3-functions

February 2, 2024

1 Python Programming (Basic-Intermediate)

1.1 Module 3 - Functions

1.2 Defining a function

[1]:
```
dir()
```

[1]:
```
['In',
 'Out',
 '_',
 '__',
 '___',
 '__builtin__',
 '__builtins__',
 '__doc__',
 '__loader__',
 '__name__',
 '__package__',
 '__spec__',
 '_dh',
 '_i',
 '_i1',
 '_ih',
 '_ii',
 '_iii',
 '_oh',
 'exit',
 'get_ipython',
 'quit']
```

[2]:
```
def greet_user():
    """Display a simple greeting."""
    print("Hello!")
```

[3]:
```
dir()
```

[3]:
```
['In',
 'Out',
 '_',
 '_1',
 '__',
 '___',
 '__builtin__',
 '__builtins__',
 '__doc__',
 '__loader__',
 '__name__',
 '__package__',
 '__spec__',
 '_dh',
 '_i',
 '_i1',
 '_i2',
 '_i3',
 '_ih',
 '_ii',
 '_iii',
 '_oh',
 'exit',
 'get_ipython',
 'greet_user',
 'quit']
```

[4]:
```
greet_user()
```

```
Hello!
```

[5]:
```
?greet_user
```

1.3 Passing information to a function

[6]:
```
def greet_user(username):
    """Display a simple greeting."""
    print(f"Hello, {username.title()}!")
```

[7]:
```
greet_user('santitham')
```

```
Hello, Santitham!
```

[15]:
```
greet_user(123)
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
```

```
<ipython-input-15-c0bf58441f0d> in <cell line: 1>()
----> 1 greet_user(123)

<ipython-input-6-3a705010c2c8> in greet_user(username)
      1 def greet_user(username):
      2   """Display a simple greeting."""
----> 3   print(f"Hello, {username.title()}!")

AttributeError: 'int' object has no attribute 'title'
```

1.4 Positional arguments

[13]:
```
def difference(arg1, arg2):
  return arg1 - arg2
```

[14]:
```
difference(5,3)
```

[14]: 2

1.5 Keyword arguments

[16]:
```
def describe_pet(animal_type, pet_name):
  """Display information about a pet."""
  print(f"\nI have a {animal_type}.")
  print(f"My {animal_type}'s name is {pet_name.title()}.")
```

[17]:
```
describe_pet(animal_type='cat', pet_name='tigris')
```

```
I have a cat.
My cat's name is Tigris.
```

[18]:
```
describe_pet(pet_name='tigris', animal_type='cat')
```

```
I have a cat.
My cat's name is Tigris.
```

1.6 Default values

[19]:
```
def describe_pet(pet_name, animal_type='cat'):
  """Display information about a pet."""
  print(f"\nI have a {animal_type}.")
  print(f"My {animal_type}'s name is {pet_name.title()}.")
```

[20]:
```
describe_pet('Cheesy')
```

```
I have a cat.
My cat's name is Cheesy.
```

[22]:
```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

[23]:
```
import pandas as pd
df=pd.read_excel('/content/drive/MyDrive/AIS_DG/Superstore.xlsx')
```

1.7 Return - simple value

[24]:
```
def get_formatted_name(first_name, last_name):
  """Return a full name, neatly formatted."""
  full_name = f"{first_name} {last_name}"
  return full_name.title()
```

[25]:
```
ajyai = get_formatted_name('santitham','prom-on')
print(ajyai)
```

```
Santitham Prom-On
```

1.8 Return - a dictionary

[26]:
```
def build_person(first_name, last_name):
  """Return a dictionary of information about a person."""
  person = {'first':first_name, 'last':last_name}
  return person
```

[27]:
```
build_person('santitham','prom-on')
```

[27]:
```
{'first': 'santitham', 'last': 'prom-on'}
```

[28]:
```
def multiple_return():
    return (1,2)
```

[29]:
```
x,y = multiple_return()
x
```

[29]:
```
1
```

[30]:
```
len([1,2,3])
```

[30]:
```
3
```

[31]:
```
sum([1,2,3])
```

[31]: 6

[32]:
```
def function_choice(name):
    if name == 'sum':
        return sum
    elif name == 'len':
        return len
```

[33]:
```
ret = function_choice('len')
ret([1,2,3])
```

[33]: 3

1.9 Multiple arguments

[34]:
```
def sum_many_args(*args):
  print(type(args))
  return sum(args)
```

[35]:
```
sum_many_args(1,2,3,4,5)
```

