

```
In [1]: #1) Write a program to check whether an year entered by an user is Leap year or not
year = int(input("Enter a year: "))

check = ( (year % 4 == 0) and (year % 100 != 0) ) or (year % 400 == 0)
if check:
    print("{} is a leap year!".format(year))
else:
    print("{} is not a leap year!".format(year))
```

Enter a year: 2020
2020 is a leap year!

```
In [2]: #2) Take a string containing punctuation marks as an input from user and write a program
str_punc = input("Enter a string with punctuation marks: ")
punctuation = ""!"()-[]{};:'"\,<>./?@$%^&*~""

for i in str_punc:
    if i in punctuation:
        str_punc = str_punc.replace(i, "")

print("Without punctuation: ", str_punc)
```

Enter a string with punctuation marks: Hello world!! How are you? I am, fine.
Without punctuation: Hello world How are you I am fine

```
In [1]: #3) Take two separate numbers as input from the user and Write a program to swap 2 numbers
num1 = int(input("Enter 1st Number: "))
num2 = int(input("Enter 2nd Number: "))

num1, num2 = num2, num1

print("1st Number: ", num1)
print("2nd Number: ", num2)
```

Enter 1st Number: 17
Enter 2nd Number: 15
1st Number: 15
2nd Number: 17

```
In [3]: #4) Write a program to print product of following variables
num_int = 10
num_str = "15"

product = num_int * int(num_str)

print("Product of ", num_int, " and ", num_str, " is ", product)
```

Product of 10 and 15 is 150

```
In [15]: #5) Write a program to print following pattern
# 65-90: capital Alpha Ascii, 97-122: smaller alpha ascii, 48-56: Numbers ascii

#without ascii
pattern = "abcdefghijklmnopqrstu"

print("Without Ascii:")
count = 0
for i in range(6):
    for j in range(i + 1):
        print(pattern[count], end=" ")
        count += 1
```

```

print()

print()

#with ascii
print("With Ascii:")
asciichr = 97
for i in range(6):
    for j in range(i + 1):
        char = chr(asciichr)
        print(char, end=" ")
        asciichr += 1
    print()

```

Without Ascii:

```

a
b c
d e f
g h i j
k l m n o
p q r s t u

```

With Ascii:

```

a
b c
d e f
g h i j
k l m n o
p q r s t u

```

In [18]: *#6) Write a program to print tables from 2 to n, where n is provided by user*

```

n = int(input("Enter tables limit: "))

for i in range(2, n + 1):
    print("Table of ", i, ": ")
    for j in range(1, 11):
        m = i * j
        print(i, " * ", j, " = ", m)
    print()

```

Enter tables limit: 5

Table of 2 :

```

2 * 1 = 2
2 * 2 = 4
2 * 3 = 6
2 * 4 = 8
2 * 5 = 10
2 * 6 = 12
2 * 7 = 14
2 * 8 = 16
2 * 9 = 18
2 * 10 = 20

```

Table of 3 :

```

3 * 1 = 3
3 * 2 = 6
3 * 3 = 9
3 * 4 = 12
3 * 5 = 15
3 * 6 = 18
3 * 7 = 21
3 * 8 = 24
3 * 9 = 27
3 * 10 = 30

```

Table of 4 :

| | | | | |
|---|---|----|---|----|
| 4 | * | 1 | = | 4 |
| 4 | * | 2 | = | 8 |
| 4 | * | 3 | = | 12 |
| 4 | * | 4 | = | 16 |
| 4 | * | 5 | = | 20 |
| 4 | * | 6 | = | 24 |
| 4 | * | 7 | = | 28 |
| 4 | * | 8 | = | 32 |
| 4 | * | 9 | = | 36 |
| 4 | * | 10 | = | 40 |

Table of 5 :

| | | | | |
|---|---|----|---|----|
| 5 | * | 1 | = | 5 |
| 5 | * | 2 | = | 10 |
| 5 | * | 3 | = | 15 |
| 5 | * | 4 | = | 20 |
| 5 | * | 5 | = | 25 |
| 5 | * | 6 | = | 30 |
| 5 | * | 7 | = | 35 |
| 5 | * | 8 | = | 40 |
| 5 | * | 9 | = | 45 |
| 5 | * | 10 | = | 50 |

In [19]: *#7) Write a program to read height (in centimeters) from user and then convert it to feet*

```
height = float(input("Enter your height: "))

h inches = 0.394 * height
h feet = 0.0328 * height

print("Height in inches: ", h inches)
print("Height in feet: ", h feet)
```

Enter your height: 153
 Height in inches: 60.282000000000004
 Height in feet: 5.018400000000001

