Numpy (pimp install numpy)

- Numpy stands for Numerical python
- Numpy is a python libraray used for working with arrays.

In [1]:

```
import numpy as np
```

In [2]:

```
np.__version__
```

Out[2]:

```
'1.16.5'
```

In [4]:

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

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

In [6]:

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

```
[[ 1  2  3]
 [ 4  5  6]
 [ 7  8  9]
 [10 11 12]]
```

In [8]:

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

```
[[ 1  2  3]
 [ 4  5  6]
 [ 7  8  9]
 [10 11 12]]
2
(4, 3)
```

In [12]:

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

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

In [15]:

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

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

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

Creating an array using range()

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

In [17]:

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

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

In [22]:

```
a1 = np.array(range(10)).reshape(5,2)
print(a1)
print(a1.ndim)
```

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

In [23]:

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

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

In [25]:

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

Creating an array using an arange()

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

In [28]:

```
a2 = np.arange(10,40).reshape(5,6)
print(a2)
print(a2.ndim)
print(a2.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 [30]:

```
a2 = np.arange(1,40,2).reshape(5,4)
print(a2)
print(a2.ndim)
print(a2.size)
```

```
[[ 1  3  5  7]
 [ 9 11 13 15]
 [17 19 21 23]
 [25 27 29 31]
 [33 35 37 39]]
2
20
```

In [32]:

```
# zeros matrix
z = np.zeros((5,5))
print(z)
```

```
[[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.]]
```

In [33]:

```
# zeros matrix
z = np.zeros((5,5))
print(z)
print(z[1][1])
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 [34]:

```
# zeros matrix
z = np.zeros((5,5),dtype='int')
print(z)
print(z[1][1])
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]:

```
# ones matrix
one = np.ones((5,4))
print(one)
```

```
[[1. 1. 1. 1.]
 [1. 1. 1. 1.]
 [1. 1. 1. 1.]
 [1. 1. 1. 1.]
 [1. 1. 1. 1.]]
```

In [37]:

```
# ones matrix
one = np.ones((5,4),dtype=int)
print(one)
```

```
[[1 1 1 1]
 [1 1 1 1]
 [1 1 1 1]
 [1 1 1 1]
 [1 1 1 1]]
```

In [38]:

```
# ones matrix
one = np.ones((5,4),dtype=str)
print(one)
```

```
[['1' '1' '1' '1']
 ['1' '1' '1' '1']
 ['1' '1' '1' '1']
 ['1' '1' '1' '1']
 ['1' '1' '1' '1']]
```

In [39]:

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

```
[[5 5 5 5]
 [5 5 5 5]
 [5 5 5 5]
 [5 5 5 5]
 [5 5 5 5]]
```

In [40]:

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

