7. User Input and While Loops

How the input() function works

In [3]:

```
message = input("Tell me something, and I will repeat it back to you: ")
print(message)
```

```
Tell me something, and I will repeat it back to you: Hi
Hi
```

- the **input() function** pauses the program and waits for the user to enter some text
- once the program receives the user's input it stores it in a variable (message)

In [4]:

```
name = input("Please enter your name: ")
print("Hello, " + name + "!")
```

```
Please enter your name: Ugur
Hello, Ugur!
```

- The promt message should be clear, easy-to-follow
- which tells the user exactly what kind of information it needs
- add space after the promt

In [5]:

```
promt = "If you tell us who you are, we can personalize the messages you see."
promt += "\nWhat is your first name? "

name = input(promt)
print("\nHello, " + name + "!")
```

```
If you tell us who you are, we can personalize the messages you see.
What is your first name? Ugur

Hello, Ugur!
```

- Here we biuld a multi-line string
- the first line is the part of the message
- the "**+=**" operator take the string that was stored in "**promt**" and adds the new string onto the end
- now we have two lines

In [6]:

```
age = input("How old are you? ")
```

```
How old are you? 30
```

In [7]:

```
type(age)
```

Out[7]:

```
str
```

- the input function interprets everything as **string**

In [ ]:

```

```

In [8]:

```
age = input("How old are you? ")
age = int(age)
```

```
How old are you? 30
```

In [9]:

```
type(age)
```

Out[9]:

```
int
```

- we can resolve this ussue by using the **int() function**
- which tells python to threat the input as a numerical value

In [10]:

```
height = input("How tall are you, in cm? ")
height = int(height)

if height >= 150:
    print("\nYou are tall enough to ride!")
else:
    print("\nYou are too small buddy!")
```

```
How tall are you, in cm? 184

You are tall enough to ride!
```

- an example for using the input in a program

In [11]:

```
4%3
```

Out[11]:

```
1
```

In [12]:

```
5%3
```

Out[12]:

2

In [13]:

```
6%3
```

Out[13]:

0

In [14]:

```
7%3
```

Out[14]:

1

- the **modulo operator** doesn't tell you how many times one number fits into another
- it just tells you what the remainder is
- when a number is divisible by another number, the remainder is 0
- you can use this fact to determine if a number is even or odd:

In [15]:

```
number = input("Enter a number, and I'll tell you if it's even or odd: ")
number = int(number)

if number % 2 == 0:
    print("\nThe number " + str(number) + " is even!")
else:
    print("\nThe number " + str(number) + " is odd!")
```

```
Enter a number, and I'll tell you if it's even or odd: 2

The number 2 is even!
```

- **even numbers are always divisible by two**
- so if the modulo of a number and two is zero (the conditional test is true)
  - the number is even
- else:
  - the number is odd

Tasks

- 7-1. Rental Car: Write a program that asks the user what kind of rental car they would like. Print a message about that car, such as “Let me see if I can find you a Subaru.”
- 7-2. Restaurant Seating: Write a program that asks the user how many people are in their dinner group. If the answer is more than eight, print a message saying they’ll have to wait for a table. Otherwise, report that their table is ready.
- 7-3. Multiples of Ten: Ask the user for a number, and then report whether the number is a multiple of 10 or not.

In [75]:

```
# 7.1
prompt = "What car do you want to drive? "
car = input(prompt)
print("\nLet me see if I can find a " + car + ".")
```

```
What car do you want to drive? Opel

Let me see if I can find a Opel.
```

In [79]:

```
# 7.2
prompt = "How many people are in the dinner group? "
number = input(prompt)
number = int(number)
if number > 8:
    print("You have to wait for a table!")
else:
    print("Your table is ready!")
```

```
How many people are in the dinner group? 9
You have to wait for a table!
```

