Digital Career Institute

Python Course: Input & Output

Digital Career Institute

Using the File System

Digital Career Institute

Using the File System

When files are being created ...

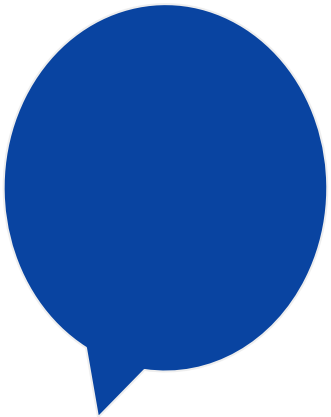
Does the file already exist?

Does the directory exist?

What other files are in the directory?

Using the File System

When files are being created ...



Create a directory

Delete a directory

Explore a directory

Delete a file

These are all operations on **paths**.

The pathlib Module

Digital Career Institute

The **`pathlib`** module is a utility library that provides access to various operations with paths in the file system.

Create a File

paths.py

```
from pathlib import Path


file = Path("/home/PythonCourse/text.txt")
file.open("w").write("My notes")
```

Files can also be created using **pathlib**.

The **Path** object has a method **open** that returns a stream.

Delete a File

Digital Career Institute

paths.py

```
from pathlib import Path


file = Path("/home/PythonCourse/test.py")
file.unlink()
```

A directory cannot be removed this way.

```
$ python3 paths.py
```

Delete a File

Digital Career Institute

paths.py

```
from pathlib import Path


file = Path("/does/not/exist.py")
file.unlink()
```

By default, the **unlink** method will raise an exception if the file does not exist.

```
$ python3 paths.py
FileNotFoundError: [Errno 2] No such file or directory:
'does/not/exist.py'
```

Delete a File

paths.py

```
from pathlib import Path


file = Path("/does/not/exist.py")
file.unlink(missing_ok=True)
```

The **missing_ok** argument will prevent the exception if it is set to **True**.

```
$ python3 paths.py
$
```

Python 3.8

Remove a Directory

paths.py

```
from pathlib import Path


directory = Path("/home/PythonCourse/test/")
directory.rmdir()
```

A directory **must be empty** before it can be deleted.

```
$ python3 paths.py
```

Create a Directory

Digital Career Institute

paths.py

```
from pathlib import Path


directory = Path("test")
directory.mkdir()
```

```
$ python3 paths.py
```

Get the Current Directory

/home/PythonCourse/paths.py

```
from pathlib import Path


print(Path.cwd())
```

```
$ python3 paths.py
/home/PythonCourse
```

Rename a File or Directory

Digital Career Institute

paths.py

```
from pathlib import Path


file = Path("foo.txt")
file.open("w").write("Some text")
new_file = Path("bar.txt")
file.replace(new_file)
print(new_file.open().read())
```

```
$ python3 paths.py
Some text
```

List the Directory Content

Digital Career Institute

paths.py

```
from pathlib import Path


books = Path("books")
for item in books.iterdir():
    print(item)
```

```
$ python3 paths.py
the_hobbit.bin
the_hobbit.txtt
todo_list.txt
io
```

Search a Directory

Digital Career Institute

paths.py

```
from pathlib import Path


books = Path("books")
for path in books.glob("*.txt"):
    print(path)
```

```
$ python3 paths.py
books/the_hobbit.txt
books/dracula.txt
books/frankenstein.txt
$
```

glob returns any path object (files and directories) matching the indicated pattern.

Search a Directory

Digital Career Institute

paths.py

```
from pathlib import Path


books = Path("books")
for path in books.glob("*/"):
    print(path)
```

```
$ python3 paths.py
books/fantasy
books/horror
books/biography
$
```

glob can be used to match subdirectories.

Search a Directory

Digital Career Institute

paths.py

```
from pathlib import Path


books = Path("books")
for path in books.glob("**/*.txt"):
    print(path)
```

```
$ python3 paths.py
books/frankenstein.txt
books/fantasy/the_hobbit.txt
books/horror/dracula.txt
$
```

glob can also be used recursively.

Get the File Path

paths.py

```
from pathlib import Path


print(__file__)
file = Path(__file__)
print(file.resolve())
```

`__file__` is a reference to the current file.

`resolve` returns the full path of the file.

```
$ python3 paths.py
paths.py
/home/DCI/PythonCourse/paths.py
```

Get the File's Directory Path

paths.py

```
from pathlib import Path


file_path = Path("the_hobbit.txt")
print(file_path.parent)
print(file_path.resolve().parent)
print(file_path.parent.resolve())
```

`parent` returns the parent directory of the provided path.

If the **`file_path`** is relative **`resolve`** can be used either before or after using **`parent`**.

```
$ python3 paths.py
.
/home/DCI/PythonCourse
/home/DCI/PythonCourse
```

Get the Object Name

paths.py

```
from pathlib import Path


file = Path("/home/DCI/main.py")
print(file.name)
directory = Path("/home/DCI")
print(directory.name)
```

name returns the name of the object.

```
$ python3 paths.py
main.py
DCI
```

Join Paths

paths.py

```
from pathlib import Path


home = Path("/home")
user = "DCI"
course = "PythonCourse"
path = home.joinpath(user, course)
print(path)
```

The **`joinpath`** method will return a new path merging the inputs into the original path.

```
$ python3 paths.py
/home/DCI/PythonCourse
```

Join Paths

paths.py

```
from pathlib import Path


home = Path("/home")
user = "DCI"
course = "PythonCourse"
path = home / user / course
print(path)
```

The `/` operator and the **`joinpath`** method serve the same purpose.

```
$ python3 paths.py
/home/DCI/PythonCourse
```

Existence of a Path

paths.py

```
from pathlib import Path


path = Path("/home/DCI/PythonCourse/")
print(path.exists())
```

We can use either a file path or a directory path.

```
$ python3 paths.py
True
```

The Pythonic Way

Digital Career Institute

Ask **forgiveness**,
not permission.

