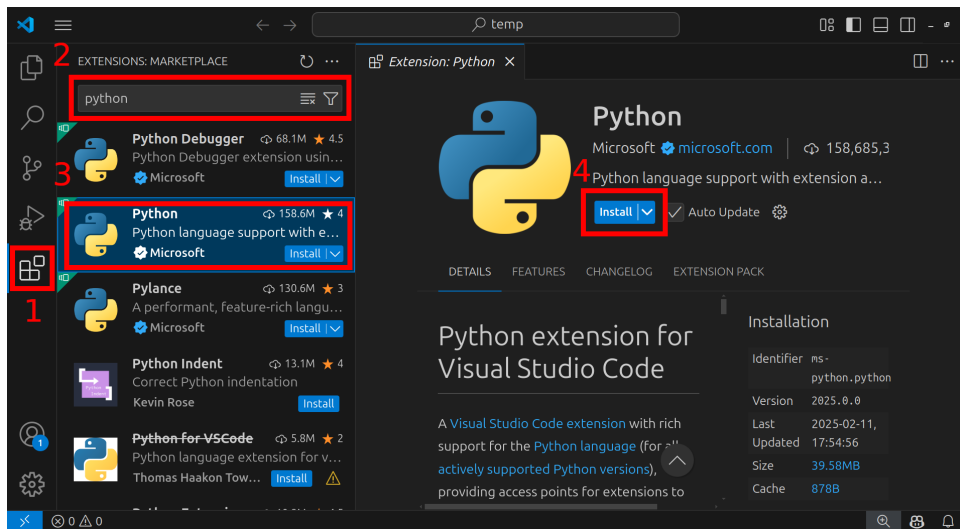


Figure 3: Installeer de Python Plugin.

6.2 python version

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 ??.

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).

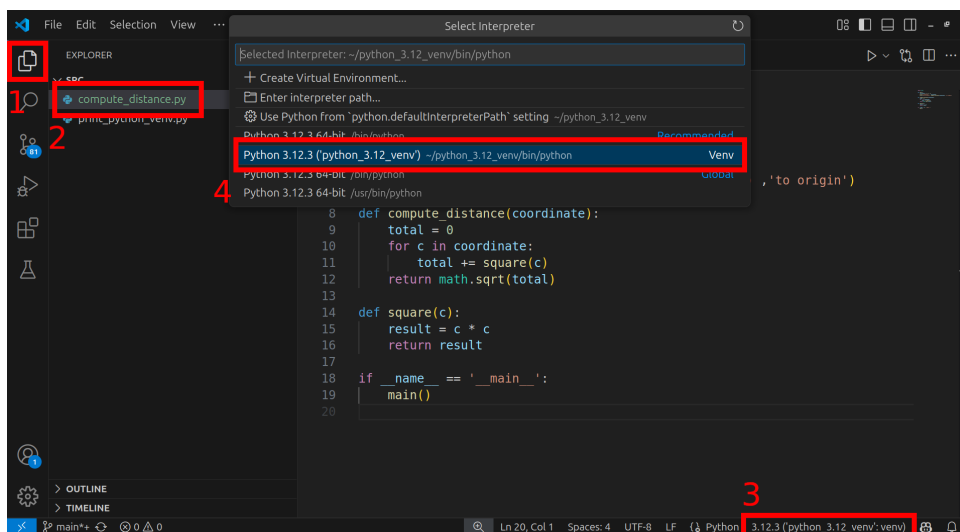


Figure 4: Selecteer de Python interpreter binnen uw virtual environment.

6.3 code schrijven

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

- [Autocomplete and IntelliSense](#)
- [Navigation](#)

7 Visual Studio Code, Debugger

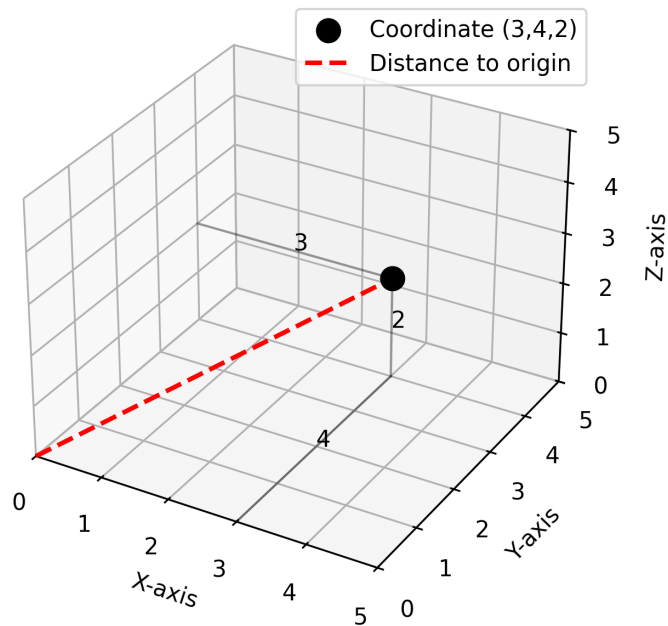


Figure 5: De afstand van de oorsprong tot het punt (3,4,2).

