

## <Exercise>

Q1. Create a program where

lst = [ 10, 30, 55]

Sum(1st)

and check the sum of all the numbers in the list. If they are not equal to 100 add the number to the list to make their sum 100

Q2. Create a program where the marks obtained by a student are

s1\_marks = 65  
s2\_marks = 30  
s3\_marks = 88

if -statements

and print the divisions of the students according to the criteria

Above or Equal to 80 marks: Ist division

Above or Equal to 60 marks: IInd division

Above or Equal to 33 marks: IIIrd division

Less than 33 marks: Failed

Q3. Shorten the following code in 1 line of code:

```
a = 5
b = 5.00

if x == y :
    c = True

else :
    c = False
```

Tip: Use conditional expression to create a variable c and something like c = True if <cond> else False

Q4.What may go wrong in the following program?

```
1 Str = "python"
2
3 if 'P' in Str:
4     pass
5 else:
6     Str = Str.capitalize()
7
```

NO  
output

Q5.What is the difference between and and or Keyword?

Both conditions must be True:

```
if 1<2 and 3<4:
    print('yay')
```

only either condition

must be True :

```
if 3>4 or 1<2:
    print('yay')
```

## <Exercise>

Q1. Write a program which prints the square of 11, 12, 13, 14, 15 using the while loop. Make sure the output should look like

121 is the square of 11

Q2. Write a program where it prints 1 to 10 using while loop but, without printing 3 and 7. Make sure to use the continue keyword.

Q3. Write a program to print the numbers using while loops:

11, 13, 14, 15, 16, 18

Tip: Make use of both the continue and break keyword.

Q4. What will be the output of the following program?

```
1 j = 0
2 n = 3
3 while j != 7:
4     print(n)
5     j += 1
6     if n == 5:
7         n += 1
8         continue
9     elif n == 7:
10        n += 3
11        continue
12        n += 1
13        if n == 13:
14            break
15    else:
16        print("15")
```

3  
4  
5  
6  
7  
10  
11  
15

## <Exercise>

Q1. Create a program using for loop, to print the maximum number in a list.

Tip: Create a variable with value 0 and iterate over the list, also use if statements

```
lst = [12,15,77,93,101]
max_num = 0
for num in lst:
    if num > max_num:
        max_num = num
```

Q2. Create a program to print all the unique combinations of a, b and c.

Q3. Create a program to calculate the sum of a numeric list.

Tip: Like above, add the num

```
sum = 0
for num in lst:
    sum += num
```

Q4. Create a program where a list

```
lst = ['C', 'C#', 'Python', 'C++', 'Java', 'C#']
```

and check the existence of 'C#' in the list, if no print False if yes print True and the number of times in the list.

Tip: Use for loop and nested if statement

```
ex = False
num_ex = 0
for ele in lst:
    if ele == 'C#':
        ex = True
        num_ex += 1
```

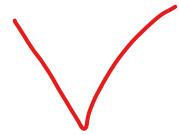
Q5.What will be the output of the following program?

```
1  lst = ['C', 'C#', 'C++']          #
2  num = list(range(6,8))           [
3  Str = '###'                     ]
4  lis,Lst = [],[]                 ]
5  for e in lst:                  ]
6      for n in num:              ]
7          for s in Str:          ]
8              lis.append(s)       ]
9              lis.append(e)       ]
10             lis.append(n)      ]
11     print(lis)                  ]
12     for ele in lis:            ]
13         if ele in Lst:        ]
14             continue           ]
15         else:                  ]
16             Lst.append(ele)    ]
17     print(Lst)                  ]
```

## <Exercise>

Q1.Create a function greet\_user() which takes the name of a user as argument and greets the user as:

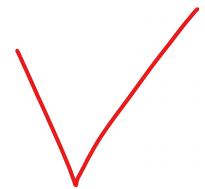
Good Morning <name>!  
Have a nice day!



