Example programs

To print HELLO WORLD

```
# This program prints Hello, world!

print('Hello, world!')
```

Output: `Hello, world!`

Add Two Numbers

```
# This program adds two numbers

num1 = 1.5
num2 = 6.3

# Add two numbers
sum = num1 + num2

# Display the sum
print('The sum of {0} and {1} is {2}'.format(num1, num2, sum))
```

Output

```
The sum of 1.5 and 6.3 is 7.8
```

Add Two Numbers With User Input

```
# Store input numbers
num1 = input('Enter first number: ')
num2 = input('Enter second number: ')

# Add two numbers
sum = float(num1) + float(num2)

# Display the sum
print('The sum of {0} and {1} is {2}'.format(num1, num2, sum))
```

Output

```
Enter first number: 1.5
Enter second number: 6.3
```

```
The sum of 1.5 and 6.3 is 7.8
```

For positive numbers- calculate the square root

```
# Python Program to calculate the square root

# Note: change this value for a different result
num = 8

# To take the input from the user
#num = float(input('Enter a number: '))

num_sqrt = num ** 0.5
print('The square root of %0.3f is %0.3f'%(num ,num_sqrt))
```

Output

```
The square root of 8.000 is 2.828
```

For real or complex numbers - Find square root

```
# Find square root of real or complex numbers
# Importing the complex math module
import cmath

num = 1+2j

# To take input from the user
#num = eval(input('Enter a number: '))

num_sqrt = cmath.sqrt(num)
print('The square root of {0} is {1:0.3f}+{2:0.3f}j'.
format(num ,num_sqrt.real,num_sqrt.imag))
```

Output

```
The square root of (1+2j) is 1.272+0.786j
```

Find the AREA OF TRIANGLE

```
# Python Program to find the area of triangle

a = 5
b = 6
c = 7

# Uncomment below to take inputs from the user
# a = float(input('Enter first side: '))
# b = float(input('Enter second side: '))
# c = float(input('Enter third side: '))

# calculate the semi-perimeter
s = (a + b + c) / 2

# calculate the area
area = (s*(s-a)*(s-b)*(s-c)) ** 0.5
print('The area of the triangle is %0.2f' %area)
```

Output

```
The area of the triangle is 14.70
```

Generate random number

```
# Program to generate a random number between 0 and 9

# importing the random module
import random

print(random.randint(0,9))
```

Run Code

Output

```
5
```

Kilometers to Miles

```
# Taking kilometers input from the user
kilometers = float(input("Enter value in kilometers: "))

# conversion factor
conv_fac = 0.621371

# calculate miles
miles = kilometers * conv_fac
print('%0.2f kilometers is equal to %0.2f miles' %(kilometers,miles))
```

Output

```
Enter value in kilometers: 3.5
3.50 kilometers is equal to 2.17 miles
```

Solve Quadratic Equation

```
# import complex math module
import cmath

a = 1
b = 5
c = 6

# calculate the discriminant
d = (b**2) - (4*a*c)

# find two solutions
sol1 = (-b-cmath.sqrt(d))/(2*a)
sol2 = (-b+cmath.sqrt(d))/(2*a)

print('The solution are {0} and {1}'.format(sol1,sol2))
```

Output

```
Enter a: 1
Enter b: 5
Enter c: 6
The solutions are (-3+0j) and (-2+0j)
```

convert temperature in Celsius to Fahrenheit

```
# change this value for a different result
celsius = 37.5

# calculate fahrenheit
fahrenheit = (celsius * 1.8) + 32
print('%0.1f degree Celsius is equal to %0.1f degree Fahrenheit'
%(celsius,fahrenheit))
```

Run Code

Output

```
37.5 degree Celsius is equal to 99.5 degree Fahrenheit
```

Swap two numbers Using a temporary variable

```
# Python program to swap two variables

x = 5
y = 10

# To take inputs from the user
#x = input('Enter value of x: ')
#y = input('Enter value of y: ')

# create a temporary variable and swap the values
temp = x
x = y
y = temp

print('The value of x after swapping: {}'.format(x))
print('The value of y after swapping: {}'.format(y))
```

