

ディレクトリパスを扱う関数は、標準モジュール `os` の `path` 名前空間に存在する。

```
>>> import os
>>> dir(os.path)
['_all_', '__builtins__', '__cached__', '__doc__', '__file__', '__name__', '__package__', '_get_altsep', '_get_bothseps', '_get_colon', '_get_dot', '_get_empty', '_get_sep', '_get_special', '_getfileinformation', '_getfinalpathname', '_getfullpathname', 'abspath', 'altsep', 'basename', 'commonprefix', 'curdir', 'defpath', 'devnull', 'dirname', 'exists', 'expanduser', 'expandvars', 'extsep', 'genericpath', 'getatime', 'getctime', 'getmtime', 'getsize', 'isabs', 'isdir', 'isfile', 'islink', 'ismount', 'join', 'lexists', 'normcase', 'normpath', 'os', 'pardir', 'pathsep', 'realpath', 'relpath', 'samefile', 'sameopenfile', 'sep', 'split', 'splitdrive', 'splittext', 'splitunc', 'stat', 'supports_unicode_filenames', 'sys']
>>>
```

以下、主なもののみ。

`abspath(path)`

相対パスから絶対パスを得る。

`basename(path)`

パスからベースネームを得る。ディレクトリパスの場合はカレントディレクトリ名。ファイルパスの場合はファイル名が得られる。

`dirname(path)`

親ディレクトリパスを得る。

`exists(path)`

存在判定。

`isdir(path)`

ディレクトリかどうかの判定。

`isfile(path)`

ファイルかどうかの判定。

`islink(path)`

シンボリックリンクかどうかの判定。シンボリックリンクがサポートされていない環境では、常に `False` を返す。

`join(path1, path2, ...)`

文字列をつなげてパスを作成する。ディレクトリ区切り文字列が自動的に適切に追加・削除される。

`normcase(path)`

Windows でパス名をすべて小文字にし、ディレクトリ区切り文字がスラッシュの場合にバックスラッシュに変換した Python 文字列を返す。Windows 以外の環境では `normcase` は入力された文字をそのまま返すだけ。

`normpath(path)`

重複したディレクトリ区切り文字を削除し、冗長なパスを削除し、Windows でスラッシュをバックスラッシュに変換した Python 文字列を返す。余談だが Windows はスラッシュ("/")もディレクトリ区切り文字として処理するので、Linux 用のディレクトリパス処理コードはほぼそのまま正しく動作する。スラッシュとバックスラッシュが混在したパスでも問題なし。