



Introduction to Python Programming

Lab Manual 2



Learning Outcomes:

- Students shall be able to define variables and their use in programming.
- Students shall learn the use of arithmetic operators.
- Students shall learn the use of print function.
- Students shall learn the use of strings.
- Students shall learn how to get the input from the user.

Introduction

In this class, we will learn about “**variables**”, “**assignment statements**”, “**arithmetic operators**”, “**print function**”, “**introduction to strings**” and “**getting input from user**” concepts in python, these concepts are used to write different programs.

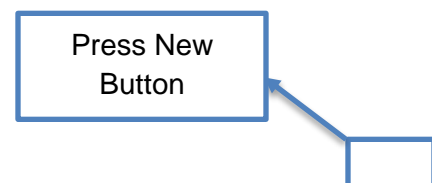
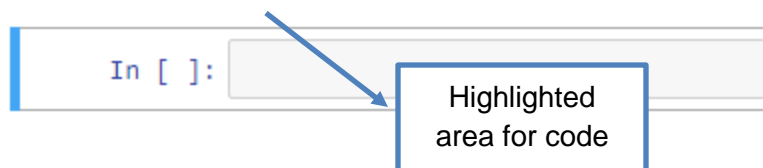
Program structure

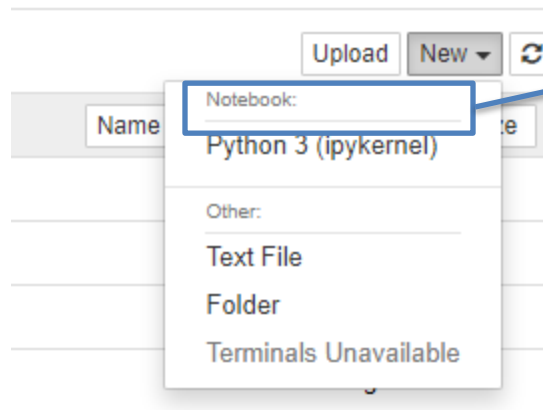
Just like any other language, the python language also has a particular structure that requires that instructions must be given according to the defined structure.

Example:

```
print("Hello, World!")
```

The jupyter notebook rules require that all the instructions must be given inside the highlighted section, the code file is saved automatically.





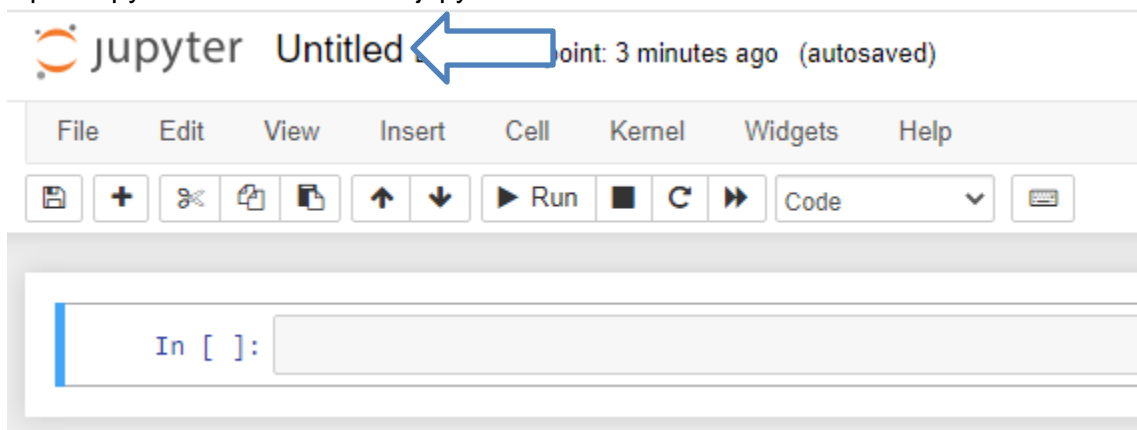
Activity: Write these lines of code in your editor (notepad) to graph the practical understanding of the basic program structure of the c++ language.

Compiler

We will use the **jupyter notebook** integrated development environment (**IDE**) to compile and execute our programs.

To execute the program, perform the following steps.

Open a python notebook in the jupyter notebook IDE.



Click on the **"Untitled"** title to change the name of your notebook.

Rename Notebook

Enter a new notebook name:

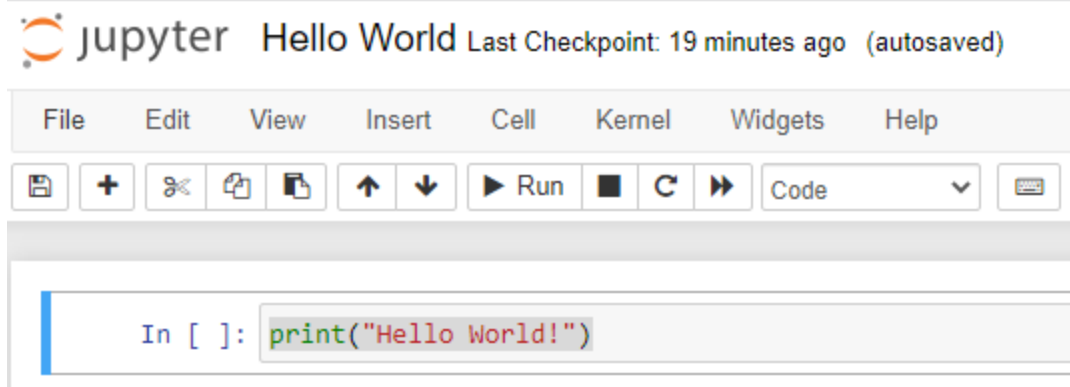
Untitled

CancelRename

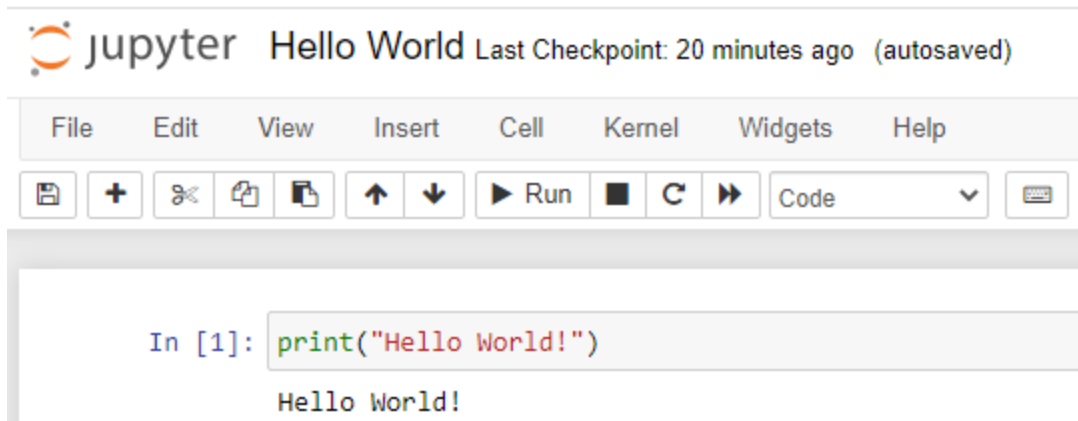
Give your desired name to the notebook.



