

Numpy

- it stands for numerical python
- Numpy is python library used for working with arrays.

In [1]:

```
1 import numpy as np
```

In [2]:

```
1 np.__version__
```

Out[2]:

```
'1.20.1'
```

In [4]:

```
1 # 1D array create
2 a1 = np.array([1,2,3,4,5])
3 print(a1)
4 print(a1.ndim)
```

```
[1 2 3 4 5]
```

```
1
```

In [5]:

```
1 # create 2d array
2 a2 = np.array([[1,2,3],[4,5,6],[7,8,9],[10,11,12]])
3 print(a2)
```

```
[[ 1  2  3]
```

```
 [ 4  5  6]
```

```
 [ 7  8  9]
```

```
[10 11 12]]
```

In [8]:

```
1 print(a2.ndim)
2 print(a2.shape)
3 type(a2)
```

```
2
```

```
(4, 3)
```

Out[8]:

```
numpy.ndarray
```

In [12]:

```
1 a3 = np.array([[[1,2,3],[4,5,6],[6,7,8]]])
2 print(a3)
3 print(a3.shape)
4 print(a3.ndim)
5 type(a3)
```

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

Out[12]:

numpy.ndarray

In [16]:

```
1 a3 = np.array([[[1,2,3],[4,5,6]],[[6,7,8],[9,10,11]]])
2 print(a3)
3 print(a3.shape)
4 print(a3.ndim)
5 type(a3)
6 print(a3.size)
7 print(a3.itemsize)
```

```
[[[ 1  2  3]
  [ 4  5  6]]

 [[ 6  7  8]
  [ 9 10 11]]]
(2, 2, 3)
3
12
4
```

creating AN array using range()

- np.array(range(start, end, step))
- np.array(range(start,end,step)).reshape(rows,columns)

In [18]:

```
1 a1 = np.array(range(10))
2 print(a1)
3 print(a1.ndim)
```

```
[0 1 2 3 4 5 6 7 8 9]
1
```

In [26]:

```
1 a1 = np.array(range(10,40)).reshape(1,6,5)
2 print(a1)
3 print(a1.ndim)
4 print(a1.shape)
```

```
[[[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]]]
3
(1, 6, 5)
```

creating an array using arange()

- `np.arange(start,end,step).reshape()`

In [28]:

```
1 a1 = np.arange(10,40).reshape(5,6)
2 print(a1)
3 print(a1.ndim)
4 print(a1.size)
```

```
[[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]]
2
30
```

In [31]:

```
1 # zeros matrix
2 z = np.zeros((5,5))
3 print(z)
4 print(z[1][1])
5 print(type(z[1][1]))
```

```
[[0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0.]]
0.0
<class 'numpy.float64'>
```

In [32]:

```
1 z = np.zeros((5,5),dtype='int')
2 print(z)
3 print(z[1][1])
4 print(type(z[1][1]))
```

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

0

<class 'numpy.int32'>

In [35]:

```
1 # ones matrix
2 one = np.ones((5,4),dtype=int)
3 print(one*8)
```

```
[[8 8 8 8]
 [8 8 8 8]
 [8 8 8 8]
 [8 8 8 8]
 [8 8 8 8]]
```

