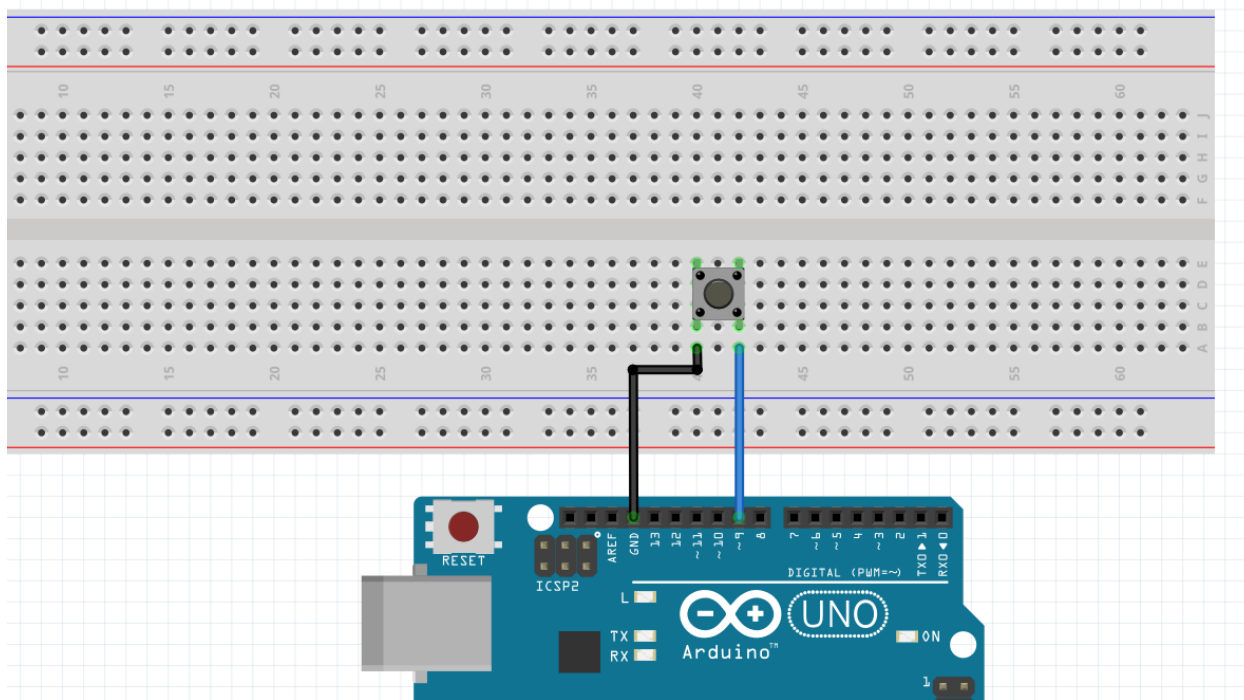


Chapter 9

How to use push button with Arduino

Push buttons are very helpful in building complex projects. Let's build circuit below:



Now upload this below code to Arduino UNO:

The image shows the Arduino IDE interface. At the top, there is a menu bar with 'File', 'Edit', 'Sketch', 'Tools', and 'Help'. Below the menu bar is a toolbar with icons for checking, running, and uploading code, followed by a dropdown menu showing 'Arduino Uno'. On the left side, there is a file explorer showing a folder icon and a file named 'sketch_jun30a.ino'. The main area displays the code for 'sketch_jun30a.ino' with line numbers 1 through 14. The code is as follows:

```
1 int x;
2
3 void setup() {
4   Serial.begin(9600);
5   pinMode(9, INPUT_PULLUP);
6
7 }
8
9 void loop() {
10  x = digitalRead(9);
11  Serial.println(x);
12  delay(50);
13
14 }
```

Now open the Serial monitor and press the button. Press the button.

