

11. File and Directory Access

The modules described in this chapter deal with disk files and directories. For example, there are modules for reading the properties of files, manipulating paths in a portable way, and creating temporary files. The full list of modules in this chapter is:

- [11.1. pathlib — Object-oriented filesystem paths](#)
 - [11.1.1. Basic use](#)
 - [11.1.2. Pure paths](#)
 - [11.1.2.1. General properties](#)
 - [11.1.2.2. Operators](#)
 - [11.1.2.3. Accessing individual parts](#)
 - [11.1.2.4. Methods and properties](#)
 - [11.1.3. Concrete paths](#)
 - [11.1.3.1. Methods](#)
- [11.2. os.path — Common pathname manipulations](#)
- [11.3. fileinput — Iterate over lines from multiple input streams](#)
- [11.4. stat — Interpreting stat\(\) results](#)
- [11.5. filecmp — File and Directory Comparisons](#)
 - [11.5.1. The dircmp class](#)
- [11.6. tempfile — Generate temporary files and directories](#)
 - [11.6.1. Examples](#)
 - [11.6.2. Deprecated functions and variables](#)
- [11.7. glob — Unix style pathname pattern expansion](#)
- [11.8. fnmatch — Unix filename pattern matching](#)
- [11.9. linecache — Random access to text lines](#)
- [11.10. shutil — High-level file operations](#)
 - [11.10.1. Directory and files operations](#)
 - [11.10.1.1. copytree example](#)
 - [11.10.1.2. rmtree example](#)
 - [11.10.2. Archiving operations](#)
 - [11.10.2.1. Archiving example](#)
 - [11.10.3. Querying the size of the output terminal](#)
- [11.11. macpath — Mac OS 9 path manipulation functions](#)

See also:

Module [os](#)

Operating system interfaces, including functions to work with files at a lower level than Python [file objects](#).

Module [io](#)

Python's built-in I/O library, including both abstract classes and some concrete classes such as file I/O.

Built-in function `open()`

The standard way to open files for reading and writing with Python.