# Week 7 - Exceptions & Files

You might notice that no record is left after the user exits from the application then starts it again. That's because all the variables in Python are stored in the memory (記憶體). Once a program terminates, it releases the memory space it allocated. Our application is far from useful. You can't ask the user to run the application for days, months, or even years. Even if a user is willing to run it forever, computers power off accidentally from time to time.

However, things stored in files stay there no matter how many times the computer restarts, unless the hard disk (硬碟) is broken. This leads to our solution: **write the records to a file before the program stops and read from it when the program restarts**.

$ python3 pymoney.py

How much money do you have? 1000

What do you want to do (add / view / delete / exit)? add

Add some expense or income records with description and amount:

desc1 amt1, desc2 amt2, desc3 amt3, ...

breakfast -50

What do you want to do (add / view / delete / exit)? add

Add some expense or income records with description and amount:

desc1 amt1, desc2 amt2, desc3 amt3, ...

lunch -70

What do you want to do (add / view / delete / exit)? add

Add some expense or income records with description and amount:

desc1 amt1, desc2 amt2, desc3 amt3, ...

salary 3500

What do you want to do (add / view / delete / exit)? view

Here's your expense and income records:

Description Amount

==================== ======

breakfast -50

lunch -70

salary 3500

===========================

Now you have 4380 dollars.

What do you want to do (add / view / delete / exit)? exit

$ python3 pymoney.py

Welcome back!

What do you want to do (add / view / delete / exit)? view

Here's your expense and income records:

Description Amount

==================== ======

breakfast -50

lunch -70

salary 3500

===========================

Now you have 4380 dollars

What do you want to do (add / view / delete / exit)? add

Add some expense or income records with description and amount:

desc1 amt1, desc2 amt2, desc3 amt3, ...

breakfast -50

What do you want to do (add / view / delete / exit)? delete

Which record do you want to delete? design your own way to specify "lunch -70"

What do you want to do (add / view / delete / exit)? add

Add some expense or income records with description and amount:

desc1 amt1, desc2 amt2, desc3 amt3, ...

dinner -120

What do you want to do (add / view / delete / exit)? exit

$ python3 pymoney.py

Welcome back!

What do you want to do (add / view / delete / exit)? view

Here's your expense and income records:

Description Amount

==================== ======

breakfast -50

salary 3500

breakfast -50

dinner -120

===========================

Now you have 4280 dollars.

What do you want to do (add / view / delete / exit)? exit

Since your code is getting longer and more complicated, many things can go wrong and crash your program. Besides some mistakes you might make in your code, more errors can occur as the user inputs something out of your expectation. Now that we included file operations in our program, we also have to consider things happen outside the program: files created by the program being modified or deleted outside the program, for example.

Here we list some exceptions that may occur in our current program:

* (1) When prompted for the initial amount of money, the user inputs a string that cannot be converted to integer.
* (2) When prompted for a command (add / view / delete / exit), the user inputs a string that is not one of the four above.
* When prompted to add a record,
  + (3) The user inputs a string that does not follow the format, or
  + (4) the second string of a record, after splitting, cannot be converted to an integer.
* When prompted to delete a record,
  + (5) the user inputs in an invalid format in respect of your design, or
  + (6) the specified record does not exist.
* When loading the records from the file,
  + (7) the file does not exist,
  + (8) no line is in the file,
  + (9) the first line cannot be interpreted as initial amount of money (i.e. cannot be converted to integer), or
  + (10) any of the other lines cannot be interpreted as a record (i.e. cannot be split into a list of two strings or the second string after splitting cannot be converted to integer).

$ python3 pymoney.py

How much money do you have? abc

Invalid value for money. Set to 0 by default.

What do you want to do (add / view / delete / exit)? hello

Invalid command. Try again.

What do you want to do (add / view / delete / exit)? add

Add some expense or income records with description and amount:

desc1 amt1, desc2 amt2, desc3 amt3, ...

salary3000

The format of a record should be like this: breakfast -50.

Fail to add a record.

What do you want to do (add / view / delete / exit)? add

Add some expense or income records with description and amount:

desc1 amt1, desc2 amt2, desc3 amt3, ...

salary 3500

What do you want to do (add / view / delete / exit)? add

Add some expense or income records with description and amount:

desc1 amt1, desc2 amt2, desc3 amt3, ...

breakfast -abc

Invalid value for money.

Fail to add a record.

What do you want to do (add / view / delete / exit)? view

Here's your expense and income records:

Description Amount

==================== ======

salary 3500

===========================

Now you have 3500 dollars.

What do you want to do (add / view / delete / exit)? delete

Which record do you want to delete? invalid in respect to your design

Invalid format. Fail to delete a record.

What do you want to do (add / view / delete / exit)? delete

Which record do you want to delete? specify a record that doesn't exist

There's no record with xxxxxxxx. Fail to delete a record.

What do you want to do (add / view / delete / exit)? exit

(modify the content of records.txt to make it invalid)

$ python3 pymoney.py

Invalid format in records.txt. Deleting the contents.

How much money do you have? 1000

What do you want to do (add / view / delete / exit)? add

Add some expense or income records with description and amount:

desc1 amt1, desc2 amt2, desc3 amt3, ...

breakfast -50

What do you want to do (add / view / delete / exit)? view

Here's your expense and income records:

Description Amount

==================== ======

breakfast -50

===========================

Now you have 950 dollars.

What do you want to do (add / view / delete / exit)? exit

## Required Steps

1. Before the program terminates, write the initial amount of money and records into a file named **'records.txt'** in a **with-as** statement.
   1. Use the **write()** method to write the initial amount of money into the file.
   2. Convert the data structure storing the records into a list of strings and use the **writelines()** method to write the list of strings into the file.
   3. Put new line characters properly so that you can read the file by **readline()** and **readlines()** later.
2. At the beginning of the program, try to open 'records.txt' by calling the **open()** function in a **try-except** statement.
   1. If the file exists, use the **readline()** method to read the first line, which is the initial amount of money, into a variable. Then use the **readlines()** method to read the records and build the data structure you used before (e.g. list of tuples, dictionary, class, etc.) to store them. Remember to **close()** the file after reading.
   2. If the file doesn't exist (i.e. a **FileNotFoundError** is raised), prompt the user for the initial amount of money and initialize the variables needed in the coming operations.
3. Handle 10 possible exceptions apart from the already handled **FileNotFoundError**.
   1. Use **try-except** (and probably **try-except-finally**) statements.
   2. Specify the error type if possible. (e.g. except **ValueError**:)
   3. Use **sys.stderr.write()** to report some error messages to the user if necessary.
   4. Prompt the user again or set related variables directly, depending on the situation.
4. Add appropriate comments for your code.

## Notes

* In the 10 exceptions we list above, "the file does not exist" is not really an unexpected condition. It inevitably happens when this application runs for the first time. Therefore, it doesn't make sense to report an error using **sys.stderr.write()**. You can simply assume that there is no record at all and prompt the user for the initial amount of money.
* In some cases, you may use the **if-else** statement instead, if it's more reasonable (and more convenient) than using the **try-except** statement.

## Related Knowledge

* File operations
  + open() function
  + readline() and readlines() method
  + write() and writelines() method
  + close() method
  + with-as statements
  + Stand error output (sys.stderr)
* Exception handling
  + try-except and try-except-finally statements
  + Common error types