pathlib

1) Constants

2) Dependent quantities

3) Auxiliary function

4) Memory allocation

5) Initialization

6) Iteratation

```
for k in range(K):
    ...
    ...
    ...
```

7) Analysis

8) Plots

1) Constants

2) Dependent quantities

3) Auxiliary function

4) Memory allocation

5) Initialization

6) Iteratation

```
for k in range(K):
    ...
    ...
    ...
```

7) Analysis

8) Plots

1) Constants

2) Dependent quantities

3) Auxiliary function

4) Memory allocation

5) Initialization

6) Iteratation

```
for k in range(K):
    ...
    ...
    ...
```

Write to files

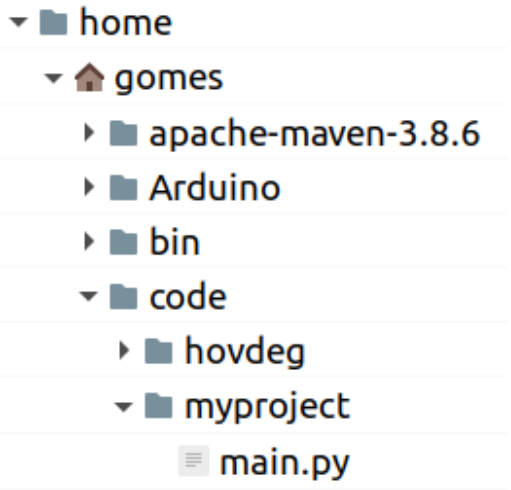
Read from files

7) Analysis

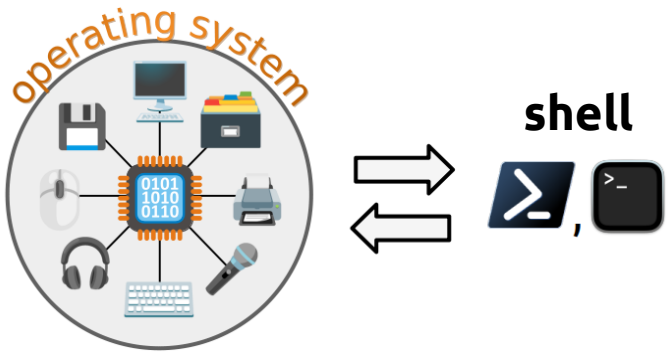
8) Plots

Terminology

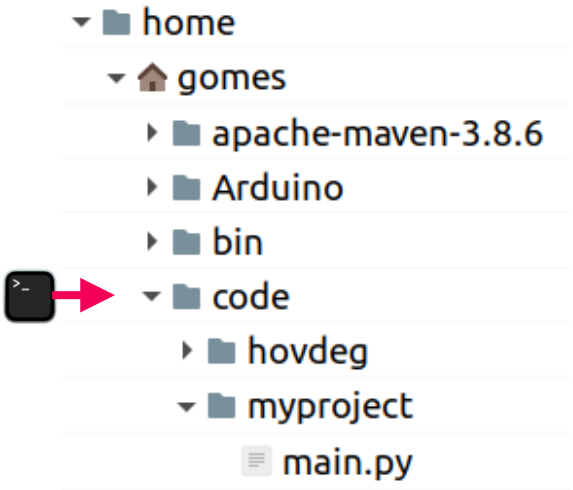
- **file**: document
- **directory**: folder (a container of files)



- **shell**: A program used to communicate with the operating system.
  - Windows: PowerShell
  - Mac/Linux: Terminal



- **path:** Address of the file or directory in the file system
    - **absolute** : with respect to the root of the file system.
    - **relative** : with respect to the shell's **current working directory**.
- **current woking directory (cwd)** : directory currently referred to by the shell.



pathlib

- Everything is based on the **Path** object:

```
from pathlib import Path
```

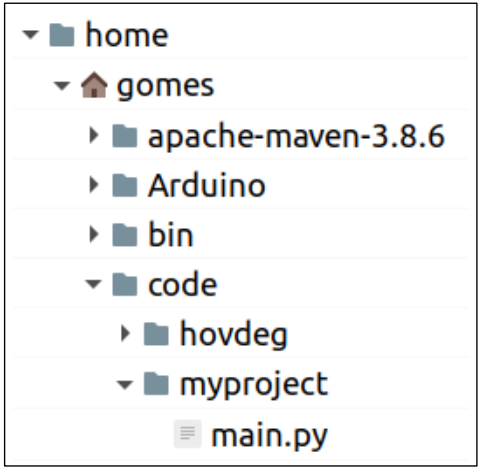
- A **Path** object represents a **file** or **directory** path.

```
path_cwd = Path.cwd()          # the directory from where the
                               # interpreter was launched
```

```
path_home = Path.home()        # the user's home folder
```

- Moving down the tree: **slash operator (/)**

```
code_dir  = path_home / 'code'
main_file = path_home / 'code' / 'myproject' / 'main.py'
```



- Moving up the tree:

```
main_file.parent
```

is equal to

```
path_home/'code'/'myproject'
```

```
main_file.parent.parent
```

is equal to

```
path_home/'code'
```

Path methods

Inspect	`Path.exists()`	… True if the path exists
	`Path.is_file()`	… True if the path is a file
	`Path.is_dir()`	… True if the path is a directory
	`Path.name`	… Last part of the path
	`Path.suffix`	… File extension, if any
Create	`Path.touch()`	… Create new file
	`Path.mkdir()`	… Create new directory
Iterate through a directory	`Path.iterdir()`	… Iterator for a directory