Chapter 15
Python User Input

Before we begin: Make sure you can run a basic Python program on your computer using Visual Studio Code. If you prefer using a different code editor, feel free to use that instead.

Let's go to python:

By user input we mean that python will ask you to give some input data from the keyboard.

Run this python code:

```
inputInPython.py > ...
1   user_input = input("Enter something: ")
2   print("You entered:", user_input)
3
```

What we have done in the upper code is we have taken a value from the user. And printed that value.

Problem is by default the user_input variable is always of string data type. Even if you give numbers like 1, 2, 12, etc. This will be of string type. You can check the data type of any variable by using this below code:

```
inputInPython.py > ...
1   user_input = input("Enter something: ")
2   print("You entered:", user_input)
3   print("Data type:", type(user_input))
4
```

You see whatever you give from the keyboard is of string type. Let's solve this problem.

Taking a specific type data from the user:

Taking integer data from the user:

inputInPython.py > ...

```
1   user_input = int(input("Enter an integer: "))
2   print("You entered:", user_input)
3
```

What we are doing in this upper code is that we are taking a data from the user and converting it into integer. Try to understand from code. If you face difficulty, it is completely fine.

Enough python today. Let's go to Arduino-Python communication.

Chapter 16
Arduino-Python Communication

In our previous Arduino tutorials, we have learned:

Step 1: How to print something on serial monitor from Arduino.

Step 2: How to send a value from serial monitor to Arduino.

In step 1 Arduino sends data to Serial monitor and in step 2 serial monitor sends data to Arduino. The fun fact is we can make python pretend like the serial monitor. It means Python will send data to Arduino through serial and Python will get data from Arduino through serial.

Sending Data from Arduino to Python:

First upload this code to Arduino from Arduino IDE:

printPinSerialforPython.ino

```
1   char value = 'p';
2
3   void setup() {
4     Serial.begin(9600);
5
6   }
7
8   void loop() {
9
10    Serial.println(value);
11    delay(1000);
12
13  }
```

What does this upper code do? Try to find out on your own.

Now run this python code: (Change ‘COM3’ as of your own.)

pythonReadDataFromArduino.py > ...

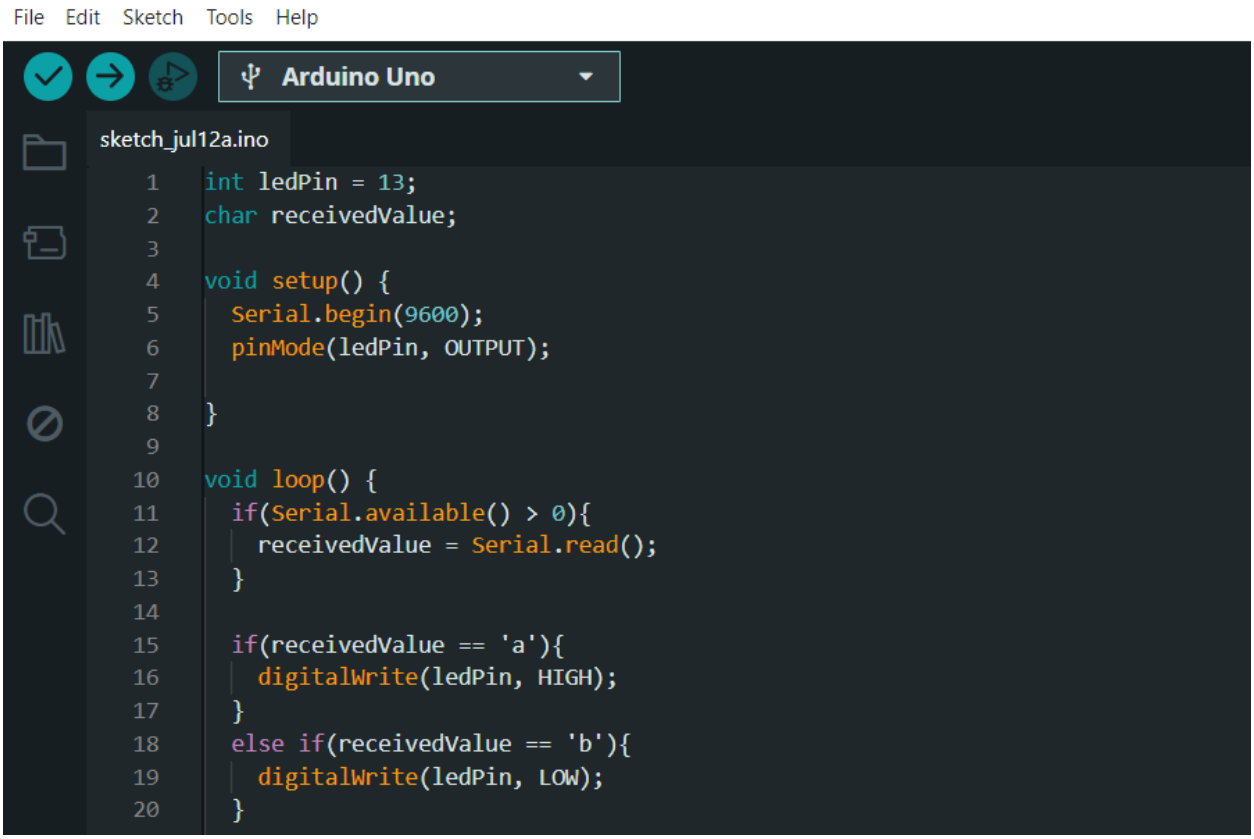
```python
import serial

port1 = serial.Serial('COM3', 9600)

while 1 == 1:
    if port1.in_waiting:
        value = port1.read().decode()
        print(value)

