Introduction

The code of large Python projects is spread among several files. One must be able to use within a file the software components stored in another file.

Python handles imports by replacing the idea of file by the idea of **module**. A set of modules constitutes a **package**.

In this part, the `numpy` package is used to demonstrate the various import methods.

Full import

With the keywork `import`, a package (group of subpackages and modules) or a module is placed into the local memory space.

We can make use of the module. For instance, list its attributes using the `dir` function.

In [1]:
```
import numpy
dir(numpy)[:10]
```

Out[1]:
```
['ALLOW_THREADS',
 'BUFSIZE',
 'CLIP',
 'DataSource',
 'ERR_CALL',
 'ERR_DEFAULT',
 'ERR_IGNORE',
 'ERR_LOG',
 'ERR_PRINT',
 'ERR_RAISE']
```

It is handy to associate a shorter name to the imported module: this is called an **alias**.

In [2]:
```
import numpy as np
dir(np)[:10]
```

Out[2]:
```
['ALLOW_THREADS',
 'BUFSIZE',
 'CLIP',
 'DataSource',
 'ERR_CALL',
 'ERR_DEFAULT',
 'ERR_IGNORE',
 'ERR_LOG',
 'ERR_PRINT',
 'ERR_RAISE']
```

The components that can be imported are available as a hierarchy from the root package (here: `numpy`):

```
In [3]: import numpy.random.bit_generator
        type(numpy.random.bit_generator)
```

```
Out[3]: module
```

Relative import

Using `from ... import ...`, some specific components are placed in the local memory space. These components can be:

- variables
- classes
- functions
- modules

In [4]:
```python
from numpy.random import randint
```

Good practices

Import only the needed content

Importing Python objects can be unnecessarily time-consuming. Thus, relative imports mist be preferred over absolute imports so that only needed components are imported.

In [5]:
```
from numpy import log, sqrt
from numpy.random import rand
```

Choose the import name wisely

If a chosen alias is also an existing variable name, the variable reference will be lost (shadowing):

```
In [6]:
rd = 5
print(rd)
from numpy.random import randint as rd
print(rd)
```

```
5
<built-in method randint of numpy.random.mtrand.RandomState object at 0x7b51
7a1e6740>
```

Move all imports to the beginning of the file

For clarity purpose, in an ideal world, all imports must be placed at the beginning of the file and be sorted:

1. By origin:
    - A. Built-in packages: `os`, `sys`, `pathlib`, etc...
    - B. Third-party packages from internet: `pandas`, `numpy`, `matplotlib`, etc...
    - C. Your local packages or modules
2. By alphabetical order

n.b.: some IDE order the imports automatically.

Example of sorted imports:

In [7]:
```
from os import getcwd, lstat
from time import sleep

from pandas import DataFrame, Interval

# from mypackage.mymodule import a, b, c
```