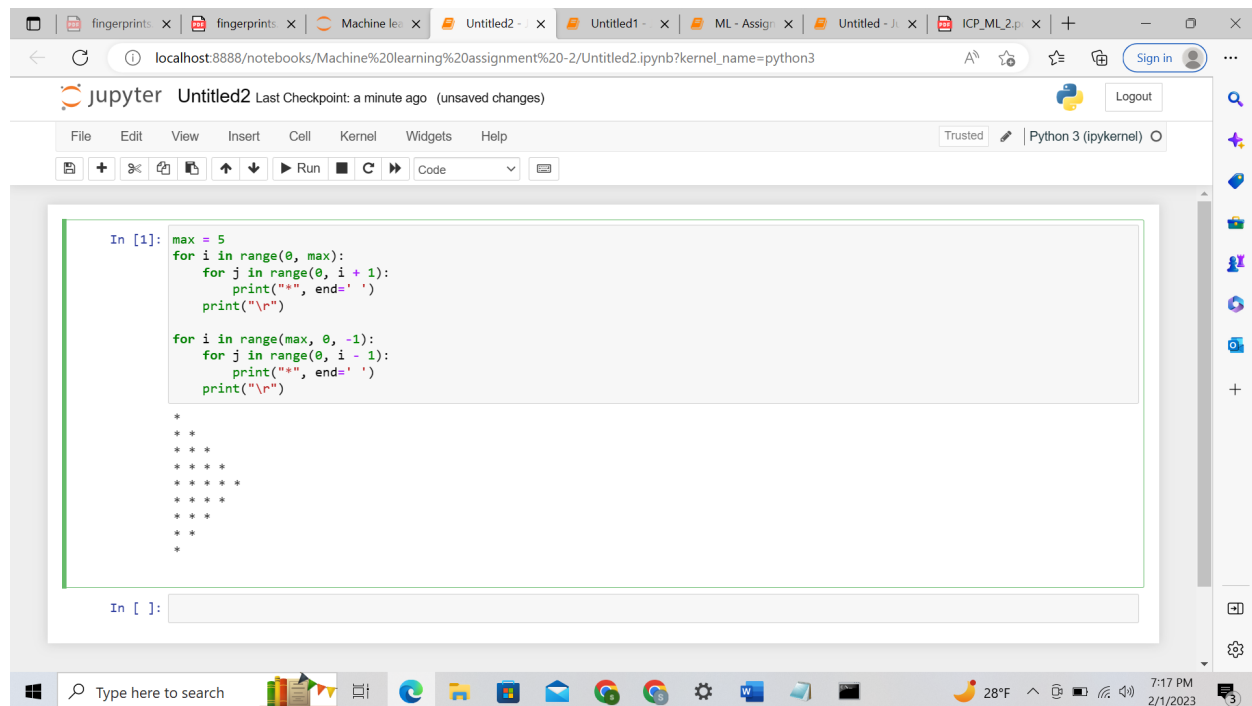


## Spring 2023: CS5710 – Machine Learning In-Class Programming Assignment-2

1. Use a python code to display the following star pattern using the for loop

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

Ans



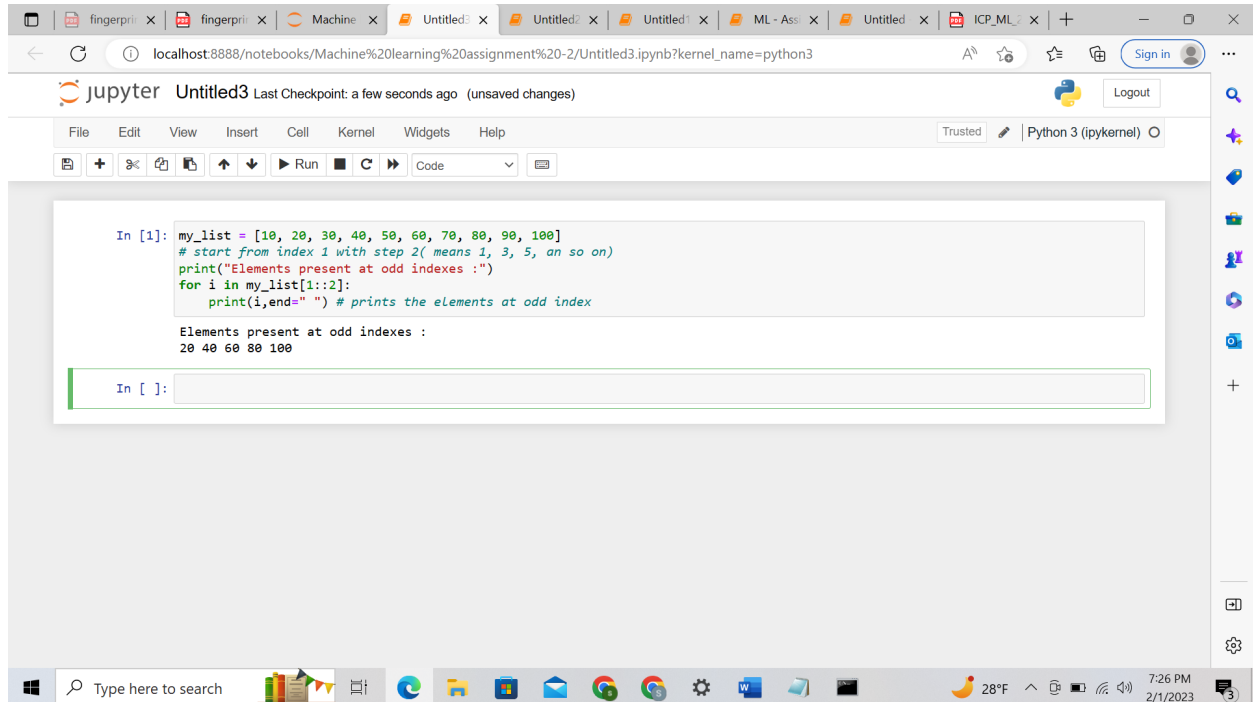
```
In [1]: max = 5  
for i in range(0, max):  
    for j in range(0, i + 1):  
        print("*", end=' ')  
    print("\n")  
  
for i in range(max, 0, -1):  
    for j in range(0, i - 1):  
        print("*", end=' ')  
    print("\n")  
  
*  
* *  
* * *  
* * * *  
* * * * *  
* * * *  
* * *  
* *  
*
```

- 1) To print the pattern the pattern as above I have used for loop
- 2) Here assigned the value for max as 5
- 3) Used 2 for loops, one inside the other
- 4) Used end=" " and "\n" to print the pattern as above.

2. Use looping to output the elements from a provided list present at odd indexes.

```
my_list = [10, 20, 30, 40, 50, 60, 70, 80, 90, 100]
```

Ans



The screenshot shows a Jupyter Notebook window titled 'Untitled3' with a Python 3 kernel. The code cell contains the following Python code:

```
In [1]: my_list = [10, 20, 30, 40, 50, 60, 70, 80, 90, 100]
# start from index 1 with step 2( means 1, 3, 5, an so on)
print("Elements present at odd indexes :")
for i in my_list[1::2]:
    print(i,end=" ") # prints the elements at odd index
```

The output of the code is displayed below the cell:

```
Elements present at odd indexes :
20 40 60 80 100
```

The Jupyter Notebook interface includes a menu bar (File, Edit, View, Insert, Cell, Kernel, Widgets, Help), a toolbar with icons for file operations and execution, and a status bar at the bottom showing the system clock and temperature.

- 1) my\_list has the input values of the problem
- 2) We have used the for loop to print the elements present at odd indexes.
- 3) Using print and end=" " all the unique elements are printed in the same line.

