- esc+m--markdowm
- esc+y--code
- shift+enter---excute the code
- esc+b---create a cell below
- esc+a---creates a cell above

Numpy

```
In [1]: print(help("modules"))
```

```
Please wait a moment while I gather a list of all available modules...


C:\Users\Alekhya\Anaconda3\lib\site-packages\IPython\kernel\__init__.py:13: Shi
mWarning: The `IPython.kernel` package has been deprecated since IPython 4.0.Yo
u should import from ipykernel or jupyter_client instead.
  "You should import from ipykernel or jupyter_client instead.", ShimWarning)
WARNING: AstropyDeprecationWarning: astropy.utils.compat.futures is now depreca
ted - use concurrent.futures instead [astropy.utils.compat.futures]
C:\Users\Alekhya\Anaconda3\lib\site-packages\dask\config.py:168: YAMLLoadWarnin
g: calling yaml.load() without Loader=... is deprecated, as the default Loader
is unsafe. Please read https://msg.pyyaml.org/load (https://msg.pyyaml.org/loa
d) for full details.
  data = yaml.load(f.read()) or {}
C:\Users\Alekhya\Anaconda3\lib\site-packages\distributed\config.py:20: YAMLLoad
Warning: calling yaml.load() without Loader=... is deprecated, as the default L
oader is unsafe. Please read https://msg.pyyaml.org/load (https://msg.pyyaml.or
g/load) for full details.
  defaults = yaml.load(f)
C:\Users\Alekhya\Anaconda3\lib\site-packages\nltk\twitter\__init__.py:22: UserW
arning: The twython library has not been installed. Some functionality from the
twitter package will not be available.
  "The twython library has not been installed. "

DEBUG:pip._internal.vcs.versioncontrol:Registered VCS backend: bzr
DEBUG:pip._internal.vcs.versioncontrol:Registered VCS backend: git
DEBUG:pip._internal.vcs.versioncontrol:Registered VCS backend: hg
DEBUG:pip._internal.vcs.versioncontrol:Registered VCS backend: svn

C:\Users\Alekhya\Anaconda3\lib\site-packages\skimage\novice\__init__.py:103: Us
erWarning: The `skimage.novice` module was deprecated in version 0.14. It will
be removed in 0.16.
  warnings.warn("The `skimage.novice` module was deprecated in version 0.14. "
C:\Users\Alekhya\Anaconda3\lib\site-packages\skimage\viewer\utils\core.py:10: U
serWarning: Recommended matplotlib backend is `Agg` for full skimage.viewer fun
ctionality.
  warn("Recommended matplotlib backend is `Agg` for full "
C:\Users\Alekhya\Anaconda3\lib\site-packages\sphinx\websupport\__init__.py:25:
RemovedInSphinx20Warning: sphinx.websupport module is now provided as sphinxcon
trib-websupport. sphinx.websupport will be removed at Sphinx-2.0. Please use th
e package instead.
  RemovedInSphinx20Warning)
C:\Users\Alekhya\Anaconda3\lib\site-packages\qtawesome\iconic_font.py:276: User
Warning: You need to have a running QApplication to use QtAwesome!
  warnings.warn("You need to have a running "
C:\Users\Alekhya\Anaconda3\lib\pkgutil.py:107: VisibleDeprecationWarning: zmq.e
ventloop.minitornado is deprecated in pyzmq 14.0 and will be removed.
    Install tornado itself to use zmq with the tornado IOLoop.