Explanation of the code we are using:

The image shows the same Arduino IDE interface as the first image, but with comments added to the code in 'sketch_jun30a.ino'. The code is as follows:

```
1 int x; // We will use this variable to store the state of the button
2
3 void setup() {
4   Serial.begin(9600); // Initiate te Serial communication at 9600 baud rate
5   pinMode(9, INPUT_PULLUP); // INPUT_PULLUP means naturally UP or HIGH. HIGH means 1.
6
7 }
8
9 void loop() {
10  x = digitalRead(9); // Read the state of pin 9 and put the state into x variable
11  Serial.println(x); // Print the x. (State of the pin 9)
12  delay(50); // Wait for 50 milliseconds before repeating again
13
14 }
```

From today we will put the explanation along with the code using comments. Hope you have learned that comments are not a part of code. They are just some random talks that helps you to understand. It is a very good habit to write comments in your own code. Write what ever you want. Just make sure that it makes sense to you later. Also, comments help others to understand your code.

Now, we are slowly becoming a programmer. Let's write the upper same code but in a more programmer's way:

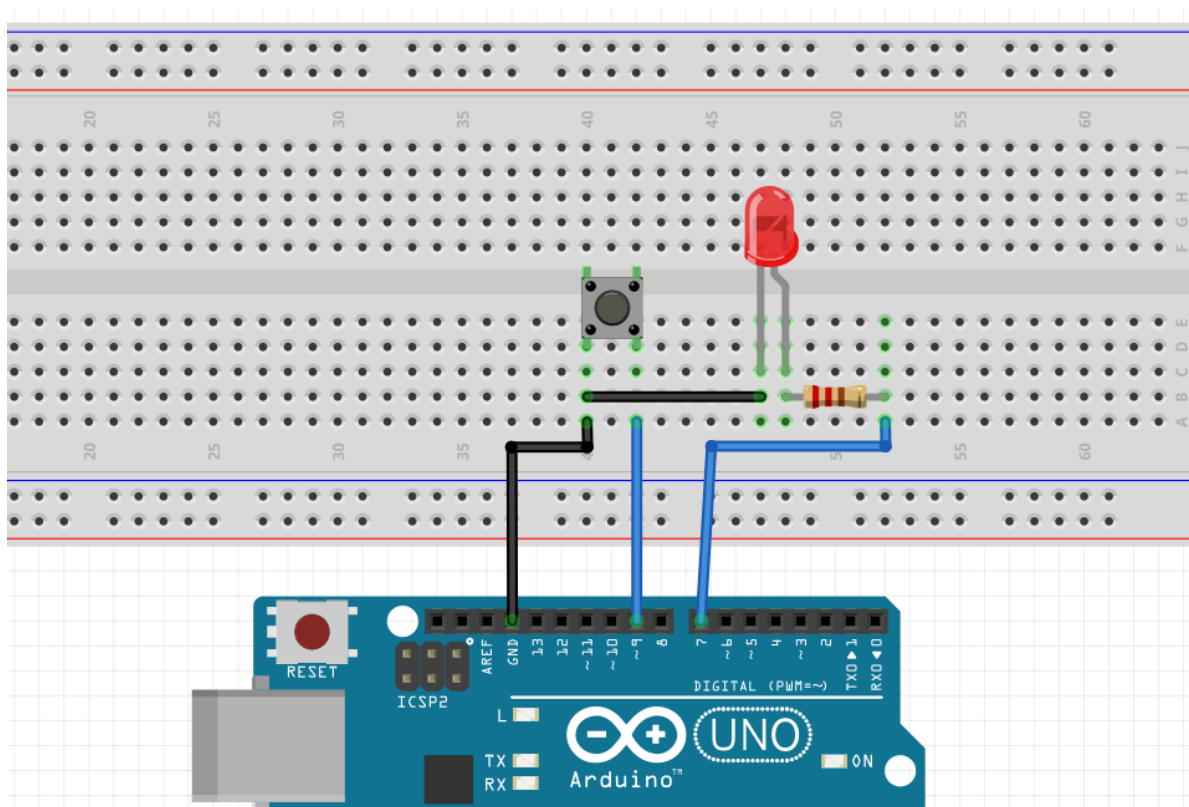


```
File Edit Sketch Tools Help
Arduino Uno
sketch_jun30a.ino
1 int x; // We will use this variable to store the state of the button
2 int buttonPin = 9;
3
4 void setup() {
5   Serial.begin(9600); // Initiate te Serial communication at 9600 baud rate
6   pinMode(buttonPin, INPUT_PULLUP); // INPUT_PULLUP means naturally UP or HIGH. HIGH means 1.
7 }
8
9
10 void loop() {
11   x = digitalRead(buttonPin); // Read the state of pin 9 and put the state into x variable
12   Serial.println(x); // Print the x. (State of the pin 9)
13   delay(50); // Wait for 50 milliseconds before repeating again
14 }
15 }
```

