Project: Introduction to Python (1)

Project: Groups, forks and setup

- Groups formed in the issue feeds

- Forks were created.

- Further questions related to the GitHub setup?

- Setup completed?

www.uni-bamberg.de/digital-work/

Learning objectives:

- Familiarize with Python syntax

- Learn good debugging and development practices

- Understand how to extend a Python package (CoLRev)

Groups

- Form groups of three to four, discuss your solutions, and solve problems together.

www.uni-bamberg.de/digital-work/

Start the Codespace

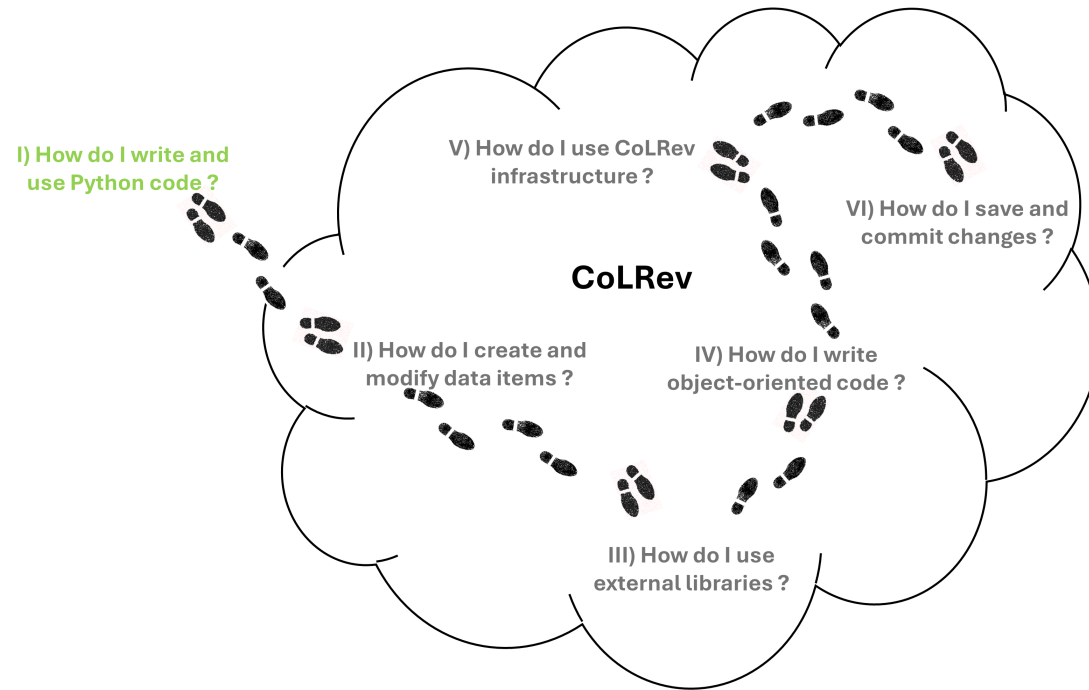Open the notebook for practicing Git branching:

Open in GitHub Codespaces

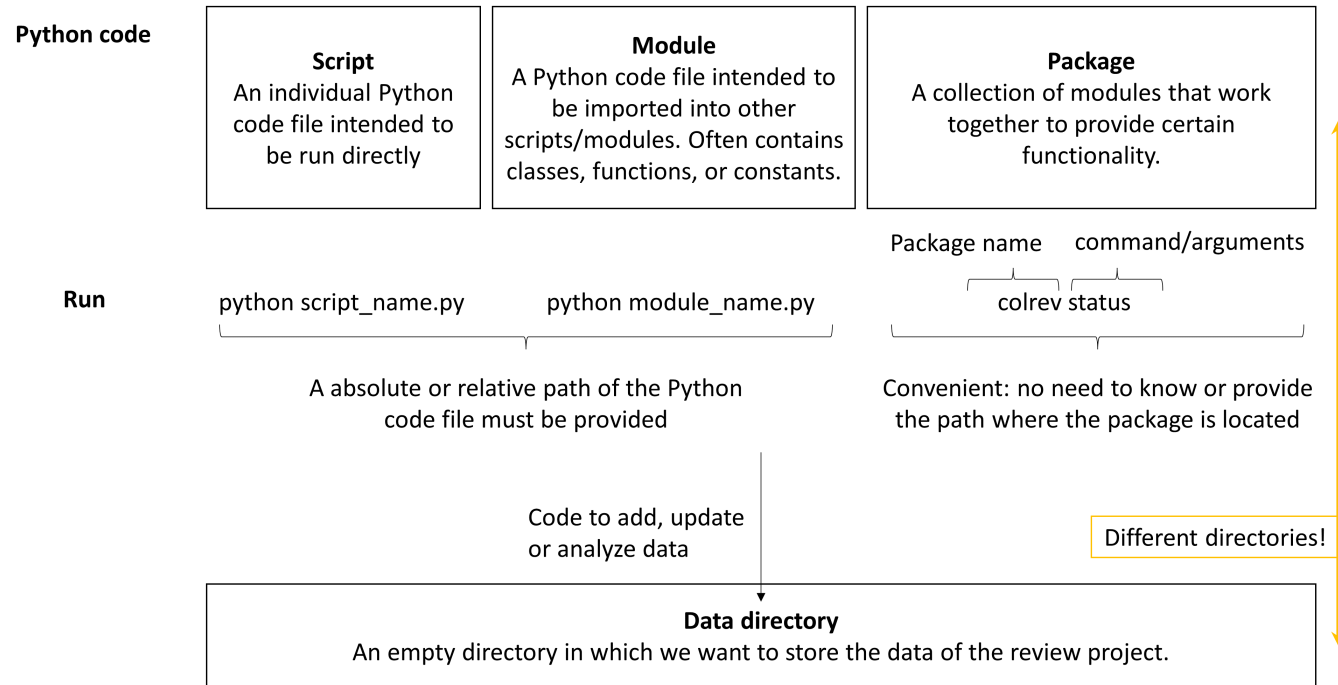The setup can run in the background, while we focus on the concepts.