```
<class 'tuple'>
```

[35]: 15

1.10 Lambda functions

[36]:
```
pow = lambda x,y : x**y
pow(2,3)
```

[36]: 8

[37]:
```
import pandas as pd
df=pd.read_excel('/content/drive/MyDrive/AIS_DG/Superstore.xlsx')
```

[38]:
```
df
```

[38]:

	Row ID	Order ID	Order Date	Ship Date	Ship Mode	\
0	1	CA-2013-152156	2014-11-09	2014-11-12	Second Class	
1	2	CA-2013-152156	2014-11-09	2014-11-12	Second Class	
2	3	CA-2013-138688	2014-06-13	2014-06-17	Second Class	
3	4	US-2012-108966	2013-10-11	2013-10-18	Standard Class	
4	5	US-2012-108966	2013-10-11	2013-10-18	Standard Class	
…	…	…	…	…	…	
9989	9990	CA-2011-110422	2012-01-22	2012-01-24	Second Class	

```
9990    9991  CA-2014-121258 2015-02-27 2015-03-04  Standard Class
9991    9992  CA-2014-121258 2015-02-27 2015-03-04  Standard Class
9992    9993  CA-2014-121258 2015-02-27 2015-03-04  Standard Class
9993    9994  CA-2014-119914 2015-05-05 2015-05-10    Second Class

     Customer ID     Customer Name    Segment        Country             City  \
0       CG-12520       Claire Gute   Consumer  United States        Henderson
1       CG-12520       Claire Gute   Consumer  United States        Henderson
2       DV-13045   Darrin Van Huff  Corporate  United States      Los Angeles
3       SO-20335    Sean O'Donnell   Consumer  United States  Fort Lauderdale
4       SO-20335    Sean O'Donnell   Consumer  United States  Fort Lauderdale
…          …               …         …            …               …
9989    TB-21400  Tom Boeckenhauer   Consumer  United States            Miami
9990    DB-13060       Dave Brooks   Consumer  United States       Costa Mesa
9991    DB-13060       Dave Brooks   Consumer  United States       Costa Mesa
9992    DB-13060       Dave Brooks   Consumer  United States       Costa Mesa
9993    CC-12220      Chris Cortes   Consumer  United States      Westminster

      … Postal Code  Region       Product ID         Category Sub-Category  \
0     …       42420   South  FUR-BO-10001798        Furniture    Bookcases
1     …       42420   South  FUR-CH-10000454        Furniture       Chairs
2     …       90036    West  OFF-LA-10000240  Office Supplies       Labels
3     …       33311   South  FUR-TA-10000577        Furniture       Tables
4     …       33311   South  OFF-ST-10000760  Office Supplies      Storage
…   …           …     …              …               …           …
9989  …       33180   South  FUR-FU-10001889        Furniture  Furnishings
9990  …       92627    West  FUR-FU-10000747        Furniture  Furnishings
9991  …       92627    West  TEC-PH-10003645       Technology       Phones
9992  …       92627    West  OFF-PA-10004041  Office Supplies        Paper
9993  …       92683    West  OFF-AP-10002684  Office Supplies   Appliances