In [36]: *#8) Write a program to print following pattern*

```
n = int(input("Enter number of rows: "))
for i in range(n):
    for j in range(i + 1):
        print("*", end=" ")
    print()

for i in range(n, 0, -1):
    for j in range(i - 1):
        print("*", end=" ")
    print()
```

Enter number of rows: 3

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

In [37]: *#9) Below List contains age and name of five students. Convert the List into a dictionary*

```
students = [26, 'Jon', 34, 'Roger', 13, 'Jessica', 24, 'Monica', 65, 'Ashton']

#Method 1
keys = students[1::2]
```

```

values = students[0::2]

student_dict = dict(zip(keys, values))
print(student_dict)

#Method 2
def Convert(lst):
    dct = { lst[i + 1]: lst[i] for i in range(0, len(lst), 2) }
    return dct

student_dict = Convert(students)
print(student_dict)

```

```

{'Jon': 26, 'Roger': 34, 'Jessica': 13, 'Monica': 24, 'Ashton': 65}
{'Jon': 26, 'Roger': 34, 'Jessica': 13, 'Monica': 24, 'Ashton': 65}

```

In [24]: *#10) Write a program to check whether two strings, taken from user, are anagram or not*

```

str1 = input("Enter 1st string: ")
str2 = input("Enter 2nd string: ")

if sorted(str1.lower()) == sorted(str2.lower()):
    print("{0} and {1} are anagram!".format(str1, str2))
else:
    print("{0} and {1} are not anagram!".format(str1, str2))

```

```

Enter 1st string: Listen
Enter 2nd string: Silent
Listen and Silent are anagram!

```

In [38]: *#11) Write a program that sorts all the elements of the list according to the first letter*

```

fruits = ["Mango", "Cherry", "Grape", "Apple", "Orange"]
n = len(fruits)

for i in range(n - 1): #0, 1, 2, 3, 4
    for j in range(n-i-1): #4, 3, 2, 1, 0
        if fruits[j].lower() > fruits[j + 1].lower():
            fruits[j], fruits[j + 1] = fruits[j + 1], fruits[j]

print(fruits)

```

```

['Apple', 'Cherry', 'Grape', 'Mango', 'Orange']

```

In [42]: *#12) Write a program that prints a list of sour fruits from the below tuple:*

```

fruits = (('Lemon', 'sour'), ('DragonFruit', 'Sweet'), ('Grapes', 'soUr'), ('Kiwi', 'Sour'))
sour_fruits = []

for k, f in fruits:
    if f.lower() == "sour":
        sour_fruits.append(k)

print("Sour Fruits: ", sour_fruits)

```

```

Sour Fruits: ['Lemon', 'Grapes', 'Kiwi', 'Orange', 'Limes']

```

In [44]: *#13) Given below are the list of positive and negative words. Also a list of tweets is*

```

positive = ['good', 'awesome', 'best', 'nice']
negative = ['worst', 'awful', 'bad']

tweets = ["This government policies are good", "bad implementation", "The way he played"]
tweet_dict = {"positive": [], "negative": []}

for tweet in tweets:

```

```

for word in tweet.split():
    if word.lower() in positive:
        tweet_dict["positive"].append(tweet)
    elif word.lower() in negative:
        tweet_dict["negative"].append(tweet)

print("Positive Tweets are: ", tweet_dict["positive"])
print("Negative Tweets are: ", tweet_dict["negative"])

```

Positive Tweets are: ['This government policies are good', 'The way he played showed th
at he is one of the best players in the world', 'Her acting in the play was awesome', "I
t's nice to hear this little kid's laugh"]
Negative Tweets are: ['bad implementation', 'The wine tastes awful']

In [3]: *#14) Write a program that performs multiplication of the following two matrix*
import numpy **as** np

```

X = [[1, 2, 3],
      [4, 5, 6],
      [7, 8, 9]]

Y = [[1, 8, 1],
      [3, 7, 3],
      [4, 5, 9]]

X_arr = np.array(X)
Y_arr = np.array(Y)

multiplication = X_arr * Y_arr
dot_product = np.dot(X_arr, Y_arr)

print("Multiplication: ")
print(multiplication)

print("Dot Product: ")
print(dot_product)

```

Multiplication:
[[1 16 3]
[12 35 18]
[28 40 81]]
Dot Product:
[[19 37 34]
[43 97 73]
[67 157 112]]

In [59]: *#15) Write a program to remove space from keys of the given dictionary*
employee = {'emp 01' : 'Raj', 'emp 02' : 'Sheela', 'emp 0 3 ' : 'Joseph'}
new_employee = {}

```

for k, v in employee.items():
    new_employee[k.replace(" ", "")] = v

print(new_employee)