Write the following line in the highlighted cell for code in the notebook **print("Hello World!")**



Press **Shift + Enter** to execute the line of code.



“Congratulations, You have successfully created, compiled, and executed your first program”

Variable and Types

In programming, there are “nicknames” that are known as “**variables**” that are used to store different kinds of values.

For example,

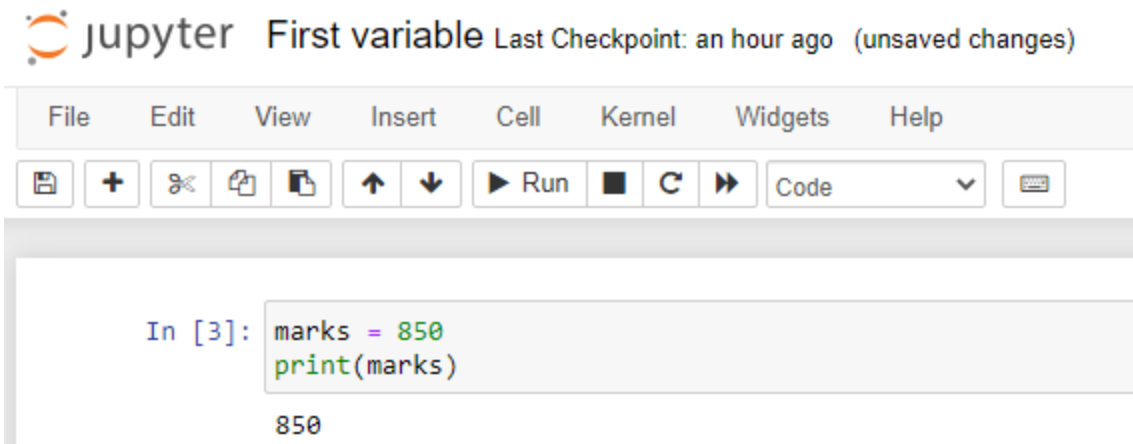
Ali had obtained 850 marks in matric. We want to store this number in computer memory. But, the problem is we cannot possibly remember the address of all the things that we stored in the memory. That’s why we use nicknames, “**variables**”.

So, we can use a variable to store the obtained marks of Ali in the storage of the computer.

For example, we can name the variable as “**marks**”.

Activity: Write your first code of python language on your jupyter notebook by declaring a variable.

Example (Integer):



The screenshot shows a Jupyter Notebook window titled "First variable". The interface includes a menu bar with options: File, Edit, View, Insert, Cell, Kernel, Widgets, and Help. Below the menu bar is a toolbar with icons for saving, adding cells, undo, redo, and running code. The main area displays a code cell with the following code:

```
In [3]: marks = 850
        print(marks)
```

The output of the code cell is the number 850.

“Congratulations, you have successfully learned how to declare a variable”.

Now, we can assign any number to this **marks** variable.

Activity: Write The above code on your computer editor to check the value stored in the variable.

Example (Integer):

Similarly, we can also declare a variable for storing **total** marks in this way.

Now, similarly, we can store **total** marks in this variable.

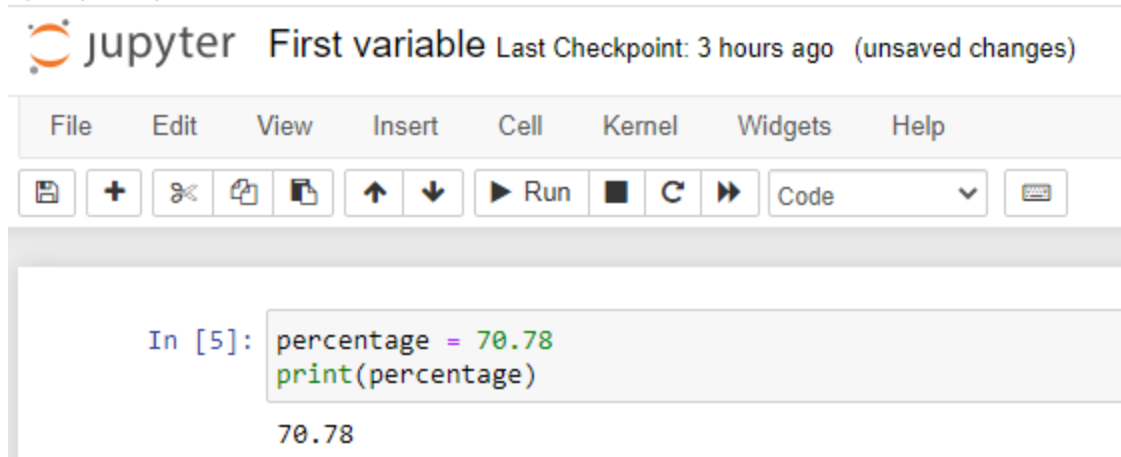
total = 1050

“This process of giving value to a variable is known as “**assigning value to variable**”.

Activity: Write the above code on your computer editor to store the value in the variable.

Similarly, let's also check the value stored in the **percentage** variable.

Example (Float):



The screenshot shows a Jupyter Notebook interface. At the top, it says "jupyter First variable Last Checkpoint: 3 hours ago (unsaved changes)". Below this is a menu bar with "File", "Edit", "View", "Insert", "Cell", "Kernel", "Widgets", and "Help". Under the menu bar is a toolbar with icons for saving, adding, deleting, copying, pasting, undo, redo, running, and other actions. The main area shows a code cell with the following code:

```
In [5]: percentage = 70.78
        print(percentage)
```

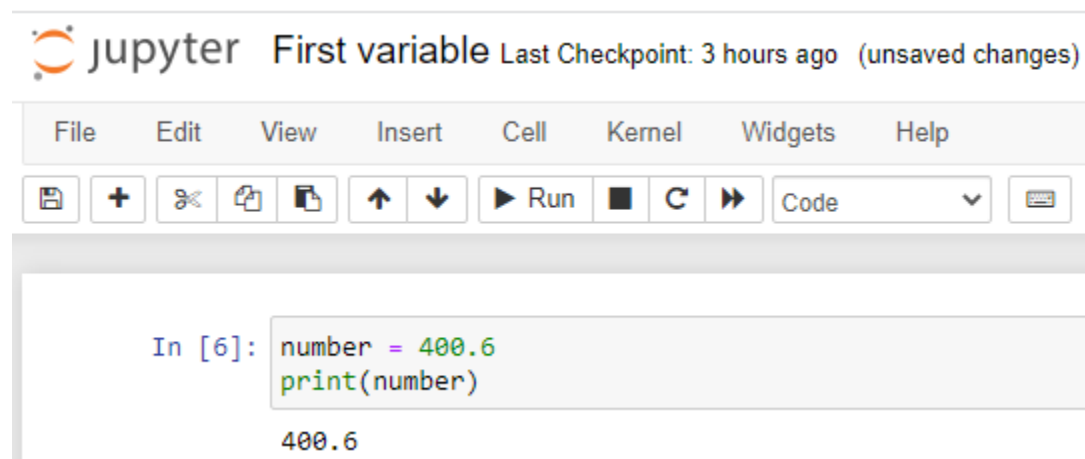
The output of the code cell is 70.78.