In [83]:

```
#7.3
prompt = "Say a number: "
number = input(prompt)
number = int(number)
if number % 10 == 0:
    print("The number: " + str(number) + " is a multiple of 10!")
else:
    print("The number: " + str(number) + " is not a multiple of 10!")
```

```
Say a number: 30
The number: 30 is a multiple of 10!
```

Introducing while Loops

In [16]:

```
current_number = 1
while current_number <= 5:
    print(current_number)
    current_number += 1
```

```
1
2
3
4
5
```

- the *for loop* takes a collection of items and executes a block of code once for each item
- the *while loop* runs as long as, or "while", a certain condition is *true*
- we have a variable which is 1
- while the conditional test is true (means when the number is <= 5 which makes it true)
- we have a loop which does here 2 things:
  - while the number is <= 5 we print the current number
  - and the increment the current_number with 1
- so at the end we reach 6 which is **not** <= 5 and the conditional test will get **False** so the loop stops
- if we don't have the last statement, we will have a loop which will go for ever, because the number would stay 1 all the time (no increment, which will not make the statement False, which will be the reason fo looping for ever) --> "endless loop"

In [24]:

```
promt = "\nTell me something, and I will repeat it back to you:"
promt += "\nEnter 'quit' to end the program. "

message = ""
while message != 'quit':
    message = input(promt)
    print(message)
```

```
Tell me something, and I will repeat it back to you:
Enter 'quit' to end the program. hallo
hallo

Tell me something, and I will repeat it back to you:
Enter 'quit' to end the program. hallo2
hallo2

Tell me something, and I will repeat it back to you:
Enter 'quit' to end the program. quit
quit
```

- **letting the user choose when to quit**
- we defined two promts, which tells us what to do
- we have a empty message
- while the user **doesn't** entet **'quit'** the while loop is true and does (means the message is not 'quit'):
  - store the input from the promt in the variable message

In [26]:

```
promt = "\nTell me something, and I will repeat it back to you:"
promt += "\nEnter 'quit' to end the program. "

message = ""
while message != 'quit':
    message = input(promt)

    if message != 'quit':
        print(message)
```

```
Tell me something, and I will repeat it back to you:
Enter 'quit' to end the program. hallo
hallo

Tell me something, and I will repeat it back to you:
Enter 'quit' to end the program. hallo2
hallo2

Tell me something, and I will repeat it back to you:
Enter 'quit' to end the program. quit
```

- now the program makes a quick test before printing out the 'quit' which is not a message, just a command
- if the user does **not** enter quit the message will be displayed (which is true whenever the user enters something else than 'quit'

In [27]:

```
promt = "\nTell me something, and I will repeat it back to you:"
promt += "\nEnter 'quit' to end the program. "

active = True
while active:
    message = input(promt)

    if message == 'quit':
        active = False
    else:
        print(message)
```

```
Tell me something, and I will repeat it back to you:
Enter 'quit' to end the program. hi
hi

Tell me something, and I will repeat it back to you:
Enter 'quit' to end the program. quit
```

- **using a flag**
- we set the variable active to True, so the program starts in active state
- the while statement has no condition means it runs
- as long as the active variable stays True the loop will continue
- if the if statement inside the loop is True (which is True when the user enters 'quit' the active variable will be False
- otherwise the program will print the message
- in the previous example we did the conditional test directly inside the while statemnent
- but now we have a flag, which gives us the same result but the flag variable is an indicator for running the while loop (making it True)
- we can now build more conditions inside the while loop one which will be useful for complicated programs, in which there may be many events that should make the program stop running
- when any of these events causes the active flag to become False, the main variable will be false and the while loop will stop

In [28]:

```
promt = "\nPlease enter the name of a city you visited:"
promt += "\n(Enter 'quit' when you are finished. "

while True:
    city = input(promt)

    if city == 'quit':
        break
    else:
        print("I'd love to go to " + city.title() + "!")
```

```
Please enter the name of a city you visited:
(Enter 'quit' when you are finished. Prag
I'd love to go to Prag!

Please enter the name of a city you visited:
(Enter 'quit' when you are finished. quit
```