Q2.Create a function which prints the multiplication table of the number passed as an arguement like,

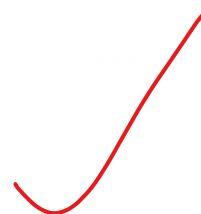
<num> X 1 = <num>

...



Q3.Create functions like,

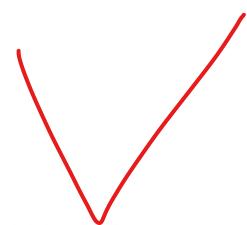
is\_str()  
is\_int()  
is\_float()  
is\_complex()



which takes a data-type as an argument and returns True and False if the argument is a string, integer, floating-point number or complex number respectively. Also, if the functions are called without given any value they should return False.

Tip: Use the type function

```
def is_str(s=0):  
    t = str(type(s))  
    if 'str' in t:  
        return True  
    else:  
        return False
```



Note the use of 0 as parameter. If nothing is passed the function evaluates 0's type and returns False.

Q4. Write your function to calculate the fibonacci of a number passed as an argument.

Tip: fibonacci is the sum of numbers from 1 to it like, fibonacci of 5 is  $5 + 4 + 3 + 2 + 1 = 15$ . Use recursion.

Q5. Create a function to delete the duplicates present in a list passed as an argument.

Tip: Check the Q5 in Exercise of for loops

or

Use set() function.

```
def remove_d(lst):
    s = set(lst)
    return list(s)
```

set() removes all the duplicates while conversion.

## <Exercise>

Q1.Create a class Bird and add attributes like name, color and weight. Also add Bird methods like fly(), flap\_wings() and chirp().

Q2.Create a class coder and add attributes like name, languages, py (for py files) and create methods like can\_codin\_Java(), can\_codin\_C(), code() and coder\_details().

Q3.What is wrong in the following code?

```
1  class Vehicle():
2      _v_key = 'my_key'
3      def __init__(self,type,color):
4          self.type = type
5          self.color = color
6      def start_engine(key):
7          if key == _v_key:
8              print('Vrrrrrrr... ')
9          else:
10             print('Wrong Key! ')
11      def stop_engine():
12          print('shut.. ')
13      def lock(code):
14          _v_key = code
15      def unlock(code):
16          _v_key = 'my_key'
```

TypeError / NameError

## <Exercise>

Q1. Create a function `msngr()` which prints a message "Have a nice day" and also its docstring (also define a documentation string before).

Q2. Create a class `Lang` that has a list like,

```
[‘Python’, ‘Java’, ‘C’, ‘C++’, ‘php’, ‘JavaScript’, ‘C#’]
```

and define a function to use the `in` operator on the `Lang` class to perform the following comparisons:

```
if ‘Python’ in <Lang_Object>:
    print(‘Pythoneer’)
elif ‘TypeScript’ in <Lang_Object>:
    print(‘TypeScriptter’)
else:
    print(‘Learning’)
```

Tip: Define a class `Lang` and initialize the list. Then define `__contains__(self,p)` function which is called by Python when we use `in` operator on `Lang` objects.

```
class Lang():
    def __init__(self):
        self.List = [‘Python’, ‘Java’, ‘C’, \
                    ‘C++’, ‘php’, ‘JavaScript’, ‘C#’]
    def __contains__(self,ele):
        return True if ele in self.List else False
```

Q3. Define a function to check whether a data-type is a iterable or iterator.

Tip: Use the `hasattr(<variable>, <method>)` function to check this, use the passed data-type in place of `<variable>` and pass `__iter__` and `__next__` in methods twice. As we already know if a object has only `__iter__` it is a iterable and if it consists of both `__iter__` and `__next__` then it is iterator.

Q4. Create a function to obtain the sum of all prime numbers under 100.

Q5. Create a function to obtain the sum of all prime numbers under 5,000,000.

