Functions

```
-> It is a block of code which only runs when it is called
- Divided into 2 types
    - built-in functions :- print() , input(), min(),max(),sum
(),..........
    - These functions are developed by the developers
 - user defined functions
        - These functions are developed by the users
```

1. with arguments with return value 2.with arguments without return value 3.without arguments with return value 4.without arguments without return value

- Advantages

---

1. Reusability of code
2. Less space taken

- Syntax to create a function

---

def functionname(parameters/arguments): # fn definition statements
functionname(parameters/arguments) # fn calling

In [1]:

```python
# 1. with arguments with return value
# Addition of 2 numbers
a,b = int(input()),int(input())
def add1(a,b):
    return a+b
add1(a,b)
```

```
45
56
```

Out[1]: 101

In [2]:

```python
# 2.with arguments without return value
x,y=12,45
def add2(x,y):
    print(x+y)
add2(x,y)
```

```
57
```

In [3]:

```
# 3.Without arguments with return value
s,e=56,99
def add3():
    return s+e
add3()
```

Out[3]: 155

In [4]:

```
# without arguments without return value
m,n=int(input()),int(input())
def add4():
    print(m+n)
add4()
```

```
78
44
122
```

In [6]:

```
add1(34,78)
```

Out[6]: 112

In [8]:

```
# i/p: 7
# o/p: odd
n = int(input())
def evenodd(n):
    if(n%2==0):
        print("even")
    else:
        print("odd")
evenodd(n)
```

```
9
odd
```

In [15]:

```
# prime number
num = int(input())
def prime(num):
    count=0
    for j in range(1,num+1):
        if(num%j==0):
            count+=1
    if(count==2):
        return True
    else:
        return False
prime(num)

```

```
8
```

Out[15]:

```
False
```

In [17]:

```
s1,s2=int(input()),int(input())
def prime_number(s1,s2): # 1 50
    for k in range(s1,s2+1):
        if(prime(k)==True):
            print(k,end=' ')
prime_number(s1,s2)
```

...

String

```
--> Group of characters (or) sequence of characters
          (or) Anything within the quotes

--> Stirng is a datatype
-->  index starts with 0
--> String is immutable
     (We can't change the data once we assigned)
```

In [24]:

```
s = "python"
print(len(s))
print(s[0],s[2],s[4])  # forward index
print(s[-1],s[-2],s[-3]) # backward index
```

```
6
p t o
n o h
```

In [34]:
```
# Slicing:-  Cutting into pieces
s1 = "workshop"
print(s1[0:4])
print(s1[4:8])
print(s1[4:])
print(s1[:4])

```

```
work
shop
shop
work
```

In [36]:
```
print(s1[0:8:2])
print(s1[0::2])
print(s1[0::3])
print(s1[::-1])
```

```
wrso
wrso
wko
pohskrow
```

In [42]:
```
# len(),min(),max(),sum(),sorted()
h = "apssdc"
print(len(h))
print(min(h))
print(max(h),sorted(h))
#print(sum(h))
```

```
6
a
s ['a', 'c', 'd', 'p', 's', 's']
```

In [43]:
```
ord('a')
```

Out[43]: 97

In [44]:
```
ord('A')
```

Out[44]: 65

In [45]:
```
ord('2')
```

Out[45]: 50

In [46]:
```
print(dir(str),end=' ')
```

```
['__add__', '__class__', '__contains__', '__delattr__', '__dir__', '__do
c__', '__eq__', '__format__', '__ge__', '__getattribute__', '__getitem_
_', '__getnewargs__', '__gt__', '__hash__', '__init__', '__init_subclass
__', '__iter__', '__le__', '__len__', '__lt__', '__mod__', '__mul__', '_
_ne__', '__new__', '__reduce__', '__reduce_ex__', '__repr__', '__rmod_
_', '__rmul__', '__setattr__', '__sizeof__', '__str__', '__subclasshook_
_', 'capitalize', 'casefold', 'center', 'count', 'encode', 'endswith',
'expandtabs', 'find', 'format', 'format_map', 'index', 'isalnum', 'isalp
ha', 'isascii', 'isdecimal', 'isdigit', 'isidentifier', 'islower', 'isnu
meric', 'isprintable', 'isspace', 'istitle', 'isupper', 'join', 'ljust',
'lower', 'lstrip', 'maketrans', 'partition', 'removeprefix', 'removesuff
ix', 'replace', 'rfind', 'rindex', 'rjust', 'rpartition', 'rsplit', 'rst
rip', 'split', 'splitlines', 'startswith', 'strip', 'swapcase', 'title',
'translate', 'upper', 'zfill']
```