  yield from walk_packages(path, info.name+'.', onerror)

Crypto              bz2                 menuinst            sortedcollections
Cython              cProfile            mimetypes           sortedcontainers
IPython             calendar            mistune             soupsieve
```

```
OpenSSL             certifi             mkl                 sphinx
PIL                 cffi                mkl_fft             sphinxcontrib
PyQt5               cgi                 mkl_random          spyder
__future__          cgitb               mmap                spyder_breakpoints
_abc                chardet             mmapfile            spyder_io_dcm
_ast                chunk               mmsystem            spyder_io_hdf5
_asyncio            click               modulefinder        spyder_kernels
_bisect             cloudpickle         more_itertools      spyder_profiler
_blake2             clyent              mpmath              spyder_pylint
_bootlocale         cmath               msgpack             sqlalchemy
_bz2                cmd                 msilib              sqlite3
_cffi_backend       code                msvcrt              sre_compile
_codecs             codecs              multipledispatch    sre_constants
_codecs_cn          codeop              multiprocessing     sre_parse
_codecs_hk          collections         navigator_updater   ssl
_codecs_iso2022     colorama            nbconvert           sspi
_codecs_jp          colorsys            nbformat            sspicon
_codecs_kr          commctrl            netbios             stat
_codecs_tw          compileall          netrc               statistics
_collections        comtypes            networkx            statsmodels
_collections_abc    concurrent          nltk                storemagic
_compat_pickle      conda               nntplib             string
_compression        conda_build         nose                stringprep
_contextvars        conda_env           notebook            struct
_csv                conda_package_handling nt                  subprocess
_ctypes             conda_verify        ntpath              sunau
_ctypes_test        configparser        ntsecuritycon       symbol
_datetime           constantly          nturl2path          sympy
_decimal            contextlib          numba               sympyprinting
_dummy_thread       contextlib2         numbers             symtable
_elementtree        contextvars         numexpr             sys
_functools          copy                numpy               sysconfig
_hashlib            copyreg             numpydoc            tables
_heapq              crypt               odbc                tabnanny
_imp                cryptography        odo                 tarfile
_io                 csv                 olefile             tblib
_json               ctypes              opcode              telnetlib
_locale             curl                openpyxl            tempfile
_lsprof             curses              operator            terminado
_lzma               cwp                 optparse            test
_markupbase         cycler              os                  test_data
_md5                cython              packaging           test_path
_msi                cythonmagic         pandas              test_pycosat
_multibytecodec     cytoolz             pandocfilters       testpath
_multiprocessing    dask                parser              tests
_nsis               dataclasses         parso               textwrap
_opcode             datashape           partd               this
_operator           datetime            past                threading
_osx_support        dateutil            path                time
_overlapped         dbi                 pathlib             timeit
_pickle             dbm                 pathlib2            timer
_py_abc             dde                 patsy               tkinter
_pydecimal          decimal             pdb                 tlz
_pyio               decorator           pep8                token
_pylief             defusedxml          perfmon             tokenize
_pyrsistent_version difflib             pickle              toolz
_pytest             dis                 pickleshare         tornado
```

```
_queue              distributed         pickletools         tqdm
_random             distutils           pip                 trace
_sha1               doctest             pipes               traceback
_sha256             docutils            pkg_resources       tracemalloc
_sha3               dummy_threading     pkginfo             traitlets
_sha512             easy_install        pkgutil             tty
_signal             email               platform            turtle
_sitebuiltins       encodings           plistlib            turtledemo
_socket             ensurepip           pluggy              twisted
_sqlite3            entrypoints         ply                 types
_sre                enum                poplib              typing
_ssl                errno               posixpath           unicodecsv
_stat               et_xmlfile          pprint              unicodedata
_string             fastcache           profile             unittest
_strptime           faulthandler        prometheus_client   urllib
_struct             filecmp             prompt_toolkit      urllib3
_symtable           fileinput           pstats              uu
_system_path        filelock            psutil              uuid
_testbuffer         flask               pty                 venv
_testcapi           flask_cors          pvectorc            warnings
_testconsole        fnmatch             py                  wave
_testimportmultiple formatter           py_compile          wcwidth
_testmultiphase     fractions           pyasn1              weakref
_thread             ftplib              pyasn1_modules      webbrowser
_threading_local    functools           pyclbr              webencodings
_tkinter            future              pycodestyle         werkzeug
_tracemalloc        gc                  pycosat             wheel