Python

- Supports multiple paradigms: object-oriented, procedural, or functional
- Python is an interpreted language: no need to compile (build jars) before running
- Uses indentation instead of brackets to separate blocks (such as if statements)
- Is strongly, dynamically typed
- Provides access to many packages on PyPI, covering machine learning, data science, web scraping, etc.
- Python is actively developed, with new versions introducing changes in functionality and old versions no longer receiving security updates
  - Python 3.6 (2016): Introduction of f-strings
  - Python 3.7 (2018): Dictionaries are now ordered
  - ...

Start our development environment on GitHub Codespaces

I) How do I write and use Python code ?

V) How do I use CoLRev infrastructure ?

VI) How do I save and commit changes ?

CoLRev

II) How do I create and modify data items ?

IV) How do I write object-oriented code ?

III) How do I use external libraries ?

Writing and running Python code

Python code

Script An individual Python code file intended to be run directly	Module A Python code file intended to be imported into other scripts/modules. Often contains classes, functions, or constants.	Package A collection of modules that work together to provide certain functionality.

Package name   command/arguments

colrev status

Run

python script_name.py   python module_name.py

A absolute or relative path of the Python code file must be provided

Convenient: no need to know or provide the path where the package is located

Code to add, update or analyze data

Different directories!

Data directory
An empty directory in which we want to store the data of the review project.

www.uni-bamberg.de/digital-work/

Python packages

Structure (directories and files)   **Package configuration**

```
colrev/…

docs/..

tests/..

README.md

LICENSE

pyproject.toml
```

```
[tool.poetry]
name = "colrev"
version = "0.12.2"

…
[tool.poetry.scripts]
colrev = "colrev.ui_cli.cli:main"

…
[tool.poetry.dependencies]
requests = "<2.32.0"
```
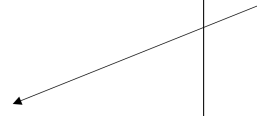
Package management with pip

pip install colrev	Install the colrev package from PyPI to **use** the package
pip install -e .	Install the package in the current local directory („.") to **edit** the package (changes are available instantly without installation)
pip install –e .[dev,docs]	Install the package with the extra dependencies
pip list	List all packages installed via pip
pip show colrev	Show details on the colrev installation

www.uni-bamberg.de/digital-work/

For the tutorial, we switch to the `tutorial_2024_04` branch:

```
git clone https://github.com/CoLRev-Environment/colrev
cd colrev
pip install -e .[dev]
git fetch
git checkout tutorial_2024_04
git reset --hard ca9902e666518af1d33a368adf055c9809004433
```

- As the session progresses, you can checkout the current commits.
- Whenever you see a `git reset --hard ...` command on the following slides, you can use it to set your repository to the required state (commit).

www.uni-bamberg.de/digital-work/

Best practices

- Carefully read tutorials, vignettes, and code examples (e.g., on GitHub)

- Start with small code segments, try whether they work, and extend them

- Add or commit working code frequently

- Use code linters to ensure high code quality (run `pre-commit run --all`)

- To debug code, check whether variables have the expected values (use assert statements)

- When exceptions are thrown, read the Traceback:

```
Traceback (most recent call last):
  File "/path/to/example.py", line 4, in <module>
    greet('Chad')
  ...
  File "/path/to/example.py", line 2, in greet
    print('Hello, ' + someon)
NameError: name 'someon' is not defined
```

read from bottom to top

www.uni-bamberg.de/digital-work/

Next steps

- Read the package development documentation.

- Study code of related CoLRev packages.

- Take notes on the CoLRev-objects or libraries that will be needed.

Tip:

You can use this tutorial for more insights in Python

www.uni-bamberg.de/digital-work/

We value your feedback and suggestions

We encourage you to share your feedback and suggestions on this slide deck:

 <img src="../assets/iconmonstr-pencil-lined.svg" alt="Edit" width="32" height="32"> Suggest specific changes by directly modifying the content
 <img src="../assets/iconmonstr-info-12.svg" alt="New Issue" width="32" height="32"> Provide feedback by submitting an issue
Your feedback plays a crucial role in helping us align with our core goals of **impact in research, teaching, and practice**. By contributing your suggestions, you help us further our commitment to **rigor**, **openness** and **participation**. Together, we can continuously enhance our work by contributing to **continuous learning** and collaboration across our community.

Visit this page to learn more about our goals: 🚀 🛠️ ♻️ 🙏 🧑‍🎓 .