

# Python FILE (I/O) Handling

# Nội Dung

- Giới thiệu về FILE
- Đọc FILE
- Ghi FILE
- Xóa FILE

# Giới thiệu về FILE

- File là một thứ rất cần thiết trong các dự án, ví dụ như chúng ta cần phải ghi log ra một file để sau này có thể kiểm soát được.
- Và ngôn ngữ lập trình nào cũng hỗ trợ chúng ta làm việc với file
- Có 3 loại file thông dụng là file văn bản, hình ảnh và âm thanh. Trong nội dung này, chúng ta chỉ hướng đến đối tượng File văn bản (\*.txt)
- Mỗi loại file có module xử lý khác nhau:
  - Module pandas dùng xử lý file CSV, Excel
  - Module PyPDF2 dùng xử lý file PDF
  - Module json dùng xử lý file JSON
  - ...

# MỞ FILE

- Python cung cấp hàm `open()` trả về một đối tượng File mà được sử dụng với các hàm khác.
- Với File đã mở, bạn có thể thực hiện các hoạt động đọc, ghi, ... trên File đó. Cú pháp của hàm `open()` là:

`file_object = open(file_name [, access_mode][, buffering])`

- **file\_name**: Đối số `file_name` là một giá trị chuỗi chứa tên của các file mà bạn muốn truy cập.
- **access\_mode**: Các `access_mode` xác định các chế độ của file được mở ra như `read`, `write`, `append`,... Đây là thông số tùy chọn và chế độ truy cập file mặc định là `read (r)`.
- **buffering**: là thông số đệm cho file mặc định thì nó sẽ là 0.

# MỞ FILE

Mode	Mô tả
r	Mở file chỉ để đọc
r+	Mở file để đọc và ghi
rb	Mở file trong chế độ đọc cho định dạng nhị phân, đây là chế độ mặc định. Con trỏ tại phần bắt đầu của file
rb+	Mở file để đọc và ghi trong định dạng nhị phân. Con trỏ tại phần bắt đầu của file
w	Tạo một file mới để ghi, nếu file đã tồn tại thì sẽ bị ghi mới
w+	Tạo một file mới để đọc và ghi, nếu file tồn tại thì sẽ bị ghi mới
wb	Mở file trong chế độ ghi trong định dạng nhị phân. Nếu file đã tồn tại, thì ghi đè nội dung của file đó, nếu không thì tạo một file mới
wb+	Mở file để đọc và ghi trong định dạng nhị phân. Nếu file tồn tại thì ghi đè nội dung của nó, nếu file không tồn tại thì tạo một file mới để đọc và ghi
a	Mở file để ghi thêm vào cuối file, nếu không tìm thấy file sẽ tạo mới một file để ghi mới
a+	Mở file để đọc và ghi thêm vào cuối file, nếu không tìm thấy file sẽ tạo mới một file để đọc và ghi mới
ab	Mở file trong chế độ append trong chế độ nhị phân. Con trỏ là ở cuối file nếu file này đã tồn tại. Nếu file không tồn tại, thì tạo một file mới để ghi
ab+	Mở file trong chế độ đọc và append trong định dạng nhị phân. Con trỏ file tại cuối nếu file đã tồn tại. Nếu không tồn tại thì tạo một file mới để đọc và ghi

# Thuộc tính của FILE

Thuộc tính	Mô tả
<code>file.closed</code>	Trả về True nếu file đã đóng, ngược lại là False
<code>file.mode</code>	Trả về chế độ truy cập của file đang được mở
<code>file.name</code>	Trả về tên của file

# Thuộc tính của FILE

- Ví dụ: Mở file myfile.txt

# Mở file

```
file = open("myfile.txt", "w+")
```

```
print("Tên của file là: ", file.name)
```

```
print("File có đóng hoặc không? : ", file.closed)
```

```
print("Chế độ mở file : ", file.mode)
```

```
Tên của file là:  myfile.txt
File có đóng hoặc không? :  False
Chế độ mở file :  w+
```

# Đóng File

- Khi bạn đã thực hiện xong các hoạt động trên file thì cuối cùng bạn cần đóng file đó.
- Sử dụng phương thức **close()** để đóng một file.

**fileObject.close()**

# Mở file

```
file = open("myfile.txt ", "r")
```

# Đóng file

```
file.close()
```



# Đọc FILE

- Phương thức **read** : `fileObject.read([size])`
- Phương thức này trả về một chuỗi có kích thước bằng size. Nếu không truyền size thì toàn bộ nội dung của file sẽ được đọc.
- Ví dụ: tạo file myfile.txt có nội dung

```
hello world
1 2 3 4 5 6
My name is Tom
```

```
f = open('myfile.txt', 'r')
str = f.read()
print (str)
f.close()
```

```
In [7]: runfile('C:/Users/sony/Desktop/python_co_ban/
test/mymodule/test_file.py', wdir='C:/Users/sony/
Desktop/python_co_ban/test/mymodule')
hello world
1 2 3 4 5 6
My name is Tom
```

# Đọc FILE

- **Phương thức readline:** `fileObject.readline()`
- Phương thức này cho phép đọc một dòng trong file và trả về chuỗi.

```
f = open('myfile.txt', 'r')  
line1 = f.readline()  
line2 = f.readline()  
print ('Dòng 1: ', line1)  
print ('Dòng 2: ', line2)  
f.close()
```

```
Dòng 1:  hello world  
Dòng 2:  1 2 3 4 5 6
```

# Đọc FILE đơn giản

with open('dog\_breeds.txt', 'r') as reader:

# Further file processing goes here

reader.read()

# Ghi FILE

- Để ghi một file ta cần mở file bằng cú pháp để ghi và sử dụng phương thức write để ghi vào: **fileObject.write(string)**
- Phương thức này cho phép ghi một chuỗi có nội dung là string vào vị trí của con trỏ trong file.