Q6. What is the difference between the following variables in a class:

```
class Lang  
    self.name = "jake"  
    num = 1024  
    _age = 17  
    __loc = "#48UFG"  
    __init__()  
    None
```

class name  
instance  
public  
protected  
private  
LD-function  
keyword

Classify the according to class name, class variable, instance variable, protected variable, private variable, public variable, Language-defined function and keyword.

## <Exercise>

Q1. Write a program that uses simple inheritance between classes Numbers and Tens. Numbers stores all numbers (integers and floats) and write methods like is\_div5() and is\_mult8() to check whether the Number object is divisible by 5 and is a multiple of 8. Tens should inherit from Number class. Tens is a class which can contain only numbers from 10 to 99. If a number other than of digits is passed, initialize it to its last tens places or if it's a one digit number add 10 to it (like 1134 to 34 and 6 to 16). It should have a method tens\_place() which returns the tenth's place of the object (like 3 from 38). Tip: Recieve the number and change it to string and perform slicing.

```

11  class Tens(Numbers):
12      def __init__(self, num):
13          self._num = str(num)
14          self.len_n = len(self._num)
15          if self.len_n > 2:
16              slic = self._num[-2:]
17              self.tens = int(slic)
18          elif self.len_n == 1:
19              self.tens = num + 10
20          else:
21              self.tens = num

```

Q2. What is the difference between containership and inheritance? *use of our parent class within our class*

Q3. Create an abstract class called Vehicle containing methods color(), max\_speed() and price(). Create child classes from it like, TwoWheeler(), FourWheeler() and Aerial(). Check whether you can create an object of class Vehicle.

Tip: Use the abc module. To use it write the following line of code at the beginning of your program:

```
from abc import ABC
```

## <Exercise>

Q1. Why should we create modules or packages in the first place? *Simplify methods*

Q2. In a program import the sys module and perform the following functions:

```
1 import sys
2 v = 1024
3 py_ver = str(sys.version)
4 api_ver = str(sys.api_version)
5
6 print('Python Version:' + py_ver)
7 print("API Version:" + api_ver)
8 print(sys.getsizeof(v))
9
```

Q3. What is the difference between the four functions?

```
1 def imp():
2     import random
3     import math
4 def imp_as():
5     import random as rnd
6     import math as mth
7 def imp_some():
8     from random import randint
9     from math import sin,cos,tan
10 def imp_all():
11     from random import * All
```

*None specific*

*specific methods*

Q4. Write a program to use all the functions in the random module. Also, which way of importing the random module is more efficient?

1 import random  
2 or  
3 from random import \*

No difference



Q5. What will be the output of the following program?

```
1 import cmath
2
3 num = 25 + 16j
4
5 print(cmath.sqrt(num)) 5.229...+1.79...j
6 print(cmath.isfinite(num)) True
7 print(cmath.asin(num)) TypeError
8 print(cmath.pi) 3.14...
```

5.229...+1.79...j

True

TypeError

3.14...

## <Exercise>

Q1. Write a program to receive numbers infinitely. If the entered number is 0 or negative or a string raise an error and halt the program.

Tip: Use if statements and raise keyword.

Q2. Write a program where you receive a number from the user through input() and print the cube of it below. If 0 is passed raise an user-defined error, and catch it with except. If string is passed, catch the error. If the passed string is "Quit" or "quit" halt the program.

Tip: Create a Error-class and use while and nested if statements

Q3. What is/are wrong in the following program?

```

1  a,b = 10  TypeError
2  k = 10 if i > 3 else 11  NameError
3  int = 1024

4
5  class Vab:  missing 'self'
6      def __init__(de):
7          self.val = de
8      def function:  SyntaxError
9          if me == 'name':
10              print("YES")
11          else:
12              print("NO")
13
14  print(str(pow(5**3)))  TypeError
15  num = input(Number:)  SyntaxError
16  print('You typed_'+num+'_')

```