```
['ALLOW_THREADS', 'AxisError', 'BUFSIZE', 'CLIP', 'ComplexWarning', 'DataSou
rce', 'ERR_CALL', 'ERR_DEFAULT', 'ERR_IGNORE', 'ERR_LOG', 'ERR_PRINT', 'ERR_
RAISE', 'ERR_WARN', 'FLOATING_POINT_SUPPORT', 'FPE_DIVIDEBYZERO', 'FPE_INVAL
ID', 'FPE_OVERFLOW', 'FPE_UNDERFLOW', 'False_', 'Inf', 'Infinity', 'MAXDIM
S', 'MAY_SHARE_BOUNDS', 'MAY_SHARE_EXACT', 'MachAr', 'ModuleDeprecationWarni
ng', 'NAN', 'NINF', 'NZERO', 'NaN', 'PINF', 'PZERO', 'RAISE', 'RankWarning',
'SHIFT_DIVIDEBYZERO', 'SHIFT_INVALID', 'SHIFT_OVERFLOW', 'SHIFT_UNDERFLOW',
'ScalarType', 'Tester', 'TooHardError', 'True_', 'UFUNC_BUFSIZE_DEFAULT', 'U
FUNC_PYVALS_NAME', 'VisibleDeprecationWarning', 'WRAP', '_NoValue', '_UFUNC_
API', '__NUMPY_SETUP__', '__all__', '__builtins__', '__cached__', '__config_
_', '__doc__', '__file__', '__git_revision__', '__loader__', '__mkl_version_
_', '__name__', '__package__', '__path__', '__spec__', '__version__', '_add_
newdoc_ufunc', '_arg', '_distributor_init', '_globals', '_mat', '_mklinit',
'_pytesttester', 'abs', 'absolute', 'absolute_import', 'add', 'add_docstrin
g', 'add_newdoc', 'add_newdoc_ufunc', 'alen', 'all', 'allclose', 'alltrue',
'amax', 'amin', 'angle', 'any', 'append', 'apply_along_axis', 'apply_over_ax
es', 'arange', 'arccos', 'arccosh', 'arcsin', 'arcsinh', 'arctan', 'arctan
2', 'arctanh', 'argmax', 'argmin', 'argpartition', 'argsort', 'argwhere', 'a
round', 'array', 'array2string', 'array_equal', 'array_equiv', 'array_repr',
'array_split', 'array_str', 'asanyarray', 'asarray', 'asarray_chkfinite', 'a
scontiguousarray', 'asfarray', 'asfortranarray', 'asmatrix', 'asscalar', 'at
least_1d', 'atleast_2d', 'atleast_3d', 'average', 'bartlett', 'base_repr',
'binary_repr', 'bincount', 'bitwise_and', 'bitwise_not', 'bitwise_or', 'bitw
ise_xor', 'blackman', 'block', 'bmat', 'bool', 'bool8', 'bool_', 'broadcas
t', 'broadcast_arrays', 'broadcast_to', 'busday_count', 'busday_offset', 'bu
sdaycalendar', 'byte', 'byte_bounds', 'bytes0', 'bytes_', 'c_', 'can_cast',
'cast', 'cbrt', 'cdouble', 'ceil', 'cfloat', 'char', 'character', 'chararra
y', 'choose', 'clip', 'clongdouble', 'clongfloat', 'column_stack', 'common_t
ype', 'compare_chararrays', 'compat', 'complex', 'complex128', 'complex64',
'complex_', 'complexfloating', 'compress', 'concatenate', 'conj', 'conjugat
e', 'convolve', 'copy', 'copysign', 'copyto', 'core', 'corrcoef', 'correlat
e', 'cos', 'cosh', 'count_nonzero', 'cov', 'cross', 'csingle', 'ctypeslib',
'cumprod', 'cumproduct', 'cumsum', 'datetime64', 'datetime_as_string', 'date
time_data', 'deg2rad', 'degrees', 'delete', 'deprecate', 'deprecate_with_do
c', 'diag', 'diag_indices', 'diag_indices_from', 'diagflat', 'diagonal', 'di
ff', 'digitize', 'disp', 'divide', 'division', 'divmod', 'dot', 'double', 'd
split', 'dstack', 'dtype', 'e', 'ediff1d', 'einsum', 'einsum_path', 'emath',
'empty', 'empty_like', 'equal', 'errstate', 'euler_gamma', 'exp', 'exp2', 'e
xpand_dims', 'expm1', 'extract', 'eye', 'fabs', 'fastCopyAndTranspose', 'ff
t', 'fill_diagonal', 'find_common_type', 'finfo', 'fix', 'flatiter', 'flatno
nzero', 'flexible', 'flip', 'fliplr', 'flipud', 'float', 'float16', 'float3
2', 'float64', 'float_', 'float_power', 'floating', 'floor', 'floor_divide',
'fmax', 'fmin', 'fmod', 'format_float_positional', 'format_float_scientifi
c', 'format_parser', 'frexp', 'frombuffer', 'fromfile', 'fromfunction', 'fro
miter', 'frompyfunc', 'fromregex', 'fromstring', 'full', 'full_like', 'fv',
'gcd', 'generic', 'genfromtxt', 'geomspace', 'get_array_wrap', 'get_includ
e', 'get_printoptions', 'getbufsize', 'geterr', 'geterrcall', 'geterrobj',
'gradient', 'greater', 'greater_equal', 'half', 'hamming', 'hanning', 'heavi
side', 'histogram', 'histogram2d', 'histogram_bin_edges', 'histogramdd', 'hs
plit', 'hstack', 'hypot', 'i0', 'identity', 'iinfo', 'imag', 'in1d', 'index_
exp', 'indices', 'inexact', 'inf', 'info', 'infty', 'inner', 'insert', 'in
t', 'int0', 'int16', 'int32', 'int64', 'int8', 'int_', 'int_asbuffer', 'int
c', 'integer', 'interp', 'intersect1d', 'intp', 'invert', 'ipmt', 'irr', 'is
_busday', 'isclose', 'iscomplex', 'iscomplexobj', 'isfinite', 'isfortran',