Activity: Write The above code on your computer editor to check the value stored in the variable.

Example (Float):

declare a variable named as “**number**” to store a value of 400.6.

number = 400.6



The screenshot shows a Jupyter Notebook interface. At the top, it says "jupyter First variable Last Checkpoint: 3 hours ago (unsaved changes)". Below this is a menu bar with "File", "Edit", "View", "Insert", "Cell", "Kernel", "Widgets", and "Help". Under the menu bar is a toolbar with icons for saving, adding, deleting, copying, pasting, undo, redo, running, and other actions. The main area shows a code cell with the following code:

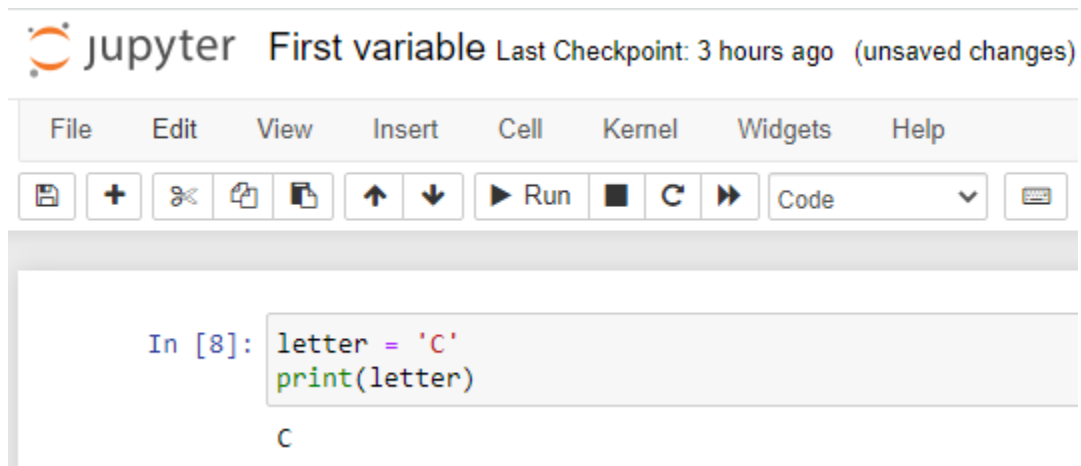
```
In [6]: number = 400.6
        print(number)
```

The output of the code cell is 400.6.

Activity: Write the above code on your computer editor to store the value in the variable.

Example (Character):

Declare a variable named **letter** to store “**C**”.



The image shows a Jupyter Notebook interface. At the top, the title bar says "jupyter First variable" followed by "Last Checkpoint: 3 hours ago (unsaved changes)". Below the title bar is a menu bar with "File", "Edit", "View", "Insert", "Cell", "Kernel", "Widgets", and "Help". Under the menu bar is a toolbar with icons for saving, adding cells, undo, redo, copy, paste, up/down arrows, a run button, a stop button, a refresh button, and a dropdown menu currently set to "Code". The main area of the notebook contains a code cell with the following text:

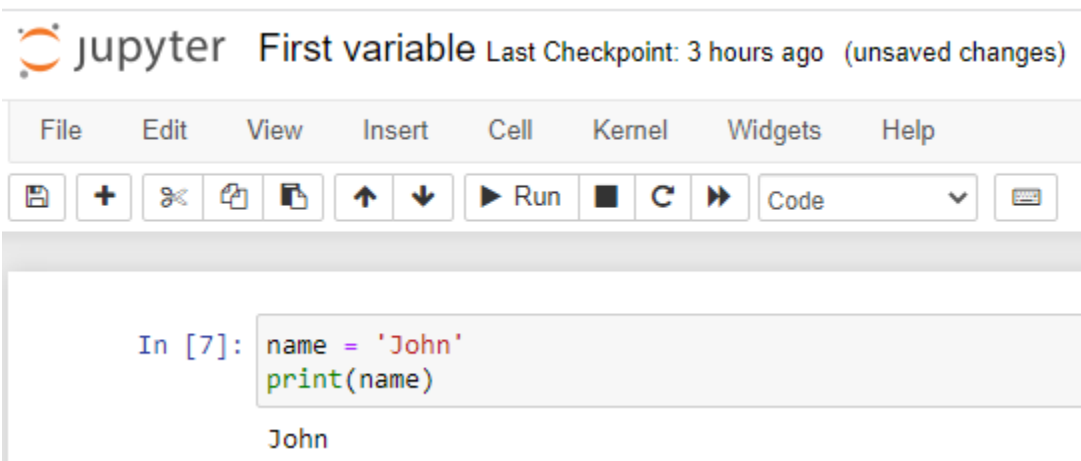
```
In [8]: letter = 'C'
        print(letter)
```

Below the code cell, the output "C" is displayed.

Activity: Write the above code in your computer to declare a char type variable in your program.

Example (String):

name = “John”



The image shows a Jupyter Notebook interface. At the top, the title bar says "jupyter First variable" followed by "Last Checkpoint: 3 hours ago (unsaved changes)". Below the title bar is a menu bar with "File", "Edit", "View", "Insert", "Cell", "Kernel", "Widgets", and "Help". Under the menu bar is a toolbar with icons for saving, adding cells, undo, redo, copy, paste, up/down arrows, a run button, a stop button, a refresh button, and a dropdown menu currently set to "Code". The main area of the notebook contains a code cell with the following text:

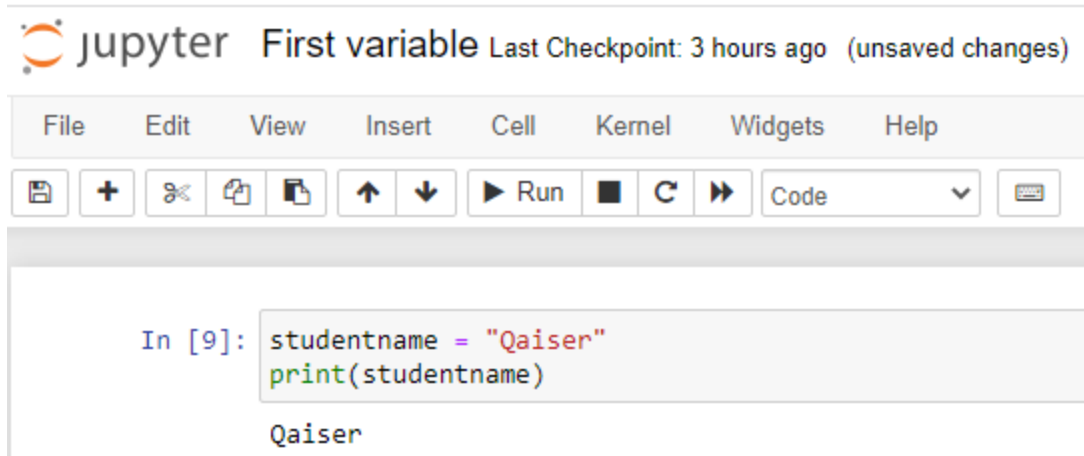
```
In [7]: name = 'John'
        print(name)
```

Below the code cell, the output "John" is displayed.

