Functions

- function is a group of related statements that performs a particular task
- types of functions
  - predefined or inbuilt functions
  - userdefined function
- Syntax for functions: def functionname(parameters):

```
    statements()
```

  - def keyword means start the function header
  - function(based on user define it)
  - parameters(arguments) through which we pass values to function(option)
  - : is end of the function

In [1]:
```
# Predefined or in bulit functions

# 1. abs() indicated fot absolute values to convert negative value to positive

a=-10
abs(a)
```

Out[1]: 10

In [2]:
```
a=15
abs(a)
```

Out[2]: 15

In [3]:
```
# 2. bin() indicated for binary version of specified integer

bin(10)
```

Out[3]: '0b1010'

In [4]:
```
bin(16)
```

Out[4]: '0b10000'

In [5]:
```
#chr() indicates character that represents the specific unicode
chr(98)
```

Out[5]: 'b'

In [7]:
```
chr(97)
```

Out[7]: 'a'

In [8]:
```
chr(101)
```

Out[8]: 'e'

In [9]:
```
#ord() # indicates char to ascii values

ord('8')
```

Out[9]: 56

In [10]:
```
ord('A')
```

Out[10]: 65

In [11]:
```
chr(65)
```

Out[11]: 'A'

In [15]:
```
# compile() indicates specific source on object ready to be exceute
# in compile 3 repesentation are there 1. aval, 2. single, 3. exec
a=compile('print(123)','JNTUACEA','single')
exec(a)
```

123

In [17]:
```
# complex() indicates to complex number by specified real and imaginary number
s

complex(3,7)
```

Out[17]: (3+7j)

In [18]:
```
str
#dir() defines a list of specified objects

dir(list)
```

Out[18]:
```
['__add__',
 '__class__',
 '__contains__',
 '__delattr__',
 '__delitem__',
 '__dir__',
 '__doc__',
 '__eq__',
 '__format__',
 '__ge__',
 '__getattribute__',
 '__getitem__',
 '__gt__',
 '__hash__',
 '__iadd__',
 '__imul__',
 '__init__',
 '__init_subclass__',
 '__iter__',
 '__le__',
 '__len__',
 '__lt__',
 '__mul__',
 '__ne__',
 '__new__',
 '__reduce__',
 '__reduce_ex__',
 '__repr__',
 '__reversed__',
 '__rmul__',
 '__setattr__',
 '__setitem__',
 '__sizeof__',
 '__str__',
 '__subclasshook__',
 'append',
 'clear',
 'copy',
 'count',
 'extend',
 'index',
 'insert',
 'pop',
 'remove',
 'reverse',
 'sort']
```

In [19]:
```
str(10)
```

Out[19]:
```
'10'
```

In [20]:
```
int('100')
```

Out[20]: 100

In [21]:
```
float(10)
```

Out[21]: 10.0

In [23]:
```
a='jntuacea-ece'
len(a) # reads length of characters in string
```

Out[23]: 12

In [24]:
```
a=['a','12','cd']
len(a)
```

Out[24]: 3

In [25]:
```
a=[1,2,45,56,101]
max(a)
```

Out[25]: 101

In [26]:
```
min(a)
```

Out[26]: 1

In [27]:
```
# boolean functions and Data types

#list()
#dict()
#set()
#tuple()
```

In [29]:
```
# List()

list((1,2,3,4,5))
```

Out[29]: [1, 2, 3, 4, 5]

In [31]:
```
# dictonary

dict(name='JNTUACEA',id=12,address='Anantapur')
```

Out[31]: {'name': 'JNTUACEA', 'id': 12, 'address': 'Anantapur'}

In [33]:
```
tuple((1,2,3,45))
```

Out[33]: (1, 2, 3, 45)

In [34]:
```
set((1,2,3,4,5,))
```

Out[34]: {1, 2, 3, 4, 5}

In [35]:
```
# functions syntax
''' def function_name(parameters(arguments)):
     statements(s)

    '''
```