3. Write a code that appends the type of elements from a given list.

Input :

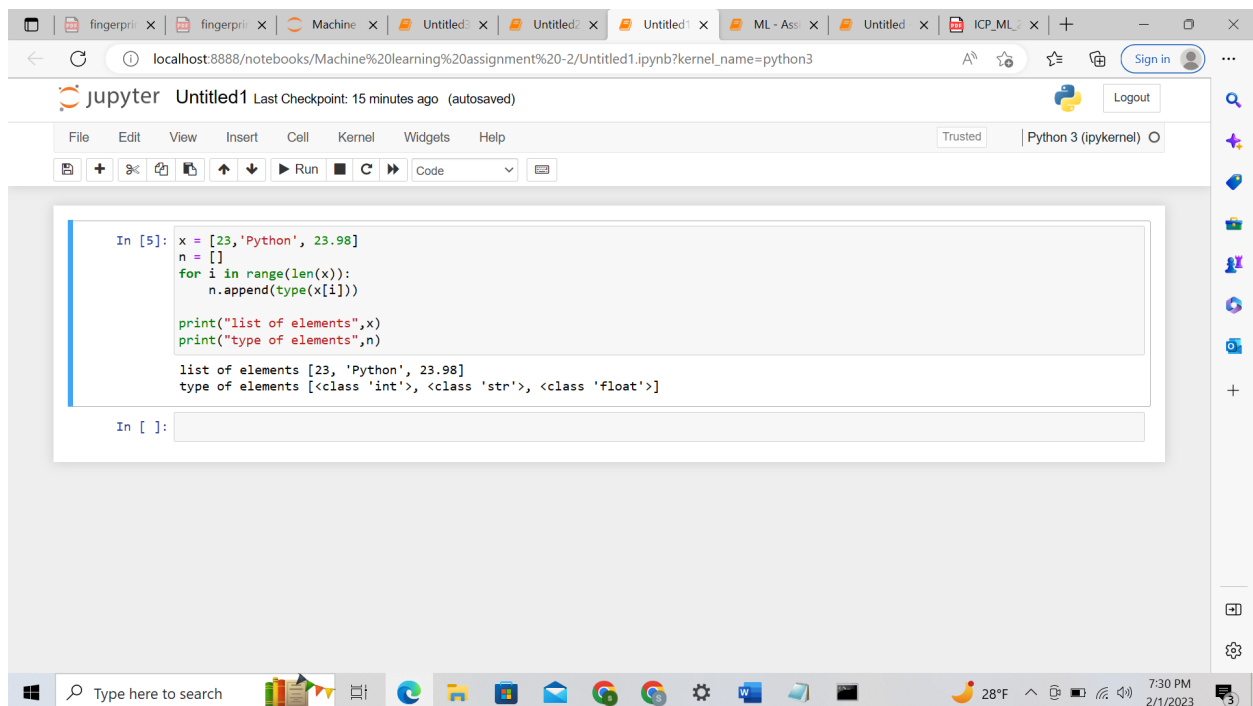
```
x = [23, 'Python', 23.98]
```

Expected output :

```
[23, 'Python', 23.98]
```

```
[<class 'int'>, <class 'str'>, <class 'float'>]
```

Ans

The screenshot shows a Jupyter Notebook window titled 'Untitled1' running on a local host. The code cell contains a list 'x' with elements 23, 'Python', and 23.98. A loop iterates over the list, and a new list 'n' is created to store the types of each element. The output of the code cell shows the list 'x' and the list 'n' containing the type objects for each element in 'x'.

```
In [5]: x = [23, 'Python', 23.98]
n = []
for i in range(len(x)):
    n.append(type(x[i]))

print("list of elements", x)
print("type of elements", n)

list of elements [23, 'Python', 23.98]
type of elements [<class 'int'>, <class 'str'>, <class 'float'>]
```

