

Navigation

- [index](#)
- [modules](#) |
- [next](#) |
- [previous](#) |
- [Python v2.6.4 documentation](#) »
- [The Python Standard Library](#) »
- [11. File and Directory Access](#) »

11.5. filecmp — File and Directory Comparisons¶

The `filecmp` module defines functions to compare files and directories, with various optional time/correctness trade-offs. For comparing files, see also the [difflib](#) module.

The `filecmp` module defines the following functions:

`filecmp.cmp(f1, f2[, shallow])`¶

Compare the files named `f1` and `f2`, returning `True` if they seem equal, `False` otherwise.

Unless `shallow` is given and is false, files with identical [os.stat\(\)](#) signatures are taken to be equal.

Files that were compared using this function will not be compared again unless their [os.stat\(\)](#) signature changes.

Note that no external programs are called from this function, giving it portability and efficiency.

`filecmp.cmpfiles(dir1, dir2, common[, shallow])`¶

Compare the files in the two directories `dir1` and `dir2` whose names are given by `common`.

Returns three lists of file names: `match`, `mismatch`, `errors`. `match` contains the list of files that match, `mismatch` contains the names of those that don't, and `errors` lists the names of files which could not be compared. Files are listed in `errors` if they don't exist in one of the directories, the user lacks permission to read them or if the comparison could not be done for some other reason.

The `shallow` parameter has the same meaning and default value as for [filecmp.cmp\(\)](#).

For example, `cmpfiles('a', 'b', ['c', 'd/e'])` will compare `a/c` with `b/c` and `a/d/e` with `b/d/e`. `'c'` and `'d/e'` will each be in one of the three returned lists.

Example:

```
>>> import filecmp
>>> filecmp.cmp('undoc.rst', 'undoc.rst')
True
>>> filecmp.cmp('undoc.rst', 'index.rst')
False
```

11.5.1. The `dircmp` class¶

`dircmp` instances are built using this constructor:

`class filecmp.dircmp(a, b[, ignore[, hide]])`¶

Construct a new directory comparison object, to compare the directories `a` and `b`. `ignore` is a list of names to ignore, and defaults to `['RCS', 'CVS', 'tags']`. `hide` is a list of names to hide, and defaults to `[os.curdir, os.pardir]`.

The `dircmp` class provides the following methods:

`report()`¶

Print (to `sys.stdout`) a comparison between `a` and `b`.

`report_partial_closure()`¶

Print a comparison between `a` and `b` and common immediate subdirectories.

`report_full_closure()`¶

Print a comparison between `a` and `b` and common subdirectories (recursively).

The `dircmp` offers a number of interesting attributes that may be used to get various bits of information about the directory trees being compared.

Note that via [`getattr\(\)`](#) hooks, all attributes are computed lazily, so there is no speed penalty if only those attributes which are lightweight to compute are used.

[`left_list\(\)`](#)

Files and subdirectories in *a*, filtered by *hide* and *ignore*.

[`right_list\(\)`](#)

Files and subdirectories in *b*, filtered by *hide* and *ignore*.

[`common\(\)`](#)

Files and subdirectories in both *a* and *b*.

[`left_only\(\)`](#)

Files and subdirectories only in *a*.

[`right_only\(\)`](#)

Files and subdirectories only in *b*.

[`common_dirs\(\)`](#)

Subdirectories in both *a* and *b*.

[`common_files\(\)`](#)

Files in both *a* and *b*

[`common_funny\(\)`](#)

Names in both *a* and *b*, such that the type differs between the directories, or names for which [`os.stat\(\)`](#) reports an error.

[`same_files\(\)`](#)

Files which are identical in both *a* and *b*.

[`diff_files\(\)`](#)

Files which are in both *a* and *b*, whose contents differ.

[`funny_files\(\)`](#)

Files which are in both *a* and *b*, but could not be compared.

[`subdirs\(\)`](#)

A dictionary mapping names in [`common_dirs`](#) to [`dircmp`](#) objects.

[Table Of Contents](#)

[11.5. filecmp — File and Directory Comparisons](#)

- [11.5.1. The `dircmp` class](#)

Previous topic

[11.4. statvfs — Constants used with `os.statvfs\(\)`](#)

Next topic

[11.6. tempfile — Generate temporary files and directories](#)

This Page

- [Show Source](#)

Navigation

- [index](#)
- [modules](#) |
- [next](#) |
- [previous](#) |
- [Python v2.6.4 documentation](#) »
- [The Python Standard Library](#) »
- [11. File and Directory Access](#) »

© [Copyright](#) 1990-2010, Python Software Foundation.

The Python Software Foundation is a non-profit corporation. [Please donate.](#)

Last updated on Feb 26, 2010. Created using [Sphinx](#) 0.6.3.