Output

```
The value of x after swapping: 10
The value of y after swapping: 5
```

Swap two numbers Without Using Temporary Variable

```
x = 5
y = 10

x, y = y, x
print("x =", x)
print("y =", y)
```

Addition and Subtraction

```
x = x + y
y = x - y
x = x - y
```

Multiplication and Division

```
x = x * y
y = x / y
x = x / y
```

XOR swap

This algorithm works for integers only

```
x = x ^ y
y = x ^ y
x = x ^ y
```

Check if a Number is Positive, Negative or 0

Using if...elif...else

```
num = float(input("Enter a number: "))
if num > 0:
   print("Positive number")
elif num == 0:
   print("Zero")
else:
   print("Negative number")
```

Output

```
Enter a number: 2
Positive number
```

Using Nested if

```
num = float(input("Enter a number: "))
if num >= 0:
   if num == 0:
       print("Zero")
   else:
       print("Positive number")
else:
   print("Negative number")
```

Output

```
Enter a number: 0
Zero
```

Check if a Number is Odd or Even

```
A number is even if division by 2 gives a remainder of 0.
# If the remainder is 1, it is an odd number.

num = int(input("Enter a number: "))
if (num % 2) == 0:
   print("{0} is Even".format(num))
else:
   print("{0} is Odd".format(num))
```

Output 1

```
Enter a number: 43
43 is Odd
```

Output 2

```
Enter a number: 18
18 is Even
```

Find the Largest Among Three Numbers

```
# uncomment following lines to take three numbers from user
num1 = float(input("Enter first number: "))
num2 = float(input("Enter second number: "))
num3 = float(input("Enter third number: "))

if (num1 >= num2) and (num1 >= num3):
   largest = num1
elif (num2 >= num1) and (num2 >= num3):
   largest = num2
else:
   largest = num3

print("The largest number is", largest)
```

Output

```
The largest number is 14.0
```

CHECK LEAP YEAR

```
# Python program to check if year is a leap year or not

year = int(input("Enter a year: "))

# divided by 100 means century year (ending with 00)
# century year divided by 400 is leap year

if (year % 400 == 0) and (year % 100 == 0):
    print("{0} is a leap year".format(year))

# not divided by 100 means not a century year
# year divided by 4 is a leap year
elif (year % 4 ==0) and (year % 100 != 0):
    print("{0} is a leap year".format(year))

# if not divided by both 400 (century year) and 4 (not century year)
# year is not leap year
else:
    print("{0} is not a leap year".format(year))
```

Output

```
2000 is a leap year
```

Python program to find the largest number among the three input numbers

```
num1 = float(input("Enter first number: "))
num2 = float(input("Enter second number: "))
num3 = float(input("Enter third number: "))

if (num1 >= num2) and (num1 >= num3):
   largest = num1
elif (num2 >= num1) and (num2 >= num3):
   largest = num2
else:
   largest = num3

print("The largest number is", largest)
```

Output

```
The largest number is 14.0
```

Program to check if a number is prime or not

```
num = int(input("Enter a number: "))

# define a flag variable
flag = False

# prime numbers are greater than 1
if num > 1:
    # check for factors
    for i in range(2, num):
        if (num % i) == 0:
            # if factor is found, set flag to True
            flag = True
            # break out of loop
            break

# check if flag is True
if flag:
    print(num, "is not a prime number")
else:
    print(num, "is a prime number")
```

Output

```
407 is not a prime number
11 times 37 is 407
```

Python program to find the factorial of a number provided by the user.

```
num = int(input("Enter a number: "))

factorial = 1

# check if the number is negative, positive or zero
if num < 0:
   print("Sorry, factorial does not exist for negative numbers")
elif num == 0:
   print("The factorial of 0 is 1")
else:
   for i in range(1,num + 1):
       factorial = factorial*i
   print("The factorial of",num,"is",factorial)
```

Output

```
The factorial of 7 is 5040
```

Multiplication table (from 1 to 10) in Python

```
# Multiplication table (from 1 to 10) in Python

num = 12

# To take input from the user
# num = int(input("Display multiplication table of? "))

# Iterate 10 times from i = 1 to 10
for i in range(1, 11):
   print(num, 'x', i, '=', num*i)
```

