

12-pre – Install Git and VS code

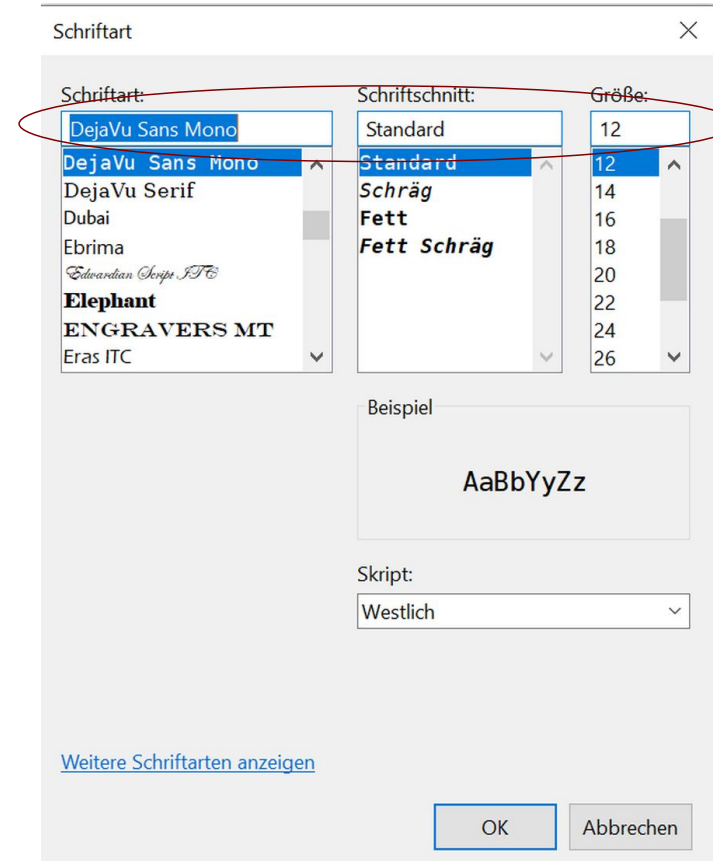
Bálint Aradi

Scientific Programming in Python (2026)

<https://atticlectures.net/scipro/python-2026/>

Prerequisites - simple text editor (Windows)

- We will use **Notepad** to enter git commit messages
- Configure Notepad to use a **mono-spaced** font (e.g. DejaVu Sans Mono)



Prerequisites - Git (Windows)

- Download and install [Git for Windows](#)
- Take default options apart of following ones:

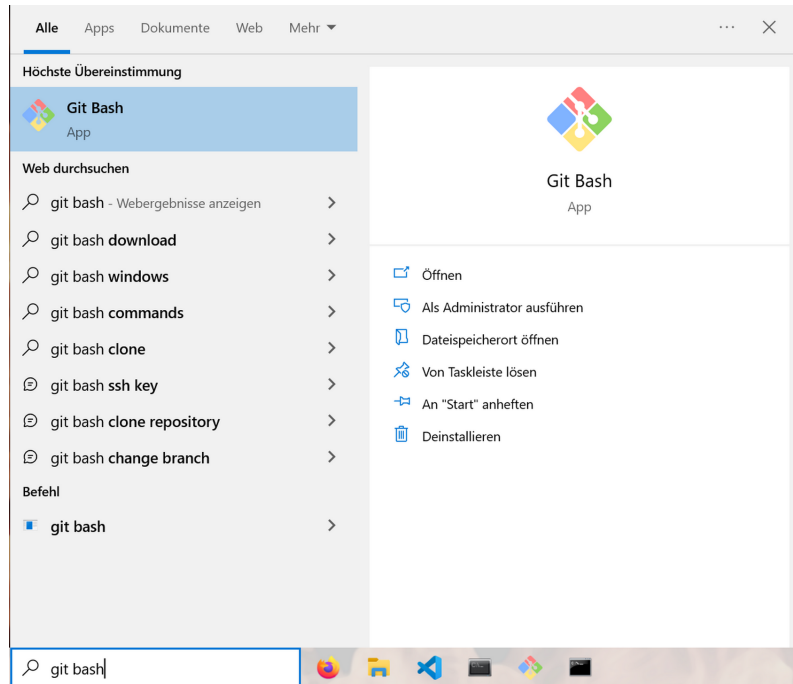
Git 2.40.1 Setup
Choosing the default editor used by Git
Which editor would you like Git to use?
Use Notepad as Git's default editor
(NEW!) Notepad is a simple GUI editor that comes with Windows.