See that this is the same thing. But way more understandable. Now anyone can see this code and tell that you have connected a button at pin 9 without even looking at the circuit.

Let's control a LED using push-button.

Build the circuit below:



And upload this code below:

```
File Edit Sketch Tools Help
Arduino Uno
sketch_jun30a.ino
1 int x; // We will use this variable to store the state of the button
2 int buttonPin = 9;
3 int ledPin = 7;
4
5 void setup() {
6   Serial.begin(9600); // Initiate te Serial communication at 9600 baud rate
7   pinMode(buttonPin, INPUT_PULLUP); // INPUT_PULLUP means naturally UP or HIGH. HIGH means 1.
8   pinMode(ledPin, OUTPUT); //ledPin means 9
9
10 }
11
12 void loop() {
13   x = digitalRead(buttonPin); // Read the state of pin 9 and put the state into x variable
14
15   if (x == 0){ // When Button is Pressed
16     digitalWrite(ledPin, HIGH);
17   }
18   else if( x == 1){ // When Button is not Pressed
19     digitalWrite(ledPin, LOW);
20   }
21 }
```

Press the button and see what happens. Things are getting complex with time. So, pay attention.

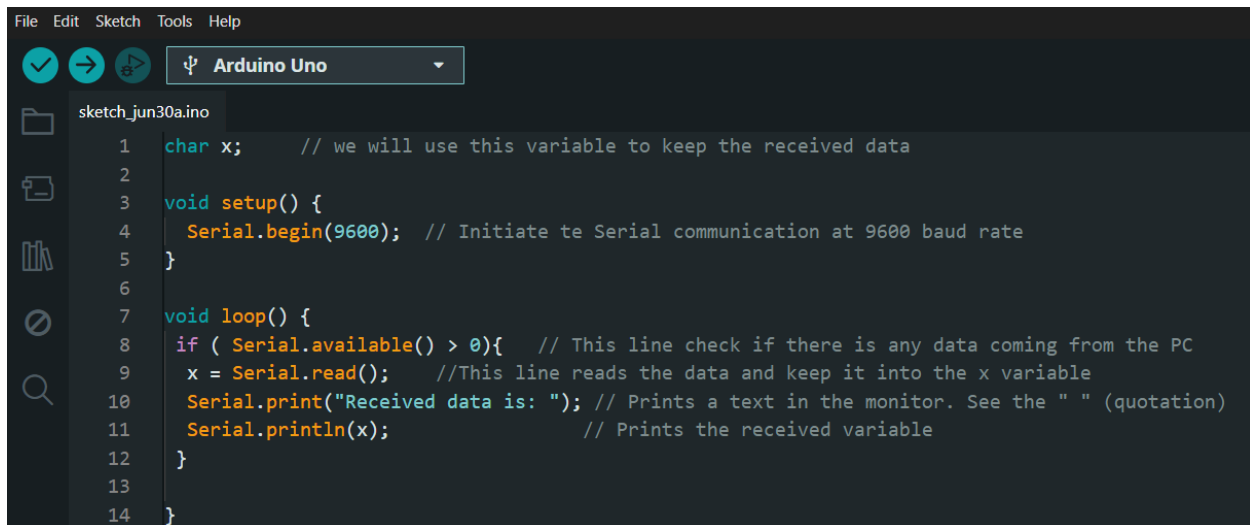
Chapter 10

Sending Data from PC to Arduino

If you remembered, we have learned how to print something on serial monitor to print some message or text or value. In that time Arduino was sending data to PC through that cable.