Output

```
12 x 1 = 12
12 x 2 = 24
12 x 3 = 36
12 x 4 = 48
12 x 5 = 60
12 x 6 = 72
12 x 7 = 84
12 x 8 = 96
12 x 9 = 108
12 x 10 = 120
```

Program to display the Fibonacci sequence up to n-th term

```
nterms = int(input("How many terms? "))

# first two terms
n1, n2 = 0, 1
count = 0

# check if the number of terms is valid
if nterms <= 0:
   print("Please enter a positive integer")
# if there is only one term, return n1
elif nterms == 1:
   print("Fibonacci sequence upto",nterms,":")
   print(n1)
# generate fibonacci sequence
else:
   print("Fibonacci sequence:")
   while count < nterms:
       print(n1)
       nth = n1 + n2
       # update values
       n1 = n2
       n2 = nth
       count += 1
```

Output

```
How many terms? 7
Fibonacci sequence:
0
1
1
2
3
5
8
```

Python program to check if the number is an Armstrong number or not

```
num = int(input("Enter a number: "))

# initialize sum
sum = 0

# find the sum of the cube of each digit
temp = num
while temp > 0:
   digit = temp % 10
   sum += digit ** 3
   temp //= 10

# display the result
if num == sum:
   print(num,"is an Armstrong number")
else:
   print(num,"is not an Armstrong number")
```

Output 1

```
Enter a number: 663
663 is not an Armstrong number
```

Output 2

```
Enter a number: 407
407 is an Armstrong number
```

Sum of natural numbers up to num

```
num = 16

if num < 0:
   print("Enter a positive number")
else:
   sum = 0
   # use while loop to iterate until zero
   while(num > 0):
       sum += num
       num -= 1
   print("The sum is", sum)
```

Output

```
The sum is 136
```

Program to Find Numbers Divisible by Another Number

```
# Take a list of numbers
my_list = [12, 65, 54, 39, 102, 339, 221,]

# use anonymous function to filter
result = list(filter(lambda x: (x % 13 == 0), my_list))

# display the result
print("Numbers divisible by 13 are",result)
```

Output

```
Numbers divisible by 13 are [65, 39, 221]
```

Program to Find Numbers Divisible by Another Number

```
# Take a list of numbers
my_list = [12, 65, 54, 39, 102, 339, 221,]

# use anonymous function to filter
result = list(filter(lambda x: (x % 13 == 0), my_list))

# display the result
print("Numbers divisible by 13 are",result)
```

Output

```
Numbers divisible by 13 are [65, 39, 221]
```

Program to Convert Decimal to Binary, Octal and Hexadecimal

```
dec = 344

print("The decimal value of", dec, "is:")
print(bin(dec), "in binary.")
print(oct(dec), "in octal.")
print(hex(dec), "in hexadecimal.")
```

Output

```
The decimal value of 344 is:
0b101011000 in binary.
0o530 in octal.
0x158 in hexadecimal.
```

Program to Find ASCII Value of Character

```
# Program to find the ASCII value of the given character

c = 'p'
print("The ASCII value of '" + c + "' is", ord(c))
```

Output

```
The ASCII value of 'p' is 112
```

Program to Find HCF or GCD

```
# Python program to find H.C.F of two numbers

# define a function
def compute_hcf(x, y):

# choose the smaller number
    if x > y:
        smaller = y
    else:
        smaller = x
    for i in range(1, smaller+1):
        if((x % i == 0) and (y % i == 0)):
            hcf = i
    return hcf

num1 = 54
num2 = 24

print("The H.C.F. is", compute_hcf(num1, num2))
```

Output

```
The H.C.F. is 6
```

Program to Find LCM

```
# Python Program to find the L.C.M. of two input number

def compute_lcm(x, y):

   # choose the greater number
   if x > y:
       greater = x
   else:
       greater = y

   while(True):
       if((greater % x == 0) and (greater % y == 0)):
           lcm = greater
           break
       greater += 1

   return lcm

num1 = 54
num2 = 24

print("The L.C.M. is", compute_lcm(num1, num2))
```

