

# **Effective Programming Practices for Economists**

## **Basic Python**

### **Running Python code via pytask**

Janoś Gabler and Hans-Martin von Gaudecker

# Preparation

- We assume you have installed anaconda and created the course environment
- Open a shell in the root directory of your project
  - On Windows, use the anaconda prompt or the powershell
  - If conda is not recognized in the powershell, check out this [stackoverflow post](#)
- Activate the environment using `conda activate epp`
- Confirm the activation worked using `conda info`

# 0: Activate and Info

```
hmg@hmg-home:/mnt/econ/epp/example
(base) → example conda activate epp
(epp) → example conda info

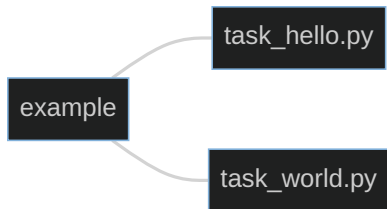
active environment : epp
  active env location : /home/hmg/mambaforge/envs/epp
    shell level      : 2
  user config file   : /home/hmg/.condarc
populated config files : /home/hmg/.condarc
  conda version      : 23.1.0
  conda-build version : not installed
  python version     : 3.10.10.final.0
  virtual packages   : __archspec=1=x86_64
                     __cuda=11.4=0
                     __glibc=2.35=0
                     __linux=6.2.0=0
                     __unix=0=0
  base environment   : /home/hmg/mambaforge (writable)
  conda av data dir  : /home/hmg/mambaforge/etc/conda
  conda av metadata url : None
  channel URLs      : https://conda.anaconda.org/conda-forge/linux-64
                     https://conda.anaconda.org/conda-forge/noarch
                     https://conda.anaconda.org/pytask/linux-64
                     https://conda.anaconda.org/pytask/noarch
                     https://repo.anaconda.com/pkgs/main/linux-64
```

# How does pytask execute code?

- Executing .py files: Run the entire file
- Executing notebooks: Run individual cells
- Pytask: Run individual functions in multiple .py files

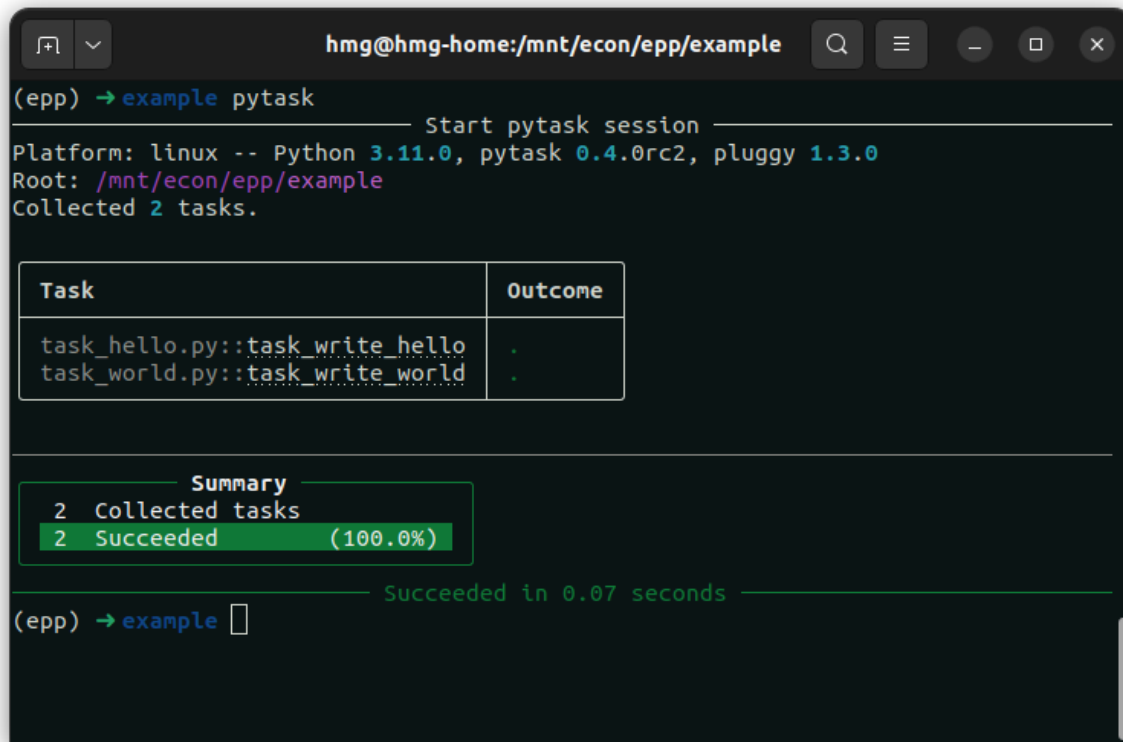
Very useful for automating research pipelines

# Example Project Structure



- Our shell is in the `example` directory
- We want to run all functions that start with `task_` in both `.py` files
- Command is `pytask`

# 1: Execute



```
hmg@hmg-home:/mnt/econ/epp/example
(epp) → example pytask
Start pytask session
Platform: linux -- Python 3.11.0, pytask 0.4.0rc2, pluggy 1.3.0
Root: /mnt/econ/epp/example
Collected 2 tasks.

Task Outcome
task_hello.py::task_write_hello .
task_world.py::task_write_world .

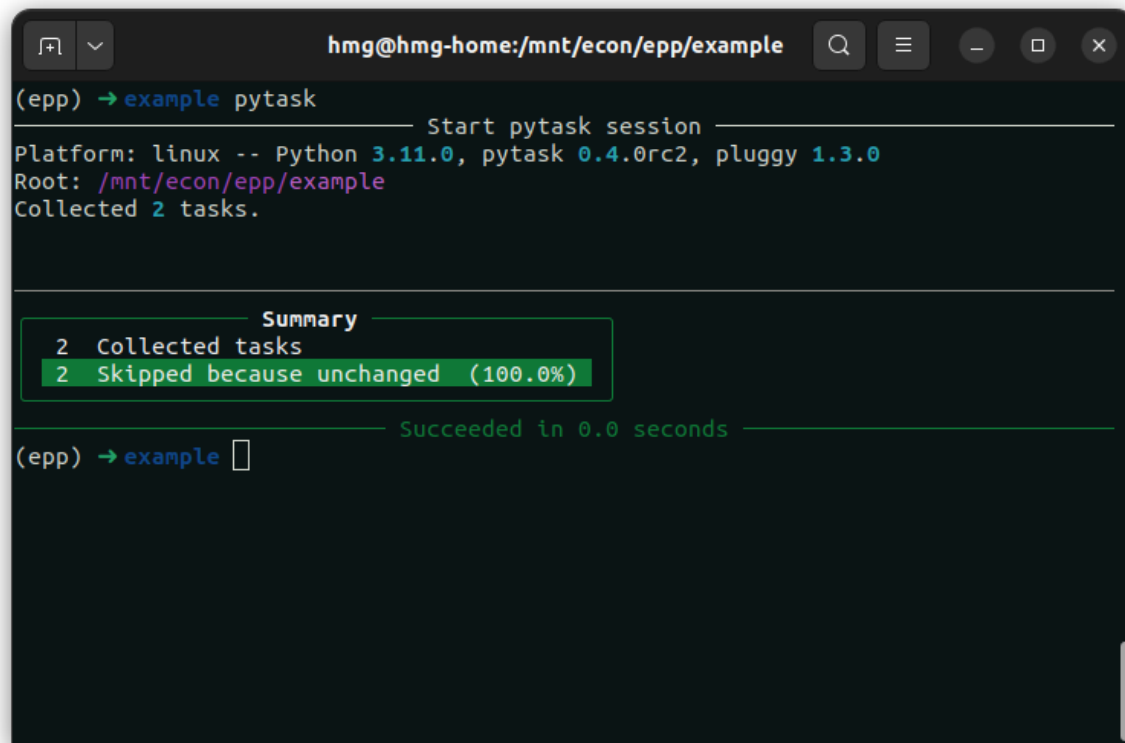
Summary
2 Collected tasks
2 Succeeded (100.0%)

Succeeded in 0.07 seconds
(epp) → example
```

Task	Outcome
task_hello.py::task_write_hello	.
task_world.py::task_write_world	.

Summary	
2	Collected tasks
2	Succeeded (100.0%)

# 1: Execute again



```
hmg@hmg-home:/mnt/econ/epp/example
(epp) → example pytask

Start pytask session
Platform: linux -- Python 3.11.0, pytask 0.4.0rc2, pluggy 1.3.0
Root: /mnt/econ/epp/example
Collected 2 tasks.

Summary
2 Collected tasks
2 Skipped because unchanged (100.0%)

Succeeded in 0.0 seconds
(epp) → example
```

The terminal window shows the execution of the 'pytask' command. It displays the platform (linux), Python version (3.11.0), pytask version (0.4.0rc2), and pluggy version (1.3.0). It also shows the root directory (/mnt/econ/epp/example) and the number of collected tasks (2). A summary box highlights that 2 tasks were collected and 2 tasks were skipped because they were unchanged (100.0%). The execution succeeded in 0.0 seconds.