

Data Structures:

- List
- Set
- Tuples
- Dictionary
- Arrays
- Multidimensional Arrays (Matrix)
- DataFrame

In [8]:

```
L = ['a', 'b', 'Delhi', 981, 12]
```

In [9]:

```
L
```

Out[9]:

```
['a', 'b', 'Delhi', 981, 12]
```

In [10]:

```
type(L)
```

Out[10]:

```
list
```

In [12]:

```
S = {'a', 'b', 'Delhi', 981, 12}
```

In [13]:

```
S
```

Out[13]:

```
{12, 981, 'Delhi', 'a', 'b'}
```

In [14]:

```
type(S)
```

Out[14]:

```
set
```

In [15]:

```
T = ('a', 'b', 'Delhi', 981, 12)  
print(type(T))
```

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

In [16]:

```
type(T)
```

Out[16]:

tuple

In [18]:

```
L[2]
```

Out[18]:

'Delhi'

In [20]:

```
T[2]
```

Out[20]:

'Delhi'

In [21]:

```
D = {'location': 'India', 'Name': 'pavan'}
```

In [22]:

```
type(D)
```

Out[22]:

dict

In [23]:

```
D
```

Out[23]:

```
{'location': 'India', 'Name': 'pavan'}
```

In [24]:

```
D['location']
```

Out[24]:

'India'

In [27]:

```
arr = np.array(['a', 'b', 'Delhi', 981, 12])
```

In [28]:

```
arr
```

Out[28]:

```
array(['a', 'b', 'Delhi', '981', '12'], dtype='<U5')
```

In [37]:

```
type(arr)
```

Out[37]:

```
numpy.ndarray
```

In [38]:

```
brr = np.array(['a','b','c'])  
brr
```

Out[38]:

```
array(['a', 'b', 'c'], dtype='<U1')
```

In [29]:

```
L
```

Out[29]:

```
['a', 'b', 'Delhi', 981, 12]
```

In [30]:

```
L[1:3]
```

Out[30]:

```
['b', 'Delhi']
```

In [31]:

```
L[1:3] = 'x'
```

In [32]:

```
L
```

Out[32]:

```
['a', 'x', 981, 12]
```

In [36]:

```
T[1]='x'
```

```
-----
-
TypeError                                Traceback (most recent call las
t)
<ipython-input-36-8fb88bb05efe> in <module>
----> 1 T[1]='x'
```

TypeError: 'tuple' object does not support item assignment

In [25]:

```
import numpy as np
```

In [33]:

```
arr[1:3] = 'x'
```

In [34]:

```
arr
```

Out[34]:

```
array(['a', 'x', 'x', '981', '12'], dtype='<U5')
```

In []:

```
X[20:89] # begin:end-1
```

In [39]:

```
X = np.array([[1,2],[3,4]])
X
```

Out[39]:

```
array([[1, 2],
       [3, 4]])
```

In [40]:

```
import pandas as pd
```

In [41]:

```
df = pd.DataFrame({'col1':[1,2,3], 'col': ['a','b','c']})
```

In [42]:

```
type(df)
```

Out[42]:

```
pandas.core.frame.DataFrame
```

In [43]:

```
df
```

Out[43]:

	col1	col
0	1	a
1	2	b
2	3	c

In [44]:

```
a = ([1,2],[3,4])  
type(a)
```

Out[44]:

```
tuple
```

In [45]:

```
a[0]
```

Out[45]:

```
[1, 2]
```

In [51]:

```
a[0][0]
```

Out[51]:

```
1
```

In [48]:

```
a[0][1] = 99
```

In [50]:

```
a[0]=[99]
```

```
-----  
-  
TypeError                                Traceback (most recent call las  
t)  
<ipython-input-50-a46346fe40ef> in <module>  
----> 1 a[0]=[99]
```

TypeError: 'tuple' object does not support item assignment

In [52]:

```
a.append([7
        ,8])
```

-
AttributeError Traceback (most recent call last)