Output

```
The L.C.M. is 216
```

Program to Find the Factors of a Number

```
# Python Program to find the factors of a number

# This function computes the factor of the argument passed
def print_factors(x):
   print("The factors of",x,"are:")
   for i in range(1, x + 1):
       if x % i == 0:
           print(i)

num = 320

print_factors(num)
```

Output

```
The factors of 320 are:
1
2
4
5
8
10
16
20
32
40
64
80
160
320
```

Program to Make a Simple Calculator

```
# Program make a simple calculator

# This function adds two numbers
def add(x, y):
    return x + y

# This function subtracts two numbers
def subtract(x, y):
    return x - y

# This function multiplies two numbers
def multiply(x, y):
    return x * y

# This function divides two numbers
def divide(x, y):
    return x / y


print("Select operation.")
print("1.Add")
print("2.Subtract")
print("3.Multiply")
print("4.Divide")

while True:
    # take input from the user
    choice = input("Enter choice(1/2/3/4): ")

    # check if choice is one of the four options
    if choice in ('1', '2', '3', '4'):
        num1 = float(input("Enter first number: "))
        num2 = float(input("Enter second number: "))

        if choice == '1':
            print(num1, "+", num2, "=", add(num1, num2))

        elif choice == '2':
            print(num1, "-", num2, "=", subtract(num1, num2))
```

```
        elif choice == '3':
            print(num1, "*", num2, "=", multiply(num1, num2))

        elif choice == '4':
            print(num1, "/", num2, "=", divide(num1, num2))

        # check if user wants another calculation
        # break the while loop if answer is no
        next_calculation = input("Let's do next calculation? (yes/no): ")
        if next_calculation == "no":
          break

    else:
        print("Invalid Input")
```

Output

```
Select operation.
1.Add
2.Subtract
3.Multiply
4.Divide
Enter choice(1/2/3/4): 3
Enter first number: 15
Enter second number: 14
15.0 * 14.0 = 210.0
Let's do next calculation? (yes/no): no
```

Program to Shuffle Deck of Cards

```
# Python program to shuffle a deck of card

# importing modules
import itertools, random

# make a deck of cards
deck = list(itertools.product(range(1,14),['Spade','Heart','Diamond','Club']))

# shuffle the cards
random.shuffle(deck)

# draw five cards
print("You got:")
for i in range(5):
   print(deck[i][0], "of", deck[i][1])
```

Output

```
You got:
5 of Heart
1 of Heart
8 of Spade
12 of Spade
4 of Spade
```

Program to Display Calendar

```
# Program to display calendar of the given month and year

# importing calendar module
import calendar

yy = 2014  # year
mm = 11    # month

# To take month and year input from the user
# yy = int(input("Enter year: "))
# mm = int(input("Enter month: "))

# display the calendar
print(calendar.month(yy, mm))
```

Output

```
   November 2014
Mo Tu We Th Fr Sa Su
                1  2
3  4  5  6 7  8  9
10 11 12 13 14 15 16
17 18 19 20 21 22 23
24 25 26 27 28 29 30
```

Program to Display Fibonacci Sequence Using Recursion

```
# Python program to display the Fibonacci sequence

def recur_fibo(n):
   if n <= 1:
       return n
   else:
       return(recur_fibo(n-1) + recur_fibo(n-2))

nterms = 10

# check if the number of terms is valid
if nterms <= 0:
   print("Plese enter a positive integer")
else:
   print("Fibonacci sequence:")
   for i in range(nterms):
       print(recur_fibo(i))
```

Output

```
Fibonacci sequence:

0
1
1
2
3
5
8
13
21
34
```

Program to Find Sum of Natural Numbers Using Recursion

```
# Python program to find the sum of natural using recursive function

def recur_sum(n):
   if n <= 1:
       return n
   else:
       return n + recur_sum(n-1)

# change this value for a different result
num = 16

if num < 0:
   print("Enter a positive number")
else:
   print("The sum is",recur_sum(num))
```