                                          Product Name     Sales  Quantity  \
0                    Bush Somerset Collection Bookcase  261.9600         2
1     Hon Deluxe Fabric Upholstered Stacking Chairs,…  731.9400         3
2     Self-Adhesive Address Labels for Typewriters b…   14.6200         2
3         Bretford CR4500 Series Slim Rectangular Table  957.5775         5
4                       Eldon Fold 'N Roll Cart System   22.3680         2
…                                                  …       …       …
9989                           Ultra Door Pull Handle   25.2480         3
9990  Tenex B1-RE Series Chair Mats for Low Pile Car…   91.9600         2
9991                            Aastra 57i VoIP phone  258.5760         2
9992  It's Hot Message Books with Stickers, 2 3/4" x 5"   29.6000         4
9993  Acco 7-Outlet Masterpiece Power Center, Wihtou…  243.1600         2

      Discount    Profit
0         0.00   41.9136
1         0.00  219.5820
```

```
2          0.00    6.8714
3          0.45 -383.0310
4          0.20    2.5164
…           …        …
9989       0.20    4.1028
9990       0.00   15.6332
9991       0.20   19.3932
9992       0.00   13.3200
9993       0.00   72.9480

[9994 rows x 21 columns]
```

[39]:
```
df[['Sales','Profit','Quantity']].apply(lambda x: x/x.max())
```

[39]:
```
          Sales    Profit  Quantity
0      0.011571  0.004990  0.142857
1      0.032332  0.026141  0.214286
2      0.000646  0.000818  0.142857
3      0.042299 -0.045599  0.357143
4      0.000988  0.000300  0.142857
…         …        …         …
9989   0.001115  0.000488  0.214286
9990   0.004062  0.001861  0.142857
9991   0.011422  0.002309  0.142857
9992   0.001308  0.001586  0.285714
9993   0.010741  0.008684  0.142857

[9994 rows x 3 columns]
```

1.11 Variable scope - global

[40]:
```
x = "global"

def foo():
  print("x inside:", x)

foo()
print("x outside:", x)
```

```
x inside: global
x outside: global
```

1.12 Error in attempting to update global

[41]:
```
x = "global"

def foo():
  x = 'x * 2'
  print(x)

foo()
print(x)
```

```
x * 2
global
```

1.13 Variable scope - local

[42]:
```
def sum(x,y):
  s = x + y
  return s

print(sum(5,10))
```

```
15
```

[43]:
```
def foo():
  y1 = "local"

foo()
print(y1)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-43-56c5298068d6> in <cell line: 5>()
      3
      4 foo()
----> 5 print(y1)

NameError: name 'y1' is not defined
```

1.14 Variable scope - nonlocal

[53]:
```
def outer():
  x = 'local'

  def inner():
    nonlocal x
    x = 'nonlocal'
```

```
    print('inner: ', x)

  inner()
  print('outer: ', x)

outer()
```

```
inner:  nonlocal
outer:  nonlocal
```

1.15 Import your own module

[54]:
```
import sys
sys.path.append('/content/drive/MyDrive/AIS_DG/lib')
```

[55]:
```
import mymodule1 as mm
```

[56]:
```
mm.STATIC_VALUE
```

[56]:
```
10
```

[57]:
```
import mymodule1 as mm
mm.build_person('Santitham','Prom-on')
```

[57]:
```
{'first': 'Santitham', 'last': 'Prom-on'}
```

[58]:
```
from mymodule1 import STATIC_VALUE
from mymodule1 import build_person, build_person_with_title
```

[59]:
```
dir()
```

[59]:
```
['In',
 'Out',
 'STATIC_VALUE',
 '_',
 '_1',
 '_10',
 '_12',
 '_14',
 '_27',
 '_29',
 '_3',
 '_30',
 '_31',
 '_33',
 '_35',
 '_36',
```

```
'_38',
'_39',
'_44',
'_47',
'_49',
'_56',
'_57',
'_9',
'__',
'___',
'__builtin__',
'__builtins__',
'__doc__',
'__loader__',
'__name__',
'__package__',
'__spec__',
'_dh',
'_i',
'_i1',
'_i10',
'_i11',
'_i12',
'_i13',
'_i14',
'_i15',
'_i16',
'_i17',
'_i18',
'_i19',
'_i2',
'_i20',
'_i21',
'_i22',
'_i23',
'_i24',
'_i25',
'_i26',
'_i27',
'_i28',
'_i29',
'_i3',
'_i30',
'_i31',
'_i32',
'_i33',
'_i34',
```