<ipython-input-52-a7d3fc44d17b> in <module>
----> 1 a.append([7,8])

AttributeError: 'tuple' object has no attribute 'append'

In [53]:

```
L
```

Out[53]:

```
['a', 'x', 981, 12]
```

In [54]:

```
L.append('Delhi')
```

In [55]:

```
L
```

Out[55]:

```
['a', 'x', 981, 12, 'Delhi']
```

In [56]:

```
L.remove('Delhi')
```

In [57]:

```
L
```

Out[57]:

```
['a', 'x', 981, 12]
```

In [59]:

```
var = L.pop(1)
```

Out[59]:

```
'x'
```

In [60]:

```
L
```

Out[60]:

```
['a', 981, 12]
```

In [61]:

```
arr = np.array([1,2,3,4,5,6,7,8,9])
```

In [62]:

```
arr.max()
```

Out[62]:

9

In [63]:

```
arr.min()
```

Out[63]:

1

In [64]:

```
arr.mean()
```

Out[64]:

5.0

In [65]:

```
arr.sum()
```

Out[65]:

45

In [66]:

```
arr.std()
```

Out[66]:

2.581988897471611

In [67]:

```
arr
```

Out[67]:

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

In [68]:

```
type(arr)
```

Out[68]:

```
numpy.ndarray
```

In [69]:

```
arr[2]
```

Out[69]:

```
3
```

In [70]:

```
arr[2:6]
```

Out[70]:

```
array([3, 4, 5, 6])
```

In [71]:

```
arr[-1]
```

Out[71]:

```
9
```

In [75]:

```
arr[4:-1]
```

Out[75]:

```
array([5, 6, 7, 8])
```

In [76]:

```
arr[2:]
```

Out[76]:

```
array([3, 4, 5, 6, 7, 8, 9])
```

In [77]:

```
arr[:4]
```

Out[77]:

```
array([1, 2, 3, 4])
```

In [78]:

```
df
```

Out[78]:

	col1	col
0	1	a
1	2	b
2	3	c

In [98]:

```
for i in range(0,10,2):  
    print(i)
```

```
0  
2  
4  
6  
8
```

In [100]:

```
df['col3'] = ['x','y','z']  
df
```

Out[100]:

	col1	col	col3
0	1	a	x
1	2	b	y
2	3	c	z

In [101]:

```
df.columns
```

Out[101]:

```
Index(['col1', 'col', 'col3'], dtype='object')
```

In [102]:

```
df.columns = ['col1', 'col2', 'col3']  
df
```

Out[102]:

	col1	col2	col3
0	1	a	x
1	2	b	y
2	3	c	z

In [103]:

```
df.rename(columns={'col1': 'col_A'})
```

Out[103]:

	col_A	col2	col3
0	1	a	x
1	2	b	y
2	3	c	z

In [104]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 3 entries, 0 to 2  
Data columns (total 3 columns):  
col1      3 non-null int64  
col2      3 non-null object  
col3      3 non-null object  
dtypes: int64(1), object(2)  
memory usage: 200.0+ bytes
```

In [105]:

```
df.describe()
```

Out[105]:

	col1
count	3.0
mean	2.0
std	1.0
min	1.0
25%	1.5
50%	2.0
75%	2.5
max	3.0

In [106]:

```
df
```

Out[106]:

	col1	col2	col3
0	1	a	x
1	2	b	y
2	3	c	z

In [108]:

```
df.iloc[0]
```

Out[108]:

```
col1      1  
col2      a  
col3      x  
Name: 0, dtype: object
```

`df.loc[row condition, list of columns to fetch]`

In [114]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 3 entries, 0 to 2  
Data columns (total 3 columns):  
col1      3 non-null int64  
col2      3 non-null object  
col3      3 non-null object  
dtypes: int64(1), object(2)  
memory usage: 200.0+ bytes
```