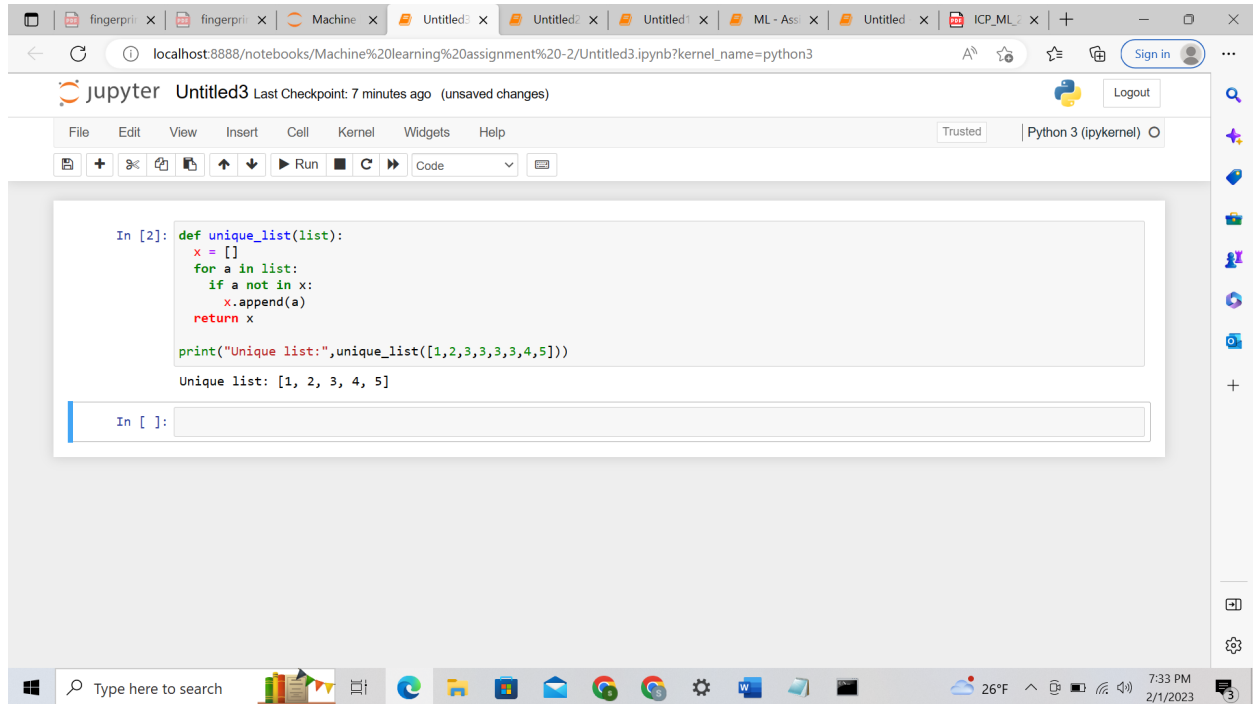
- 1) List X stored the elements 23,'Python',23.98.
- 2) We have used for loop and type() function to find out the type of each & every element in the list.
- 3) And appended those numbers consequently to a list.
- 4) Using print() function printed on the console.

4. Write a function that takes a list and returns a new list with unique items of the first list.

Sample List: [1,2,3,3,3,3,4,5]

Unique List: [1, 2, 3, 4, 5]

Ans



The screenshot shows a Jupyter Notebook window titled 'Untitled3' with a 'Python 3 (ipykernel)' kernel. The code in the notebook is as follows:

```
In [2]: def unique_list(list):  
        x = []  
        for a in list:  
            if a not in x:  
                x.append(a)  
        return x  
  
        print("Unique list:", unique_list([1,2,3,3,3,3,4,5]))  
  
        Unique list: [1, 2, 3, 4, 5]
```

The output of the code is 'Unique list: [1, 2, 3, 4, 5]'. The notebook interface includes a top toolbar with 'File', 'Edit', 'View', 'Insert', 'Cell', 'Kernel', 'Widgets', and 'Help' menus. The bottom status bar shows the system clock as 7:33 PM on 2/1/2023.

- 1) Created a function named `unique_list()` which takes the list as the input.
- 2) Inside the function using for loop and if condition , the unique elements are stored in the list `x[ ]`
- 3) The function `unique_list()` is called from the main in the print function
- 4) And the desired output is printed.