- **using break to exit a loop**
- a loop with **while True** will run forever unless it reaches a break statement
- which will be the case for, if the conditional test for city == 'quit' is True

In [31]:

```
current_number = 0
while current_number < 10:
    current_number += 1
    if current_number % 2 == 0:
        continue

    print(current_number)
```

```
1
3
5
7
9
```

- **using continue in a loop**
- continue statment will return to the beginning of the loop based on the result of the conditional test
- we have a variable number stored
- our while loop is True untill we reach 10 and does
    - incrementing the current number + 1
    - has a conditional test which is true when we have odd numbers (odd numbers we have when the modulo of the number is equal to 0)
        - our if statement will continue if its True means
        - it will return to the beginning of the loop
- our if statement is True means it will ignore the rest and go to the beginning (which is case when the number is even)
- when the if statemen is False it will not continue, means it will print the odd numbers

In [65]:

```
numbers = range (0, 11)
for number in numbers:
    if number % 2 == 0:
        print("The equation: " + str(number) + " / 2 is = " + str(number / 2 ))
        print(str(number) + " is a even number")
        print("So " + str(number) + " %2 is 0!\n")

    else:
        print("The equation: " + str(number) + " / 2 is " + str(number / 2 ))
        print(str(number) + " is an odd number")
        print("So " + str(number) + " % 2 is 1!\n")
```

```
The equation: 0 / 2 is = 0.0
0 is a even number
So 0 %2 is 0!

The equation: 1 / 2 is 0.5
1 is an odd number
So 1 % 2 is 1!

The equation: 2 / 2 is = 1.0
2 is a even number
So 2 %2 is 0!

The equation: 3 / 2 is 1.5
3 is an odd number
So 3 % 2 is 1!

The equation: 4 / 2 is = 2.0
4 is a even number
So 4 %2 is 0!

The equation: 5 / 2 is 2.5
5 is an odd number
So 5 % 2 is 1!

The equation: 6 / 2 is = 3.0
6 is a even number
So 6 %2 is 0!

The equation: 7 / 2 is 3.5
7 is an odd number
So 7 % 2 is 1!

The equation: 8 / 2 is = 4.0
8 is a even number
So 8 %2 is 0!

The equation: 9 / 2 is 4.5
9 is an odd number
So 9 % 2 is 1!

The equation: 10 / 2 is = 5.0
10 is a even number
So 10 %2 is 0!
```

- **excursion**
- made a program which explains modulo and remainder

In [66]:

```
x = 1
while x <= 5:
    print(x)
    x += 1
```

```
1
2
3
4
5
```

- **avoiding infinite loops**
- this will run from 1 to 5
- because we have the increment which counts to 5
- otherwise it would count for ever because it would be true for ever
- we need a condition which makes the while loop false
- this will be the case after 5
- we reach that just by incrementing the x variable

Tasks

- 7-4. Pizza Toppings: Write a loop that prompts the user to enter a series of pizza toppings until they enter a 'quit' value. As they enter each topping, print a message saying you'll add that topping to their pizza.
- 7-5. Movie Tickets: A movie theater charges different ticket prices depending on a person's age. If a person is under the age of 3, the ticket is free; if they are between 3 and 12, the ticket is 10 dollar; and if they are over age 12, the ticket is 15 dollar. Write a loop in which you ask users their age, and then tell them the cost of their movie ticket.
- 7-6. Three Exits: Write different versions of either Exercise 7-4 or Exercise 7-5 that do each of the following at least once:
  - Use a conditional test in the while statement to stop the loop.
  - Use an active variable to control how long the loop runs.
  - Use a break statement to exit the loop when the user enters a 'quit' value.
- 7-7. Infinity: Write a loop that never ends, and run it. (To end the loop, press ctrl-C or close the window displaying the output.)