In [36]:

```
1 print(dir(np))
```

```
['ALLOW_THREADS', 'AxisError', 'BUFSIZE', 'Bytes0', 'CLIP', 'ComplexWarning',
'DataSource', 'Datetime64', 'ERR_CALL', 'ERR_DEFAULT', 'ERR_IGNORE', 'ERR_
R_LOG', 'ERR_PRINT', 'ERR_RAISE', 'ERR_WARN', 'FLOATING_POINT_SUPPORT', 'FPE
_DIVIDEBYZERO', 'FPE_INVALID', 'FPE_OVERFLOW', 'FPE_UNDERFLOW', 'False_', 'I
nf', 'Infinity', 'MAXDIMS', 'MAY_SHARE_BOUNDS', 'MAY_SHARE_EXACT', 'MachAr',
'ModuleDeprecationWarning', 'NAN', 'NINF', 'NZERO', 'NaN', 'PINF', 'PZERO',
'RAISE', 'RankWarning', 'SHIFT_DIVIDEBYZERO', 'SHIFT_INVALID', 'SHIFT_OVERFL
OW', 'SHIFT_UNDERFLOW', 'ScalarType', 'Str0', 'Tester', 'TooHardError', 'Tru
e_', 'UFUNC_BUFSIZE_DEFAULT', 'UFUNC_PYVALS_NAME', 'Uint64', 'VisibleDepreca
tionWarning', 'WRAP', '_NoValue', '_UFUNC_API', '__NUMPY_SETUP__', '__all_
_', '__builtins__', '__cached__', '__config__', '__deprecated_attrs__', '__d
ir__', '__doc__', '__expired_functions__', '__file__', '__getattr__', '__git
_revision__', '__loader__', '__mk1_version__', '__name__', '__package__', '_
_path__', '__spec__', '__version__', '_add_newdoc_ufunc', '_distributor_ini
t', '_financial_names', '_globals', '_mat', '_pytesttester', 'abs', 'absolut
e', 'add', 'add_docstring', 'add_newdoc', 'add_newdoc_ufunc', 'alen', 'all',
'allclose', 'alltrue', 'amax', 'amin', 'angle', 'any', 'append', 'apply_alon
g_axis', 'apply_over_axes', 'arange', 'arccos', 'arccosh', 'arcsin', 'arcsin
h', 'arctan', 'arctan2', 'arctanh', 'argmax', 'argmin', 'argpartition', 'arg
sort', 'argwhere', 'around', 'array', 'array2string', 'array_equal', 'array_
equiv', 'array_repr', 'array_split', 'array_str', 'asanyarray', 'asarray',
'asarray_chkfinite', 'ascontiguousarray', 'asfarray', 'asfortranarray', 'asm
atrix', 'asscalar', 'atleast_1d', 'atleast_2d', 'atleast_3d', 'average', 'ba
rtlett', 'base_repr', 'binary_repr', 'bincount', 'bitwise_and', 'bitwise_no
t', 'bitwise_or', 'bitwise_xor', 'blackman', 'block', 'bmat', 'bool8', 'bool
_', 'broadcast', 'broadcast_arrays', 'broadcast_shapes', 'broadcast_to', 'bu
sday_count', 'busday_offset', 'busdaycalendar', 'byte', 'byte_bounds', 'byte
s0', 'bytes_', 'c_', 'can_cast', 'cast', 'cbrt', 'cdouble', 'ceil', 'cfloa
t', 'char', 'character', 'chararray', 'choose', 'clip', 'clongdouble', 'clon
gfloat', 'column_stack', 'common_type', 'compare_chararrays', 'compat', 'com
plex128', 'complex64', 'complex_', 'complexfloating', 'compress', 'concatena
te', 'conj', 'conjugate', 'convolve', 'copy', 'copysign', 'copyto', 'core',
'corrcoef', 'correlate', 'cos', 'cosh', 'count_nonzero', 'cov', 'cross', 'cs
ingle', 'ctypeslib', 'cumprod', 'cumproduct', 'cumsum', 'datetime64', 'datet
ime_as_string', 'datetime_data', 'deg2rad', 'degrees', 'delete', 'deprecat
e', 'deprecate_with_doc', 'diag', 'diag_indices', 'diag_indices_from', 'diag
flat', 'diagonal', 'diff', 'digitize', 'disp', 'divide', 'divmod', 'dot', 'd
ouble', 'dsplit', 'dstack', 'dtype', 'e', 'ediff1d', 'einsum', 'einsum_pat
h', 'emath', 'empty', 'empty_like', 'equal', 'errstate', 'euler_gamma', 'ex
p', 'exp2', 'expand_dims', 'expm1', 'extract', 'eye', 'fabs', 'fastCopyAndTr
anspose', 'fft', 'fill_diagonal', 'find_common_type', 'finfo', 'fix', 'flatl
iter', 'flatnonzero', 'flexible', 'flip', 'fliplr', 'flipud', 'float16', 'flo
at32', 'float64', 'float_', 'float_power', 'floating', 'floor', 'floor_divid
e', 'fmax', 'fmin', 'fmod', 'format_float_positional', 'format_float_scienti
fic', 'format_parser', 'frexp', 'frombuffer', 'fromfile', 'fromfunction', 'f
romiter', 'frompyfunc', 'fromregex', 'fromstring', 'full', 'full_like', 'gc
d', 'generic', 'genfromtxt', 'geomspace', 'get_array_wrap', 'get_include',
'get_printoptions', 'getbufsize', 'geterr', 'geterrcall', 'geterrobj', 'grad