$$\sqrt{x^2 + y^2 + z^2}$$

TODO: make/find video

- run programma normaal, klopt het?
- Vscode heeft een debugger tool
- we willen zien hoe het wordt berekend zonder debug print statements
- set break at main
- start debugger, run debug, f5,
- click 'Step Into', de main functie wordt aangeroepen
- click 'Step Into', de variable coordinate krijgt een waarde, zie links
- click 'Step Into', coordinate wordt geprint in terminal
- click 'Step Into', de compute_distance wordt aangeroepen
- click 'Step Into', variable 'total' krijgt een waarde
- click 'Step Into', variable 'c' krijgt een waarde van eerst element in 'coordinate'
- click 'Step into', de main functie wordt aangeroepen
- click 'Step Over',
- zet een breakpoint op regel6
- click 'Continue', uitvoeren tot eerst volgende breakpoint
- 'Step out' continue totdat met return uit de huidige functie wordt geprongen
- 'Retart' herstart de debugger, stop weer bij eerste breakpoint, bij code verandering pas actie bij restart

- 'Stop' stop de debugger

Play button

Ctrl-click filename:line to jump to source

[meer over debugger tool](#)

8 SoftEng Installatie

Installeer de “softeng” module door in directory `softwareengineering` het commando:

```
pip install --upgrade .
```

uit te voeren. De output hiervan zou bij succes moeten eindigen met de regel “Successfully installed softeng-0.0.1”.

9 Opdracht Addition Testen

In directory `softwareengineering/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 -v 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 `softwareengineering/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 -v
```

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

10 Beoordeling Opdrachten

De beoordeling van opdrachten is afhankelijk van:

- controle door nakijkers
- automatische tests

10.1 controle door nakijkers

Bij de controle door nakijkers wordt vooral gekeken of er aanwijzingen zijn dat het ingeleverde geen eigen werk is. In dat geval kan u worden uitgenodigd om uw werk uit te leggen. Als u dat onvoldoende kunt, dan moeten wij dit melden bij de examencommissie en kan dit worden gezien als [Fraude en Plagiat](#) 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 (ChatGPT, Github Copilot, Gemini, DeepSeek, ...) of programmeerfora en tutorial-websites (Stack Overflow, GeeksForGeeks, Real Python, ...). Werk kopiëren zal daarnaast ook onvoldoende oefening blijken als voorbereiding op het tentamen.

10.2 automatische tests

Automatische tests staan in Python bestanden die beginnen met `test_`. Zoals besproken kunt u alle tests in een directory zoals `softwareengineering/week1/1_setup/assignments` uitvoeren met commando:

```
pytest -v
```

Het voorlopige cijfer van opdrachten is naar ratio van de tests die slagen en wordt onderaan geprint bij uitvoeren van commando:

```
python compute_grade.py
```

Als u deze twee commando's bijvoorbeeld in directory `softwareengineering/week1` uitvoert, dan wordt elke onderliggende test van week1 uitgevoerd en kunt u uw voorlopige cijfer van week1 zien. Hetzelfde geldt voor latere weken. Als u deze commando's in directory `softwareengineering` uitvoert, dan worden alle tests van dit vak uitgevoerd en kunt u uw voorlopige cijfer van alle opdrachten zien.

11 Opdrachten Inleveren

Opdrachten moeten op Canvas worden ingeleverd als “softeng.zip” file. Deze zip-file kan gemaakt door in directory `softwareengineering` het volgende script uit te voeren:

- Ubuntu: `./zip_linux.sh`
- MacOS: `./zip_mac.sh`
- Windows: `.\zip_windows.bat`

Het is ook een goed idee om uw werk regelmatig te backup-en door deze zip-file te kopiëren naar bijvoorbeeld uw [OneDrive](#), voor het geval uw laptop het spontaan begeeft of u deze kwijtraakt.