```

If you get error, close the Arduino IDE. The Arduino IDE has kept the COM port occupied. So, python may not be able to use that same COM port if Arduino IDE is open. Note that this python code never stops running. It’s because of the while loop. Try to understand why this code will run forever. If you want to stop this code delete the terminal.

So, now you know how to send data from Arduino to Python.

Sending data from Python to Arduino through serial:

Upload this code to Arduino from Arduino IDE:




```
int ledPin = 13;
char receivedValue;

void setup() {
  Serial.begin(9600);
  pinMode(ledPin, OUTPUT);

}

void loop() {
  if(Serial.available() > 0){
    receivedValue = Serial.read();
  }

  if(receivedValue == 'a'){
    digitalWrite(ledPin, HIGH);
  }
  else if(receivedValue == 'b'){
    digitalWrite(ledPin, LOW);
  }
```

What this Arduino code does is reads incoming data from serial. If that data is ‘a’ it turns on the built in LED. And if the data send by python is ‘b’ it turns off the built in LED.

Now run this python code to send data to Arduino through serial.

sendDatafromPythontoSerial.py > ...

```python
import serial

ser = serial.Serial('COM3', 9600)
while 1 == 1:
    data = input('Please enter a character: ')
    data = data +'\r'
    ser.write(data.encode())

```

This upper python code will ask for a data. Hit a and press Enter. You should see the LED turning ON on Arduino. If you send b from Python, the LED should turn OFF. Note that this python code runs forever unless you delete the terminal. What is the reason behind this forever looping?

Assignment 08:

Python practice problem:

1. Write a python code that will take a number form you and check if it is odd or even.
2. Write a python code that will take a year and check if it is a leap year or not.
3. Write python code to check if a number is positive or not.
4. Write python code that will take a number from you and add from 0 to up to that number.
5. Write a python code that will take a number from you and find it's factorial.
6. Write a python code to check age group: If age < 13 → Child
   If 13–19 → Teenager
   Else → Adult
7. Write python code to convert a temperature in degree Celsius to Fahrenheit and kelvin.

Arduino-Python practice problem:

1. Control 2 LED from Python.
2. Control a buzzer from Python.
3. Rotate a servo by sending data from Python.
4. Send value from python to Arduino from 0 to 9. And write Arduino code that will read that data from python and show it on SSD.
5. Send character from a to z from Arduino to python. If you want to write println() function for 26 times, you can do that. But there are better ways. Note: You have to write Arduino code to send. And python code to read.