In Python, Exceptions have a very small cost in performance. Smaller than most operations.

Existence of a Path: Asking Forgiveness

Digital Career Institute

paths.py

```
from pathlib import Path


path = Path("/path/")
if path.exists():
    file = path.open()
else:
    print("Does not exist")
```

Asking permission

path_example.py

```
from pathlib import Path


path = Path("/path/")
try:
    file = path.open()
except FileNotFoundError:
    print("Does not exist")
```

Asking forgiveness

Nature of a Path

Digital Career Institute

paths.py

```
from pathlib import Path


path = Path("/home/DCI/PythonCourse/")
print(path.is_absolute())
print(Path("the_hobbit.txt").is_absolute())
```

is_absolute returns **True** if the path is **absolute**.

```
$ python3 paths.py
True
False
```

Nature of a Path

Digital Career Institute

paths.py

```
from pathlib import Path


path = Path("/home/DCI/PythonCourse/")
print(path.is_file())
```

is_file returns **True**
if the path is a **file**.

```
$ python3 paths.py
False
```

Nature of a Path

Digital Career Institute

paths.py

```
from pathlib import Path


path = Path("/home/DCI/PythonCourse/")
print(path.is_dir())
```

is_dir returns **True**
if the path is a **directory**.

```
$ python3 paths.py
True
```

The os Module

Digital Career Institute

The **os** module is a utility interface to various operations that concern the **operating system** where the code is being executed.

The **os** module can perform many operations on the operating system, such as getting information, working with environment variables or executing system processes.

It also has some feature to work with paths.

Walk the Directory Tree

Digital Career Institute

os_example.py

```
import os
from pathlib import Path


for dir_name, subdirs, files in os.walk(Path.cwd()):
    print("*" * 20)
    print("Directory:", dir_name)
    print("Subdirectories:", subdirs)
    print("Files:", files)
```

walk loops through all the directory tree and returns every directory, its subdirectories and its files.

Walk the Directory Tree

Digital Career Institute

os_example.py

```
import

for dir
    pri
    pri
    pri
    pri
```

```
$ python3 os_example.py
**********
Directory: /home/DCI/PythonCourse
Subdirectories: ['io']
Files: ['the_hobbit.bin', 'the_hobbit.txt', 'todo_list.txt']
**********
Directory: /home/DCI/PythonCourse/io
Subdirectories: []
Files: ['test.py']
```

The os.path Module

Digital Career Institute

The **`os.path`** module is a utility interface to various operations that concern the **File System** where the code is being executed.

This module is scarcely used and it is being replaced by **`pathlib`**, but it has some additional features.

Properties of a Path

Digital Career Institute

path_example.py

```
import os


print(os.path.getsize("the_hobbit.txt"))
print(os.path.getmtime("the_hobbit.txt"))
```

getsize returns the size of a path.

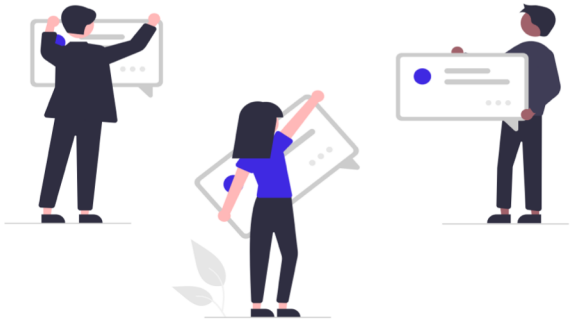
getmtime returns the timestamp of the last modification.

```
$ python3 path_example.py
247
1629893207.7288237
```

We learned

...

- That we can use the **`pathlib`** module to create, delete and explore directories or delete files.
- That **`pathlib`** also provides information about paths, their nature and their properties.
- That it is better to catch the **`FileNotFoundError`** exception than using **`os.path.exists`** to prevent it.
- That the **`os`** module provides additional features to work with paths and the operating system.



Reflection Round

Digital Career Institute

Suggested Topics

- Discuss advantages and disadvantages of storing data in files rather than in the database.
- Name 10 possible uses of a file based approach.
- Discuss the convenience (or not) of using IO streams to manipulate simple text. Compare it to using simple `str` objects.

Documentation

Documentation

- File I/O
  https://docs.python.org/3/tutorial/inputoutput.html#reading-and-writing-files
  https://www.tutorialspoint.com/python/python_files_io.htm
  https://www.w3schools.com/python/python_file_handling.asp
  https://www.w3schools.com/python/python_file_open.asp
  https://www.w3schools.com/python/python_file_write.asp
- Bytes-like objects
  https://www.w3resource.com/python/python-bytes.php
  https://docs.python.org/3/library/stdtypes.html#bytes
  https://docs.python.org/3/library/stdtypes.html#bytearray
  https://www.python.org/dev/peps/pep-0257/

Documentation

- Streams
  https://docs.python.org/3/library/io.html
- File system
  https://docs.python.org/3/library/os.html#module-os
  https://docs.python.org/3/library/os.path.html#module-os.path
- User I/O
  https://docs.python.org/3/library/functions.html#print
- Application I/O
  https://pypi.org/project/requests/

Contact Details
DCI Digital Career Institute gGmbH

Digital Career Institute