In [122]:

```
df.loc[df['col1']>=2, ['col3','col1','col2']]
```

Out[122]:

	col3	col1	col2
1	y	2	b
2	z	3	c

In [120]:

```
df
```

Out[120]:

	col1	col2	col3
0	1	a	x
1	2	b	y
2	3	c	z

In [124]:

```
df.iloc[2]
```

Out[124]:

```
col1      3  
col2      c  
col3      z  
Name: 2, dtype: object
```

In [125]:

```
df.isnull()
```

Out[125]:

	col1	col2	col3
0	False	False	False
1	False	False	False
2	False	False	False

In [126]:

```
df
```

Out[126]:

	col1	col2	col3
0	1	a	x
1	2	b	y
2	3	c	z

In [131]:

```
df.loc[1, 'col2'] = np.nan
```

In [132]:

```
df
```

Out[132]:

	col1	col2	col3
0	1	a	x
1	2	NaN	y
2	3	c	z

In [137]:

```
df.isna()
```

Out[137]:

	col1	col2	col3
0	False	False	False
1	False	True	False
2	False	False	False

In [138]:

```
df.isnull()
```

Out[138]:

	col1	col2	col3
0	False	False	False
1	False	True	False
2	False	False	False

In [139]:

```
df.fillna('b')
```

Out[139]:

	col1	col2	col3
0	1	a	x
1	2	b	y
2	3	c	z

In [140]:

```
df.isna().sum()
```

Out[140]:

```
col1    0
col2    1
col3    0
dtype: int64
```

In [141]:

```
np.ones(shape=(2,3))
```

Out[141]:

```
array([[1., 1., 1.],
       [1., 1., 1.]])
```

In [142]:

```
for i in range(0,10,2):
    print(i)
```

```
0
2
4
6
8
```

In [143]:

```
np.arange(0,10,2)
```

Out[143]:

```
array([0, 2, 4, 6, 8])
```

In [145]:

```
np.linspace(0, 2, 5)
```

Out[145]:

```
array([0. , 0.5, 1. , 1.5, 2. ])
```

In [146]:

```
range(100)
```

Out[146]:

```
range(0, 100)
```

In [147]:

```
np.arange(0,100)
```

Out[147]:

```
array([ 0,  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, 31, 32, 33,
        34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50,
        51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67,
        68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84,
        85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99])
```

In [148]:

```
np.zeros((3,3))
```

Out[148]:

```
array([[0., 0., 0.],
       [0., 0., 0.],
       [0., 0., 0.]])
```

In [149]:

```
arr
```

Out[149]:

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

In [150]:

```
arr.ndim
```

Out[150]:

```
1
```

In [152]:

```
m_arr = np.array([np.arange(0,10),np.arange(10,20)])  
m_arr
```

Out[152]:

```
array([[ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9],  
       [10, 11, 12, 13, 14, 15, 16, 17, 18, 19]])
```

In [153]:

```
m_arr.ndim
```

Out[153]:

```
2
```

In [154]:

```
m_arr.shape
```

Out[154]:

```
(2, 10)
```

In [155]:

```
arr.shape
```

Out[155]:

```
(9,)
```

In [156]:

```
arr
```

Out[156]:

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

In [162]:

```
arr.reshape((3,3))
```

Out[162]:

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

In [165]:

```
arr.shape
```

Out[165]:

```
(9,)
```

In [166]:

```
arr.T
```

Out[166]:

