Read File

Zijing

Reading from file steps

Reading from a file involves three main steps:

- **Step 1**: Open the file. The general syntax is:

  open(<file name>, <mode>)

- Mode:
  1. "r": reads from existing file

```
# open existing file to read data
# Step 1: open file with mode "r"
file=open("fileName.txt", "r")
```

- **Step 2**: Read data from the file. The general syntax is:

  <file object>.read() # read whole file
  <file object>.readlines() # read line by line
  <file object>.readline() # read a single line

```
# Step 1: open file with mode "r"
file=open("fileName.txt", "r")
# Step 2: read data from file
string =file.read()
# or we can use readlines()
string =file.readlines()
# or we can use readline()
string =file.readline()
```

- **Step 3**: Close the file. The general syntax is:

  <file object>.close()

```
# Step 1: open file with mode "r"
file=open("fileName.txt", "r")
# Step 2: read data from file
string=file.read ()
# Step 3: Close the file
file.close()
```

Examples

Write a python program to reads data from files using **read()** and **readlines()**.

Example: This program opens a file (test.txt) to read and then display the string 'text'.

```
# Open the file to read
fileName = "test.txt"
inFile = open(fileName, "r")
# Read the whole file and return a single string
text = inFile.read()
# Have read the file. Can close it
inFile.close()
# Display the string 'text'.
print(text)
>>>
This is line 1
This is line 2
This is line 3
```

Example: Write a Python program that reads the whole file and return a list of strings, one string for each line, including the newline character.

```
# Open the file to read
fileName = "test.txt"
inFile = open(fileName, "r")
# Read the whole file and return a list of strings
lines = inFile.readlines()
# Have read the file. Can close it
inFile.close()
# Display the list 'lines' for inspection
print(lines)
```

Examples

Write a python program to reads data from files using **readline()** function.

Example: Write a Python program that reads one line at a time. When we reach the end of the file, it returns an empty string. To loop through file lines, we need to use `while` loop.

```
# Open the file to read
fileName = "test.txt"
inFile = open(fileName, "r")
while True:
    line = inFile.readline()
        if line == "":
            break
    print(line)
# Close the file
inFile.close()
```

Example: Write a Python program that reads products information. Each line represents one product ID, name and prices. Use **strip()** function to split each line based on space character. The **strip()** function removes any spaces or specified characters at the start and end of a string. To loop through file lines, we need to use `while` loop.

```
filename = "products.txt"
file = open(filename)
while True:
    line = file.readline().strip()
        if line == "":
            break
    productID = line
    name = file.readline().strip()
    price = float(file.readline().strip())
    file.readline()
    print(productID, name, price)
file.close()
```

Examples

Write a python program to reads data from files using **readline()**, **strip()** and **split()** functions.

Example: Write a Python program that reads authors information which separated by semicolon (;). Each line represents ID, name and hobbies. Use **strip()** and **split()** functions to split each line based on space character. The **strip()** function removes any spaces or specified characters at the start and end of a string. The **split()** function splits the string at the specified separator, and returns a list.
To loop through file lines, we need to use `while` loop.

```
fname = "authors.txt"
file = open(fname)
    while True:
    line = file.readline().strip()
    if line == "":
        break
    tokens = line.split(";")
    ID = tokens[0].strip()
    name = tokens[1].strip()
    hobbies = tokens[2].strip()
    print(ID)
    print(name)
    tokens = hobbies.split(",")
    for s in tokens:
        hobby = s.strip()
        print(hobby)
file.close()
```

Write File

Zijing

Writing to file steps

Writing to a file involves **three** main steps:

- **Step 1**: Open the file. The general syntax is:

  open(<file name>, <mode>)

- Mode:
  1. "**w**": creates a new file for writing
  2. "**a**": appends data into existing file

```
# create a new file for writing
# Step 1: open file with mode "w"
file=open("fileName.txt", "w")

# writ (append) to existing file
# Step 1: open file with mode "a"
file=open("fileName.txt", "a")
```

- **Step 2**: Write to the file. The general syntax is:

  <file object>.write(<string>)

```
# Step 1: open file with mode "w"
file=open("fileName.txt", "w")
# Step 2: write to file
file.write (string)
# Step 1: open file with mode "a"
file=open("fileName.txt", "a")
# Step 2: write to file
file.write (string)
```

- **Step 3**: Close the file. The general syntax is:

  <file object>.close()

```
# Step 1: open file with mode "w"
file=open("fileName.txt", "w")
# Step 2: write to file
file.write (string)
# Step 3: Close the file
file.close()
# Step 1: open file with mode "a"
file=open("fileName.txt", "a")
# Step 2: write to file
file.write (string)
# Step 3: Close the file
file.close()
```

Examples

Write a Python program to write data to file.

Example: This program opens a file (test.txt) and writes 5 lines of text to it.

```
# open a file to write to
outfile = open("test.txt", "w")
# write 5 lines
outfile.write("This is line 1\n")
outfile.write("This is line 2\n")
outfile.write("This is line 3\n")
outfile.write("This is the last line\n")
# close the file
outfile.close()
```

Example: This program opens a file (test.txt) and writes 5 lines of text to it. We can use `for` loop or `while` loop.

```
# open a file to write to
outfile = open("test.txt", "w")
# write 5 lines
for i in range(5):
    outfile.write("This is line {}".format(i),"\n")
# close the file
outfile.close()
```

Example: This program opens a file (test.txt) and asks user to writes "**n**" lines of text to it. Use `while` loop.

```
# open a file to write to
outfile = open("test.txt", "w")
n= int(input("Enter number of lines: "))
i=0
while i < n:
    outfile.write("This is line {}".format(i),"\n")
    i=i+1
# close the file
outfile.close()
```

Examples

Write a python program to append text (data) into files.

Example: Write a Python program to append text to existing file. We assume the file exist and might have some data (text).

```
# Open the file to append text
fileName = "test.txt"
outFile = open(fileName, "a")
# Add two more lines
outFile.write("Extra line 1\n")
outFile.write("Extra line 2\n")
# Close the file
outFile.close()
```

Example: Write a Python program that asks the user to enter data and then append it to existing file. We assume the file exists and might have some data (text). To read from user, we can use `while()` or `for()` loops and the `input()` function.

```
# Open the file to append text
fileName = "test.txt"
outFile = open(fileName, "a")
while True:
    data= input ("Enter data: ")
    # Add one line
    outFile.write(data)
    outFile.write("\n")
    s=input ("Do you want to stop? (y/n):")
    if (s=='y' or s=='Y'):
        break
# Close the file
outFile.close()
```