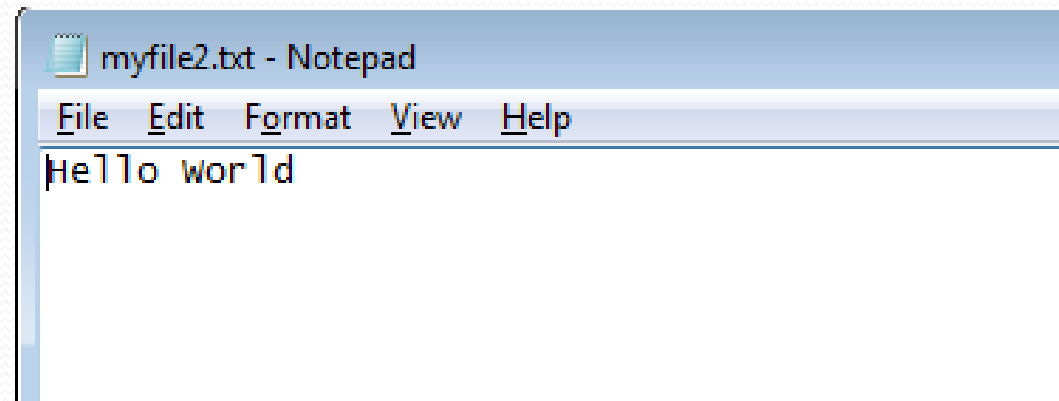
# Mở file

```
file = open("myfile2.txt", "w+")
```

```
file.write("Hello World");
```

# Đóng file

```
file.close()
```



# Ghi FILE

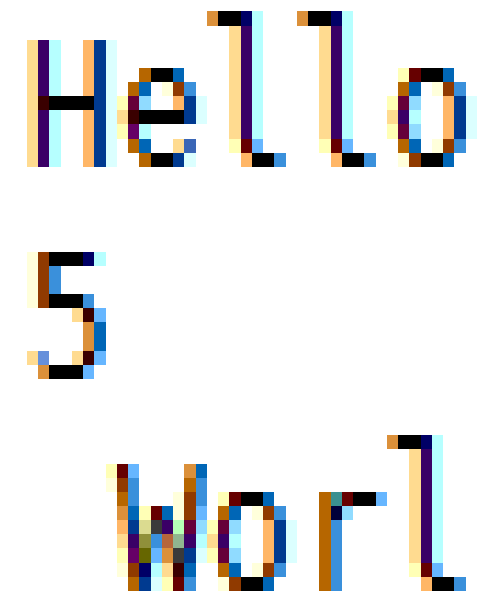
## ❖ Vị trí File trong Python

- Phương thức **tell()** nói cho bạn biết vị trí hiện tại bên trong file. Nói cách khác, việc đọc và ghi tiếp theo sẽ diễn ra trên các byte đó.
- Phương thức **seek(offset[, from])** thay đổi vị trí hiện tại bên trong file.
  - Tham số *offset* chỉ số byte để được di chuyển.
  - Tham số **from** xác định vị trí tham chiếu mà từ đó byte được di chuyển.
  - Nếu *from* được thiết lập là **0** nghĩa là sử dụng phần đầu file như là vị trí tham chiếu và **1** nghĩa là sử dụng vị trí hiện tại như là vị trí tham chiếu và nếu là **2** thì sử dụng phần cuối file như là vị trí tham chiếu.

# Ghi FILE

- Vị trí File trong Python

```
# Mở file
file = open("myfile2.txt", "r+") # Hello World
str = file.read(5)
print(str)
# Kiểm tra con trỏ hiện tại
vitri = file.tell()
print(vitri)
# Đặt lại vị trí con trỏ tại vị trí hiện tại
vitri = file.seek(0, 1)
str = file.read(5)
print(str)
# Đóng file
file.close()
```



Hello  
5  
Worl

# Ghi FILE

- Vị trí File trong Python

```
# Mở file
file = open("myfile2.txt", "rb+") # Hello World
str = file.read(5)
print(str)
# Kiểm tra con trỏ hiện tại
vitri = file.tell()
print(vitri)

vitri = file.seek(-5,2)
str = file.read(5)
print(str)
# Đóng file
file.close()
```

```
b'Hello'
5
b'World'
```

# Xóa FILE

```
import os
if os.path.exists("demofile.txt"):
    os.remove("demofile.txt")
else:
    print("The file does not exist")
```