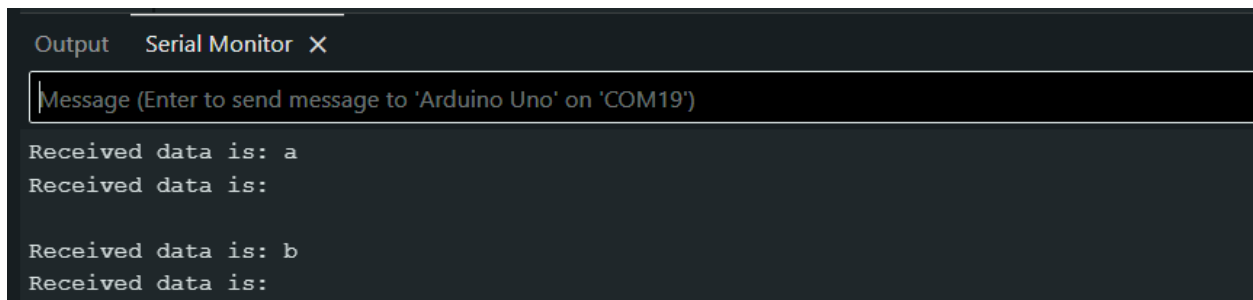
The PC can also send data to Arduino using the same process. Here the Serial monitor will send data and Arduino will receive that.

Upload this code to Arduino UNO:



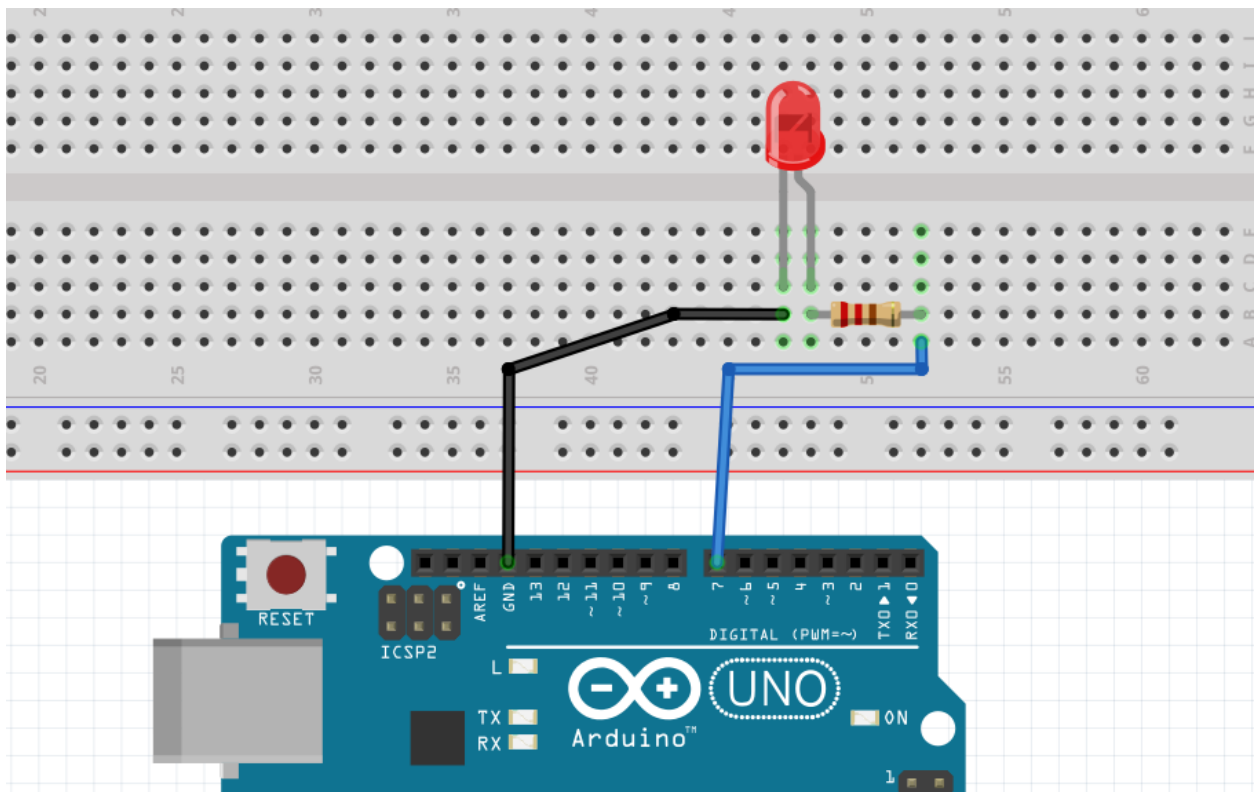
```
File Edit Sketch Tools Help
sketch_jun30a.ino
1 char x; // we will use this variable to keep the received data
2
3 void setup() {
4   Serial.begin(9600); // Initiate te Serial communication at 9600 baud rate
5 }
6
7 void loop() {
8   if ( Serial.available() > 0){ // This line check if there is any data coming from the PC
9     x = Serial.read(); //This line reads the data and keep it into the x variable
10    Serial.print("Received data is: "); // Prints a text in the monitor. See the " " (quotation)
11    Serial.println(x); // Prints the received variable
12  }
13
14 }
```

