

```
In [1]: #copy

In [2]: sfile=input("Enter Source File:")
sfo=open(sfile,"r")
(ffile = input("Enter Target File:"))
tf=open(ffile,"w")
tf.write(sf.read())
sf.close()
tf.close()
print("File Copied...")
fp=open("myfile1.txt")
print(fp.read())

File Copied...
Hello There
This is Python
Program
class days
today class about file handling

In [3]: # opening the file
file1 = open('myfile.txt', 'r')

# creating another file to store odd lines
file2 = open('myfile1.txt', 'w')

# reading content of the files
# and writing odd lines to another file
lines = file1.readlines()
print(type(lines))
for i in range(0, len(lines)):
    if(i % 2 != 0):
        file2.write(lines[i])

# closing the files
file1.close()
file2.close()

# opening the files and printing their content
file1 = open('myfile.txt', 'r')
file2 = open('myfile1.txt', 'r')

# reading and printing the files content
str1 = file1.read()
str2 = file2.read()

print("file1 content...")
print(str1)

print() # to print new line

print("file2 content...")
print(str2)

# closing the files
file1.close()
file2.close()

<class 'list'>
file1 content...
Hello There
This is Python
Program
class days
today class about file handling

file2 content...
This is Python
class days

In [6]: #merging two files:
file1 = open('myfile.txt', 'r')
file2 = open('myfile1.txt', 'r')
merge_file=open("merge_file.txt","w")
merge_file.write(file1.read())
merge_file.write(file2.read())
print("merged successfully")
file1.close()
file2.close()
merge_file.close()
merge_file=open("merge_file.txt","r")
print(merge_file.read())

merged successfully
Hello There
This is Python
Program
class days
today class about file handlingHello There
This is Python
Program
class days
today class about file handling

In [23]: fo=open("even_numbers.txt","r")
fo=open("odd_numbers.txt","r")
fm=open("merged_file_numbers.txt","w")

fe_lines=fe.readlines()
fo_lines=fo.readlines()
print(fe_lines)
print(fo_lines)
if (len(fe_lines)<len(fo_lines)):
    length=len(fe_lines)
    print("odd_file_length",length)
else:
    length=len(fo_lines)
    print("even_file_length",length)
for i in range(0,length):
    fm.write(fo_lines[i])
    fm.write(fe_lines[i])
    # fm.write("")
if(length < len(fe_lines)):
    fm.writelines(fe_lines[length:])
elif(length < len(fo_lines)):
    fm.writelines(fe_lines[length:])
else:
    pass

fm.close()
fo.close()
fe.close()

fm=open("merged_file_numbers.txt","r")
print(fm.read())

['2\n', '4\n', '6\n', '8\n', '10\n', '12\n', '14\n', '16']
['1\n', '3\n', '5\n', '7\n', '9']
even_file_length 5
1
2
3
4
5
6
7
8
910
12
14
16

In [27]: fo=open("even_numbers.txt","r")
fo=open("odd_numbers.txt","r")
fm=open("merged_file_numbers.txt","w")

fe_lines=fe.readlines()
fo_lines=fo.readlines()
if (len(fe_lines)<len(fo_lines)):
    length=len(fe_lines)
    print("odd_file_length",length)
else:
    length=len(fo_lines)
    print("even_file_length",length)
for i in range(0,length):
    fm.write(fo_lines[i].strip())
    fm.write("\n")
    fm.write(fe_lines[i].strip())
    fm.write("\n")
if(length < len(fe_lines)):
    fm.writelines(fe_lines[length:])
elif(length < len(fo_lines)):
    fm.writelines(fe_lines[length:])
else:
    pass

fm.close()
fo.close()
fe.close()

fm=open("merged_file_numbers.txt","r")
print(fm.read())
fm.close()

even_file_length 5
1
2
3
4
5
6
7
8
9
10
12
14
16

In [86]: fbo=open("byte.txt","r")
fbi=open("byte1.txt","w")

byte_lines=fb.readlines()
for i in range(0,len(byte_lines),4):
    byte_lines=byte_lines[i:i+4]
    l=[]
    #print(byte_lines)
    for k in range (0,len(byte_lines)-1,0-1,-1):
        byte_lines[k]=byte_lines[k].strip()
        l.append(byte_lines[k])
    #print(l)
    m=""
    m="".join(l)
    #print(m)
    fbi.write(m)
    fbi.write("\n")
fbi.close()
fbi=open("byte1.txt","r")
print(fbi.read())
fbi.close()

34201f21
99007436
b4234578
09

In [87]: fbo=open("byte.txt","r")
fbi=open("byte1.txt","w")

byte_lines=fb.readlines()
for i in range(0,len(byte_lines),4):
    byte_lines=byte_lines[i:i+4]
    l=[]
    #print(byte_lines)
    for k in range (0,len(byte_lines)):
        byte_lines[k]=byte_lines[k].strip()
        l.append(byte_lines[k])
    #print(l)
    m=""
    m="".join(l)
    #print(m)
    fbi.write(m)
    fbi.write("\n")
fbi.close()
fbi=open("byte1.txt","r")
print(fbi.read())
fbi.close()

211f2034
36749008
78452304
69

In [83]: #count words in two files
def count_words(filename):
    with open(filename, 'r') as file:
        content = file.read()
        words = content.split()
        return len(words)

# Count words in both files
word_count_file1 = count_words('myfile.txt')
word_count_file2 = count_words('merge_file.txt')

print(f"Word count in file1.txt: {word_count_file1}")
print(f"Word count in file2.txt: {word_count_file2}")

Word count in file1.txt: 25
word count in file2.txt: 13

In [ ]:

In [25]: def swap_files_content(file1, file2):
    with open(file1, 'r') as f1, open(file2, 'r') as f2:
        content1 = f1.read()
        content2 = f2.read()

    with open(file1, 'w') as f1, open(file2, 'w') as f2:
        f1.write(content2)
        f2.write(content1)

# Usage
swap_files_content('myfile.txt', 'merge_file.txt')

In [24]: def find_and_replace(file1, file2, search_string, replace_string):
    def replace_in_file(file_path):
        with open(file_path, 'r') as file:
            content = file.read()
            content = content.replace(search_string, replace_string)
        with open(file_path, 'w') as file:
            file.write(content)

    replace_in_file(file1)
    replace_in_file(file2)

# Usage
find_and_replace('myfile.txt', 'merge_file.txt', 'class', 'session')
with open('merge_file.txt', 'r') as file:
    print(file.read())

Hello There
This is Python
Program
session days
today session about file handlingHello There
This is Python
Program
session days
today session about file handling

Handling binary files in Python involves reading and writing data in binary format. Unlike text files, binary files are not human-readable and contain data in bytes.