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

In [168]:

```
arr.size
```

Out[168]:

```
9
```

In [169]:

```
arr.itemsize
```

Out[169]:

```
4
```

In [170]:

```
arr.data
```

Out[170]:

```
<memory at 0x000002218FBB2408>
```

In [171]:

```
arr.dtype
```

Out[171]:

```
dtype('int32')
```

In [172]:

```
a = np.array([20,30,40,50])  
b = np.arange(4)  
b
```

Out[172]:

```
array([0, 1, 2, 3])
```

In [173]:

```
c = a - b  
c
```

Out[173]:

```
array([20, 29, 38, 47])
```


In [174]:

```
b*2
```

Out[174]:

```
array([0, 2, 4, 6])
```

In [175]:

```
b**2
```

Out[175]:

```
array([0, 1, 4, 9], dtype=int32)
```

In [176]:

```
import numpy
```

In [177]:

```
dir(numpy)
```

Out[177]:

```
['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_',
 'Inf',
 'Infinity',
 'MAXDIMS',
 'MAY_SHARE_BOUNDS',
 'MAY_SHARE_EXACT',
 'MachAr',
 'ModuleDeprecationWarning',
 'NAN',
 'NINF',
 'NZERO',
 'NaN',
 'PINF',
 'PZERO',
 'PackageLoader',
 'RAISE',
 'RankWarning',
 'SHIFT_DIVIDEBYZERO',
 'SHIFT_INVALID',
 'SHIFT_OVERFLOW',
 'SHIFT_UNDERFLOW',
 'ScalarType',
 'Tester',
 'TooHardError',
 'True_',
 'UFUNC_BUFSIZE_DEFAULT',
 'UFUNC_PYVALS_NAME',
 'VisibleDeprecationWarning',
 'WRAP',
 '_NoValue',
 '__NUMPY_SETUP__',
 '__all__',
 '__builtins__',
 '__cached__',
 '__config__',
 '__doc__',
 '__file__',
 '__git_revision__',
 '__loader__',
 '__mkl_version__',
 '__name__',
```

```
'__package__',  
'__path__',  
'__spec__',  
'__version__',  
'_distributor_init',  
'_globals',  
'_import_tools',  
'_mat',  
'_mklinit',  
'abs',  
'absolute',  
'absolute_import',  
'add',  
'add_docstring',  
'add_newdoc',  
'add_newdoc_ufunc',  
'add_newdocs',  
'alen',  
'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',  
'asscalar',  
'atleast_1d',  
'atleast_2d',  
'atleast_3d',  
'average',
```

'bartlett',
'base_repr',
'binary_repr',
'bincount',
'bitwise_and',
'bitwise_not',
'bitwise_or',
'bitwise_xor',
'blackman',
'block',
'bmat',
'bool',
'bool8',
'bool_',
'broadcast',
'broadcast_arrays',
'broadcast_to',
'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',
'compat',
'complex',
'complex128',
'complex64',
'complex_',
'complexfloating',
'compress',
'concatenate',
'conj',
'conjugate',
'convolve',
'copy',
'copysign',
'copyto',
'core',
'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',  
'diag',  
'diag_indices',  
'diag_indices_from',  
'diagflat',  
'diagonal',  
'diff',  
'digitize',  
'disp',  
'divide',  
'division',  
'divmod',  
'dot',  
'double',  
'dsplit',  
'dstack',  
'dtype',  
'e',  
'ediff1d',  
'einsum',  
'einsum_path',  
'emath',  
'empty',  
'empty_like',  
'equal',  
'erf',  
'errstate',  
'euler_gamma',  
'exp',  
'exp2',  
'expand_dims',  
'expm1',  
'extract',  
'eye',  
'fabs',  
'fastCopyAndTranspose',  
'fft',  
'fill_diagonal',  
'find_common_type',  
'finfo',  
'fix',  
'flatiter',  
'flatnonzero',  
'flexible',  
'flip',  
'fliplr',  
'flipud',
```