ient', 'greater', 'greater_equal', 'half', 'hamming', 'hanning', 'heavisid
e', 'histogram', 'histogram2d', 'histogram_bin_edges', 'histogramdd', 'hspli
t', 'hstack', 'hypot', 'i0', 'identity', 'iinfo', 'imag', 'in1d', 'index_ex
p', 'indices', 'inexact', 'inf', 'info', 'infty', 'inner', 'insert', 'int0',
'int16', 'int32', 'int64', 'int8', 'int_', 'intc', 'integer', 'interp', 'int
ersect1d', 'intp', 'invert', 'is_busday', 'isclose', 'iscomplex', 'iscomplex
obj', 'isfinite', 'isfortran', 'isin', 'isinf', 'isnan', 'isnat', 'isnegin
f', 'isposinf', 'isreal', 'isrealobj', 'isscalar', 'issctype', 'issubclass
```

```
_, 'issubdtype', 'issubsctype', 'iterable', 'ix_', 'kaiser', 'kron', 'lcm',
'ldexp', 'left_shift', 'less', 'less_equal', 'lexsort', 'lib', 'linalg', 'li
nspac', 'little_endian', 'load', 'loads', 'loadtxt', 'log', 'log10', 'log1
p', 'log2', 'logaddexp', 'logaddexp2', 'logical_and', 'logical_not', 'logica
l_or', 'logical_xor', 'logspace', 'longcomplex', 'longdouble', 'longfloat',
'longlong', 'lookfor', 'ma', 'mafromtxt', 'mask_indices', 'mat', 'math', 'ma
tmul', 'matrix', 'matrixlib', 'max', 'maximum', 'maximum_sctype', 'may_share
_memory', 'mean', 'median', 'memmap', 'meshgrid', 'mgrid', 'min', 'min_scala
r_type', 'minimum', 'mintypecode', 'mkl', 'mod', 'modf', 'moveaxis', 'msor
t', 'multiply', 'nan', 'nan_to_num', 'nanargmax', 'nanargmin', 'nancumprod',
'nancumsum', 'nanmax', 'nanmean', 'nanmedian', 'nanmin', 'nanpercentile', 'n
anprod', 'nanquantile', 'nanstd', 'nansum', 'nanvar', 'nbytes', 'ndarray',
'ndenumerate', 'ndfromtxt', 'ndim', 'ndindex', 'nditer', 'negative', 'nested
_iters', 'newaxis', 'nextafter', 'nonzero', 'not_equal', 'numarray', 'numbe
r', 'obj2sctype', 'object0', 'object_', 'ogrid', 'oldnumeric', 'ones', 'ones
_like', 'os', 'outer', 'packbits', 'pad', 'partition', 'percentile', 'pi',
'piecewise', 'place', 'poly', 'poly1d', 'polyadd', 'polyder', 'polydiv', 'po
lyfit', 'polyint', 'polymul', 'polynomial', 'polysub', 'polyval', 'positiv
e', 'power', 'prntoptions', 'prod', 'product', 'promote_types', 'ptp', 'pu
t', 'put_along_axis', 'putmask', 'quantile', 'r_', 'rad2deg', 'radians', 'ra
ndom', 'ravel', 'ravel_multi_index', 'real', 'real_if_close', 'rec', 'recarr
ay', 'recfromcsv', 'recfromtxt', 'reciprocal', 'record', 'remainder', 'repea
t', 'require', 'reshape', 'resize', 'result_type', 'right_shift', 'rint', 'r
oll', 'rollaxis', 'roots', 'rot90', 'round', 'round_', 'row_stack', 's_', 's
afe_eval', 'save', 'savetxt', 'savez', 'savez_compressed', 'sctype2char', 's
ctypeDict', 'sctypes', 'searchsorted', 'select', 'set_numeric_ops', 'set_pri
ntoptions', 'set_string_function', 'setbufsize', 'setdiff1d', 'seterr', 'set
errcall', 'seterrobj', 'setxor1d', 'shape', 'shares_memory', 'short', 'show_
config', 'sign', 'signbit', 'signedinteger', 'sin', 'sinc', 'single', 'singl
ecomplex', 'sinh', 'size', 'sometrue', 'sort', 'sort_complex', 'source', 'sp
acing', 'split', 'sqrt', 'square', 'squeeze', 'stack', 'std', 'str0', 'str
_', 'string_', 'subtract', 'sum', 'swapaxes', 'sys', 'take', 'take_along_axi
s', 'tan', 'tanh', 'tensordot', 'test', 'testing', 'tile', 'timedelta64', 't
race', 'tracemalloc_domain', 'transpose', 'trapz', 'tri', 'tril', 'tril_indi
ces', 'tril_indices_from', 'trim_zeros', 'triu', 'triu_indices', 'triu_indic
es_from', 'true_divide', 'trunc', 'typeDict', 'typecodes', 'typename', 'ubyt
e', 'ufunc', 'uint', 'uint0', 'uint16', 'uint32', 'uint64', 'uint8', 'uint
c', 'uintp', 'ulonglong', 'unicode_', 'union1d', 'unique', 'unpackbits', 'un
ravel_index', 'unsignedinteger', 'unwrap', 'use_hugepage', 'ushort', 'vande
r', 'var', 'vdot', 'vectorize', 'version', 'void', 'void0', 'vsplit', 'vstac
k', 'warnings', 'where', 'who', 'zeros', 'zeros_like']
```