Output

```
The sum is 136
```

Program to Find Factorial of Number Using Recursion

```
# Factorial of a number using recursion

def recur_factorial(n):
   if n == 1:
       return n
   else:
       return n * recur_factorial(n-1)

num = 7

# check if the number is negative
if num < 0:
   print("Sorry, factorial does not exist for negative numbers")
elif num == 0:
   print("The factorial of 0 is 1")
else:
   print("The factorial of", num, "is", recur_factorial(num))
```

Output

```
The factorial of 7 is 5040
```

Program to Convert Decimal to Binary Using Recursion

```
# Function to print binary number using recursion
def convertToBinary(n):
   if n > 1:
       convertToBinary(n//2)
   print(n % 2,end = '')

# decimal number
dec = 34

convertToBinary(dec)
print()
```

Output

```
100010
```

Program to Add Two Matrices

```
# Program to add two matrices using nested loop

X = [[12,7,3],
    [4 ,5,6],
    [7 ,8,9]]

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

result = [[0,0,0],
          [0,0,0],
          [0,0,0]]

# iterate through rows
for i in range(len(X)):
   # iterate through columns
   for j in range(len(X[0])):
       result[i][j] = X[i][j] + Y[i][j]

for r in result:
   print(r)
```

Output

```
[17, 15, 4]
[10, 12, 9]
[11, 13, 18]
```

Python Program to Transpose a Matrix

```
# Program to transpose a matrix using a nested loop

X = [[12,7],
    [4 ,5],
    [3 ,8]]

result = [[0,0,0],
         [0,0,0]]

# iterate through rows
for i in range(len(X)):
   # iterate through columns
   for j in range(len(X[0])):
       result[j][i] = X[i][j]

for r in result:
   print(r)
```

Output

```
[12, 4, 3]
[7, 5, 8]
```

Python Program to Multiply Two Matrices

```
# Program to multiply two matrices using nested loops

# 3x3 matrix
X = [[12,7,3],
    [4 ,5,6],
    [7 ,8,9]]
# 3x4 matrix
Y = [[5,8,1,2],
    [6,7,3,0],
    [4,5,9,1]]
# result is 3x4
result = [[0,0,0,0],
         [0,0,0,0],
         [0,0,0,0]]

# iterate through rows of X
for i in range(len(X)):
   # iterate through columns of Y
   for j in range(len(Y[0])):
       # iterate through rows of Y
       for k in range(len(Y)):
           result[i][j] += X[i][k] * Y[k][j]

for r in result:
   print(r)
```

Output

```
[114, 160, 60, 27]
[74, 97, 73, 14]
[119, 157, 112, 23]
```

Python Program to Check Whether a String is Palindrome or Not

```
# Program to check if a string is palindrome or not

my_str = 'aIbohPhoBiA'

# make it suitable for caseless comparison

my_str = my_str.casefold()

# reverse the string
rev_str = reversed(my_str)

# check if the string is equal to its reverse
if list(my_str) == list(rev_str):
   print("The string is a palindrome.")
else:
   print("The string is not a palindrome.")
```

Output

```
The string is a palindrome.
```

Python Program to Remove Punctuations From a String

```
# define punctuation
punctuations = '''!()-[]{};:'"\,<>./?@#$%^&*_~'''

my_str = "Hello!!!, he said ---and went."

# To take input from the user
# my_str = input("Enter a string: ")

# remove punctuation from the string
no_punct = ""
for char in my_str:
   if char not in punctuations:
       no_punct = no_punct + char

# display the unpunctuated string
print(no_punct)
```

Output

```
Hello he said and went
```

Python Program to Sort Words in Alphabetic Order

```
# Program to sort alphabetically the words form a string provided by the user

my_str = "Hello this Is an Example With cased letters"

# To take input from the user
#my_str = input("Enter a string: ")

# breakdown the string into a list of words
words = [word.lower() for word in my_str.split()]

# sort the list
words.sort()

# display the sorted words

print("The sorted words are:")
for word in words:
   print(word)
```

Output

```
The sorted words are:
an
cased
example
hello
is
letters
this
with
```