Example (String):

Declare a variable named “**studentname**”.

```
studentname = “Qaiser”
```



The image shows a Jupyter Notebook window titled "First variable". The toolbar includes options like File, Edit, View, Insert, Cell, Kernel, Widgets, and Help. Below the toolbar, a code cell is shown with the following code:

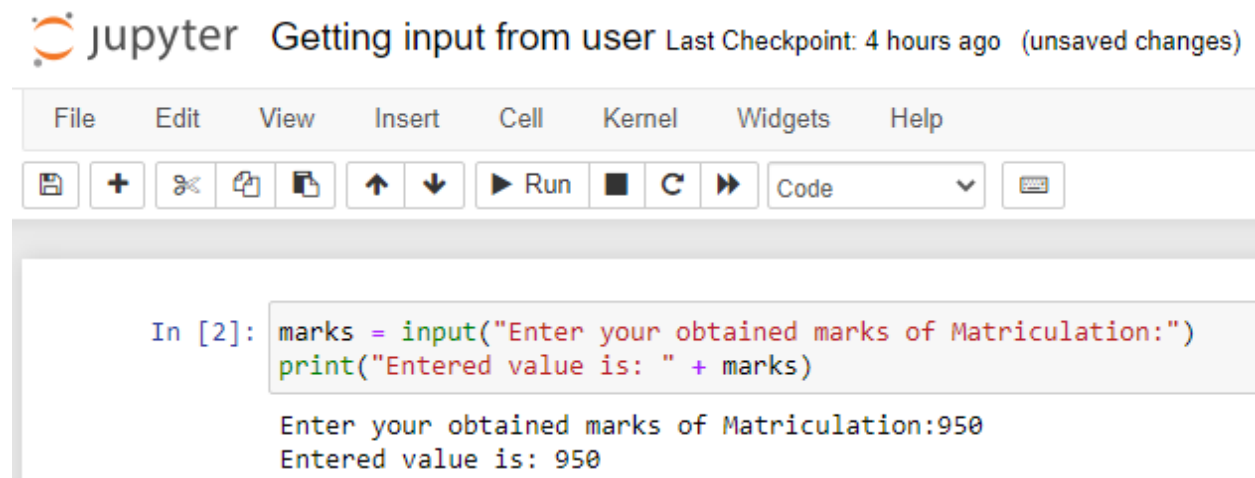
```
In [9]: studentname = "Qaiser"
        print(studentname)
```

The output of the code cell is "Qaiser".

Activity: Write the above code on your computer editor to declare a string-type variable in your program.

Example (Getting input from user):

Ask the user to enter his obtained matric marks through the console.



The image shows a Jupyter Notebook window titled "Getting input from user". The toolbar includes options like File, Edit, View, Insert, Cell, Kernel, Widgets, and Help. Below the toolbar, a code cell is shown with the following code:

```
In [2]: marks = input("Enter your obtained marks of Matriculation:")
        print("Entered value is: " + marks)
```

The output of the code cell is:

```
Enter your obtained marks of Matriculation:950
Entered value is: 950
```

Activity: Write the above code on your computer to print the above-mentioned line on your computer screen.

Write a python program that prints a box using stars (*).

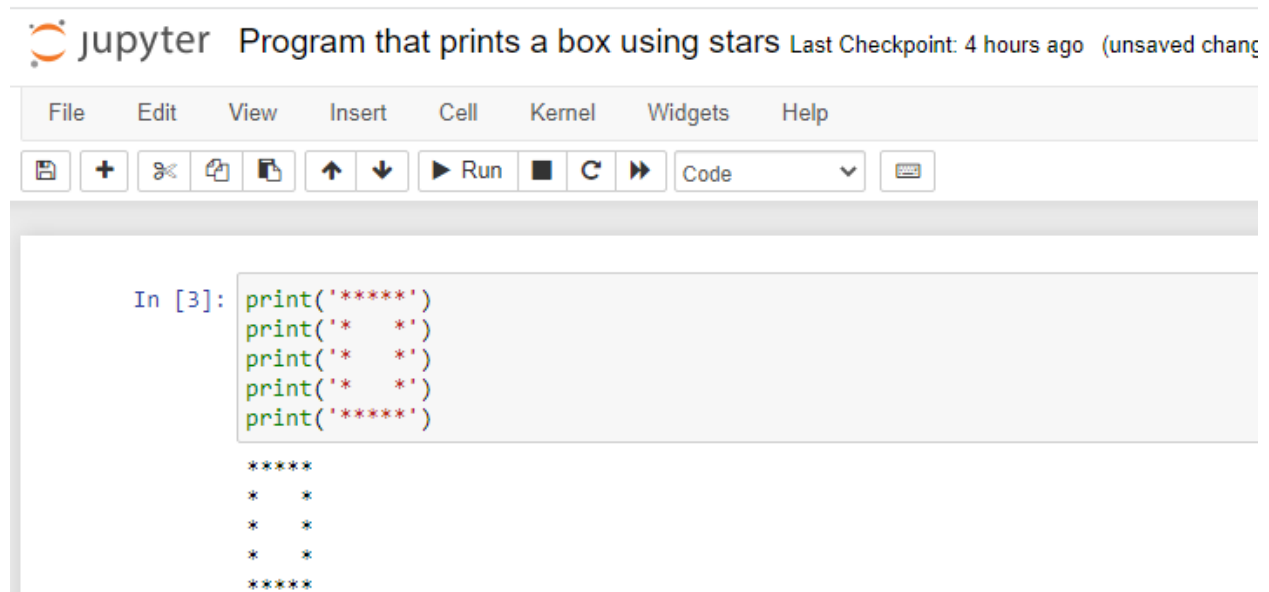
```
*****  
  
*   *  
*   *  
*   *  
  
*****
```

First Try by yourself.

Activity: Try to solve the above question by using the knowledge you have learned to this point.

Don't worry.

The solution is attached below.



The screenshot shows a Jupyter Notebook window titled "Program that prints a box using stars" with a "Last Checkpoint: 4 hours ago (unsaved changes)" status. The interface includes a menu bar (File, Edit, View, Insert, Cell, Kernel, Widgets, Help) and a toolbar with icons for saving, adding cells, undo, redo, copy, paste, and running code. The code cell, labeled "In [3]:", contains the following Python code:

```
print('*****')  
print('*   *')  
print('*   *')  
print('*   *')  
print('*   *')  
print('*****')
```

Below the code, the output of the program is displayed as a box of stars:

```
*****  
  
*   *  
*   *  
*   *  
  
*****
```

How to get input?