In [49]:
```
## capitalize()
d = "python workshop"
d.capitalize()
```

Out[49]: 'Python workshop'

In [50]:
```
# title()
d1 = "hi gd mrng"
d1.title()
```

Out[50]: 'Hi Gd Mrng'

In [51]:
```
# count()
d1.count("g")
```

Out[51]: 2

In [52]:
```
# upper()
j = "work"
j.upper()
```

Out[52]: 'WORK'

In [53]:
```
# lower()
j1 = "WORK"
j1.lower()
```

Out[53]: 'work'

In [54]:

```
# swapcase()
j2 = "JuPyTEr"
j2.swapcase()
```

Out[54]: 'jUpYteR'

In [61]:

```
# index()
k = "python workshop"
print(k.index('p'))
print(k.rindex('p'))
print(k.index('o'))
print(k.rindex('o'))
print(k.index('z'))
```

```
0
14
4
13

---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Cell In[61], line 7
      5 print(k.index('o'))
      6 print(k.rindex('o'))
----> 7 print(k.index('z'))

ValueError: substring not found
```

In [63]:

```
# find()
k = "python workshop"
print(k.find('p'))
print(k.rfind('p'))
print(k.find('o'))
print(k.rfind('o'))
print(k.find('x'))
```

```
0
14
4
13
-1
```

In [67]:
```
# startswith(),endswith()
c = "abcd"
print(c.startswith('a'))
print(c.startswith('b'))
print(c.endswith('c'))
print(c.endswith('d'))
```

```
True
False
False
True
```

In [73]:
```
# isalpha(),isalnum()
b,a = "kldh###rje","478hfgf"
print(b.isalpha())
print(b.isalnum())
print(a.isalnum())
```

```
False
False
True
```

In [78]:
```
# isdigit(),isspace()
k = "8427683"
print(k.isdigit())
k1 = "   "
print(k1.isspace())
```

```
True
True
```

In [81]:
```
# istitle()
n = "Usha Rama College"
print(n.istitle())
```

```
True
```

In [83]:
```
# casefold()
m = "thon"
m.casefold()
```

Out[83]:
```
'thon'
```

In [84]:
```
# split()
v = "abc def ghi"
v.split()
```

Out[84]:
```
['abc', 'def', 'ghi']
```

In [89]:
```
# join()
s = "apssdc"
d = " ".join(s)
d
```

Out[89]: 'a p s s d c'

In [90]:
```
v = "ab#c de#f g#hi"
v.split("#")
```

Out[90]: ['ab', 'c de', 'f g', 'hi']

In [93]:
```
# strip(),lstrip(),rstrip()
k = "     work     "
print(k.strip())
k1 = "    apple"
print(k1.lstrip())
k2 = "apple    "
print(k2.rstrip())
```

```
work
apple
apple
```

In [95]:
```
# zfill()
n = "apple"
n.zfill(20)
```

Out[95]: '000000000000000apple'

In [96]:
```
s = input('Enter any string: ')
print(s,type(s))
```

```
Enter any string: apssdc@123
apssdc@123 <class 'str'>
```

In [99]:
```
# i/p: mom,dad,madam,level
# o/p: palindrom
st = input()
if(st==st[::-1]):
    print("Palindrom")
else:
    print("Not palindrom")
```

```
kkk
Palindrom
```

In [103]:

```
# i/p: APssdC@4^#764
s = input()
up=lw=dg=sp=""
for j in s:
    if(j.isupper()):
        up=up+j
    elif(j.islower()):
        lw=lw+j
    elif(j.isdigit()):
        dg=dg+j
    else:
        sp=sp+j
print("Upeercase Letters are: ",up)
print("Lowercase Letters are: ",lw)
print("Digits are : ",dg)
print("Special Characters are : ",sp)
```

```
APssDc@37846&^%#^!
Upeercase Letters are:  APD
Lowercase Letters are:  ssc
Digits are :  37846
Special Characters are :  @&^%#^!
```

In [102]:

```
g = "python"
for i in g:
    print(i,end=' ')
```

```
p y t h o n
```

In [ ]:

```

```

In [ ]:

```

```

In [ ]:

```

```

In [ ]:

```

```

In [ ]:

```

```

In [ ]:

```

```