```
'_i35',
'_i36',
'_i37',
'_i38',
'_i39',
'_i4',
'_i40',
'_i41',
'_i42',
'_i43',
'_i44',
'_i45',
'_i46',
'_i47',
'_i48',
'_i49',
'_i5',
'_i50',
'_i51',
'_i52',
'_i53',
'_i54',
'_i55',
'_i56',
'_i57',
'_i58',
'_i59',
'_i6',
'_i7',
'_i8',
'_i9',
'_ih',
'_ii',
'_iii',
'_oh',
'ajyai',
'build_person',
'build_person_with_title',
'describe_pet',
'df',
'difference',
'drive',
'exit',
'find_max',
'foo',
'function_choice',
'get_formatted_name',
```

```
 'get_ipython',
 'greet_user',
 'max_x',
 'mm',
 'multiple_return',
 'outer',
 'pd',
 'pow',
 'quit',
 'ret',
 'sum',
 'sum_many_args',
 'sys',
 'x',
 'y']
```

[60]:
```
build_person_with_title('Dr.','Santitham','Prom-on')
```

[60]:
```
{'title': 'Dr.', 'first': 'Santitham', 'last': 'Prom-on'}
```

[61]:
```
STATIC_VALUE
```

[61]:
```
10
```

[62]:
```
from mymodule1 import *

print(build_person('Santitham','Prom-on'))
print(build_person_with_title('Dr.','Santitham','Prom-on'))
```

```
{'first': 'Santitham', 'last': 'Prom-on'}
{'title': 'Dr.', 'first': 'Santitham', 'last': 'Prom-on'}
```

---

[63]:
```
from numpy import *
```

[64]:
```
dir()
```

[64]:
```
['ALLOW_THREADS',
 'AxisError',
 'BUFSIZE',
 'CLIP',
 'ComplexWarning',
 'DataSource',
 'ERR_CALL',
 'ERR_DEFAULT',
 'ERR_IGNORE',
 'ERR_LOG',
```

```
'ERR_PRINT',
'ERR_RAISE',
'ERR_WARN',
'FLOATING_POINT_SUPPORT',
'FPE_DIVIDEBYZERO',
'FPE_INVALID',
'FPE_OVERFLOW',
'FPE_UNDERFLOW',
'False_',
'In',
'Inf',
'Infinity',
'MAXDIMS',
'MAY_SHARE_BOUNDS',
'MAY_SHARE_EXACT',
'ModuleDeprecationWarning',
'NAN',
'NINF',
'NZERO',
'NaN',
'Out',
'PINF',
'PZERO',
'RAISE',
'RankWarning',
'SHIFT_DIVIDEBYZERO',
'SHIFT_INVALID',
'SHIFT_OVERFLOW',
'SHIFT_UNDERFLOW',
'STATIC_VALUE',
'ScalarType',
'TooHardError',
'True_',
'UFUNC_BUFSIZE_DEFAULT',
'UFUNC_PYVALS_NAME',
'VisibleDeprecationWarning',
'WRAP',
'_',
'_1',
'_10',
'_12',
'_14',
'_27',
'_29',
'_3',
'_30',
'_31',
```

```
'_33',
'_35',
'_36',
'_38',
'_39',
'_44',
'_47',
'_49',
'_56',
'_57',
'_59',
'_60',
'_61',
'_9',
'_UFUNC_API',
'__',
'___',
'__builtin__',
'__builtins__',
'__doc__',
'__loader__',
'__name__',
'__package__',
'__spec__',
'__version__',
'_add_newdoc_ufunc',
'_dh',
'_i',
'_i1',
'_i10',
'_i11',
'_i12',
'_i13',
'_i14',
'_i15',
'_i16',
'_i17',
'_i18',
'_i19',
'_i2',
'_i20',
'_i21',
'_i22',
'_i23',
'_i24',
'_i25',
'_i26',
```

