

FILE HANDLING

PIITR

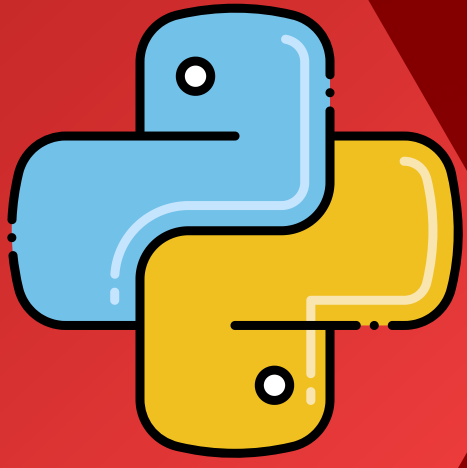
 7268941266

 8765310399

 piitroffice@ gmail.com

 Pragya Institute of IT & Research 

CLICK HERE



TOPIC

PYTHON FILE HANDLING

Opening Files

Reading Files

Writing Files

Appending Files

Closing Files

Working with Binary Files

Checking File Existence



PYTHON FILE HANDLING

Python provides a variety of built-in methods for handling files, allowing you to read, write, and manage files efficiently. Here's an overview of the key operations and concepts:

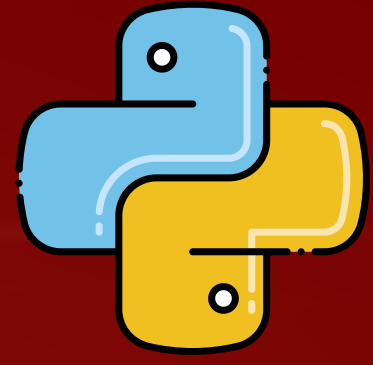
Opening Files

Python uses the `open()` function to open files. The syntax is:

```
file = open(filename, 'mode')
```

- 'r': Read (default mode). File must exist.
- 'w': Write. Creates a new file or truncates an existing file.
- 'x': Exclusive creation. Fails if the file exists.
- 'a': Append. Adds data to the end of the file.
- 'b': Binary mode.
- 't': Text mode (default).
- '+': Read and write.

Example:



```
n="Pragya Institute of IT & Research"
```

```
f=open("data.txt",'w')
```

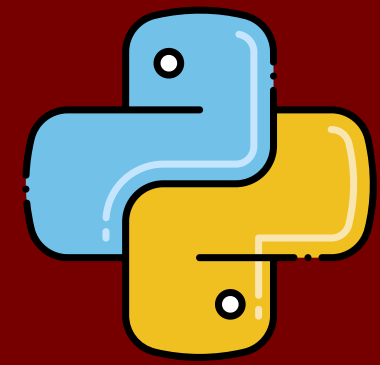
```
f.write(n)
```

```
print("your name is store:")
```

```
f.close()
```



Reading Files ✨



Read Entire File: ✨

```
f=open("data1.txt",'r')
print("your Information",f.read())
f.close()
```

Read Line by Line: ✨

```
f=open("data1.txt",'r')
print("your information",f.readline())
f.close()
```

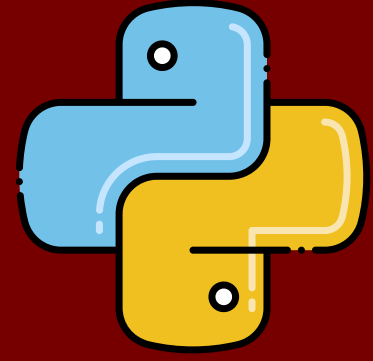
Read All Lines into a List ✨

```
f=open("data1.txt",'r')
print("your information",f.readlines())
f.close()
```

```
your information ['Pragya Institute OF IT &
Research']
```

```
>>>
```


✨ Writing Files ✨



★ Write String

```
f.write("Hello, world!")
```

Write Multiple Lines



```
f.writelines(["Line 1\n", "Line 2\n"])
```

```
f=open("data1.txt",'w')
```

```
f.writelines(["Line 1\n", "Line 2\n"])
```

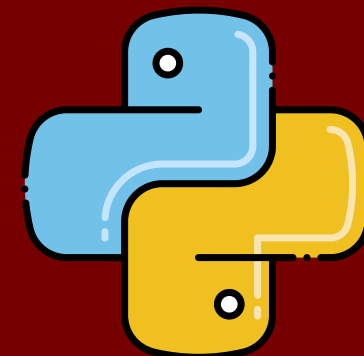
```
print("add data")
```

```
f.close()
```





Appending Files



You can add content to an existing file without overwriting it.

```
v="visit us : www.piitr.org | www.piitr.com"
```

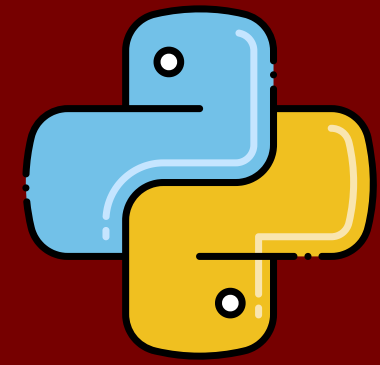
```
f=open("data1.txt",'a')
```

```
f.write(v)
```

```
print("add data")
```

```
f.close()
```





🌟 Closing Files

f.close()

Using a with block ensures the file is closed automatically.

Checking File Existence 🌟

```
import os
```

```
if os.path.exists("da.txt"):
```

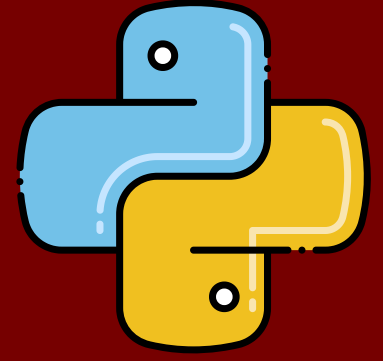
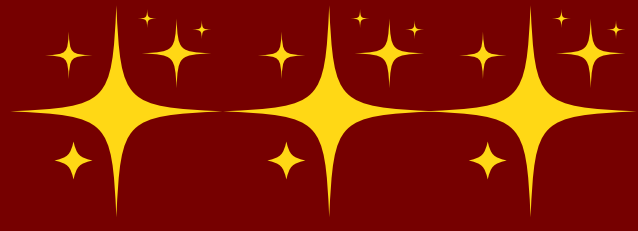
```
    print("File exists.")
```

```
else:
```

```
    print("no exists file")
```

A decorative yellow underline consisting of a wavy line, positioned below the code block.

Working with Binary Files



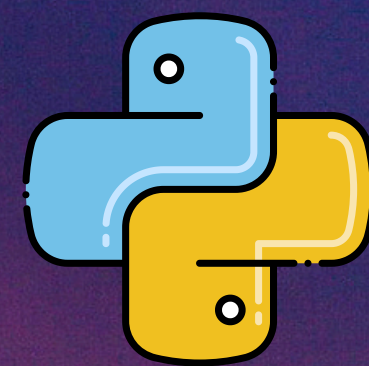
Working with binary files in Python involves reading or writing non-text data such as images, audio files, or other formats where data is stored in binary form. Python uses the rb (read binary) and wb (write binary) modes to handle binary files.

```
with open("example.jpg", "rb") as file:  
    binary_data = file.read()  
print(f"File size: {len(binary_data)} bytes")
```

File size: 140817 bytes

>>>

PYTHON FILE HANDLING



FUTURE

TECHNOLOGY



PIITR