Now send data from PC to Arduino using Serial monitor. Use this box named ‘message’

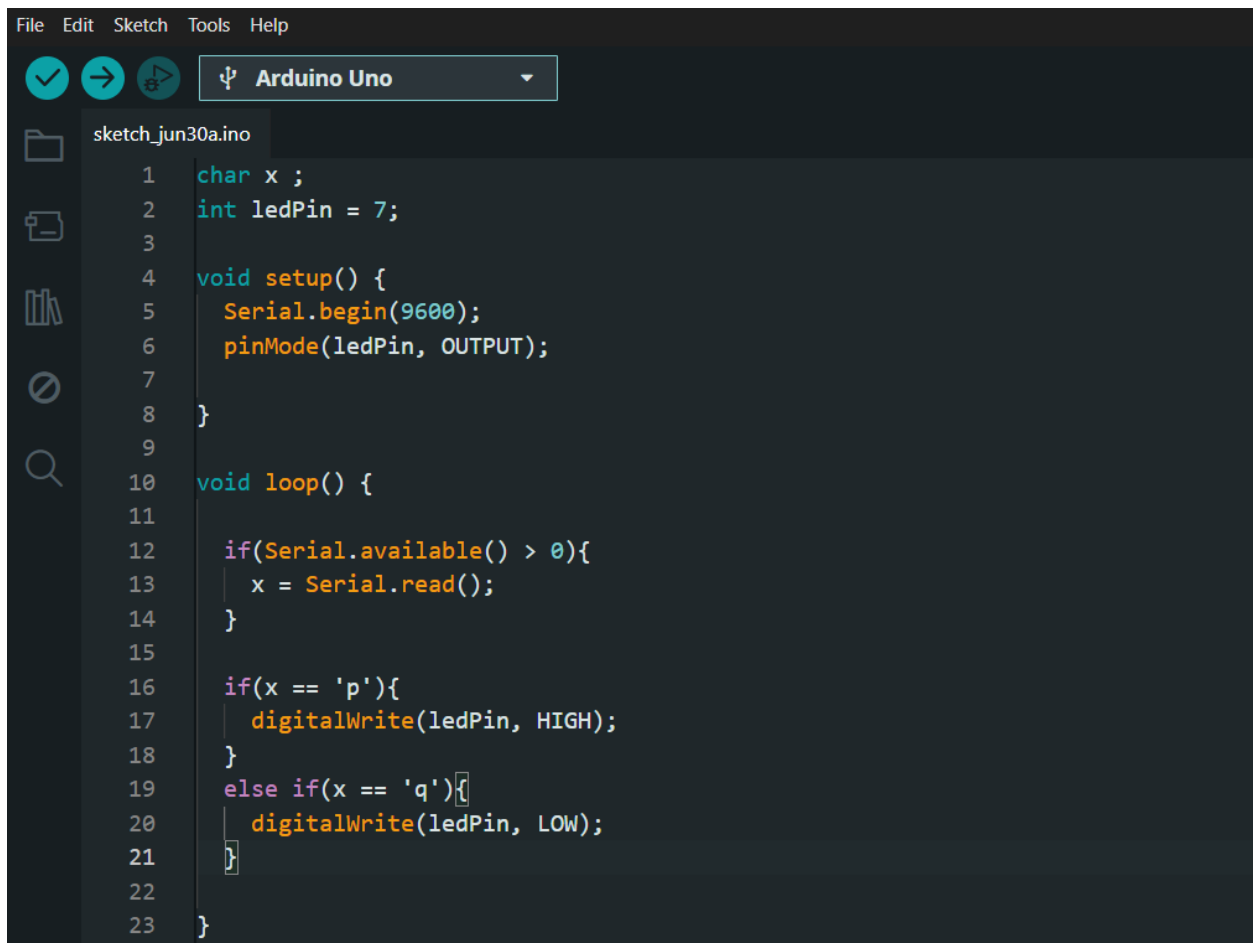


Let's use this Serial Monitor to control a LED.

Build the below circuit.



And now upload this code:



```
File Edit Sketch Tools Help
ψ Arduino Uno
sketch_jun30a.ino
1 char x ;
2 int ledPin = 7;
3
4 void setup() {
5     Serial.begin(9600);
6     pinMode(ledPin, OUTPUT);
7
8 }
9
10 void loop() {
11
12     if(Serial.available() > 0){
13         x = Serial.read();
14     }
15
16     if(x == 'p'){
17         digitalWrite(ledPin, HIGH);
18     }
19     else if(x == 'q'){
20         digitalWrite(ledPin, LOW);
21     }
22
23 }
```

Send p and q from PC to Arduino and see what happens.

Let me give you a very hard task. Connect the SSD circuit and Write code that will Show 1 if you send 1 from PC. 2 if you send 2 from PC. 3 if you send 3 from PC.

And interesting thing: When you send data from PC to Arduino through Serial a LED on Arduino lights up for a moment. See beside that LED. It has a beautiful name. Does the name make any sense about what it does? Also, when Arduino sends data to PC another LED lights up for a moment. See its name and think about this for a minute.