```
'_i27',
'_i28',
'_i29',
'_i3',
'_i30',
'_i31',
'_i32',
'_i33',
'_i34',
'_i35',
'_i36',
'_i37',
'_i38',
'_i39',
'_i4',
'_i40',
'_i41',
'_i42',
'_i43',
'_i44',
'_i45',
'_i46',
'_i47',
'_i48',
'_i49',
'_i5',
'_i50',
'_i51',
'_i52',
'_i53',
'_i54',
'_i55',
'_i56',
'_i57',
'_i58',
'_i59',
'_i6',
'_i60',
'_i61',
'_i62',
'_i63',
'_i64',
'_i7',
'_i8',
'_i9',
'_ih',
'_ii',
```

```
'_iii',
'_oh',
'absolute',
'add',
'add_docstring',
'add_newdoc',
'add_newdoc_ufunc',
'ajyai',
'all',
'allclose',
'alltrue',
'amax',
'amin',
'angle',
'any',
'append',
'apply_along_axis',
'apply_over_axes',
'arange',
'arccos',
'arccosh',
'arcsin',
'arcsinh',
'arctan',
'arctan2',
'arctanh',
'argmax',
'argmin',
'argpartition',
'argsort',
'argwhere',
'around',
'array',
'array2string',
'array_equal',
'array_equiv',
'array_repr',
'array_split',
'array_str',
'asanyarray',
'asarray',
'asarray_chkfinite',
'ascontiguousarray',
'asfarray',
'asfortranarray',
'asmatrix',
'atleast_1d',
```

```
'atleast_2d',
'atleast_3d',
'average',
'bartlett',
'base_repr',
'binary_repr',
'bincount',
'bitwise_and',
'bitwise_not',
'bitwise_or',
'bitwise_xor',
'blackman',
'block',
'bmat',
'bool8',
'bool_',
'broadcast',
'broadcast_arrays',
'broadcast_shapes',
'broadcast_to',
'build_person',
'build_person_with_title',
'busday_count',
'busday_offset',
'busdaycalendar',
'byte',
'byte_bounds',
'bytes0',
'bytes_',
'c_',
'can_cast',
'cast',
'cbrt',
'cdouble',
'ceil',
'cfloat',
'char',
'character',
'chararray',
'choose',
'clip',
'clongdouble',
'clongfloat',
'column_stack',
'common_type',
'compare_chararrays',
'complex128',
```

```
'complex256',
'complex64',
'complex_',
'complexfloating',
'compress',
'concatenate',
'conj',
'conjugate',
'convolve',
'copy',
'copysign',
'copyto',
'corrcoef',
'correlate',
'cos',
'cosh',
'count_nonzero',
'cov',
'cross',
'csingle',
'ctypeslib',
'cumprod',
'cumproduct',
'cumsum',
'datetime64',
'datetime_as_string',
'datetime_data',
'deg2rad',
'degrees',
'delete',
'deprecate',
'deprecate_with_doc',
'describe_pet',
'df',
'diag',
'diag_indices',
'diag_indices_from',
'diagflat',
'diagonal',
'diff',
'difference',
'digitize',
'disp',
'divide',
'divmod',
'dot',
'double',
```

```
'drive',
'dsplit',
'dstack',
'dtype',
'e',
'ediff1d',
'einsum',
'einsum_path',
'emath',
'empty',
'empty_like',
'equal',
'errstate',
'euler_gamma',
'exit',
'exp',
'exp2',
'expand_dims',
'expm1',
'extract',
'eye',
'fabs',
'fastCopyAndTranspose',
'fft',
'fill_diagonal',
'find_common_type',
'find_max',
'finfo',
'fix',
'flatiter',
'flatnonzero',
'flexible',
'flip',
'fliplr',
'flipud',
'float128',
'float16',
'float32',
'float64',
'float_',
'float_power',
'floating',
'floor',
'floor_divide',
'fmax',
'fmin',
'fmod',
```