```
'float',  
'float16',  
'float32',  
'float64',  
'float_',  
'float_power',  
'floating',  
'floor',  
'floor_divide',  
'fmax',  
'fmin',  
'fmod',  
'format_float_positional',  
'format_float_scientific',  
'format_parser',  
'frexp',  
'frombuffer',  
'fromfile',  
'fromfunction',  
'fromiter',  
'frompyfunc',  
'fromregex',  
'fromstring',  
'full',  
'full_like',  
'fv',  
'gcd',  
'generic',  
'genfromtxt',  
'geomspace',  
'get_array_wrap',  
'get_include',  
'get_printoptions',  
'getbufsize',  
'geterr',  
'geterrcall',  
'geterrobj',  
'gradient',  
'greater',  
'greater_equal',  
'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',
'int',
'int0',
'int16',
'int32',
'int64',
'int8',
'int_',
'int_asbuffer',
'intc',
'integer',
'interp',
'intersect1d',
'intp',
'invert',
'ipmt',
'irr',
'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',
'lib',
'linalg',
'linspace',
'little_endian',
'load',
'loads',
'loadtxt',
'log',
'log10',
'log1p',
'log2',
'logaddexp',
'logaddexp2',


```
'logical_and',  
'logical_not',  
'logical_or',  
'logical_xor',  
'logspace',  
'long',  
'longcomplex',  
'longdouble',  
'longfloat',  
'longlong',  
'lookfor',  
'ma',  
'mafromtxt',  
'mask_indices',  
'mat',  
'math',  
'matmul',  
'matrix',  
'matrixlib',  
'max',  
'maximum',  
'maximum_sctype',  
'may_share_memory',  
'mean',  
'median',  
'memmap',  
'meshgrid',  
'mgrid',  
'min',  
'min_scalar_type',  
'minimum',  
'mintypecode',  
'mirr',  
'mod',  
'modf',  
'moveaxis',  
'msort',  
'multiply',  
'nan',  
'nan_to_num',  
'nanargmax',  
'nanargmin',  
'nancumprod',  
'nancumsum',  
'nanmax',  
'nanmean',  
'nanmedian',  
'nanmin',  
'nanpercentile',  
'nanprod',  
'nanquantile',  
'nanstd',  
'nansum',  
'nanvar',  
'nbytes',  
'ndarray',  
'ndenumerate',  
'ndfromtxt',  
'ndim',  
'ndindex',  
'nditer',
```

'negative',
'nested_iters',
'newaxis',
'nextafter',
'nonzero',
'not_equal',
'nper',
'npv',
'numarray',
'number',
'obj2sctype',
'object',
'object0',
'object_',
'ogrid',
'oldnumeric',
'ones',
'ones_like',
'outer',
'packbits',
'pad',
'partition',
'percentile',
'pi',
'piecewise',
'pkgload',
'place',
'pmt',
'poly',
'poly1d',
'polyadd',
'polyder',
'polydiv',
'polyfit',
'polyint',
'polymul',
'polynomial',
'polysub',
'polyval',
'positive',
'power',
'ppmt',
'print_function',
'printoptions',
'prod',
'product',
'promote_types',
'ptp',
'put',
'put_along_axis',
'putmask',
'pv',
'quantile',
'r_',
'rad2deg',
'radians',
'random',
'rank',
'rate',
'ravel',
'ravel_multi_index',

'real',
'real_if_close',
'rec',
'recarray',
'recfromcsv',
'recfromtxt',
'reciprocal',
'record',
'remainder',
'repeat',
'require',
'reshape',
'resize',
'result_type',
'right_shift',
'rint',
'roll',
'rollaxis',
'roots',
'rot90',
'round',
'round_',
'row_stack',
's_',
'safe_eval',
'save',
'savetxt',
'savez',
'savez_compressed',
'sctype2char',
'sctypeDict',
'sctypeNA',
'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',
'str',
'str0',
'str_',
'string_',
'subtract',
'sum',
'swapaxes',
'sys',
'take',
'take_along_axis',
'tan',
'tanh',
'tensordot',
'test',
'testing',
'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',
'typeDict',
'typeNA',
'typecodes',
'typename',
'ubyte',
'ufunc',
'uint',
'uint0',
'uint16',
'uint32',
'uint64',
'uint8',
'uintc',
'uintp',
'ulonglong',
'unicode',
'unicode_',
'union1d',
'unique',
'unpackbits',
'unravel_index',
'unsignedinteger',
'unwrap',
```