```

{'emp01': 'Raj', 'emp02': 'Sheela', 'emp03': 'Joseph'}

In [2]: *#16) Write a program to display the subjects for which the scores are between 300 and 5*
import pandas **as** pd

```

subjects = {"Subject": ["Maths", "Physics", "Chemistry", "Biology", "Geology"], "Scores
df_subjects = pd.DataFrame(subjects)

```

```
df_subjects[(df_subjects["Scores"] >= 300) & (df_subjects["Scores"] <= 500)]
```

Out[2]:

| | Subject | Scores |
|---|-----------|--------|
| 0 | Maths | 345 |
| 1 | Physics | 432 |
| 2 | Chemistry | 456 |

In [63]:

```
#17) Write a program to find numbers between 10000 to 20000 from a given array
Sales = [1432, 143223, 2043332, 1642357, 54327, 754338, 12459, 75435, 224454, 1433, 124
sales_arr = np.array(Sales)

print(sales_arr[(sales_arr >= 10000) & (sales_arr <= 20000)])
```

[12459 12447 12678]

In [74]:

```
#18) Write a program to sort the dataframe by Rank (in ascending order)
students = {"Name": ["Joy", "Mia", "Karan", "Tom"], "Rank": [115, 451, 341, 84]}
df_students = pd.DataFrame(students)

df_students.sort_values(["Rank"])
```

Out[74]:

| | Name | Rank |
|---|-------|------|
| 3 | Tom | 84 |
| 0 | Joy | 115 |
| 2 | Karan | 341 |
| 1 | Mia | 451 |

In [75]:

```
#19) Use the data given in the question 18, to do the following: Write a program to swap
print(df_students)

df_students.loc[0], df_students.loc[1] = df_students.loc[1], df_students.loc[0]

df_students
```

| | Name | Rank |
|---|-------|------|
| 0 | Joy | 115 |
| 1 | Mia | 451 |
| 2 | Karan | 341 |
| 3 | Tom | 84 |

Out[75]:

| | Name | Rank |
|---|-------|------|
| 0 | Mia | 451 |
| 1 | Joy | 115 |
| 2 | Karan | 341 |
| 3 | Tom | 84 |

In [77]:

```
#20) Write a program to replace 0 and 1 in Gender variable with M and F
gender_data = {"Id": np.arange(1,5), "Name": ["Mia", "Sara", "Joy", "Karan"], "Gender":
df_gender = pd.DataFrame(gender_data)
```

```
df_gender["Gender"] = df_gender["Gender"].replace(to_replace=[1, 0], value=["F", "M"])
df_gender
```

Out[77]:

| | Id | Name | Gender |
|----------|-----------|-------------|---------------|
| 0 | 1 | Mia | F |
| 1 | 2 | Sara | F |
| 2 | 3 | Joy | M |
| 3 | 4 | Karan | M |

```
In [47]: #21) Convert a 1D array to a 2D array containing 2 rows
dim_1 = np.array([331, 245, 246, 475, 348, 229])

dim_1 = dim_1.reshape(2, 3)
dim_1
```

```
Out[47]: array([[331, 245, 246],
               [475, 348, 229]])
```

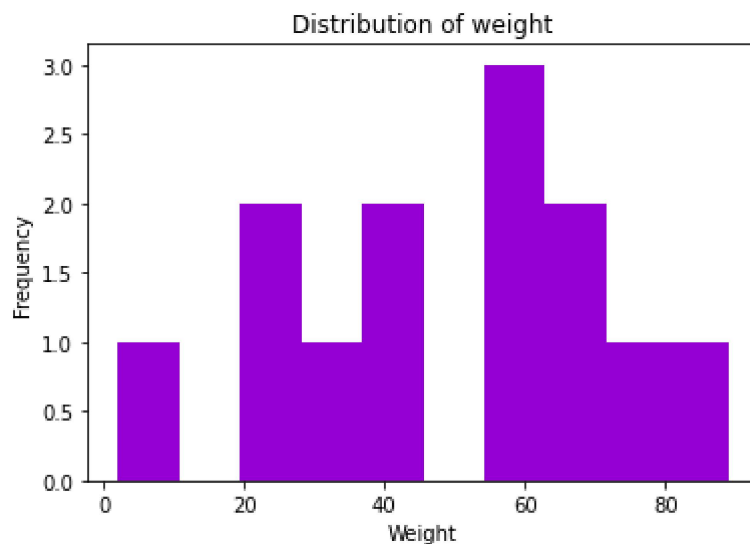
```
In [27]: #22) Find the average cost of each variety of product available in the three different
sales_df = pd.DataFrame([[10, 20, 30], [17, 14, 21], [25, 15, 12]], columns=['product_a',
print(sales_df.mean())
```

```
product_a    17.333333
product_b    16.333333
product_c    21.000000
dtype: float64
```

```
In [30]: #23) Write a program to create a histogram using given data
import matplotlib.pyplot as plt

weight = [34,56,43,22,2,58,89,78,67,56,65,43,23]
df_weight = pd.Series(weight)

plt.hist(x=df_weight, color="darkviolet")
plt.title("Distribution of weight")
plt.xlabel("Weight")
plt.ylabel("Frequency")
plt.show()
```



```
In [37]: #24) Write a program to delete items from series_a that are present in series_b and print series_a
series_a = pd.Series([11, 12, 13, 14, 15])
series_b = pd.Series([14, 15, 16, 17, 18])

series_a = pd.Series(np.setdiff1d(series_a, series_b))
series_a
```

```
Out[37]: 0    11
         1    12
         2    13
dtype: int64
```

```
In [42]: #25) Create a series of 40 random numbers and transform it into a dataframe containing
ser = pd.Series(np.random.random(size=40))
ser_df = pd.DataFrame(ser.values.reshape(5, 8))
ser_df
```

```
Out[42]:
```

