

7_user input and while loop

- Introduction
- Input function
 - How it works
 - input a number
- While loops
 - multiple conditions
 - exit the loop
 - restart the loop
- while loop with lists and dictionaries
 - move items in one list to another list.
 - remove repeated value
 - input to dictionaries

Introduction

Our programs need to get information from the user. In this section we will cover how users input their data to the programs.

We will also introduce while loop, which ensures the program keeps running as long as user want them to.

Input function

How it works

```
message=input("write down the message you want to repeat")
print(message)
```

in the example, "write down the message you want to repeat" is the prompt that will show up at the terminal. a variable is associated with the user's input to be further manipulated.

you may want to try your program on terminal to "pretend" a user.

you can set a variable for your prompt to write longer prompt.

input a number

Python understand everything our users input as strings, which forbids us from doing any mathematical operation. Therefore we need to set up a way for users to enter numbers.

```

weight=input('enters your weight in kg below:')
weight=int(weight)
#int() converts the variable inside into integers. for floating number, use float()
if weight>=100
    print('you are overweight')
else:
    print('you are so slim')
#input:120
#output:you are overweight

```

While loops

The while loop runs as long as, or while, a certain condition is true.

```

prompt = 'Enter anything and I will respond with swearing.\nType "shut up" to quit
the program. '
message = ""
while message != 'shut up':
    message = input(prompt)
    print(f'Stop saying {message}, shit!')
# we need to initialize the message variable before entering the while loop for it
to function. Later, it receives the value user inputted and replaces the old one.

```

multiple conditions

In while loops, the program will stop running when a certain condition fails. However, there are circumstances where the program needs to stop as long as one of many conditions fails. The solution is to set a variable called flag.

```

prompt='are you alive?'
alive=True
while alive:
    message=input(prompt)
    if message=='suicide':
        alive=False
    else:
        print('you are still alive')
# in this example, alive is the 'flag'. as long as alive is still true, the program
runs. by using if statement, our flag could become false in many cases. whereas the
only thing while loop needs to do is to check if the flag is true.

```

exit the loop

```

while True:
    name=input('please write your name below')
    if name=='quit':
        break
    else:
        print(f'congratulations,{name}')
# the break statement will exit the loop immediately once the condition is met. you
can break any loop with break.

```

restart the loop

instead of leaving the rest of the code unexecuted, one can use continue to go back to the start of the loop

```

compliments = ['genius', 'astonishing', 'generous', 'great', 'reputable', 'elegant']
n = 1
while True:
    name = input('Write your name and we will brag you: ')
    if name == 'ok that is enough':
        break
    elif name=='not enough':
        compliment = compliments[n]
        print(f'you are also {compliment}')
        n+=1
        if n==len(compliments):
            n=0
        continue
    else:
        print(f'{name} are genius!')
#here i build a "compliment machine" where one can type the name and receive a
compliment, type "not enough" to recieve more and type "ok that is enough to end the
program"
# everytime the user type 'not enough',the program will print a new compliment and
go back to the start of the loop where n can increase by 1.

```

while loop with lists and dictionaries

work with many pieces of information at one time

move items in one list to another list.

```

man_city=['aguero','haaland','cancelo','bernado']
liverpool=[]
while man_city:
    new_player=man_city.pop()
    liverpool.append(new_player)
for players in liverpool:
    print(f'{players} has been added to Liverpool ')
#bernado has been added to Liverpool
#cancelo has been added to Liverpool
#haaland has been added to Liverpool
#aguero has been added to Liverpool

```

remove repeated value

```

cute_animals=['cat','dog','snake','snake','alpaca','snake','panda']
# we want to remove all the 'snake' in the list
while 'snake' in cute_animals:
    cute_animals.remove('snake')
print (cute_animals)
#['cat', 'dog', 'alpaca', 'panda']

```

input to dictionaries

```

dream={}
while True:
    name=input('Hi, what is your name')
    job=input('what you want to become')
    dream[name]=job
    repeat=input('someone else to answer(yes/no)?')
    if repeat=='no':
        break
for name,job in dream.items():
    print(f'{name} want to become {job} in the future')

```