_warnings           genericpath         pycparser           widgetsnbextension
_weakref            getopt              pycurl              win2kras
_weakrefset         getpass             pydoc               win32api
_win32sysloader     gettext             pydoc_data          win32clipboard
_winapi             gevent              pydotplus           win32com
_winxptheme         glob                pyexpat             win32con
_yaml               glob2               pyflakes            win32console
abc                 graphviz            pygments            win32cred
adodbapi            greenlet            pylab               win32crypt
afxres              gzip                pylint              win32cryptcon
aifc                h5py                pyodbc              win32event
alabaster           hamcrest            pyparsing           win32evtlog
anaconda_navigator  hashlib             pyreadline          win32evtlogutil
anaconda_project    heapdict            pyrsistent          win32file
antigravity         heapq               pytest              win32gui
appdirs             hmac                pytest_arraydiff    win32gui_struct
argparse            html                pytest_doctestplus  win32help
array               html5lib            pytest_openfiles    win32inet
asn1crypto          http                pytest_remotedata   win32inetcon
ast                 hyperlink           pythoncom           win32job
astroid             idlelib             pytz                win32lz
astropy             idna                pywin               win32net
asynchat            imageio             pywin32_testutil    win32netcon
asyncio             imagesize           pywintypes          win32pdh
asyncore            imaplib             pywt                win32pdhquery
atexit              imghdr              pyximport           win32pdhutil
atomicwrites        imp                 qtawesome           win32pipe
attr                importlib           qtconsole           win32print
audioop             importlib_metadata  qtpy                win32process
automat             incremental         queue               win32profile
```

```
autoreload          inspect             quopri              win32ras
babel               io                  random              win32rcparser
backcall            ipaddress           rasutil             win32security
backports           ipykernel           re                  win32service
base64              ipykernel_launcher  readline            win32serviceutil
bcrypt              ipython_genutils    regcheck            win32timezone
bdb                 ipywidgets          regutil             win32trace
binascii            isapi               reprlib             win32traceutil
binhex              isort               requests            win32transaction
binstar_client      isympy              rlcompleter         win32ts
bisect              itertools           rmagic              win32ui
bitarray            itsdangerous        rope                win32uiole
bkcharts            jdcal               ruamel_yaml         win32verstamp
blaze               jedi                run                 win32wnet
bleach              jinja2              runpy               win_inet_pton
bokeh               json                sched               win_unicode_console
boto                jsonschema          scipy               wincertstore
bottleneck          jupyter             scripts             winerror
brain_argparse      jupyter_client      seaborn             winioctlcon
brain_attrs         jupyter_console     secrets             winnt
brain_builtin_inference jupyter_core        select              winperf
brain_collections   jupyterlab          selectors           winpty
brain_curses        jupyterlab_launcher send2trash          winreg
brain_dateutil      jupyterlab_server   service_identity    winsound
brain_fstrings      keyring             servicemanager      winxpgui
brain_functools     keyword             setuptools          winxptheme
brain_gi            kiwisolver          shelve              wrapt
brain_hashlib       lazy_object_proxy   shlex               wsgiref
brain_http          lib2to3             shutil              xdrlib
brain_io            libarchive          signal              xlrd
brain_mechanize     libfuturize         simplegeneric       xlsxwriter
brain_multiprocessing libpasteurize       singledispatch      xlwings
brain_namedtuple_enum lief                singledispatch_helpers xlwt
brain_nose          linecache           sip                 xml
brain_numpy         llvmlite            sipconfig           xmlrpc
brain_pkg_resources locale              sipdistutils        xxsubtype
brain_pytest        locket              site                yaml
brain_qt            logging             six                 zict
brain_random        lxml                skimage             zipapp
brain_re            lzma                sklearn             zipfile
brain_six           macpath             smtpd               zipimport
brain_ssl           mailbox             smtplib             zipp
brain_subprocess    mailcap             sndhdr              zlib
brain_threading     markupsafe          snowballstemmer     zmq
brain_typing        marshal             socket              zope
brain_uuid          math                socketserver
bs4                 matplotlib          socks
builtins            mccabe              sockshandler

Enter any module name to get more help.  Or, type "modules spam" to search
for modules whose name or summary contain the string "spam".

None
DEBUG:matplotlib.pyplot:Loaded backend module://ipykernel.pylab.backend_inline
version unknown.
```

In [2]:
```
pip list
```

```
...
```

In [3]:
```
import numpy
```

In [4]:
```
pip install numpy
```

```
Requirement already satisfied: numpy in c:\users\alekhya\anaconda3\lib\site-pac
kages (1.16.2)

WARNING: You are using pip version 20.2.3; however, version 21.1.1 is availabl
e.
You should consider upgrading via the 'C:\Users\Alekhya\Anaconda3\python.exe -m
pip install --upgrade pip' command.

Note: you may need to restart the kernel to use updated packages.