In [3]: # Writing binary data to a file
data = b'\x00\x01\x02\x03\x04\x05' # Example binary data

with open('example.bin', 'wb') as file:
    file.write(data)
    print("Successfully writing into the file",data)

successfully writing into the file b'\x00\x01\x02\x03\x04\x05'

In [4]: # Reading binary data from a file
with open('example.bin', 'rb') as file:
    binary_data = file.read()
    print(binary_data)

b'\x00\x01\x02\x03\x04\x05'

In [6]: # Appending binary data to a file
more_data = b'\x06\x07\x08\x09'

with open('example.bin', 'ab') as file:
    file.write(more_data)
with open('example.bin', 'rb') as file:
    binary_data = file.read()
    print(binary_data)

b'\x00\x01\x02\x03\x04\x05\x06\x07\x08\x09'

In [7]: # Reading binary file line by line
with open('example.bin', 'rb') as file:
    for line in file:
        print(line)

b'\x00\x01\x02\x03\x04\x05\x06\x07\x08\t\x06\x07\x08\t'

In [10]: # Example binary data
data1 = b'\x00\x01\x02\x03\x04'
data2 = b'\x05\x06\x07\x08\x09'

# Write binary data to a file
with open('example.bin', 'wb') as file:
    file.write(data1)

# Append more binary data to the file
with open('example.bin', 'ab') as file:
    file.write(data2)

# Read the binary data from the file
with open('example.bin', 'rb') as file:
    binary_data = file.read()
    print("Binary data:", binary_data)
    print(type(binary_data))

# Convert binary data to a list of integers
integer_list = list(binary_data)
print("Integer list:", integer_list)

integer_list = tuple(binary_data)
print("Integer list:", integer_list)

Binary data: b'\x00\x01\x02\x03\x04\x05\x06\x07\x08\t'
<class 'bytes'>
Integer list: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
Integer list: (0, 1, 2, 3, 4, 5, 6, 7, 8, 9)

In [11]: # Writing list of integers as bytes
data = [1, 2, 3, 4, 5]
byte_data = bytearray(data)

with open('example_list.bin', 'wb') as file:
    file.write(byte_data)

In [15]: import pickle

# Data to be pickled
data = {
    'name': 'Alice',
    'age': 30,
    'scores': [95, 88, 92],
    'is_active': True
}

# Pickling the data
with open('data.bin', 'wb') as file:
    pickle.dump(data, file)

# Unpickling the data
with open('data.bin', 'rb') as file:
    loaded_data = pickle.load(file)
    print(loaded_data)

{'name': 'Alice', 'age': 30, 'scores': [95, 88, 92], 'is_active': True}

In [16]: import pickle

# Data to be pickled
data1 = {'name': 'Charlie', 'age': 40}
data2 = [1, 2, 3, 4, 5]
data3 = ('tuple', 'with', 'several', 'elements')

# Pickling multiple objects
with open('multiple_data.pkl', 'wb') as file:
    pickle.dump(data1, file)
    pickle.dump(data2, file)
    pickle.dump(data3, file)

# Unpickling multiple objects
with open('multiple_data.pkl', 'rb') as file:
    loaded_data1 = pickle.load(file)
    loaded_data2 = pickle.load(file)
    loaded_data3 = pickle.load(file)

print(loaded_data1)
print(loaded_data2)
print(loaded_data3)

{'name': 'Charlie', 'age': 40}
[1, 2, 3, 4, 5]
('tuple', 'with', 'several', 'elements')

In [18]: import csv

# Data to be written to CSV
data = [
    {'name': 'Alice', 'age': 30, 'city': 'New York'},
    {'name': 'Bob', 'age': 25, 'city': 'Los Angeles'},
    {'name': 'Charlie', 'age': 35, 'city': 'Chicago'}
]

# Field names (CSV header)
fieldnames = ['name', 'age', 'city']

# Writing to CSV file
with open('data.csv', 'w', newline='') as file:
    writer = csv.DictWriter(file, fieldnames=fieldnames)

    writer.writeheader() # Write the header
    writer.writerows(data) # Write the data rows
    print("Successfully")

successfully

In [8]: with open("myfile.txt", "r") as fp:
    # Read line number 3 to 5
    # Index starts from 0
    s = fp.readlines()[2:5]
    print(s)

['Program\n', 'class days\n', 'today class about file handling']

In [5]: upperCount =0
f=open("myfile.txt","r")
while(True):
    data=f.read(1)

    if(data==""):
        break
    if (ord(data) >= 65 and ord(data) <= 90):
        upperCount = upperCount +1
print(data,ends='')

print("Total Upper Case:",upperCount)

f.close()

Total Upper Case: 5

In [ ]:

In [ ]:

In [ ]:

```