Chapter 11

Introduction to python

We are going to learn another programming language.



So, we are going to build some robots. Actually, toy but we, engineers call everything robot. The robot may have a camera. You need to do some image processing types of things. Or you may have to do very complex calculation about your environment. But the problem is Arduino has only 2KB of RAM. We have to use some other things like PC, Raspberry pi, Jetson Nano to do this heavy task. All of these things mainly runs on Python. Not only on python but python is the best option.

So, the sad news is we have to learn python. But relax. We are just going to see the basics as we have seen in C++ for Arduino. Today we will learn those same C++ thing but in Python. And you will see python is much easier than C++.

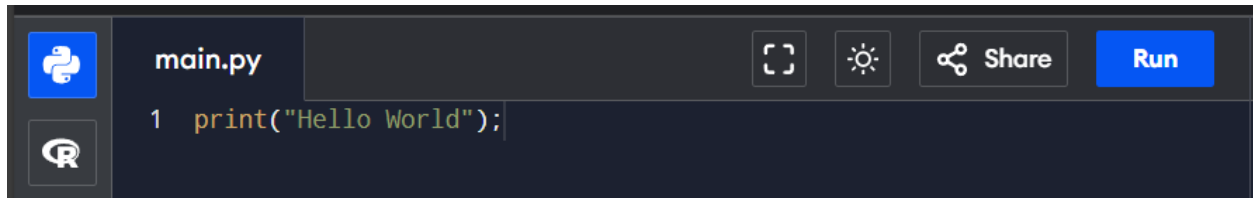
What we are going to learn in python today:

1. How to print something in python
2. Different variables
3. Condition: if...else if...else
4. Loop: For loop, While loop

How to print something in python

Go to <https://www.programiz.com/python-programming/online-compiler/>

And then write this code without thinking much. And click Run.