For this purpose, the python language has a method that is **Input() -> input method**

Now, Let's use it to get your first variable input from the user.


```
File Edit View Insert Cell Kernel Widgets Help
[Save] [New] [Close] [Copy] [Paste] [Up] [Down] [Run] [Stop] [Refresh] [Next] Code [Terminal]

In [1]: name=input("Enter Your Name: ")
        print("Hello", name)

        Enter Your Name: Nauman
        Hello Nauman
```

Activity: Write the above code on your editor to get the value of the “name” variable from the user.

Example (Addition of two numbers by taking user input):

```
File Edit View Insert Cell Kernel Widgets Help
[Save] [New] [Close] [Copy] [Paste] [Up] [Down] [Run] [Stop] [Refresh] [Next] Code [Terminal]

In [3]: print("This program adds two numbers")
        num1=input("Enter Number 1: ")
        num2=input("Enter Number 2: ")
        addition = int(num1) + int(num2)
        print("Sum = ", addition)

        This program adds two numbers
        Enter Number 1: 10
        Enter Number 2: 5
        Sum = 15
```

Activity: Write the above code on your editor to get the values of variables “num1” and “num2” variables from the user and perform their addition.

Similarly, we can use these variables to perform various mathematical tasks as well.

Example:

Consider the following question.

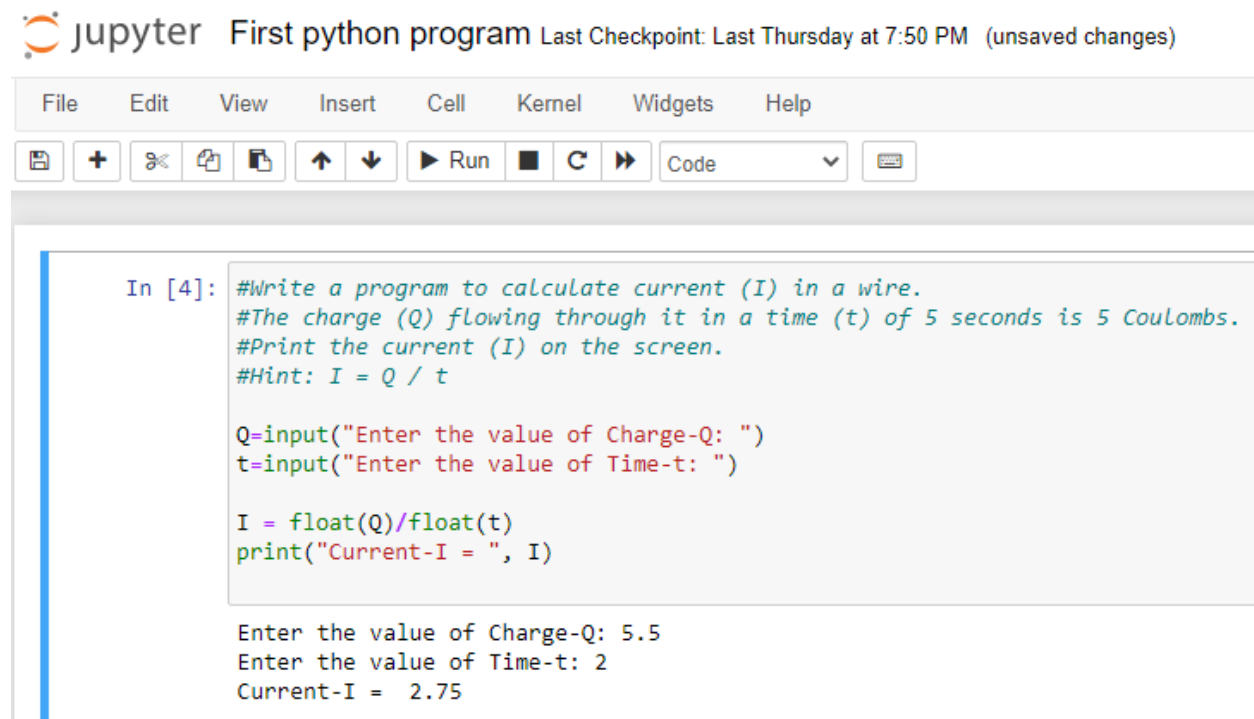
Write a program to calculate current (I) in a wire. The charge (Q) flowing through it in a time (t) of 5 seconds is 5 Coulombs. Print the current (I) on the screen.

Hint: $I = Q / t$

First, try yourself.

Don't worry.

The solution is attached below. For Now, just try for yourself.



The image shows a Jupyter Notebook interface with the title "First python program" and a status bar indicating "Last Checkpoint: Last Thursday at 7:50 PM (unsaved changes)". The interface includes a menu bar (File, Edit, View, Insert, Cell, Kernel, Widgets, Help) and a toolbar with icons for saving, adding cells, undo, redo, and running code. The code cell contains the following Python code:

```
In [4]: #Write a program to calculate current (I) in a wire.
#The charge (Q) flowing through it in a time (t) of 5 seconds is 5 Coulombs.
#Print the current (I) on the screen.
#Hint: I = Q / t

Q=input("Enter the value of Charge-Q: ")
t=input("Enter the value of Time-t: ")

I = float(Q)/float(t)
print("Current-I = ", I)
```

The output of the code is displayed below the code cell:

```
Enter the value of Charge-Q: 5.5
Enter the value of Time-t: 2
Current-I = 2.75
```

Activity: Write the above code on your editor to get the desired answer.

Scenario

Assume that Ali is a student who wants to calculate his aggregate for taking admission in UET. we shall use a computer program that would take his obtained marks and after processing, it would tell Ali his aggregate.

Firstly, we shall need to ask Ali about his marks and store those values somewhere so we can calculate the final aggregate at the end.

In programming, there are “nicknames” that are known as “**variables**” that are used to store such values.

So, we shall use variables to store the obtained marks of Ali.

Let's name the variable as “**matric**” for storing matric marks.

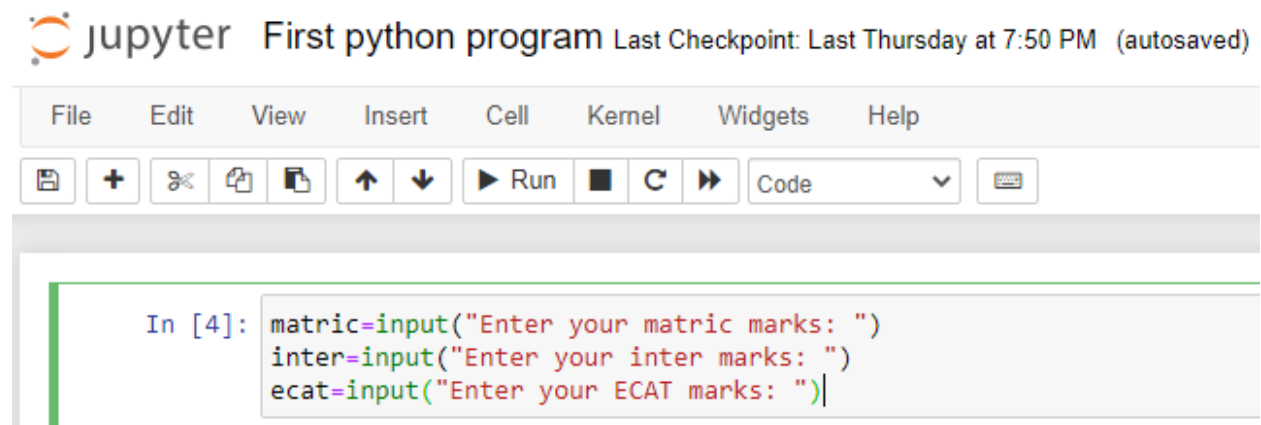
Let's name the variable as “**inter**” for storing first-year marks.