```

In [5]:
```
import numpy as np
```

In [6]:
```
print(dir(np))
```

```
['ALLOW_THREADS', 'AxisError', 'BUFSIZE', 'CLIP', 'ComplexWarning', 'DataSourc
e', 'ERR_CALL', 'ERR_DEFAULT', 'ERR_IGNORE', 'ERR_LOG', 'ERR_PRINT', 'ERR_RAIS
E', 'ERR_WARN', 'FLOATING_POINT_SUPPORT', 'FPE_DIVIDEBYZERO', 'FPE_INVALID', 'F
PE_OVERFLOW', 'FPE_UNDERFLOW', 'False_', 'Inf', 'Infinity', 'MAXDIMS', 'MAY_SHA
RE_BOUNDS', 'MAY_SHARE_EXACT', 'MachAr', 'ModuleDeprecationWarning', 'NAN', 'NI
NF', 'NZERO', 'NaN', 'PINF', 'PZERO', 'RAISE', 'RankWarning', 'SHIFT_DIVIDEBYZE
RO', 'SHIFT_INVALID', 'SHIFT_OVERFLOW', 'SHIFT_UNDERFLOW', 'ScalarType', 'Teste
r', 'TooHardError', 'True_', 'UFUNC_BUFSIZE_DEFAULT', 'UFUNC_PYVALS_NAME', 'Vis
ibleDeprecationWarning', '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', '_distribut
or_init', '_globals', '_mat', '_mklinit', '_pytesttester', 'abs', 'absolute',
'absolute_import', 'add', 'add_docstring', 'add_newdoc', 'add_newdoc_ufunc', 'a
len', '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', 'as
array_chkfinite', 'ascontiguousarray', 'asfarray', 'asfortranarray', 'asmatri
x', 'asscalar', 'atleast_1d', 'atleast_2d', 'atleast_3d', 'average', 'bartlet
t', 'base_repr', 'binary_repr', 'bincount', 'bitwise_and', 'bitwise_not', 'bitw
ise_or', 'bitwise_xor', 'blackman', 'block', 'bmat', 'bool', 'bool8', 'bool_',
'broadcast', 'broadcast_arrays', 'broadcast_to', 'busday_count', 'busday_offse
t', 'busdaycalendar', 'byte', 'byte_bounds', 'bytes0', 'bytes_', 'c_', 'can_cas
t', 'cast', 'cbrt', 'cdouble', 'ceil', 'cfloat', 'char', 'character', 'chararra
y', 'choose', 'clip', 'clongdouble', 'clongfloat', 'column_stack', 'common_typ
e', 'compare_chararrays', 'compat', 'complex', 'complex128', 'complex64', 'comp
lex_', 'complexfloating', 'compress', 'concatenate', 'conj', 'conjugate', 'conv
olve', 'copy', 'copysign', 'copyto', 'core', 'corrcoef', 'correlate', 'cos', 'c
osh', 'count_nonzero', 'cov', 'cross', 'csingle', 'ctypeslib', 'cumprod', 'cump
roduct', 'cumsum', 'datetime64', 'datetime_as_string', 'datetime_data', 'deg2ra
d', 'degrees', 'delete', 'deprecate', 'deprecate_with_doc', 'diag', 'diag_indic
es', 'diag_indices_from', 'diagflat', 'diagonal', 'diff', 'digitize', 'disp',
'distutils', 'divide', 'division', 'divmod', 'doc', 'dot', 'double', 'dsplit',
'dstack', 'dtype', 'dual', 'e', 'ediff1d', 'einsum', 'einsum_path', 'emath', 'e
mpty', 'empty_like', 'equal', 'errstate', 'euler_gamma', 'exp', 'exp2', 'expand
_dims', 'expm1', 'extract', 'eye', 'f2py', 'fabs', 'fastCopyAndTranspose', 'ff
t', 'fill_diagonal', 'find_common_type', 'finfo', 'fix', 'flatiter', 'flatnonze
ro', 'flexible', 'flip', 'fliplr', 'flipud', 'float', 'float16', 'float32', 'fl
oat64', 'float_', 'float_power', 'floating', 'floor', 'floor_divide', 'fmax',
'fmin', 'fmod', 'format_float_positional', 'format_float_scientific', 'format_p
arser', 'frexp', 'frombuffer', 'fromfile', 'fromfunction', 'fromiter', 'frompyf
unc', 'fromregex', 'fromstring', 'full', 'full_like', 'fv', 'gcd', 'generic',
'genfromtxt', 'geomspace', 'get_array_wrap', 'get_include', 'get_printoptions',
'getbufsize', 'geterr', 'geterrcall', 'geterrobj', 'gradient', 'greater', 'grea
ter_equal', 'half', 'hamming', 'hanning', 'heaviside', 'histogram', 'histogram2