```
'ushort',  
'vander',  
'var',  
'vdot',  
'vectorize',  
'version',  
'void',  
'void0',  
'vsplit',  
'vstack',  
'warnings',  
'where',  
'who',  
'zeros',  
'zeros_like']
```

In [178]:

```
arr
```

Out[178]:

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

In [180]:

```
arr.argmax()
```

Out[180]:

```
8
```

In [182]:

```
dir(pd)
```

Out[182]:

```
['Categorical',  
 'CategoricalDtype',  
 'CategoricalIndex',  
 'DataFrame',  
 'DateOffset',  
 'DatetimeIndex',  
 'DatetimeTZDtype',  
 'ExcelFile',  
 'ExcelWriter',  
 'Float64Index',  
 'Grouper',  
 'HDFStore',  
 'Index',  
 'IndexSlice',  
 'Int16Dtype',  
 'Int32Dtype',  
 'Int64Dtype',  
 'Int64Index',  
 'Int8Dtype',  
 'Interval',  
 'IntervalDtype',  
 'IntervalIndex',  
 'MultiIndex',  
 'NaT',  
 'NamedAgg',  
 'Period',  
 'PeriodDtype',  
 'PeriodIndex',  
 'RangeIndex',  
 'Series',  
 'SparseArray',  
 'SparseDataFrame',  
 'SparseDtype',  
 'SparseSeries',  
 'Timedelta',  
 'TimedeltaIndex',  
 'Timestamp',  
 'UInt16Dtype',  
 'UInt32Dtype',  
 'UInt64Dtype',  
 'UInt64Index',  
 'UInt8Dtype',  
 '__builtins__',  
 '__cached__',  
 '__doc__',  
 '__docformat__',  
 '__file__',  
 '__getattr__',  
 '__git_version__',  
 '__loader__',  
 '__name__',  
 '__package__',  
 '__path__',  
 '__spec__',  
 '__version__',  
 '_config',  
 '_hashtable',  
 '_lib',  
 '_libs',
```

```
'_np_version_under1p14',  
'_np_version_under1p15',  
'_np_version_under1p16',  
'_np_version_under1p17',  
'_tslib',  
'_typing',  
'_version',  
'api',  
'array',  
'arrays',  
'bdate_range',  
'compat',  
'concat',  
'core',  
'crosstab',  
'cut',  
'date_range',  
'datetime',  
'describe_option',  
'errors',  
'eval',  
'factorize',  
'get_dummies',  
'get_option',  
'infer_freq',  
'interval_range',  
'io',  
'isna',  
'isnull',  
'lreshape',  
'melt',  
'merge',  
'merge_asof',  
'merge_ordered',  
'notna',  
'notnull',  
'np',  
'offsets',  
'option_context',  
'options',  
'pandas',  
'period_range',  
'pivot',  
'pivot_table',  
'plotting',  
'qcut',  
'read_clipboard',  
'read_csv',  
'read_excel',  
'read_feather',  
'read_fwf',  
'read_gbq',  
'read_hdf',  
'read_html',  
'read_json',  
'read_msgpack',  
'read_parquet',  
'read_pickle',  
'read_sas',  
'read_spss',  
'read_sql',
```



```
'read_sql_query',  
'read_sql_table',  
'read_stata',  
'read_table',  
'reset_option',  
'set_eng_float_format',  
'set_option',  
'show_versions',  
'test',  
'testing',  
'timedelta_range',  
'to_datetime',  
'to_msgpack',  
'to_numeric',  
'to_pickle',  
'to_timedelta',  
'tseries',  
'unique',  
'util',  
'value_counts',  
'wide_to_long']
```

In []: