Numpy

NumPy, short for Numerical Python, is a fundamental package for scientific computing in Python. It provides support for large, multi-dimensional arrays and matrices, along with a collection of mathematical functions to operate on these arrays.

[5]:
```
!pip install numpy
```

```
Requirement already satisfied: numpy in c:\users\chandrashekar\anaconda3\lib\site-packages (1.26.4)
```

[7]:
```
import numpy as np
```

[61]:
```
print(dir(np),end=" ")
```

```
['ALLOW_THREADS', 'BUFSIZE', 'CLIP', 'DataSource', 'ERR_CALL', 'ERR_DEFAULT', 'ERR_IGNORE', 'ERR_LOG', 'ERR_PRINT', 'ERR_RAISE', 'ERR_WARN', 'FLOATING_POIN
T_SUPPORT', 'FPE_DIVIDEBYZERO', 'FPE_INVALID', 'FPE_OVERFLOW', 'FPE_UNDERFLOW', 'False_', 'Inf', 'Infinity', 'MAXDIMS', 'MAY_SHARE_BOUNDS', 'MAY_SHARE_EXAC
T', 'NAN', 'NINF', 'NZERO', 'NaN', 'PINF', 'PZERO', 'RAISE', 'RankWarning', 'SHIFT_DIVIDEBYZERO', 'SHIFT_INVALID', 'SHIFT_OVERFLOW', 'SHIFT_UNDERFLOW', 'Sc
alarType', 'True_', 'UFUNC_BUFSIZE_DEFAULT', 'UFUNC_PYVALS_NAME', 'WRAP', '_CopyMode', '_NoValue', '_UFUNC_API', '__NUMPY_SETUP__', '__all__', '__builtins_
_', '__cached__', '__config__', '__deprecated_attrs__', '__dir__', '__doc__', '__expired_functions__', '__file__', '__former_attrs__', '__future_scalars_
_', '__getattr__', '__loader__', '__name__', '__package__', '__path__', '__spec__', '__version__', '_add_newdoc_ufunc', '_builtins', '_distributor_init',
'_financial_names', '_get_promotion_state', '_globals', '_int_extended_msg', '_mat', '_no_nep50_warning', '_pyinstaller_hooks_dir', '_pytesttester', '_set_
promotion_state', '_specific_msg', '_typing', '_using_numpy2_behavior', '_utils', 'abs', 'absolute', 'add', 'add_docstring', 'add_newdoc', 'add_newdoc_ufun
c', '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', 'ar
ray_equiv', 'array_repr', 'array_split', 'array_str', 'asanyarray', 'asarray', 'asarray_chkfinite', 'ascontiguousarray', 'asfarray', 'asfortranarray', 'asm
atrix', 'atleast_1d', 'atleast_2d', 'atleast_3d', 'average', 'bartlett', 'base_repr', 'binary_repr', 'bincount', 'bitwise_and', 'bitwise_not', 'bitwise_o
r', 'bitwise_xor', 'blackman', 'block', 'bmat', 'bool_', 'broadcast', 'broadcast_arrays', 'broadcast_shapes', 'broadcast_to', 'busday_count', 'busday_offse
t', 'busdaycalendar', 'byte', 'byte_bounds', 'bytes_', 'c_', 'can_cast', 'cast', 'cbrt', 'cdouble', 'ceil', 'cfloat', 'char', 'character', 'chararray', 'ch
oose', 'clip', 'clongdouble', 'clongfloat', 'column_stack', 'common_type', 'compare_chararrays', 'compat', 'complex128', 'complex64', 'complex_', 'complexf
loating', 'compress', 'concatenate', 'conj', 'conjugate', 'convolve', 'copy', 'copysign', 'copyto', 'corrcoef', 'correlate', 'cos', 'cosh', 'count_nonzer
o', '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', 'di
vmod', 'dot', 'double', 'dsplit', 'dstack', 'dtype', 'dtypes', 'e', 'ediff1d', 'einsum', 'einsum_path', 'emath', 'empty', 'empty_like', 'equal', 'errstat
e', 'euler_gamma', 'exceptions', 'exp', 'exp2', 'expand_dims', 'expm1', 'expm1x', 'extract', 'eye', 'fabs', 'fastCopyAndTranspose', 'fft', 'fill_diagonal',
'find_common_type', 'finfo', 'fix', 'flatiter', 'flatnonzero', 'flexible', 'flip', 'fliplr', 'flipud', 'float16', 'float32', 'float64', 'float_', 'float_po
wer', 'floating', 'floor', 'floor_divide', 'fmax', 'fmin', 'fmod', 'format_float_positional', 'format_float_scientific', 'format_parser', 'frexp', 'from_dl
pack', 'frombuffer', 'fromfile', 'fromfunction', 'fromiter', 'frompyfunc', 'fromregex', 'fromstring', 'full', 'full_like', 'gcd', 'generic', 'genfromtxt',
