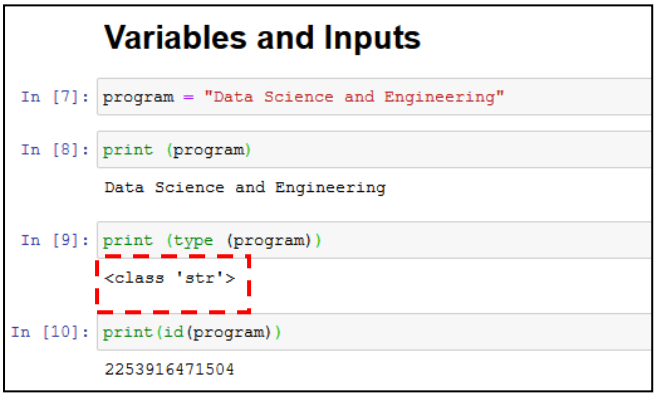
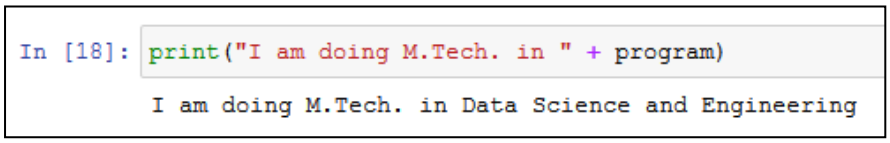
Objective: This worksheet provides a deeper insight into basic data types through variables and a mechanism to receive user inputs that is used in the program execution.

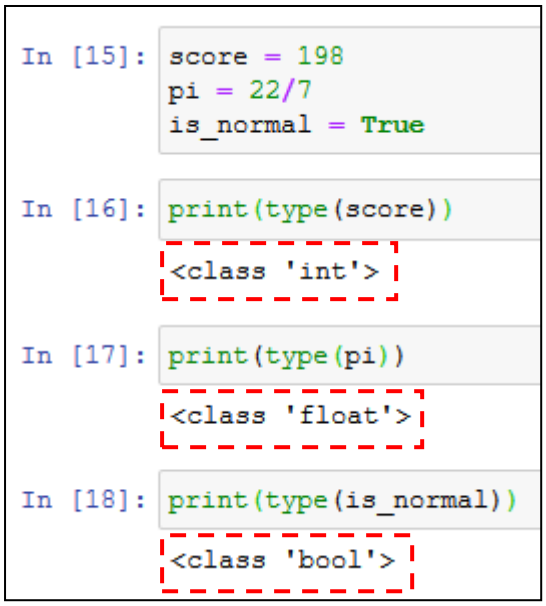
Different Basic Data Types and Variables



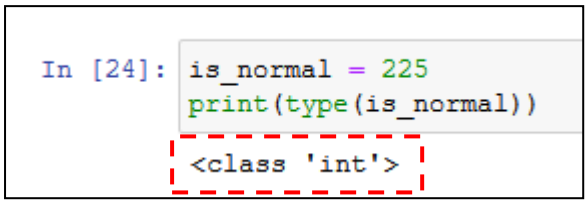
- Variables and Inputs is a heading created selecting the cell as heading with a preceding #.
- A variable ***program*** is assigned to a string ***"Data Science and Engineering"***. This string is an object which is stored in memory with some id. Here ***program*** variable has now become a reference to this particular string object.
- The function ***print()*** can directly be used on the variable ***program*** that will print the string.
- The variable ***program*** can be passed to another function ***type()*** and then to ***print()***. It prints that ***program*** variable belongs to string class.
- Similarly, the variable ***program*** can be passed to another function ***id ()*** and then to ***print()***. It prints the ***program*** variable's id; that is some address in the memory where this string object is stored.
- The operator + can be used with the variable ***program*** and another literal string. Literal string means a string that is not yet assigned to a variable. For example in the screenshot below, ***"I am doing M.Tech. in "*** is a literal string. The + operator concatenates (joins) both.



- Now we will examine the data type of integer and floating point numbers as well as boolean variables using ***type()*** functions as shown below:

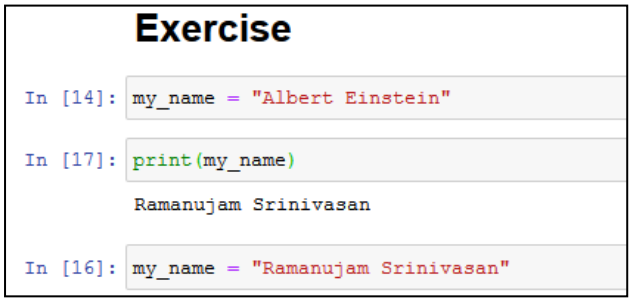


- Python is a dynamic data type programming language. A variable can dynamically assume the data type depending on what value it is last assigned to. For example, now ***is_normal*** is an integer because it is assigned to an integer value in the last.



Exercise:

1. Attempt and emulate the following. How and why did it happen?



Receiving User Input

- User can be prompted and its input can be saved in a variable using ***input()*** function as shown below. When this cell is executed, a *text box* is displayed and the program flow expects user to provide an input in the *text box*.

```
In [*]: my_name = input("What is your name: ")
        What is your name:
```

```
In [31]: my_name = input("What is your name: ")
         What is your name: JC Bose

In [32]: print(my_name)
         JC Bose
```

- Here one point needs to be kept in mind that whatever input is received from the text box is a **string**. It cannot be directly used for mathematical calculations. Look at the code below. Emulate it and can you answer why there is an error in the red marked portion below? The answer that your were expecting is 70 hours but it printed 10101010101010 (10 is written 7 times consecutively).

```
In [33]: my_name = input("What is your name: ")
         What is your name: JC Bose

In [35]: print(my_name)
         JC Bose

In [36]: hours_per_day = input("How many hours per day he used to work: ")
         How many hours per day he used to work: 10

In [41]: print(my_name + " used to work " + hours_per_day + " hours per day.")
         JC Bose used to work 10 hours per day.

In [43]: print(my_name + " used to work " + 7*hours_per_day + " hours per week.")
         JC Bose used to work 10101010101010 hours per week.
```

- The answer why this error has occurred is that ***hours_per_day*** is a variable that stores the user input and it is a string. When it is multiplied with a number, the string value will be replicated that many number times.

```
In [46]: print(type(hours_per_day))
         <class 'str'>
```

- Now the question is how the numeric user input can be received? This can be done using the ***int()*** function as shown below. The function converts the string input into an integer.

```
In [57]: hours_per_day = int(hours_per_day)
         print(my_name + " used to work " + (7*hours_per_day) + " hours per week.")

         ---------------------------------------------------------------------------
         TypeError                                 Traceback (most recent call last)
         <ipython-input-57-8f2ebc694cee> in <module>
               1 hours_per_day = int(hours_per_day)
         ----> 2 print(my_name + " used to work " + (7*hours_per_day) + " hours per week.")

         TypeError: can only concatenate str (not "int") to str
```

- The above code looks fine, then why did we get the error?
- If we notice the **TypeError;** it is fairly pin pointing that an integer cannot be concatenated with a string. So it means that although ***hours_per_day*** is converted in an integer as desired, it cannot be used to print the desired output. It needs to be converted into string back. This can be done using ***str()*** function as shown below:

```
In [58]: hours_per_day = int(hours_per_day)
         print(my_name + " used to work " + str(7*hours_per_day) + " hours per week.")

         JC Bose used to work 70 hours per week.
```

Exercise:

1. If user has entered that JC Bose used to work 10.5 hours per day (a floating point value), will you get similar errors? How will you fix it?