Python Program to Illustrate Different Set Operations

```
# Program to perform different set operations like in mathematics

# define three sets
E = {0, 2, 4, 6, 8};
N = {1, 2, 3, 4, 5};

# set union
print("Union of E and N is",E | N)

# set intersection
print("Intersection of E and N is",E & N)

# set difference
print("Difference of E and N is",E - N)

# set symmetric difference
print("Symmetric difference of E and N is",E ^ N)
```

Output

```
Union of E and N is {0, 1, 2, 3, 4, 5, 6, 8}
Intersection of E and N is {2, 4}
Difference of E and N is {8, 0, 6}
Symmetric difference of E and N is {0, 1, 3, 5, 6, 8}
```

Python Program to Count the Number of Each Vowel

```
# Program to count the number of each vowels

# string of vowels
vowels = 'aeiou'

ip_str = 'Hello, have you tried our tutorial section yet?'

# make it suitable for caseless comparisions
ip_str = ip_str.casefold()

# make a dictionary with each vowel a key and value 0
count = {}.fromkeys(vowels,0)

# count the vowels
for char in ip_str:
   if char in count:
       count[char] += 1

print(count)
```

Output

```
{'o': 5, 'i': 3, 'a': 2, 'e': 5, 'u': 3}
```

Python Program to Create Pyramid Patterns

Example 1: Program to print half pyramid using *

```
*

* *

* * *

* * * *

* * * * *
```

Source Code

```
rows = int(input("Enter number of rows: "))

for i in range(rows):
    for j in range(i+1):
        print("* ", end="")
    print("\n")
```

Python Program to Merge Two Dictionaries

```
dict_1 = {1: 'a', 2: 'b'}
dict_2 = {2: 'c', 4: 'd'}

print(dict_1 | dict_2)
```

Run Code

Output

```
{1: 'a', 2: 'c', 4: 'd'}
```

Python Program to Access Index of a List Using for Loop

```
my_list = [21, 44, 35, 11]

for index, val in enumerate(my_list):
    print(index, val)
```

Run Code

Output

```
0 21
1 44
2 35
3 11
```

Python Program to Flatten a Nested List

```
my_list = [[1], [2, 3], [4, 5, 6, 7]]

flat_list = [num for sublist in my_list for num in sublist]
print(flat_list)
```

Run Code

Output

```
[1, 2, 3, 4, 5, 6, 7]
```

Python Program to Access Index of a List Using for Loop

```
my_list = [21, 44, 35, 11]

for index, val in enumerate(my_list):
    print(index, val)
```

Run Code

Output

```
0 21
1 44
2 35
3 11
```

Python Program to Sort a Dictionary by Value

```
dt = {5:4, 1:6, 6:3}

sorted_dt = {key: value for key, value in sorted(dt.items(), key=lambda item: item[1])}

print(sorted_dt)
```

Run Code

Output

```
{6: 3, 5: 4, 1: 6}
```

Python Program to Iterate Over Dictionaries Using for Loop

```
dt = {'a': 'juice', 'b': 'grill', 'c': 'corn'}

for key, value in dt.items():
    print(key, value)
```

Run Code

Output

```
a juice
b grill
c corn
```

Python Program to Check If a List is Empty

```
my_list = []
if not my_list:
    print("the list is empty")
```

Run Code

Output

```
the list is empty
```

Python Program to Catch Multiple Exceptions in One Line

```
string = input()

try:
    num = int(input())
    print(string+num)
except (TypeError, ValueError) as e:
    print(e)
```

Run Code

Input

```
a
2
```

Output

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

Python Program to Concatenate Two Lists

```
list_1 = [1, 'a']
list_2 = [3, 4, 5]

list_joined = list_1 + list_2
print(list_joined)
```

Run Code

Output

```
[1, 'a', 3, 4, 5]
```

Python Program to Check if a Key is Already Present in a Dictionary

```
my_dict = {1: 'a', 2: 'b', 3: 'c'}

if 2 in my_dict:
    print("present")
```

Run Code

Output

```
present
```

Python Program to Split a List Into Evenly Sized Chunks

```
def split(list_a, chunk_size):

  for i in range(0, len(list_a), chunk_size):
    yield list_a[i:i + chunk_size]

chunk_size = 2
my_list = [1,2,3,4,5,6,7,8,9]
print(list(split(my_list, chunk_size)))
```