Git 2.40.1 Setup
Adjusting the name of the initial branch in new repositories
What would you like Git to name the initial branch after "git init"?
 Let Git decide
Let Git use its default branch name (currently: "master") for the initial branch in newly created repositories. The Git project intends to change this default to a more inclusive name in the near future.
 Override the default branch name for new repositories
NEW! Many teams already renamed their default branches; common choices are "main", "trunk" and "development". Specify the name "git init" should use for the initial branch:
main
This setting does not affect existing repositories.

Git 2.45.0 Setup
Choose the default behavior of `git pull`
What should `git pull` do by default?
 Fast-forward or merge
Fast-forward the current branch to the fetched branch when possible, otherwise create a merge commit.
 Rebase
Rebase the current branch onto the fetched branch. If there are no local commits to rebase, this is equivalent to a fast-forward.
 Only ever fast-forward
Fast-forward to the fetched branch. Fail if that is not possible. This is the standard behavior of `git pull`.

Prerequisites - Git (Windows)

- We will use the Git Bash application (part of Git for Windows) to communicate with Git



- You can even configure Git Bash to automatically initialize Conda when started by creating a `~/.bashrc` file with following two lines:

```
source ~/miniconda3/etc/profile.d/conda.sh
conda activate scipro
```

~/.bashrc

This must match the path where Miniconda had been installed into

Prerequisites - diff-viewer (Windows)

- Download and install a suitable diff-viewer for Windows ([Meld](#) or [KDiff3](#))
- We might need the location, where the actual executable can be found

The screenshot shows a Windows File Explorer window with a search for 'KDiff3'. The context menu is open over the 'KDiff3 App' file. The 'Dateispeicherort öffnen' option is selected. A red arrow points from this option to a separate box containing the file path: `C:\Users\aradi\AppData\Local\Programs\KDiff3`. Another red arrow points from the 'Eigenschaften' option in the context menu to the 'Ziel:' field of the path box.