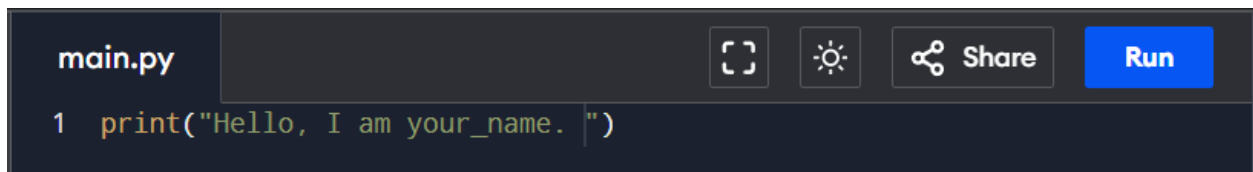
A screenshot of an online Python compiler interface. On the left, there is a sidebar with a Python logo and a file icon. The main area is titled 'main.py' and contains a single line of code: `1 print("Hello World");`. To the right of the code editor are four icons: a square with a plus sign, a sun icon, a share icon, and a blue 'Run' button.

Hurrah! You have just written your first python code. Now write this code and click Run.

A screenshot of an online Python compiler interface. The file name is 'main.py'. The code editor shows the same line of code as before, but the semicolon has been removed: `1 print("Hello World")`. The 'Run' button is still present on the right.

What do you see? Python actually does not care about this annoying semicolone after each line. Just this simple property of python is enough to love python.

Now write this code and run: (Replace your_name by your name.)

A screenshot of an online Python compiler interface. The file name is 'main.py'. The code editor shows a new line of code: `1 print("Hello, I am your_name. ")`. The 'Run' button is still present on the right.

Variables in python:

There are many variables and each of them has different use case. But for now, we are going to learn about integers and strings only.

Write this code and try to understand how to use a variable.

```
main.py  [Full Screen] [Theme] [Share] [Run]
1 x = int(3)
2 print(x)
```

Now, try this code:

```
main.py  [Full Screen] [Theme] [Share] [Run]
1 x = int(3)
2 y = int(5)
3 print(x)
4 print(y)
```

“x = int(5)” in python is similar to “int x = 5;” in Arduino.

Try this code:

```
main.py  [Full Screen] [Theme] [Share] [Run]
1 print("Hello, world")
```

This same thing can be done by using a string type variable. String type variable keeps text inside them. Integer type variables keep numbers inside them.

Try this code:

```
main.py  [Full Screen] [Theme] [Share] [Run]
1 x = str("Hello, world")
2 print(x)
```

Try this code:

```
main.py  [Full Screen] [Theme] [Share] [Run]
1 x = str("your_name")
2 print("My name is " + x)
```

Conditions in python:

Remember we have written if...else if ...else in Arduino. We can write similar things in python:

Write and run this code below:

```
main.py  [Full Screen] [Theme] [Share] [Run]
1 x = int(2)
2 if x==2:
3     print("You have written 2.")
4 else:
5     print("You have written something other than 2.")
```

Try this code below:

```
main.py  [Full Screen] [Theme] [Share] [Run]
1 x = int(56)
2 if x==2:
3     print("You have written 2.")
4 else:
5     print("You have written something other than 2.")
```

Basic math in python:

We won't talk much. Just write these codes and run them and try to understand.

```
main.py  [Full Screen] [Theme] [Share] [Run]

1 a = 2
2 b = 5
3 c = a + b
4 print(c)
```