```
'foo',
'format_float_positional',
'format_float_scientific',
'format_parser',
'frexp',
'from_dlpack',
'frombuffer',
'fromfile',
'fromfunction',
'fromiter',
'frompyfunc',
'fromregex',
'fromstring',
'full',
'full_like',
'function_choice',
'gcd',
'generic',
'genfromtxt',
'geomspace',
'get_array_wrap',
'get_formatted_name',
'get_include',
'get_ipython',
'get_printoptions',
'getbufsize',
'geterr',
'geterrcall',
'geterrobj',
'gradient',
'greater',
'greater_equal',
'greet_user',
'half',
'hamming',
'hanning',
'heaviside',
'histogram',
'histogram2d',
'histogram_bin_edges',
'histogramdd',
'hsplit',
'hstack',
'hypot',
'i0',
'identity',
'iinfo',
```

```
'imag',
'in1d',
'index_exp',
'indices',
'inexact',
'inf',
'info',
'infty',
'inner',
'insert',
'int0',
'int16',
'int32',
'int64',
'int8',
'int_',
'intc',
'integer',
'interp',
'intersect1d',
'intp',
'invert',
'is_busday',
'isclose',
'iscomplex',
'iscomplexobj',
'isfinite',
'isfortran',
'isin',
'isinf',
'isnan',
'isnat',
'isneginf',
'isposinf',
'isreal',
'isrealobj',
'isscalar',
'issctype',
'issubclass_',
'issubdtype',
'issubsctype',
'iterable',
'ix_',
'kaiser',
'kron',
'lcm',
'ldexp',
```

```
'left_shift',
'less',
'less_equal',
'lexsort',
'linalg',
'linspace',
'little_endian',
'load',
'loadtxt',
'log',
'log10',
'log1p',
'log2',
'logaddexp',
'logaddexp2',
'logical_and',
'logical_not',
'logical_or',
'logical_xor',
'logspace',
'longcomplex',
'longdouble',
'longfloat',
'longlong',
'lookfor',
'ma',
'mask_indices',
'mat',
'math',
'matmul',
'matrix',
'max_x',
'maximum',
'maximum_sctype',
'may_share_memory',
'mean',
'median',
'memmap',
'meshgrid',
'mgrid',
'min_scalar_type',
'minimum',
'mintypecode',
'mm',
'mod',
'modf',
'moveaxis',
```

```
'msort',
'multiple_return',
'multiply',
'nan',
'nan_to_num',
'nanargmax',
'nanargmin',
'nancumprod',
'nancumsum',
'nanmax',
'nanmean',
'nanmedian',
'nanmin',
'nanpercentile',
'nanprod',
'nanquantile',
'nanstd',
'nansum',
'nanvar',
'nbytes',
'ndarray',
'ndenumerate',
'ndim',
'ndindex',
'nditer',
'negative',
'nested_iters',
'newaxis',
'nextafter',
'nonzero',
'not_equal',
'number',
'obj2sctype',
'object0',
'object_',
'ogrid',
'ones',
'ones_like',
'outer',
'packbits',
'pad',
'partition',
'pd',
'percentile',
'pi',
'piecewise',
'place',
```

```
'poly',
'poly1d',
'polyadd',
'polyder',
'polydiv',
'polyfit',
'polyint',
'polymul',
'polysub',
'polyval',
'positive',
'pow',
'power',
'printoptions',
'prod',
'product',
'promote_types',
'ptp',
'put',
'put_along_axis',
'putmask',
'quantile',
'quit',
'r_',
'rad2deg',
'radians',
'random',
'ravel',
'ravel_multi_index',
'real',
'real_if_close',
'rec',
'recarray',
'recfromcsv',
'recfromtxt',
'reciprocal',
'record',
'remainder',
'repeat',
'require',
'reshape',
'resize',
'result_type',
'ret',
'right_shift',
'rint',
'roll',
```