Let's name the variable as “**ecat**” for storing ecat marks.

Let's declare the variables.

```
matric;  
inter;  
ecat;
```

Now, ask Ali to input the values for these variables.



The screenshot shows a Jupyter Notebook window titled "First python program" with a last checkpoint of "Last Thursday at 7:50 PM (autosaved)". The interface includes a menu bar (File, Edit, View, Insert, Cell, Kernel, Widgets, Help) and a toolbar with icons for saving, adding cells, undo, redo, and running code. The code cell, labeled "In [4]:", contains the following Python code:

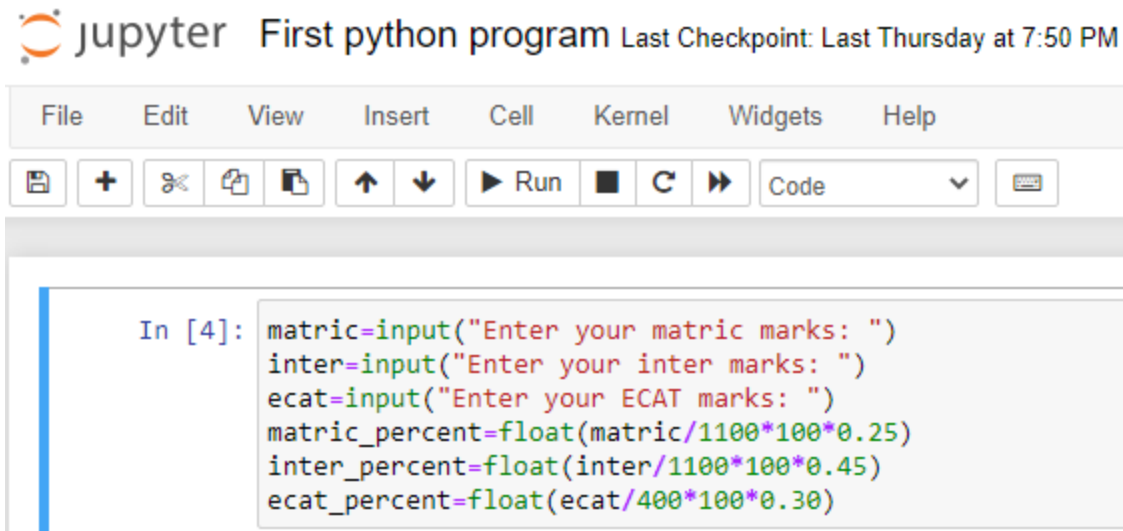
```
matric=input("Enter your matric marks: ")  
inter=input("Enter your inter marks: ")  
ecat=input("Enter your ECAT marks: ")|
```

Now, we need to calculate his aggregate.

This is where we shall incorporate the concepts of “**arithmetic expressions**”.

Now we need to calculate 25% of matric marks, 45% of first-year marks, and 30% of ecat marks to calculate Ali's final aggregate.

We shall more variables to calculate the respective percentage of each result.
Let's declare these variables as well and use the arithmetic expressions to calculate the percentage.

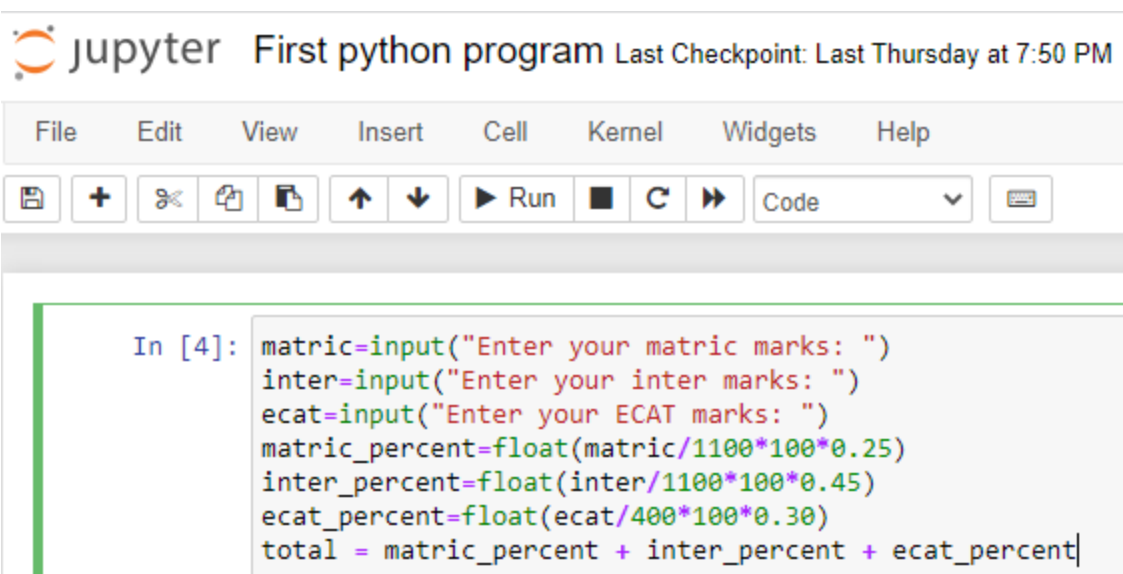


The image shows a Jupyter Notebook window titled "First python program" with a last checkpoint of "Last Thursday at 7:50 PM". The interface includes a menu bar (File, Edit, View, Insert, Cell, Kernel, Widgets, Help) and a toolbar with icons for saving, adding cells, undo, redo, and running code. The code cell contains the following Python code:

```
In [4]: matric=input("Enter your matric marks: ")
inter=input("Enter your inter marks: ")
ecat=input("Enter your ECAT marks: ")
matric_percent=float(matric/1100*100*0.25)
inter_percent=float(inter/1100*100*0.45)
ecat_percent=float(ecat/400*100*0.30)
```

You are almost there.
We have successfully built logic for your question.

Now, Let's add the three individual percentages to calculate the final aggregate.



