

fileHandlingQuickRevision

January 27, 2026

```
[ ]: # quick file handling examples

f=open("data.txt","r+") # open file for writing. r for reading
# r+ for reading and writing, a for appending
# x for creating a new file
# write lines
f.write("Hello, World!\n")
f.writelines(["Line 1\n", "Line 2\n", "Line 3\n"])

content=f.read() # read entire file
lines=f.readlines() # read lines into a list
line=f.readline() # read a single line

# seek() --> move file pointer to a specific position
f.seek(0) # move to beginning of file
f.seek(10) # move to byte 10
f.seek(10,2) # move to 10 bytes from end of file
f.seek(-10,2) # move to 10 bytes before end of file
# tell() --> get current file pointer position
position=f.tell()
print("Current position:",position)

# header=next(f) # get next line from file iterator
# file object is moved to next line after next()

# ex: header=next(f) --> first line - column names
f.close() # close the file
```

```
[5]: # another way to handle files using 'with' statement
with open("data.txt","r") as f:
    for line in f:
        print(line.strip()) # process each line
```

Hello, World!
Line 1
Line 2
Line 3

```
[ ]: # example of header
with open("data.txt","r") as f:
    header=next(f) # read first line
    print("Header:", header.strip())
    for line in f:
        print("Line:", line.strip())

# general data manipulation example
# like adding new fields, updating a field value in a file
# and write into a new file or overwrite existing file
new_lines=[]
with open("data.txt","r") as f:
    lines=f.readlines()
    for line in lines:
        data=line.strip().split(",") # assuming comma-separated values
        # example: add a new field
        data.append("NewField")
        new_lines.append(",".join(data)+"\n")

with open("data_updated.txt","w") as f:
    f.writelines(new_lines)
```

Header: Hello, World!

Line: Line 1

Line: Line 2

Line: Line 3

0.1 UNIVERSAL PATTERN

```
[10]: with open("input.txt","r") as infile,open("output.txt","w") as outfile:
        for line in infile:
            cols=line.strip().split(" ")
            # example: update a field value
            cols[1]="UpdatedValue" # update second field
            # transform data here as needed
            outfile.write(" ".join(cols)+"\n")
```

```
[ ]: # Modify a column value

# Example: increase age (index 1) by 1
cols[2] = str(int(cols[2]) + 1)
```

```
[13]: # Update based on condition

# Example: if gender is M, change to Male
if cols[2]=="M":
    cols[2]="Male"
```

```
# Add a new computed column
cols.append("India")
# or insert a position
cols.insert(3,"India")
```

```
[ ]: # Delete a particular field
```

```
del cols[2] # remove age field (index 2)
# or
cols.pop(2)
```

```
[ ]: # rearrange columns
```

```
cols=[cols[0],cols[3],cols[1],cols[2]] # new order
```

```
# select only required columns
cols=[cols[0],cols[2]]
```

```
# replace a column completely
fare=int(cols[3])
cols[3]=str(fare*1.1) # increase fare by 10%
```

```
[ ]: # skip rows
```

```
if int(cols[1])<21:
    continue
```

```
# add a calculated column
```

```
fare=int(cols[3])
if fare>500:
    cols.append("HighFare")
else:
    cols.append("LowFare")
```

```
[ ]: # delete a particular record
```

```
if cols[0]=="Rahul":
    continue # skip this record
```

Task	Shortcut
Read file	<code>`for line in f:`</code>
Split	<code>`cols = line.split()`</code>
Modify	<code>`cols[i] = value`</code>
Add	<code>`append / insert`</code>
Delete column	<code>`del cols[i]`</code>
Rearrange	<code>`cols = [...]`</code>
Filter rows	<code>`continue`</code>
Write	<code>`" ".join(cols)`</code>

0.2 Example

Increase Age by 1

Convert Gender

M → Male

F → Female

Add a new column Fare_INR

Fare + 10% tax

Delete records where Age < 21

Rearrange columns to this order:

ID Name Gender Age Fare Fare_INR

Write the output to output.txt

Include a header in the output file

```
[ ]: # Answer

with open("data.txt", "r") as fin, open("output.txt", "w") as fout:
    # write header
    header=next(fin)
    fout.write("ID Name Gender Age Fare Fare_INR\n")

    for line in fin:
        cols = line.strip().split()

        # 1 increase age by 1
        cols[2] = str(int(cols[2]) + 1)

        # 2 remove age < 21
        if int(cols[2]) < 21:
            continue

        # 3 gender conversion
        if cols[3] == "M":
            cols[3] = "Male"
        elif cols[3] == "F":
            cols[3] = "Female"
```

```
# 4 add Fare_INR (fare + 10%)
fare = int(cols[4])
fare_inr = fare * 1.10
cols.append(str(fare_inr))

# 5 rearrange columns
cols = [cols[0], cols[1], cols[3], cols[2], cols[4], cols[5]]

# write result
fout.write(" ".join(cols) + "\n")
```