'geomspace', 'get_array_wrap', 'get_include', 'get_printoptions', 'getbufsize', 'geterr', 'geterrcall', 'geterrobj', 'gradient', 'greater', 'greater_equa
l', 'half', 'hamming', 'hanning', 'heaviside', 'histogram', 'histogram2d', 'histogram_bin_edges', 'histogramdd', 'hsplit', 'hstack', 'hypot', 'i0', 'identi
ty', 'iinfo', 'imag', 'in1d', 'index_exp', 'indices', 'inexact', 'inf', 'info', 'infty', 'inner', 'insert', 'int16', 'int32', 'int64', 'int8', 'int_', 'int
c', 'integer', 'interp', 'intersect1d', 'intp', 'invert', 'is_busday', 'isclose', 'iscomplex', 'iscomplexobj', 'isfinite', 'isfortran', 'isin', 'isinf', 'i
snan', 'isnat', 'isneginf', 'isposinf', 'isreal', 'isrealobj', 'isscalar', 'issctype', 'issubclass_', 'issubdtype', 'issubsctype', 'iterable', 'ix_', 'kais
er', 'kron', 'lcm', 'ldexp', 'left_shift', 'less', 'less_equal', 'lexsort', 'lib', 'linalg', 'linspace', 'little_endian', 'load', 'loadtxt', 'log', 'log1
0', 'log1p', 'log2', 'logaddexp', 'logaddexp2', 'logical_and', 'logical_not', 'logical_or', 'logical_xor', 'logspace', 'longcomplex', 'longdouble', 'longfl
oat', 'longlong', 'lookfor', 'ma', 'mask_indices', 'mat', 'matmul', 'matrix', 'max', 'maximum', 'maximum_sctype', 'may_share_memory', 'mean', 'median', 'me
mmap', 'meshgrid', 'mgrid', 'min', 'min_scalar_type', 'minimum', 'mintypecode', 'mod', 'modf', 'moveaxis', 'msort', 'multiply', 'nan', 'nan_to_num', 'nanar
gmax', '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', 'num
array', 'number', 'obj2sctype', 'object_', 'ogrid', 'oldnumeric', 'ones', 'ones_like', 'outer', 'packbits', 'pad', 'partition', 'percentile', 'pi', 'piecew
ise', 'place', 'poly', 'poly1d', 'polyadd', 'polyder', 'polydiv', 'polyfit', 'polyint', 'polymul', 'polynomial', 'polysub', 'polyval', 'positive', 'power',
'printoptions', 'prod', 'product', 'promote_types', 'ptp', 'put', 'put_along_axis', 'putmask', 'quantile', 'r_', 'rad2deg', 'radians', 'random', 'ravel',
'ravel_multi_index', 'real', 'real_if_close', 'rec', 'recarray', 'recfromcsv', 'recfromtxt', 'reciprocal', 'record', 'remainder', 'repeat', 'require', 'res
hape', 'resize', 'result_type', 'right_shift', 'rint', 'roll', 'rollaxis', 'roots', 'rot90', 'round', 'round_', 'row_stack', 's_', 'safe_eval', 'save', 'sa
vetxt', 'savez', 'savez_compressed', 'sctype2char', 'sctypeDict', 'sctypes', 'searchsorted', 'select', 'set_numeric_ops', 'set_printoptions', 'set_string_f
unction', 'setbufsize', 'setdiff1d', 'seterr', 'seterrcall', 'seterrobj', 'setxor1d', 'shape', 'shares_memory', 'short', 'show_config', 'show_runtime', 'si
gn', 'signbit', 'signedinteger', 'sin', 'sinc', 'single', 'singlecomplex', 'sinh', 'size', 'sometrue', 'sort', 'sort_complex', 'source', 'spacing', 'spli
t', 'sqrt', 'square', 'squeeze', 'stack', 'std', 'str_', 'string_', 'subtract', 'sum', 'swapaxes', 'take', 'take_along_axis', 'tan', 'tanh', 'tensordot',
'test', 'testing', 'tile', 'timedelta64', 'trace', 'tracemalloc_domain', 'transpose', 'trapz', 'tri', 'tril', 'tril_indices', 'tril_indices_from', 'trim_ze
ros', 'triu', 'triu_indices', 'triu_indices_from', 'true_divide', 'trunc', 'typecodes', 'typename', 'ubyte', 'ufunc', 'uint', 'uint16', 'uint32', 'uint64',
'uint8', 'uintc', 'uintp', 'ulonglong', 'unicode_', 'union1d', 'unique', 'unpackbits', 'unravel_index', 'unsignedinteger', 'unwrap', 'ushort', 'vander', 'v
ar', 'vdot', 'vectorize', 'version', 'void', 'vsplit', 'vstack', 'where', 'who', 'zeros', 'zeros_like']
```

