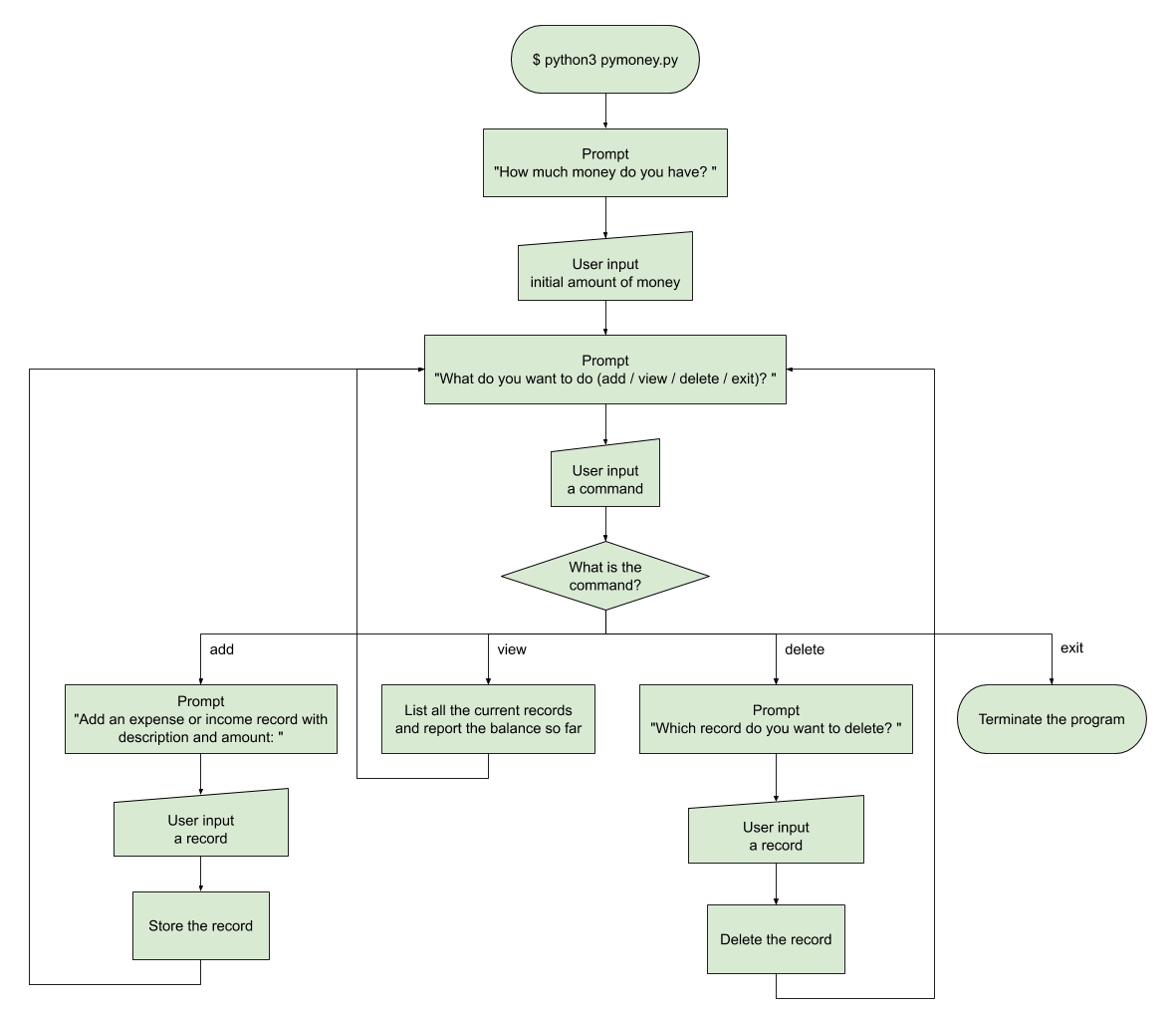
# Week 6 - Control Constructs

Currently our application only has one flow: user entering initial amount of money, user entering the records, program listing the records, and program reporting the balance.

Most of the time, the user should be able to decide what to do next. Let's provide 4 basic commands which the user can execute at any time.

* Add a record
* View the records and the balance
* Delete a record
* Exit the application

After asking for the initial amount of money, the program should keep prompting the user for one of the above commands until the user chooses to exit. The following flow chart illustrates the overall flow of the program.



$ 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, ...

dinner -100

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

Here's your expense and income records: you may design your own printing format

Description Amount

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

breakfast -50

lunch -70

dinner -100

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

Now you have 780 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)? 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: you may design your own printing format

Description Amount

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

breakfast -50

lunch -70

dinner -100

breakfast -50

salary 3500

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

Now you have 4230 dollars.

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

Which record do you want to delete? design your own way to specify the "breakfast -50" record between "dinner -100" and "salary 3500"

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

Here's your expense and income records: you may design your own printing format

Description Amount

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

breakfast -50

lunch -70

dinner -100

salary 3500

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

Now you have 4280 dollars.

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

## Required Steps

1. Prompt the user for the initial amount of money.
2. Prepare a data structure (e.g. list of tuples, dictionary, class, etc.) to store the records.
3. Create a **while** loop. In the **while** loop,
   1. Prompt the user for a command.
   2. Create a **if-elif-else** statement to handle different commands.
      1. Handle the "add" command.
      2. Handle the "view" command. Try to print the records in a neat format.
      3. Handle the "delete" command. You should also remove the amount of this record when calculating the balance (e.g. 50 is added back to the balance after deleting "breakfast -50").
      4. Handle the "exit" command if necessary.
   3. Leave the **while** loop if the command is "exit".
4. Add appropriate comments for your code.

## Think and Solve

You might run into a question when designing the "delete" command: how should the user specify which record to delete?

Apparently by saying "breakfast" or "breakfast -50" is not enough because there are possibly more than one records with the same description and amount. It's not feasible to assume that the user is deleting the first, last, or all records as "breakfast -50".

As a software developer, you need to come up with a solution and implement it in your code.

## Related Knowledge

* while loop
* if-elif-else statements
* list, tuple, dictionary and their methods
* List comprehension
* enumerate() function
* str.split() and str.join() methods
* string formatting