```
'rollaxis',
'roots',
'rot90',
'round_',
'row_stack',
's_',
'safe_eval',
'save',
'savetxt',
'savez',
'savez_compressed',
'sctype2char',
'sctypeDict',
'sctypes',
'searchsorted',
'select',
'set_numeric_ops',
'set_printoptions',
'set_string_function',
'setbufsize',
'setdiff1d',
'seterr',
'seterrcall',
'seterrobj',
'setxor1d',
'shape',
'shares_memory',
'short',
'show_config',
'sign',
'signbit',
'signedinteger',
'sin',
'sinc',
'single',
'singlecomplex',
'sinh',
'size',
'sometrue',
'sort',
'sort_complex',
'source',
'spacing',
'split',
'sqrt',
'square',
'squeeze',
```

```
'stack',
'std',
'str0',
'str_',
'string_',
'subtract',
'sum',
'sum_many_args',
'swapaxes',
'sys',
'take',
'take_along_axis',
'tan',
'tanh',
'tensordot',
'tile',
'timedelta64',
'trace',
'tracemalloc_domain',
'transpose',
'trapz',
'tri',
'tril',
'tril_indices',
'tril_indices_from',
'trim_zeros',
'triu',
'triu_indices',
'triu_indices_from',
'true_divide',
'trunc',
'typecodes',
'typename',
'ubyte',
'ufunc',
'uint',
'uint0',
'uint16',
'uint32',
'uint64',
'uint8',
'uintc',
'uintp',
'ulonglong',
'unicode_',
'union1d',
'unique',
```

```
 'unpackbits',
 'unravel_index',
 'unsignedinteger',
 'unwrap',
 'ushort',
 'vander',
 'var',
 'vdot',
 'vectorize',
 'void',
 'void0',
 'vsplit',
 'vstack',
 'where',
 'who',
 'x',
 'y',
 'zeros',
 'zeros_like']
```

[65]:
```
import mymodule
```

[66]:
```
from importlib import reload
reload(mymodule)
```

[66]:
```
<module 'mymodule' from '/content/drive/MyDrive/AIS_DG/lib/mymodule.py'>
```

[67]:
```
!pip show pandas
```

```
Name: pandas
Version: 1.5.3
Summary: Powerful data structures for data analysis, time series, and statistics
Home-page: https://pandas.pydata.org
Author: The Pandas Development Team
Author-email: pandas-dev@python.org
License: BSD-3-Clause
Location: /usr/local/lib/python3.10/dist-packages
Requires: numpy, python-dateutil, pytz
Required-by: altair, arviz, bigframes, bokeh, bqplot, cmdstanpy, cufflinks,
datascience, db-dtypes, dopamine-rl, fastai, geemap, geopandas, google-colab,
gspread-dataframe, holoviews, ibis-framework, lida, mizani, mlxtend, pandas-
datareader, pandas-gbq, panel, pins, plotnine, prophet, pymc, seaborn, sklearn-
pandas, statsmodels, vega-datasets, xarray, yfinance
```

1.16 Activity

Write a function in a file 'myutils.py' that perform: - Receive list as argument - Find maximum value/location - Return value, location as a tuple

Import and test it.

[68]:
```
x = [1,2,3,10,0,3,4]
```

[69]:
```
max_x = max(x)
[(k,v) for k,v in enumerate(x) if v == max_x]
```

[69]:
```
[(3, 10)]
```

[77]:
```
%%writefile myutils.py

def find_max(x):
    """
    Find index and value of maximum values in the list

    Parameter
    ------------
    x: list
       Input argument to be used

    Return
    ------------
    List of tuples that have maximum values

    Example
    ------------
    >>> x = [1,2,3,10,0,3,4]
    >>> find_max(x)
        [(3, 10)]
    """
    max_x = max(x)
    return [(k,v) for k,v in enumerate(x) if v == max_x]
```

```
Writing myutils.py
```

[71]:
```
find_max(x)
```

[71]:
```
[(3, 10)]
```

[72]:
```
from google.colab import drive
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call
drive.mount("/content/drive", force_remount=True).
```

[73]:
```
import sys
sys.path.append('/content/drive/MyDrive/AIS_DG/lib')
```

[78]:
```
# work here
from myutils import find_max
```

[79]:
```
x = [1,2,3,10,0,3,4]
find_max(x)
```

[79]:
```
[(3, 10)]
```