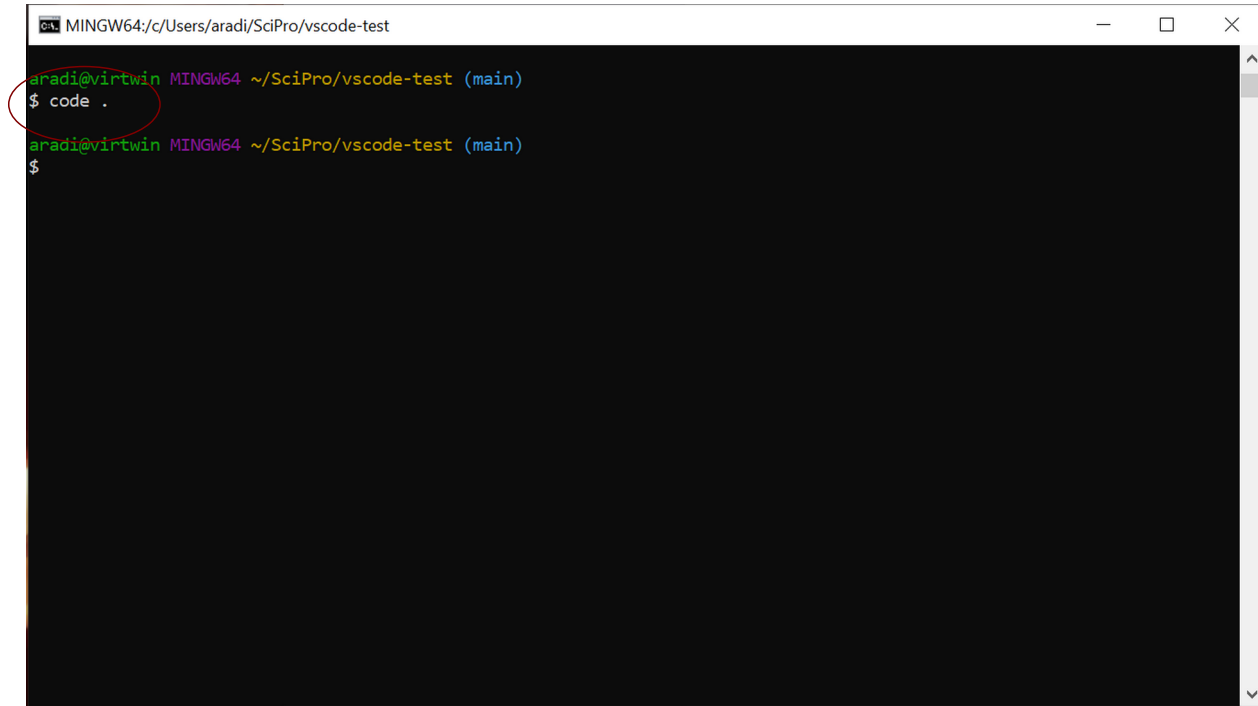
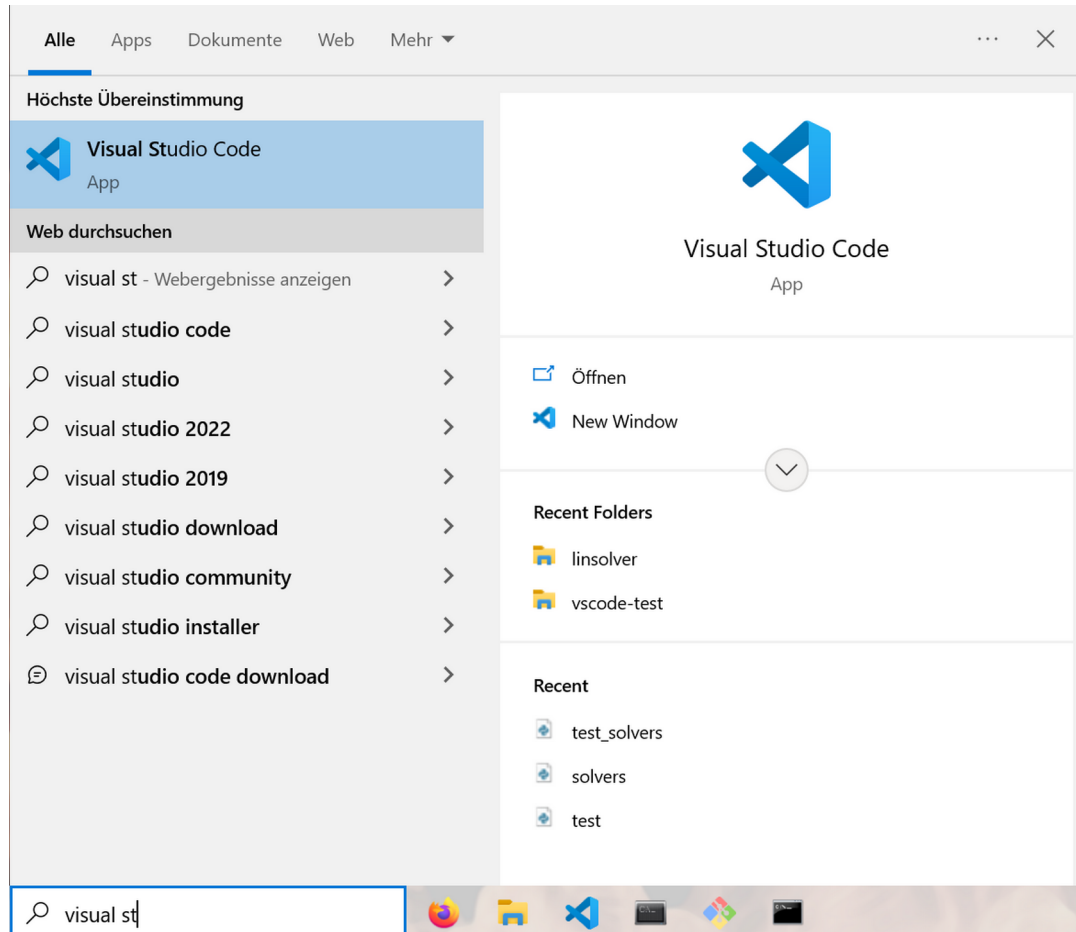
Ziel: `C:\Users\aradi\AppData\Local\Programs\KDiff3`

Prerequisites - Git & Git related tools (Linux)

- Install **git** via the package manager of your Linux distribution
- Install also a git-viewer package: **qgit** or **gitk**
- Install a diff-viewer tool: **kdifff3 (kdifff3-qt)** or **meld**
- Make sure, that a simple text editor (**nano, featherpad, leafpad, kate gedit, vim, emacs, etc.**) is installed

Prerequisites - Visual Studio Code

- Download and install [Visual Studio Code](#) for your operating system
- You can start Visual Studio Code via menu or by entering “code” in the Git Bash shell



- See [Getting started with Visual Studio Code](#) for a quick introduction