Run Code

Output

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

Python Program to Parse a String to a Float or Int

```
balance_str = "1500"
balance_int = int(balance_str)

# print the type
print(type(balance_int))

# print the value
print(balance_int)
```

Run Code

Output

```
<class 'int'>
1500
```

Example 2: Parse string into float

```
balance_str = "1500.4"
balance_float = float(balance_str)

# print the type
print(type(balance_float))

# print the value
print(balance_float)
```

Run Code

Output

```
<class 'float'>
1500.4
```

Example 3: A string float numeral into integer

```
balance_str = "1500.34"
balance_int = int(float(balance_str))

# print the type
print(type(balance_int))

# print the value
print(balance_int)
```

Run Code

Output

```
<class 'int'>
1500
```

Python Program to Convert String to Datetime

Example 1: Using datetime module

```
from datetime import datetime

my_date_string = "Mar 11 2011 11:31AM"

datetime_object = datetime.strptime(my_date_string, '%b %d %Y %I:%M%p')

print(type(datetime_object))
print(datetime_object)
```

Run Code

Output

```
<class 'datetime.datetime'>
2011-03-11 11:31:00
```

Example 2: Using dateutil module

```
from dateutil import parser

date_time = parser.parse("Mar 11 2011 11:31AM")

print(date_time)
print(type(date_time))
```

Run Code

Output

```
2011-03-11 11:31:00
<class 'datetime.datetime'>
```

Python Program to Get the Last Element of the List

```
my_list = ['a', 'b', 'c', 'd', 'e']

# print the last element
print(my_list[-1])
```

Run Code

Output

```
e
```

Python Program to Get a Substring of a String

```
my_string = "I love python."

# prints "love"
print(my_string[2:6])

# prints "love python."
print(my_string[2:])

# prints "I love python"
print(my_string[:-1])
```

Run Code

Output

```
love
love python.
I love python
```

Python Program to Check If a String Is a Number (Float)

```
def isfloat(num):
    try:
        float(num)
        return True
    except ValueError:
        return False

print(isfloat('s12'))
print(isfloat('1.123'))
```

Run Code

Output

```
False
True
```

Python Program to Count the Occurrence of an Item in a List

```
freq = ['a', 1, 'a', 4, 3, 2, 'a'].count('a')
print(freq)
```

Run Code

Output

```
3
```

Python Program to Delete an Element From a Dictionary

Example 1: Using del keyword

```
my_dict = {31: 'a', 21: 'b', 14: 'c'}

del my_dict[31]

print(my_dict)
```

Run Code

Output

```
{21: 'b', 14: 'c'}
```

Example 2: Using pop()

```
my_dict = {31: 'a', 21: 'b', 14: 'c'}

print(my_dict.pop(31))

print(my_dict)
```

Run Code

Output

```
a
{21: 'b', 14: 'c'}
```

Python Program to Create a Long Multiline String

Example 1: Using triple quotes

```
my_string = '''The only way to
learn to program is
by writing code.'''

print(my_string)
```
Run Code

Output

```
The only way to
learn to program is
by writing code.
```

Example 2: Using parentheses and a single/double quotes

```
my_string = ("The only way to \n"
         "learn to program is \n"
         "by writing code.")

print(my_string)
```
Run Code

Output

```
The only way to
learn to program is
by writing code.
```

Python Program to Reverse a Number

Example 1: Reverse a Number using a while loop

```
num = 1234
reversed_num = 0

while num != 0:
    digit = num % 10
    reversed_num = reversed_num * 10 + digit
    num //= 10

print("Reversed Number: " + str(reversed_num))
```

Run Code

Output

```
4321
```

Example 2: Using String slicing

```
num = 123456
print(str(num)[::-1])
```

Run Code

Output

```
654321
```

Python Program to Compute the Power of a Number

Example 1: Calculate power of a number using a while loop

```
base = 3
exponent = 4

result = 1

while exponent != 0:
    result *= base
    exponent-=1

print("Answer = " + str(result))
```