Out[35]:
```
' def function_name(parameters(arguments)):\n     statements(s)\n     \n    '
```

In [36]:
```
# to print sum of two numbers

def addsum():
    a=10
    b=20
    print(a+b)
addsum()
```

```
30
```

In [37]:
```
# to print name with use of parameters(arguments)
def username(name):
    print('My name is:' +name )
username(name =input('enter your name'))
```

```
enter your nameJNTUACEA
My name is:JNTUACEA
```

In [38]:
```
# 3 TYPES OF PARAMETERS OR ARGUMENT FUNCTIONS
'''
1. DEFAULT ARGUMENT
2. KEYWORD ARGUMENT
3. VARIABLELENGTH ARGUMENT '''
```

Out[38]:
```
'\n1. DEFAULT ARGUMENT\n2. KEYWORD ARGUMENT\n3. VARIABLELENGTH ARGUMENT '
```

In [42]:
```
#DEFAULT ARGUMENT

def defargment(name,contact, age):
    print('i am ',name )
    print('contact is:', contact)
    print('age is:',age)
```

In [43]:
```
defargment('JNTUACEA',1234564,75)
```

```
i am  JNTUACEA
contact is: 1234564
age is: 75
```

In [44]:
```
#2. KEYWORD ARGUMENT

def key_argument(name, email,phone,address):
    print('my name is:', name )
    print('my email is:', email )
    print('my phone is:', phone )
    print('my address is:', address )
```

In [45]:
```
key_argument(email='patnamrajeshrai@gmail.com',phone=9989786119,name='RAJESH R
AI', address= 'JNTUA')
```

```
my name is: RAJESH RAI
my email is: patnamrajeshrai@gmail.com
my phone is: 9989786119
my address is: JNTUA
```

In [47]:
```
# variablelength arguments

def user_name(*name):
    for i in name :
        print('my name is:',i)
user_name('Rajesh',123,'JNTUA','JNTUACEA')
```

```
my name is: Rajesh
my name is: 123
my name is: JNTUA
my name is: JNTUACEA
```

Strings

In [48]:
```
# 'jntuacea', "jntuacea",'123',"123@#46"
```

In [49]:
```
a='anantapur'
b='atp515001'
c='atp515001@anantapur#'
print(type(c))
```

```
<class 'str'>
```

In [50]:
```
dir(a)
```

Out[50]:
```
['__add__',
 '__class__',
 '__contains__',
 '__delattr__',
 '__dir__',
 '__doc__',
 '__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',
 'isalpha',
 'isascii',
 'isdecimal',
 'isdigit',
 'isidentifier',
 'islower',
 'isnumeric',
 'isprintable',
 'isspace',
 'istitle',
 'isupper',
 'join',
```

```
 'ljust',
 'lower',
 'lstrip',
 'maketrans',
 'partition',
 'replace',
 'rfind',
 'rindex',
 'rjust',
 'rpartition',
 'rsplit',
 'rstrip',
 'split',
 'splitlines',
 'startswith',
 'strip',
 'swapcase',
 'title',
 'translate',
 'upper',
 'zfill']
```

In [51]:
```
# single line string

a='asskakjbskabfkjsbaksbfyebanouwnonfiuaebbaa'
print(a)
```

```
asskakjbskabfkjsbaksbfyebanouwnonfiuaebbaa
```

In [52]:
```
# multiple line strings

b='''
anantapur
gooty
kadapa
'''
print(b)
```

```

anantapur
gooty
kadapa

```

In [57]:
```
# accessing characters in string
a='anantapur'

# 2 types of accessing index
#1. forward index , 2. backward index

# a n a n t a p u r
# 0 1 2 3 4 5 6 7 8 ( forward index)

# a n a n t a p u r
# -9 -8 -7 -6 -5 -4 -3 -2 -1 (backward index)


a[4]
```