'isin', 'isinf', 'isnan', 'isnat', 'isneginf', 'isposinf', 'isreal', 'isreal
obj', 'isscalar', 'issctype', 'issubclass_', 'issubdtype', 'issubsctype', 'i
terable', 'ix_', 'kaiser', 'kron', 'lcm', 'ldexp', 'left_shift', 'less', 'le
ss_equal', 'lexsort', 'lib', 'linalg', 'linspace', 'little_endian', 'load',
'loads', 'loadtxt', 'log', 'log10', 'log1p', 'log2', 'logaddexp', 'logaddexp
2', 'logical_and', 'logical_not', 'logical_or', 'logical_xor', 'logspace',
'long', 'longcomplex', 'longdouble', 'longfloat', 'longlong', 'lookfor', 'm
a', 'mafromtxt', 'mask_indices', 'mat', 'math', 'matmul', 'matrix', 'matrixl
ib', 'max', 'maximum', 'maximum_sctype', 'may_share_memory', 'mean', 'media
n', 'memmap', 'meshgrid', 'mgrid', 'min', 'min_scalar_type', 'minimum', 'min
typecode', 'mirr', 'mod', 'modf', 'moveaxis', 'msort', 'multiply', 'nan', 'n
an_to_num', 'nanargmax', 'nanargmin', 'nancumprod', 'nancumsum', 'nanmax',
'nanmean', 'nanmedian', 'nanmin', 'nanpercentile', 'nanprod', 'nanquantile',
'nanstd', 'nansum', 'nanvar', 'nbytes', 'ndarray', 'ndenumerate', 'ndfromtx
t', 'ndim', 'ndindex', 'nditer', 'negative', 'nested_iters', 'newaxis', 'nex
tafter', 'nonzero', 'not_equal', 'nper', 'npv', 'numarray', 'number', 'obj2s
ctype', 'object', 'object0', 'object_', 'ogrid', 'oldnumeric', 'ones', 'ones
_like', 'outer', 'packbits', 'pad', 'partition', 'percentile', 'pi', 'piecew
ise', 'place', 'pmt', 'poly', 'poly1d', 'polyadd', 'polyder', 'polydiv', 'po
lyfit', 'polyint', 'polymul', 'polynomial', 'polysub', 'polyval', 'positiv
e', 'power', 'ppmt', 'print_function', 'printoptions', 'prod', 'product', 'p
romote_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', 'recfromtx
t', 'reciprocal', 'record', 'remainder', 'repeat', 'require', 'reshape', 're
size', 'result_type', 'right_shift', 'rint', 'roll', 'rollaxis', 'roots', 'r
ot90', 'round', 'round_', 'row_stack', 's_', 'safe_eval', 'save', 'savetxt',
'savez', 'savez_compressed', 'sctype2char', 'sctypeDict', 'sctypeNA', 'sctyp
es', 'searchsorted', 'select', 'set_numeric_ops', 'set_printoptions', 'set_s
tring_function', 'setbufsize', 'setdiff1d', 'seterr', 'seterrcall', 'seterro
bj', 'setxor1d', 'shape', 'shares_memory', 'short', 'show_config', 'sign',
'signbit', 'signedinteger', 'sin', 'sinc', 'single', 'singlecomplex', 'sin
h', 'size', 'sometrue', 'sort', 'sort_complex', 'source', 'spacing', 'spli
t', 'sqrt', 'square', 'squeeze', 'stack', 'std', 'str', 'str0', 'str_', 'str
ing_', 'subtract', 'sum', 'swapaxes', 'sys', 'take', 'take_along_axis', 'ta
n', '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_fro
m', 'true_divide', 'trunc', 'typeDict', 'typeNA', 'typecodes', 'typename',
'ubyte', 'ufunc', 'uint', 'uint0', 'uint16', 'uint32', 'uint64', 'uint8', 'u
intc', 'uintp', 'ulonglong', 'unicode', 'unicode_', 'union1d', 'unique', 'un
packbits', 'unravel_index', 'unsignedinteger', 'unwrap', 'ushort', 'vander',
'var', 'vdot', 'vectorize', 'version', 'void', 'void0', 'vsplit', 'vstack',
'warnings', 'where', 'who', 'zeros', 'zeros_like']
```

In [42]:

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

Out[42]:

```
8
```

In [43]:

```
r = np.random.randint(10,20,5)
r
```

Out[43]:

```
array([19, 15, 12, 12, 19])
```

In [45]:

```
r = np.random.randint(10,40,20).reshape(5,4)
print(r)
```

```
[[30 17 31 36]
 [22 36 23 26]
 [18 18 21 33]
 [24 31 11 37]
 [16 27 27 33]]
```

In [46]:

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

```
[[18 20 11 20 17]
 [33 39 25 32 18]
 [11 24 38 19 20]
 [31 37 27 17 17]
 [11 30 16 15 37]
 [21 15 31 14 27]]
```

In [49]:

```
r = np.random.random()
r
```

Out[49]:

```
0.01038519105762481
```

In [55]:

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

Out[55]:

```
0.45455568703877014
```

In [56]:

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

```

Out[56]:

```
array([[0.38521123, 0.66580552, 0.98591182],
       [0.51047593, 0.3192177 , 0.21280419]])
```

In [58]:

```
a1
```

Out[58]:

```
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 [60]:

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

```
24.5
```

In [61]:

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

```
0.0
```

In [ ]:

```

```