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|----------|----------|----------|----------|----------|----------|----------|----------|
| 0 | 0.108897 | 0.345569 | 0.385397 | 0.333040 | 0.664339 | 0.050035 | 0.060078 | 0.107290 |
| 1 | 0.731980 | 0.522058 | 0.983275 | 0.850354 | 0.431804 | 0.293495 | 0.213857 | 0.892084 |
| 2 | 0.873360 | 0.108110 | 0.703995 | 0.004795 | 0.060764 | 0.352044 | 0.412313 | 0.090656 |
| 3 | 0.066782 | 0.867666 | 0.332604 | 0.555976 | 0.147835 | 0.028657 | 0.036007 | 0.959599 |
| 4 | 0.032164 | 0.705055 | 0.700672 | 0.381422 | 0.737426 | 0.366489 | 0.383652 | 0.464642 |

```
In [45]: #26) Get common elements from the two arrays
test_a = np.array([11, 12, 13, 22, 23, 44, 43, 24, 45, 36])
test_b = np.array([17, 22, 13, 22, 47, 34, 39, 44, 29, 28])

print(np.intersect1d(test_a, test_b))

[13 22 44]
```

```
In [49]: #27) Import dataset 'flights' from library seaborn. Check for datatypes of all variables
import seaborn as sns

df = sns.load_dataset('flights')
print(df.head())
df.dtypes
```

```
   year month  passengers
0  1949   Jan           112
1  1949   Feb           118
2  1949   Mar           132
3  1949   Apr           129
4  1949   May           121
```

```
Out[49]: year           int64
month         category
passengers    int64
dtype: object
```

```
In [52]: #28) Compute total sales for each product and country
sales_dic = {"Product": ["A", "J", "A", "L", "R", "A", "R", "J", "L", "A"],
              "Country": ["USA", "Japan", "Europe", "Japan", "Europe", "USA", "USA", "Ja",
                           "Sales": [44000, 43000, 54000, 95600, 79000, 67000, 68000, 38000, 59000, 6
```



```
sales_df = pd.DataFrame(sales_dic)
sales_df.groupby(by=["Product", "Country"])["Sales"].sum()
```

```
Out[52]: Product Country
A      Europe    114000
      USA      111000
J      Japan     81000
L      Japan     95600
      USA      59000
R      Europe     79000
      USA      68000
Name: Sales, dtype: int64
```

```
In [53]: #29) Get the descriptive statistics of the sales for each Country
sales_df.describe()
```

```
Out[53]:
```

| | Sales |
|--------------|--------------|
| count | 10.000000 |
| mean | 60760.000000 |
| std | 17643.330499 |
| min | 38000.000000 |
| 25% | 46500.000000 |
| 50% | 59500.000000 |
| 75% | 67750.000000 |
| max | 95600.000000 |

```
In [56]: #30) Reverse each of the rows of the 2D array
num_array = np.arange(20).reshape(4,5)

print("Array: ")
print(num_array)

print()

print("Reversed Array: ")
print(np.flip(num_array))
```

```
Array:
[[ 0  1  2  3  4]
 [ 5  6  7  8  9]
 [10 11 12 13 14]
 [15 16 17 18 19]]
```

```
Reversed Array:
[[19 18 17 16 15]
 [14 13 12 11 10]
 [ 9  8  7  6  5]
 [ 4  3  2  1  0]]
```

```
In [ ]:
```