

# From scripting to coding in 30 minutes

How to refactor, document, and distribute your code (example using Python)

---

Dr. Dieter Werthmüller

Trial Lecture and Interview; 12 July 2022

Dep. Geoscience & Engineering, CEG, TU Delft

## After today's lecture you should know...

- when and why to refactor your scripts/code;
- when and why to move from a notebook to a Python file;
- how to document your functions;
- how to create a module;
- how to easily distribute your module.

⇒ Let's start in a Jupyter Notebook!

# Checklist

- If similar statements repeat over and over in a script  
⇒ create a function
- Document your functions; for your future self and others
- If you use similar statements or functions in different notebooks of a directory  
⇒ create a python-file that you can import
- If you use similar statements or functions in different places  
⇒ create a python module that you can import anywhere
- Put it online (e.g. GitHub), so anyone can easily install it via  
`pip install git+https://github.com/{username}/{reponame}`

## Advanced topics and further info

- Deploying to PyPi and Conda
- Testing (plus linting, coverage)
- Continuous Integration
- Benchmarking
- Creating a manual

Further info:

- You can find the material of this lecture on [github.com/prisae/tlecture](https://github.com/prisae/tlecture)
- For more information about creating a module, see [packaging.python.org/en/latest/tutorials/packaging-projects](https://packaging.python.org/en/latest/tutorials/packaging-projects)