The image shows the same Jupyter Notebook window, but the code cell now includes an additional line to calculate the total aggregate:

```
In [4]: matric=input("Enter your matric marks: ")
inter=input("Enter your inter marks: ")
ecat=input("Enter your ECAT marks: ")
matric_percent=float(matric/1100*100*0.25)
inter_percent=float(inter/1100*100*0.45)
ecat_percent=float(ecat/400*100*0.30)
total = matric_percent + inter_percent + ecat_percent|
```

Let's tell Ali about his calculated aggregate.

```
File Edit View Insert Cell Kernel Widgets Help
```

```

In [20]: matric=int(input("Enter your matric marks: "))
         inter=int(input("Enter your inter marks: "))
         ecac=int(input("Enter your ecac marks: "))
         matric_percent=float(((matric/1100)*100)*0.25)
         inter_percent=float(((inter/1100)*100)*0.45)
         ecac_percent=float(((ecac/400)*100)*0.30)
         aggregate = float(matric_percent) + float(inter_percent) + float(ecac_percent)
         print("Average: ", aggregate)

Enter your matric marks: 1025
Enter your inter marks: 1050
Enter your ecac marks: 325
Average: 90.625
    
```

“Congratulations, you have finished your first complete c++ program”.

Now, we shall use all these concepts to perform the tasks that are listed below.

Task # 1.

Write a python program to print the first three multiples of the given number.

For example, if the number is 3.

Enter the number: 3

The output should be

The multiples are: 3 6 9

Solution:

```
File Edit View Insert Cell Kernel Widgets Help
```

```

In [22]: number=int(input("Enter a number: "))
         print("First three multiples of", number, "are: ", number, number+number, number+number+number)

Enter a number: 3
First three multiples of 3 are: 3 6 9
    
```

Task # 2.

Write a python program to print the first three multiples of two given numbers.

For example, if the input is 3 and 5.

Enter the first number: 5

Enter the second number: 3

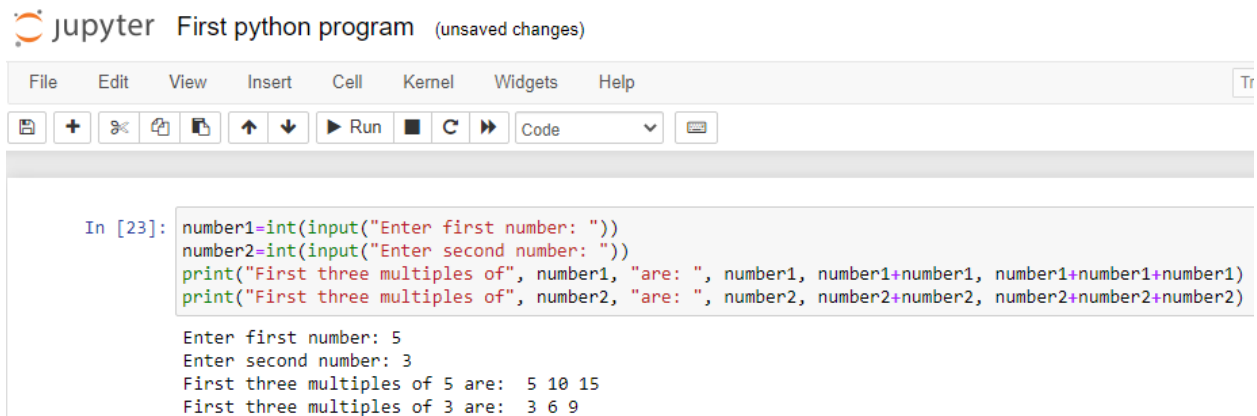
The output should be

The multiples of the first number are: 5 10 15

The multiples of the second number are: 3 6 9

Solution:

Output:



The screenshot shows a Jupyter Notebook window titled "First python program (unsaved changes)". The interface includes a menu bar (File, Edit, View, Insert, Cell, Kernel, Widgets, Help) and a toolbar with icons for file operations, running, and code execution. The code cell contains the following Python code:

```
In [23]: number1=int(input("Enter first number: "))
number2=int(input("Enter second number: "))
print("First three multiples of", number1, "are: ", number1, number1+number1, number1+number1+number1)
print("First three multiples of", number2, "are: ", number2, number2+number2, number2+number2+number2)
```

The output of the code is displayed below the code cell:

```
Enter first number: 5
Enter second number: 3
First three multiples of 5 are:  5 10 15
First three multiples of 3 are:  3 6 9
```

Task # 3.

Write a python program to print the sum of the first three multiples of two given numbers.

For example, if the input is 3 and 5.

Enter the first number: 3

Enter the second number: 5

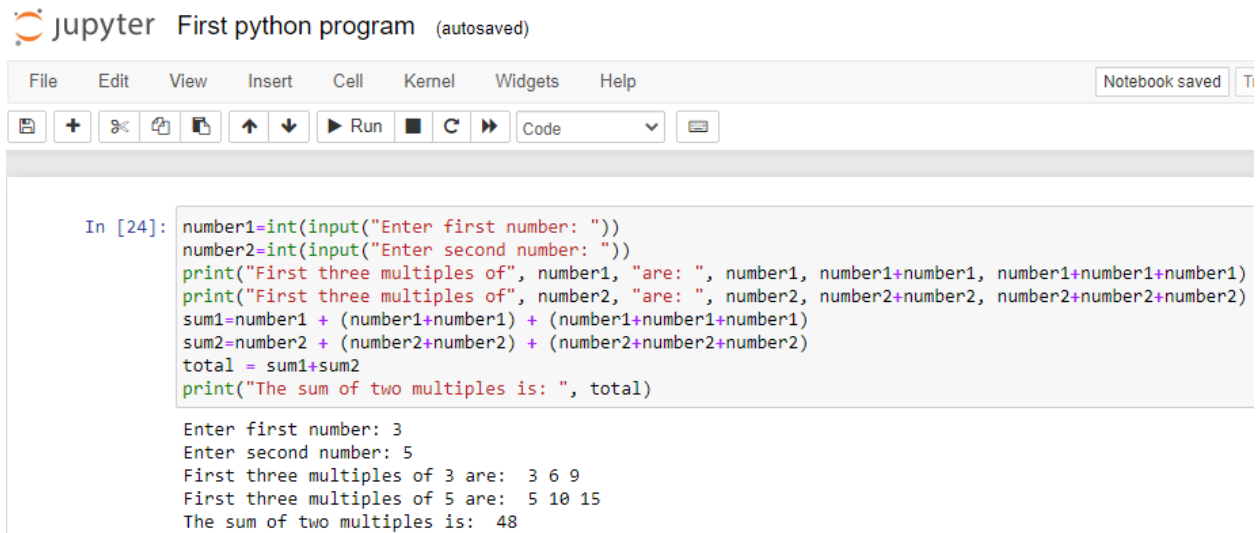
The output should be

The multiples of the first number are: 3 6 9

The multiples of the second number are: 5 10 15

The sum of the two multiples is: 48

Solution:



The image shows a Jupyter Notebook window titled "First python program (autosaved)". The interface includes a menu bar with File, Edit, View, Insert, Cell, Kernel, Widgets, and Help. Below the menu is a toolbar with icons for saving, adding cells, undo, redo, running, and other functions. The main area displays a code cell with the following Python code:

```
In [24]: number1=int(input("Enter first number: "))
number2=int(input("Enter second number: "))
print("First three multiples of", number1, "are: ", number1, number1+number1, number1+number1+number1)
print("First three multiples of", number2, "are: ", number2, number2+number2, number2+number2+number2)
sum1=number1 + (number1+number1) + (number1+number1+number1)
sum2=number2 + (number2+number2) + (number2+number2+number2)
total = sum1+sum2
print("The sum of two multiples is: ", total)
```

The output of the code is as follows:

```
Enter first number: 3
Enter second number: 5
First three multiples of 3 are: 3 6 9
First three multiples of 5 are: 5 10 15
The sum of two multiples is: 48
```

Task # 4.

The sequence of numbers (1, 2, 3, ... , 100) is arithmetic and when we are looking for the sum of a sequence, we call it a series. Thanks to Gauss, there is a special formula we can use to find the sum of a series:

$$S = \frac{n(n + 1)}{2}$$

Write a program that takes input from the user and prints the sum of consecutive numbers to the input value.

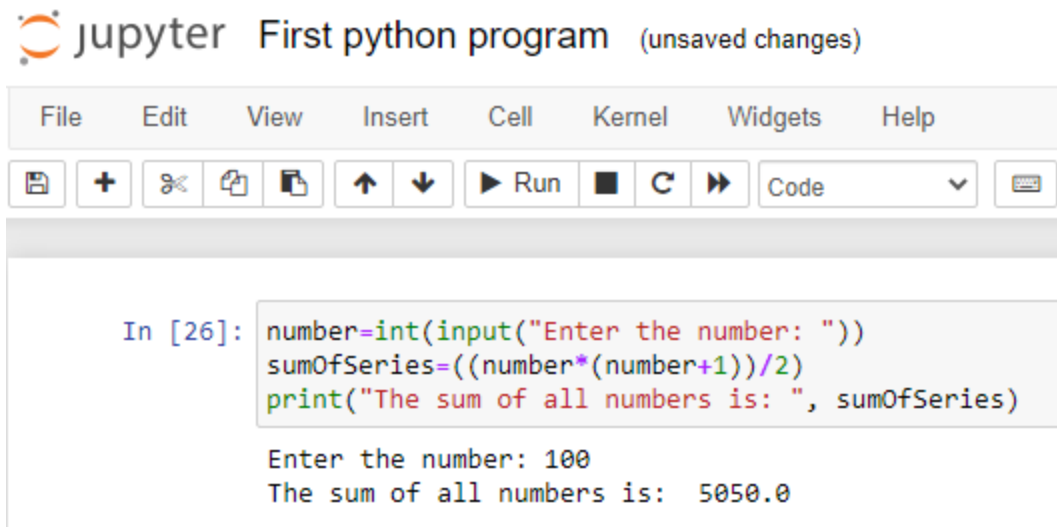
For example, if the input is

Enter the number: 100

The output should be:

The sum of all the numbers is: 5050

Solution:



The image shows a Jupyter Notebook window titled "First python program (unsaved changes)". The menu bar includes File, Edit, View, Insert, Cell, Kernel, Widgets, and Help. The toolbar contains icons for saving, adding cells, undo, redo, copy, paste, up/down arrows, a run button, a stop button, a refresh button, a dropdown menu set to "Code", and a console icon. The code cell contains the following Python code:

```
In [26]: number=int(input("Enter the number: "))
sumOfSeries=((number*(number+1))/2)
print("The sum of all numbers is: ", sumOfSeries)
```

The output of the code is:

```
Enter the number: 100
The sum of all numbers is: 5050.0
```

2Program #5.

Take two numbers as input and find the sum between these two numbers.

Input n1= 2

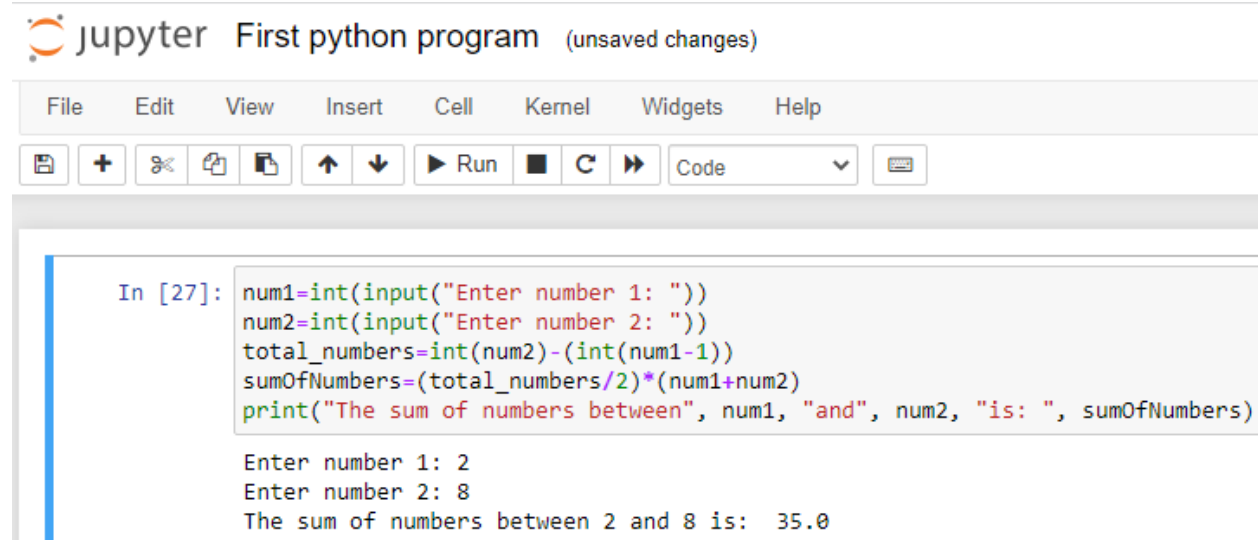
Input n2=8

Processing answer=2+3+4+5+6+7+8 = 35

Hint:

Formula: $(n / 2)(\text{first number} + \text{last number}) = \text{sum}$, where n is the number of integers between two numbers and $n = \text{last number} - (\text{first number} - 1)$

Solution:



The image shows a Jupyter Notebook window titled "First python program (unsaved changes)". The menu bar includes File, Edit, View, Insert, Cell, Kernel, Widgets, and Help. The toolbar contains icons for saving, adding cells, undo, redo, copy, paste, up/down arrows, a run button, a stop button, a refresh button, a dropdown menu set to "Code", and a console icon. The code cell contains the following Python code:

```
In [27]: num1=int(input("Enter number 1: "))
num2=int(input("Enter number 2: "))
total_numbers=int(num2)-(int(num1)-1)
sumOfNumbers=(total_numbers/2)*(num1+num2)
print("The sum of numbers between", num1, "and", num2, "is: ", sumOfNumbers)
```

The output of the code is:

```
Enter number 1: 2
Enter number 2: 8
The sum of numbers between 2 and 8 is: 35.0
```