[23]:
```
a = np.array(30)
```

[25]:
```
type(a)
```

[25]:
```
numpy.ndarray
```

[27]:
```
a
```

[27]:
```
array(30)
```

[29]:
```
b=np.array([10,20,30,40,50])
```

[31]:
```
b
```

[31]:
```
array([10, 20, 30, 40, 50])
```

[51]:
```
c=np.array([[10, 20, 30, 40],[60, 70, 80, 90]])
```

[53]:
```
c
```

[53]:
```
array([[10, 20, 30, 40],
       [60, 70, 80, 90]])
```

[57]:
```
d=np.array([
         [[10, 20, 30, 40],[60, 70, 80, 90]],
         [[10, 20, 30, 40],[60, 70, 80, 90]],
         [[10, 20, 30, 40],[60, 70, 80, 90]],
         [[10, 20, 30, 40],[60, 70, 80, 90]]
])
```

[59]:
```
d
```

[59]:
```
array([[[10, 20, 30, 40],
        [60, 70, 80, 90]],

       [[10, 20, 30, 40],
        [60, 70, 80, 90]],

       [[10, 20, 30, 40],
        [60, 70, 80, 90]],

       [[10, 20, 30, 40],
        [60, 70, 80, 90]]])
```

[49]:
```
print(np.ndim(a))
```

```
print(np.ndim(b))
print(np.ndim(c))
print(np.ndim(d))
```

```
0
1
2
3
```

[65]:
```
np.shape(b)
```

[65]:
```
(5,)
```

[67]:
```
np.shape(c)
```

[67]:
```
(2, 4)
```

[69]:
```
np.shape(d)
```

[69]:
```
(4, 2, 4)
```

[73]:
```
c
```

[73]:
```
array([[10, 20, 30, 40],
       [60, 70, 80, 90]])
```

[79]:
```
np.reshape(c,[4,2])
```

[79]:
```
array([[10, 20],
       [30, 40],
       [60, 70],
       [80, 90]])
```

