Effective Programming Practices for Economists

Basic Python

Importing, Modules, Namespaces

Janoś Gabler and Hans-Martin von Gaudecker

Topics

- What are libraries?
- Different ways to import libraries or parts of libraries
- Namespaces
- Modules
- Typical errors when importing

Third party libraries

- Python is a general purpose programming language
- The base language you have seen so far is extended by labraries
  - Standard library (e.g. pathlib, functools)
  - Third party libraries (e.g. numpy, pandas, scipy)
- Libraries need to be imported to use them
- Third party libraries need to be installed

The example

- We will use the numpy library as example
- Numpy is a library that provides efficient data structures and functions for working with matrices and higher dimensional arrays
- You can watch the numpy screencast for details but otherwise just ignore the numpy specific details for now

Different ways to import

Import one function / object

```
from numpy import array
```

Import an entire library

```
import numpy
```

Import entire library and rename it

```
import numpy as np
```

Import everything from a library

```
from numpy import *
```

- Use single import if you need one thing
- Use library import if you need many functions
- Use shorthand if there is a convention, e.g. numpy (np), pandas (pd), seaborn (sns)
- Never ever use `` `import *` ``

Namespaces or why not use `` `import *` ``

```
# bad option
>>> from math import log
>>> log(2.718281828459045)
1.0

# better option
>>> import math
>>> math.log(2.718281828459045)
1.0
```

- Multiple libraries could implement `` `log` ``
  - math library: The natural logarithm
  - web development library: Write a log file
- Importing an entire library makes it very explicit from which namespace you use a function
- Namespaces are one of the reasons why Python can succeed in so many different areas!

Modules

- So far we imported from packages or the standad library
- You can import from any module (a module is a `` `.py` `` file)
- In larger projects you will split code across multiple modules and import from them

ModuleNotFoundError

```
>>> from numpai import array
---------------------------------------------------------------------------
ModuleNotFoundError                       Traceback (most recent call last)
Cell In[32], line 1
----> 1 from nampai import array

ModuleNotFoundError: No module named 'nampai'
```

- Meaning: The library you asked for is not found
- Do you have a typo in the library name?
- Is the library installed in your environment?
- **Is the correct environment activated?**

ImportError

```
from numpy import arrrrray
---------------------------------------------------------------------------
ImportError                               Traceback (most recent call last)
Cell In[33], line 1
----> 1 from numpy import arrrrray

ImportError: cannot import name 'arrrrray' from 'numpy' (/home/janos/miniconda3/
envs/dl_intro/lib/python3.11/site-packages/numpy/__init__.py)
```

- Something went wrong during import
- Do you have typos in what you want to import?
- Is the correct version of the library installed?