Get and change the current working directory in Python

In Python, you can get and change (set) the current working directory with `os.getcwd()` and `os.chdir()`.

`os` module is included in the standard library, so no additional installation is required.

- os — Miscellaneous operating system interfaces — Python 3.7.4 documentation

This article describes the following contents.

- Get the current working directory: `os.getcwd()`
- Change the current working directory: `os.chdir()`

The path of the script file (`.py`) can be obtained with `__file__`. See the following article.

- Get the path of running file (.py) in Python: `__file__`

Get the current working directory: os.getcwd()

`os.getcwd()` returns the absolute path of the working directory where Python is currently running as a string `str`.

`getcwd` stands for "get current working directory", and the Unix command `pwd` stands for "print working directory".

Of course, you can print the working directory with `print()`.

```
import os

path = os.getcwd()

print(path)
# /Users/mbp/Documents/my-project/python-snippets/notebook

print(type(path))
# <class 'str'>
```

Use `os.path` to manipulate the path string. See the following article for details.

- Extract the file, dir, extension name from a path string in Python

Change the current working directory: `os.chdir()`

You can change the current working directory with `os.chdir()`.

Specify the destination path in the argument. It can be absolute or relative. Use `'../'` to move up.

You can change the current directory like the Unix command `cd`.

```
os.chdir('../')
```

```python
print(os.getcwd())
# /Users/mbp/Documents/my-project/python-snippets
```

Both `chdir` and `cd` stands for "change directory".

With the `__file__` and `os.path` functions, you can change the current directory to the directory where the running script file (`.py`) exists.

```python
os.chdir(os.path.dirname(os.path.abspath(__file__)))
```