[87]:
```
np.square(4)
```

[87]:
```
16
```

[104]:
```
np.min([55,35,46,20])
```

[104]:
```
20
```

[106]:
```
np.ones([4,3])
```

[106]:
```
array([[1., 1., 1.],
       [1., 1., 1.],
       [1., 1., 1.],
       [1., 1., 1.]])
```

[110]:
```
np.zeros([4,3])
```

[110]:
```
array([[0., 0., 0.],
       [0., 0., 0.],
       [0., 0., 0.],
       [0., 0., 0.]])
```

[116]:
```
np.full([4,3],50)
```

[116]:
```
array([[50, 50, 50],
       [50, 50, 50],
       [50, 50, 50],
       [50, 50, 50]])
```

[118]:
```
np.eye(5)
```

[118]:
```
array([[1., 0., 0., 0., 0.],
       [0., 1., 0., 0., 0.],
       [0., 0., 1., 0., 0.],
       [0., 0., 0., 1., 0.],
       [0., 0., 0., 0., 1.]])
```

[122]:
```
for i in range(1,11):
    print(i)
```

```
1
2
3
4
5
6
7
8
9
10
```

[136]:
```
e = np.arange(1,101)
```

[138]:
```
type(e)
```

[138]:
```
numpy.ndarray
```

[140]:
```
e
```

[140]:
```
array([  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, 100])
```

[144]:
```
np.linspace(1,20,10)
```

[144]:
```
array([ 1.        ,  3.11111111,  5.22222222,  7.33333333,  9.44444444,
       11.55555556, 13.66666667, 15.77777778, 17.88888889, 20.        ])
```

[146]:
```
np.logspace(1,10)
```

[146]:
```
array([1.00000000e+01, 1.52641797e+01, 2.32995181e+01, 3.55648031e+01,
       5.42867544e+01, 8.28642773e+01, 1.26485522e+02, 1.93069773e+02,
       2.94705170e+02, 4.49843267e+02, 6.86648845e+02, 1.04811313e+03,
       1.59985872e+03, 2.44205309e+03, 3.72759372e+03, 5.68986603e+03,
       8.68511374e+03, 1.32571137e+04, 2.02358965e+04, 3.08884360e+04,
       4.71486636e+04, 7.19685673e+04, 1.09854114e+05, 1.67683294e+05,
       2.55954792e+05, 3.90693994e+05, 5.96362332e+05, 9.10298178e+05,
       1.38949549e+06, 2.12095089e+06, 3.23745754e+06, 4.94171336e+06,
       7.54312006e+06, 1.15139540e+07, 1.75751062e+07, 2.68269580e+07,
       4.09491506e+07, 6.25055193e+07, 9.54095476e+07, 1.45634848e+08,
```

```
       2.22598648e+08, 3.59322177e+08, 5.17947468e+08, 7.96684521e+08,
       1.20679264e+09, 1.84206997e+09, 2.81176870e+09, 4.29193426e+09,
       6.55128557e+09, 1.00000000e+10])
```

[148]:
```
b
```

[148]:
```
array([10, 20, 30, 40, 50])
```

[150]:
```
c
```

[150]:
```
array([[10, 20, 30, 40],
       [60, 70, 80, 90]])
```

[152]:
```
np.sum(b)
```

[152]:
```
150
```

[158]:
```
np.mean(b)
```

[158]:
```
30.0
```

[162]:
```
np.mean(b)
```

[162]:
```
30.0
```

[164]:
```
np.median(b)
```

[164]:
```
30.0
```

[168]:
```
np.std(c)
```

[168]:
```
27.386127875258307
```

[170]:
```
np.min(b),np.max(b)
```

[170]:
```
(10, 50)
```

[172]:
```
np.corrcoef(c)
```

[172]:
```
array([[1., 1.],
       [1., 1.]])
```

[184]:
```
k =np.arange(1,10)
```

[190]:
```
np.where(k>5)
```

[190]:
```
(array([5, 6, 7, 8], dtype=int64),)
```

[ ]:
```

```