5. Write a function that accepts a string and calculate the number of upper-case letters and lower-case letters.

Input String:

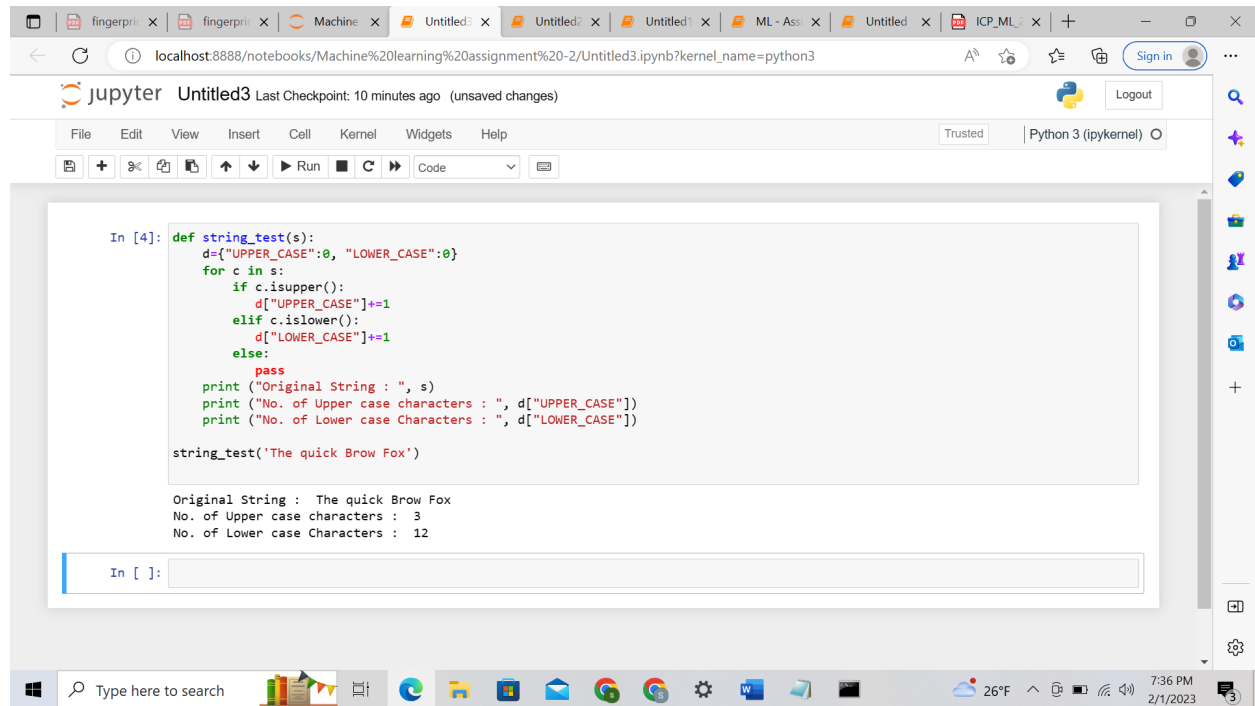
'The quick Brow Fox'

Expected Output:

No. of Upper-case characters: 3

No. of Lower-case Characters: 12

Ans



```
In [4]: def string_test(s):
d={"UPPER_CASE":0, "LOWER_CASE":0}
for c in s:
    if c.isupper():
        d["UPPER_CASE"]+=1
    elif c.islower():
        d["LOWER_CASE"]+=1
    else:
        pass
print ("Original String : ", s)
print ("No. of Upper case characters : ", d["UPPER_CASE"])
print ("No. of Lower case Characters : ", d["LOWER_CASE"])

string_test('The quick Brow Fox')

Original String : The quick Brow Fox
No. of Upper case characters : 3
No. of Lower case Characters : 12

In [ ]:
```

- 1) Created the string\_test() function to take string as input.
- 2) Using for loop and if else statement the number of upper case and lower case letters are retrieved and stored in key value pair
- 3) The number of Upper case characters are stored in 'UPPER\_CASE' key.
- 4) The number of Lower case characters are stored in 'LOWER\_CASE' key.
- 5) The output is printed using print() function.

Git hub link : <https://github.com/ShruthiVallapReddy/Machine-learning-assignment--2.git>