Run Code

Output

```
Answer = 81
```

Example 2: Calculate power of a number using a for loop

```
base = 3
exponent = 4

result = 1

for exponent in range(exponent, 0, -1):
    result *= base

print("Answer = " + str(result))
```

Run Code

Output

```
Answer = 81
```

Example 3: Calculate the power of a number using pow() function

```
base = 3
exponent = -4

result = pow(base, exponent)

print("Answer = " + str(result))
```

Run Code

Output

```
Answer = 0.012345679012345678
```

Python Program to Count the Number of Digits Present In a Number

Example 1: Count Number of Digits in an Integer using while loop

```
num = 3452
count = 0

while num != 0:
    num //= 10
    count += 1

print("Number of digits: " + str(count))
```

Run Code

Output

```
Number of digits: 4
```

Example 2: Using inbuilt methods

```
num = 123456
print(len(str(num)))
```

Run Code

Output

```
6
```

Python Program to Check If Two Strings are Anagram

```
str1 = "Race"
str2 = "Care"

# convert both the strings into lowercase
str1 = str1.lower()
str2 = str2.lower()

# check if length is same
if(len(str1) == len(str2)):

    # sort the strings
    sorted_str1 = sorted(str1)
    sorted_str2 = sorted(str2)

    # if sorted char arrays are same
    if(sorted_str1 == sorted_str2):
        print(str1 + " and " + str2 + " are anagram.")
    else:
        print(str1 + " and " + str2 + " are not anagram.")

else:
    print(str1 + " and " + str2 + " are not anagram.")
```

Run Code

Output

```
race and care are anagram.
```

Python Program to Capitalize the First Character of a String

Example 1: Using list slicing

```
my_string = "programiz is Lit"


print(my_string[0].upper() + my_string[1:])
```
Run Code

Output

```
Programiz is Lit
```

Example 2: Using inbuilt method capitalize()

```
my_string = "programiz is Lit"


cap_string = my_string.capitalize()


print(cap_string)
```
Run Code

Output

```
Programiz is lit
```

Python Program to Create a Countdown Timer

```
import time

def countdown(time_sec):
    while time_sec:
        mins, secs = divmod(time_sec, 60)
        timeformat = '{:02d}:{:02d}'.format(mins, secs)
        print(timeformat, end='\r')
        time.sleep(1)
        time_sec -= 1

    print("stop")

countdown(5)
```

Python Program to Count the Number of Occurrence of a Character in String

Example 1: Using a for loop

```
count = 0

my_string = "Programiz"
my_char = "r"

for i in my_string:
    if i == my_char:
        count += 1

print(count)
```

Run Code

Output

```
2
```

Example 2: Using method count()

```
my_string = "Programiz"
my_char = "r"

print(my_string.count(my_char))
```

Run Code

Output

```
2
```

Python Program to Remove Duplicate Element From a List

Example 1: Using set()

```
list_1 = [1, 2, 1, 4, 6]

print(list(set(list_1)))
```

Run Code

Output

```
[1, 2, 4, 6]
```

Example 2: Remove the items that are duplicated in two lists

```
list_1 = [1, 2, 1, 4, 6]
list_2 = [7, 8, 2, 1]

print(list(set(list_1) ^ set(list_2)))
```

Run Code

Output

```
[4, 6, 7, 8]
```

Python Program to Iterate Through Two Lists in Parallel

Example 1: Using zip (Python 3+)

```
list_1 = [1, 2, 3, 4]
list_2 = ['a', 'b', 'c']

for i, j in zip(list_1, list_2):
    print(i, j)
```

Run Code

Output

```
1 a
2 b
3 c
```

Example 2: Using itertools (Python 2+)

```
import itertools

list_1 = [1, 2, 3, 4]
list_2 = ['a', 'b', 'c']

# loop until the short loop stops
for i,j in zip(list_1,list_2):
    print(i,j)

print("\n")

# loop until the longer list stops
for i,j in itertools.zip_longest(list_1,list_2):
    print(i,j)
```

Run Code

Output

```
1 a
2 b
3 c


1 a
2 b
3 c
4 None
```