In [39]:

```
1 r = np.random.randint(10)
2 r
```

Out[39]:

6

In [44]:

```
1 r = np.random.randint(10,20,9)
2 r
```

Out[44]:

array([14, 17, 15, 19, 19, 19, 18, 18, 17])

In [46]:

```
1 r = np.random.randint(10,40,25).reshape(5,5)
2 print(r)
```

```
[[25 37 18 14 14]
 [35 10 21 20 26]
 [27 11 30 17 24]
 [35 14 31 34 24]
 [26 24 23 30 29]]
```

In [47]:

```
1 r = np.random.randint(10,40,(6,5))
2 print(r)
```

```
[[12 19 37 35 23]
 [19 12 24 25 33]
 [35 29 17 25 20]
 [39 32 38 31 36]
 [14 39 20 13 12]
 [22 23 36 31 25]]
```

In [48]:

```
1 r = np.random.randint(10,40,(4,5))
2 print(r)
```

```
[[20 10 26 38 14]
 [10 38 15 16 38]
 [23 10 20 24 12]
 [25 28 26 33 10]]
```

In [49]:

```
1 np.random.random()
```

Out[49]:

0.6186619384081085

In [50]:

```
1 np.random.random((2,3))
```

Out[50]:

```
array([[0.87009515, 0.52363945, 0.09614886],
       [0.28301518, 0.15172366, 0.12348926]])
```

In [51]:

```
1 a1
```

Out[51]:

```
array([[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]])
```

In [52]:

```
1 print(np.mean(a1))
```

24.5

In [53]:

```
1 print(np.median(a1))
```

24.5

In [54]:

```
1 print(np.log(1))
```

0.0

In []:

```
1
```