In [86]:

```
#7.4
promt = "\nTell me, which Toppings you like: "
promt += "\nEnter 'quit' to end the program. "

message = ""
while message != 'quit':
    message = input(promt)
    print("I'll add that to you Pizza!")
    if message != 'quit':
        print(message)
```

```
Tell me, which Toppings you like:
Enter 'quit' to end the program. cheese
I'll add that to you Pizza!
cheese

Tell me, which Toppings you like:
Enter 'quit' to end the program. ham#
I'll add that to you Pizza!
ham#

Tell me, which Toppings you like:
Enter 'quit' to end the program. quit
I'll add that to you Pizza!
```

In [116]:

```
#7.5
prompt = "\nTell me your age: "


while True:
    age = input(prompt)
    age = int(age)
    if age < 3:
        price = 0
        print("The price is: " + "$" + str(price))
        break
    elif age < 12:
        price = 10
        print("The price is: " + "$" + str(price))
        break
    elif age >= 12:
        price = 15
        print("The price is: " + "$" + str(price))
        break
```

```
Tell me your age: 12
The price is: $15
```

Using a while loop with Lists and Dictionaries

In [69]:

```
# Start with users that need to be verified,
# and an empty list to hold conformed users.

unconfirmed_users = ['alice', 'brian', 'candace']
confirmed_users = []

# Verify each user until there are nor more uncorfirmed users.
# Move each verified user into the list of confirmed users.

while unconfirmed_users:
    current_user = unconfirmed_users.pop()

    print("Verifying user: " + current_user.title())
    confirmed_users.append(current_user)

# Display all confirmed users.

print("\nThe following users have been confirmed:")
for confirmed_user in confirmed_users:
    print(confirmed_user.title())
```

```
Verifying user: Candace
Verifying user: Brian
Verifying user: Alice

The following users have been confirmed:
Candace
Brian
Alice
```

- **moving items from one list to another**
- the while loop is True
- we have two lists one of them is empty
- the while loop is True default and does:
  - stores the last element from the uncormied list with the **pop() function** in the variable current_user
  - prints this variable out
  - with the **append() function** we store the popped element in the confirmed_users list
- we have a for loop which does basically:
  - printing all confirmed users, which we generated with the while loop before in our new list confirmed_users

In [70]:

```
pets = ['dog', 'cat', 'dog', 'goldfish', 'cat', 'rabbit', 'cat']
print(pets)

while 'cat' in pets:
    pets.remove('cat')

print(pets)
```

```
['dog', 'cat', 'dog', 'goldfish', 'cat', 'rabbit', 'cat']
['dog', 'dog', 'goldfish', 'rabbit']
```

- **removing all instances of specific values from a list**
- we have a list containing several instances of the element 'cat'
- we want to remove all of them
- our while loop is True when there is 'cat' in our list and if that's the case it simply does:
  - remove the 'cat' element from the list with the **remove()** function from our list pets

In [72]:

```
responses = {}

# Set a flag to indicate that polling is active
polling_active = True

while polling_active:
    # Promt for the Person's name and response.
    name = input("\nWhat is you name? ")
    response = input("Which mountain would you like to climb someday? ")

    # Store the respinse in the dictionary:
    responses[name] = response

    # Find out if anyone else is going to take the poll.
    repeat = input("Would you like to let another person respond? (yes/ no)")
    if repeat == 'no':
        polling_active = False

# Polling is complete. Show the results.
print("\n--- Poll Results ---")
for name, response in responses.items():
    print(name + " would like to climb " + response + ".")
```

```
What is you name? Ugur
Which mountain would you like to climb someday? Erciyes
Would you like to let another person respond? (yes/ no)yes

What is you name? Nico
Which mountain would you like to climb someday? Mount Everest
Would you like to let another person respond? (yes/ no)no

--- Poll Results ---
Ugur would like to climb Erciyes.
Nico would like to climb Mount Everest.
```

