

Course Prerequisites

Best Practices in  python™ Programming, Scientific Software Center (SSC), Heidelberg University



Prerequisite slides

Last updated: 2026-09-07

Outline

1 Overview	2
2 Install The Tools	5
3 First-Time GitHub Setup	12
4 Classroom50	15
5 Check Your Setup	19
6 Optional Environment Check	23
7 Editor Or IDE	27

1 Overview

What to set up before the course

Required:

- GitHub account
- Python 3.11 or newer
- Git
- GitHub CLI (`gh`)
- Classroom50 student extension
- terminal or shell access

Recommended: a Python-aware editor or IDE.

Why this matters

The exercises use GitHub and Classroom50.

A working command-line setup lets you:

- accept assignment repositories
- clone starter code
- run tools locally
- commit and push changes
- receive classroom feedback

2 Install The Tools

Python 3.11+

Install Python 3.11 or newer.

Official downloads:

python.org/downloads

Verification:

```
python3 --version  
python3 -m pip --version
```

On Windows, use `py` if `python3` is not available.

Install Git.

Official downloads:

git-scm.com/downloads

Verification:

```
git --version
```


GitHub CLI

Install GitHub CLI (`gh`).

Official installation instructions:

[GitHub CLI installation](#)

Verification:

```
gh --version
```

Linux examples

Suggested, not required: Conda can be convenient for managing Python and environments. A compact installer is [Miniconda](#).

Alternatively, use your distribution's package manager.

Debian or Ubuntu:

```
sudo apt update  
sudo apt install python3 python3-venv python3-pip git
```

Install `gh` from the official GitHub CLI instructions.

macOS examples

Official installers:

- Python from python.org
- Git from git-scm.com
- GitHub CLI from the official `gh` instructions

Or with Homebrew:

```
brew install python git gh
```

Windows examples

Suggested, not required: Conda can be convenient for managing Python and environments on Windows. A compact installer is [Miniconda](#).

Conda does not replace Git or GitHub CLI.

Official installers:

- Python from python.org; enable “add Python to PATH”
- Git for Windows
- GitHub CLI installer

Or with PowerShell and `winget`:

```
winget install --id Python.Python.3.14.7 -e  
winget install --id Git.Git -e  
winget install --id GitHub.cli -e
```

3 First-Time GitHub Setup

Create and verify your account

If you do not already use GitHub:

1. Create an account at github.com.
2. Verify the email address for the account.
3. Record your exact GitHub username.
4. Send your username or profile URL to the course instructor (e.g., via email to thomas.isensee@iwr.uni-heidelberg.de).

Your display name and GitHub username are not necessarily the same.

Already have a working GitHub account? Keep using it and skip this slide.

Configure your Git identity

Git records an author name and email address in every commit.

Check your global configuration:

```
git config --global --get user.name  
git config --global --get user.email
```

If either command prints nothing, configure it:

```
git config --global user.name "Your Name"  
git config --global user.email "your-email@example.org"
```

Use an email associated with your GitHub account, or your GitHub-provided `noreply` address. These settings identify commits; they are not login credentials.

4 Classroom50

Install Classroom50 student CLI

The course assignments use Classroom50 through a GitHub CLI extension.

Install:

```
gh extension install foundation50/gh-student
```

If the extension is already installed, update it with:

```
gh extension upgrade gh-student
```

Verify:

```
gh student --help
```

Log in for assignments

```
gh student login
```

This starts the GitHub login flow with the permissions needed to accept and submit Classroom50 assignments. Browser-based login is usually easiest. Your Git transport can remain either HTTPS or SSH.

Choose HTTPS or SSH

Git can connect to GitHub using either HTTPS or SSH.

- **First-time users:** HTTPS is usually the easiest option. After running `gh student login`, configure Git to use the stored GitHub CLI credential:

```
gh auth setup-git
```

- **Existing GitHub users:** keep your working HTTPS or SSH setup. There is no need to change protocols or replace working SSH keys.

You do not need to create a personal access token for this course.

5 Check Your Setup

Verify on Linux or macOS

```
python3 --version  
python3 -m pip --version  
git --version  
gh --version  
gh student --help  
gh auth status  
gh api user --jq .login
```

If `python` works but `python3` does not, use the command that points to Python 3.11 or newer.

Verify on Windows PowerShell

```
py --version  
py -m pip --version  
git --version  
gh --version  
gh student --help  
gh auth status  
gh api user --jq .login
```

With Conda, activate the environment first and use `python` instead of `py`.

Expected result

- Python reports version 3.11 or newer.
- `pip`, `git`, and `gh` print version information.
- `gh student --help` prints Classroom50 command help.
- `gh auth status` shows that you are logged in to GitHub.
- `gh api user --jq .login` prints the GitHub username sent to the instructor.

6 Optional Environment Check

venv check: Linux or macOS

```
python3 -m venv .venv
source .venv/bin/activate
python -m pip install --upgrade pip
python -m pip install ruff pytest mypy
python -m pip list
deactivate
```

This is optional; no shared course environment is required before arrival.

venv check: Windows

```
py -m venv .venv
.venv\Scripts\Activate.ps1
python -m pip install --upgrade pip
python -m pip install ruff pytest mypy
python -m pip list
deactivate
```

If activation is blocked, run `.venv\Scripts\python` directly.

Optional Conda environment

Conda can be especially convenient on Linux and Windows, but it is not required. To check an equivalent environment:

```
conda create -n pbp-course python=3.14.7
conda activate pbp-course
python -m pip install ruff pytest mypy
```

7 Editor Or IDE

Editor recommendation

An IDE is not required, but it makes the course smoother.

Good options include:

- VS Code 
- PyCharm 
- Spyder 
- any editor where you can edit Python + a terminal