Out[57]: 't'

In [60]:
```
# slicing

b=' anantapurjntuacea absasshaodsa'
b[7:22]
```

Out[60]: 'purjntuacea abs'

In [61]:
```
b[:14]
```

Out[61]: ' anantapurjntu'

In [63]:
```
b[-5:-3]
```

Out[63]: 'ao'

In [64]:
```
a= 'anantapur'
a.upper()
```

Out[64]: 'ANANTAPUR'

In [65]:
```
a='JNAKABBER'
```

In [66]:
```
a.lower()
```

Out[66]: 'jnakabber'

In [1]:
```
a='Jntuacea Anantapur'
a.title()
```

Out[1]: 'Jntuacea Anantapur'

In [2]:
```
a='JNTUACEA'
b='Anantapur'
a+' '+b
```

Out[2]: 'JNTUACEA Anantapur'

In [3]:
```
a='jntuacea'
b='12'
a+b
```

Out[3]: 'jntuacea12'

In [4]:
```
# Split()
a= 'anantapur@ap'
a.split('@')
```

Out[4]: ['anantapur', 'ap']

In [8]:
```
a='I am Rajesh,from Dept of ECE'
a.split(',')
```

Out[8]: ['I am Rajesh', 'from Dept of ECE']

In [9]:
```
a='1,2,3,4,5,6,7'
a.split()
```

Out[9]: ['1,2,3,4,5,6,7']

In [13]:
```
# Format () represents specified values in string
a= 'i am rajesh,from which id {}'
id= 120
a.format(id)
```

Out[13]: 'i am rajesh,from which id 120'

In [14]:
```
# isalpha() represents True is all characters in a string are in alphabets
a= 'jntuaceaanantapur'
a.isalpha()
```

Out[14]: True

In [15]:
```
# isalpha() represents True is all characters in a string are in alphabets
a= 'jntuace12345'
a.isalpha()
```

Out[15]: False

In [17]:
```
# count()
a='jntuacea anantapur'
a.count('a')
```

Out[17]: 5

In [18]:
```
from matplotlib import pyplot as plt
import numpy as np
```

```
---------------------------------------------------------------------------
OSError                                   Traceback (most recent call last)
<ipython-input-18-4c1db7e8d3b2> in <module>
----> 1 from matplotlib import pyplot as plt
      2 import numpy as np

~\AppData\Roaming\Python\Python37\site-packages\matplotlib\__init__.py in <mo
dule>
    136 # cbook must import matplotlib only within function
    137 # definitions, so it is safe to import from it here.
--> 138 from . import cbook, rcsetup
    139 from matplotlib.cbook import (
    140     MatplotlibDeprecationWarning, dedent, get_label, sanitize_sequenc
e)

~\AppData\Roaming\Python\Python37\site-packages\matplotlib\cbook\__init__.py
 in <module>
     29 from weakref import WeakMethod
     30
---> 31 import numpy as np
     32
     33 import matplotlib

~\AppData\Roaming\Python\Python37\site-packages\numpy\__init__.py in <module>
    140     from . import _distributor_init
    141
--> 142     from . import core
    143     from .core import *
    144     from . import compat

~\AppData\Roaming\Python\Python37\site-packages\numpy\core\__init__.py in <mo
dule>
     21             # NOTE: would it change behavior to load ALL
     22             # DLLs at this path vs. the name restriction?
---> 23             WinDLL(os.path.abspath(filename))
     24             DLL_filenames.append(filename)
     25     if len(DLL_filenames) > 1:

C:\anaconda3\lib\ctypes\__init__.py in __init__(self, name, mode, handle, use
_errno, use_last_error)
    362
    363         if handle is None:
--> 364             self._handle = _dlopen(self._name, mode)
    365         else:
    366             self._handle = handle

OSError: [WinError 193] %1 is not a valid Win32 application
```

In [ ]:
```
```