Try this:

```
main.py  [Full Screen] [Theme] [Share] [Run]

1 a = 2
2 b = 5
3 d = 10
4 e = 20
5 c = a + b + d + e
6 print(c)
```

Then this:

```
main.py  [Full Screen] [Theme] [Share] [Run]

1 a = 2
2 b = 5
3 c = a - b
4 print(c)
```

Write and run this code:

```
main.py  [Full Screen] [Theme] [Share] [Run]

1 a = 12
2 b = 5
3 c = a * b
4 print(c)
```

Write and run this code: (Reminder)

```
main.py  [ ] [ ] [ ] Share Run
1 a = 12
2 b = 5
3 c = a % b
4 print(c)
```

Write and run this code:

```
main.py  [ ] [ ] [ ] Share Run
1 a = 34
2 b = 5
3 c = a % b
4 print(c)
```

Then this code below:

```
main.py  [ ] [ ] [ ] Share Run
1 a = 58
2 b = 12
3 c = a / b
4 print(c)
```

Do you remember we faced a problem for this type of math in Arduino. But python creates no problem here.

Loops in python:

While:

Write and run this:

```
main.py  [ ] [ ] [ ] Share Run
1 count = 1
2 while count <= 5:
3     print("Count is:", count)
4     count = count + 1
```

Then run this:

```
main.py  [ ] [ ] [ ] Share Run
1 count = 10
2 while count <= 500:
3     print(count)
4     count = count + 1
```

Run this code below:

```
main.py  [ ] [ ] [ ] Share Run
1 count = 10
2 while count <= 100:
3     print(count)
4     count = count + 3
```

For loop in python:

Surprise, we are not going to learn for loop now. It won't be necessary at this moment. We will learn one day if we get into trouble but not today.

While loop again:

Try this:


```
main.py  [Full Screen] [Theme] [Share] [Run]
1 i = 100
2 while i >= 0:
3     print(i)
4     i = i - 1
```

Try this code:

```
main.py  [Full Screen] [Theme] [Share] [Run]
1 i = 100
2 while i >= 0:
3     print(i)
4     i = i - 2
```

Try this code:

```
main.py  [Full Screen] [Theme] [Share] [Run]
1 i = 100
2 while i >= 0:
3     print(i)
4     i = i - 5
```

Assignment 04

1. Control a LED with push button. LED ON when button pressed.
2. Control a LED with push button. LED OFF when button pressed.
3. Control a LED using Serial monitor. LED ON if you send a and OFF if you send b
4. Connect 4 LED to Arduino and control them from Serial monitor. If you send 1 one LED will be ON. If you send 2, two LED will be ON. If you send 3, three LED will be ON. So on.
5. Control SSD using Serial monitor. If you send 1 it will display 1. If you send 2 it will display 2. Thus 3, 4, 5, 6, 7, 8, 9, 0, A, B, C, D, E, F. (Note: these are the digits of hexadecimal.)
6. Print details of your favourite aircraft/robot in python in 10 lines.
7. Solve these below maths using Python (if you know already, you can skip.)
 - a. $100 + 700 = ?$ (These are all those same maths you have done in arduino.)
 - b. $3000 - 2100 = ?$
 - c. $300 * 6 = ?$
 - d. $110 / 10 = ?$
 - e. $100 + 120 - 200 = ?$
 - f. $100 - 200 + 300 * 2 = ?$ (Remember this math from class 2 ?)
 - g. $300*2 + 200*3 + 400/2 = ?$ (Seems hard, isn't it?)
 - h. $(20 + 100) \% 7 = ?$
 - i. $(200*40) / (100*20) = ?$
 - j. $((300*5) \% (200/4)) + (30 - 20)*(30+20) = ?$
 - k. Print 1 to 500 in the serial monitor using for loop in python.
 - l. Print 1 to 500 using while loop in python.
 - m. Print 100 to 200 using for loop in python.
 - n. Print 100 to 200 using while loop in python.
 - o. Print 300 to 100 using for loop in python
 - p. Print 300 to 100 using while loop in python.
8. Control Buzzer by sending data from PC to Arduino.