- **filling a dictionary with user input**
- we want to connect each user with a response
- we need an dictionary therefore
- we set the flag of the while to *True*
- our while loop stores the input *name* and *response* in two seperate variables
- out if statement is False when the User types 'no' so this makes our flag False and this will stop our while loop
- at the end we make a for loop to print out the items from our generated new dictionary responses, which now has some items in it

Tasks

- 7-8. Deli: Make a list called sandwich_orders and fill it with the names of various sandwiches. Then make an empty list called finished_sandwiches. Loop through the list of sandwich orders and print a message for each order, such as I made your tuna sandwich. As each sandwich is made, move it to the list of finished sandwiches. After all the sandwiches have been made, print a message listing each sandwich that was made.
- 7-9. No Pastrami: Using the list sandwich_orders from Exercise 7-8, make sure the sandwich 'pastrami' appears in the list at least three times. Add code near the beginning of your program to print a message saying the deli has run out of pastrami, and then use a while loop to remove all occurrences of 'pastrami' from sandwich_orders. Make sure no pastrami sandwiches end up in finished_sandwiches.
- 7-10. Dream Vacation: Write a program that polls users about their dream vacation. Write a prompt similar to If you could visit one place in the world, where would you go? Include a block of code that prints the results of the poll.

In [119]:

```
#7.8
# Start with sandwiches that need to be done,
# and an empty list to hold made sandwiches.

sandwiches = ['tuna', 'salami', 'vegan', 'egg', 'cheese']
finished_sandwiches = []

# Make each sandwich until there are nor more sandwiches to do.
# Move each done sandwich into the list of finished sandwiches.

while sandwiches:
    current_sandwich = sandwiches.pop()

    print("Doing Sandwich: " + current_sandwich.title())
    finished_sandwiches.append(current_sandwich)

# Display all done sandwiches.

print("\nThe following sandwiches have been done:")
for finished_sandwich in finished_sandwiches:
    print(finished_sandwich.title())
```

```
Doing Sandwich: Cheese
Doing Sandwich: Egg
Doing Sandwich: Vegan
Doing Sandwich: Salami
Doing Sandwich: Tuna

The following sandwiches have been done:
Cheese
Egg
Vegan
Salami
Tuna
```

In [144]:

```
#7.9
sandwiches = ['pastrami', 'tuna', 'pastrami', 'salami', 'pastrami', 'vegan', 'eg
g', 'cheese']
print("We are out of Pastrami!")

while 'pastrami' in sandwiches:
    sandwiches.remove('pastrami')

my_sep = ", "
x = my_sep.join(sandwiches[:-1])
y = my_sep.join(sandwiches[-1:])
print("We just have " + str(x).title() + " and " + str(y).title() + " Sandwiche
s!")
```

```
We are out of Pastrami!
We just have Tuna, Salami, Vegan, Egg and Cheese Sandwiches!
```

- 7-10. Dream Vacation: Write a program that polls users about their dream vacation. Write a prompt similar to If you could visit one place in the world, where would you go? Include a block of code that prints the results of the poll.

In [145]:

```
#7.10

responses = {}

# Set a flag to indicate that polling is active!
polling_active = True

while polling_active:
    # Prompt for the Person's name and response.
    name = input("\nWhat is your name? ")
    response = input("Where is you dream vacation?")

    # Store the response in the dictionary:
    responses[name] = response

    # Find out if anyone else is going to take the poll.
    repeat = input("Would you like to let another person respond? (yes/ no)")
    if repeat == 'no':
        polling_active = False

# Polling is complete. Show the results.
print("\n--- Poll Results ---")
for name, response in responses.items():
    print(name + " would like to go " + response + ".")
```

```
What is your name? Ugur
Where is you dream vacation?Italy
Would you like to let another person respond? (yes/ no)yes

What is your name? Nora
Where is you dream vacation?Germany
Would you like to let another person respond? (yes/ no)no

--- Poll Results ---
Ugur would like to go Italy.
Nora would like to go Germany.
```