d', 'histogram_bin_edges', 'histogramdd', 'hsplit', 'hstack', 'hypot', 'i0', 'i
dentity', 'iinfo', 'imag', 'in1d', 'index_exp', 'indices', 'inexact', 'inf', 'i
nfo', 'infty', 'inner', 'insert', 'int', 'int0', 'int16', 'int32', 'int64', 'in
t8', 'int_', 'int_asbuffer', 'intc', 'integer', 'interp', 'intersect1d', 'int
p', 'invert', 'ipmt', 'irr', 'is_busday', 'isclose', 'iscomplex', 'iscomplexob
j', 'isfinite', 'isfortran', 'isin', 'isinf', 'isnan', 'isnat', 'isneginf', 'is
posinf', 'isreal', 'isrealobj', 'isscalar', 'issctype', 'issubclass_', 'issubdt
ype', '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', 'logaddex
p', 'logaddexp2', 'logical_and', 'logical_not', 'logical_or', 'logical_xor', 'l
ogspace', 'long', 'longcomplex', 'longdouble', 'longfloat', 'longlong', 'lookfo
r', 'ma', 'mafromtxt', 'mask_indices', 'mat', 'math', 'matmul', 'matrix', 'matr
ixlib', 'max', 'maximum', 'maximum_sctype', 'may_share_memory', 'mean', 'media
n', 'memmap', 'meshgrid', 'mgrid', 'min', 'min_scalar_type', 'minimum', 'mintyp
ecode', 'mirr', 'mod', 'modf', 'moveaxis', 'msort', 'multiply', 'nan', 'nan_to_
num', 'nanargmax', 'nanargmin', 'nancumprod', 'nancumsum', 'nanmax', 'nanmean',
'nanmedian', 'nanmin', 'nanpercentile', 'nanprod', 'nanquantile', 'nanstd', 'na
nsum', 'nanvar', 'nbytes', 'ndarray', 'ndenumerate', 'ndfromtxt', 'ndim', 'ndin
dex', 'nditer', 'negative', 'nested_iters', 'newaxis', 'nextafter', 'nonzero',
'not_equal', 'nper', 'npv', 'numarray', 'number', 'obj2sctype', 'object', 'obje
ct0', 'object_', 'ogrid', 'oldnumeric', 'ones', 'ones_like', 'outer', 'packbit
s', 'pad', 'partition', 'percentile', 'pi', 'piecewise', 'place', 'pmt', 'pol
y', 'poly1d', 'polyadd', 'polyder', 'polydiv', 'polyfit', 'polyint', 'polymul',
'polynomial', 'polysub', 'polyval', 'positive', 'power', 'ppmt', 'print_functio
n', 'printoptions', 'prod', 'product', 'promote_types', 'ptp', 'put', 'put_alon
g_axis', 'putmask', 'pv', 'quantile', 'r_', 'rad2deg', 'radians', 'random', 'ra
ndom_intel', 'rank', 'rate', 'ravel', 'ravel_multi_index', 'real', 'real_if_clo
se', 'rec', 'recarray', 'recfromcsv', 'recfromtxt', 'reciprocal', 'record', 're
mainder', 'repeat', 'require', 'reshape', 'resize', 'result_type', 'right_shif
t', 'rint', 'roll', 'rollaxis', 'roots', 'rot90', 'round', 'round_', 'row_stac
k', 's_', 'safe_eval', 'save', 'savetxt', 'savez', 'savez_compressed', 'sctype2
char', 'sctypeDict', 'sctypeNA', 'sctypes', 'searchsorted', 'select', 'set_nume
ric_ops', 'set_printoptions', 'set_string_function', 'setbufsize', 'setdiff1d',
'seterr', 'seterrcall', 'seterrobj', 'setxor1d', 'shape', 'shares_memory', 'sho
rt', 'show_config', 'sign', 'signbit', 'signedinteger', 'sin', 'sinc', 'singl
e', 'singlecomplex', 'sinh', 'size', 'sometrue', 'sort', 'sort_complex', 'sourc
e', 'spacing', 'split', 'sqrt', 'square', 'squeeze', 'stack', 'std', 'str', 'st
r0', 'str_', 'string_', 'subtract', 'sum', 'swapaxes', 'sys', 'take', 'take_alo
ng_axis', 'tan', 'tanh', 'tensordot', 'test', 'testing', 'tests', 'tile', 'time
delta64', 'trace', 'tracemalloc_domain', 'transpose', 'trapz', 'tri', 'tril',
'tril_indices', 'tril_indices_from', 'trim_zeros', 'triu', 'triu_indices', 'tri
u_indices_from', 'true_divide', 'trunc', 'typeDict', 'typeNA', 'typecodes', 'ty
pename', 'ubyte', 'ufunc', 'uint', 'uint0', 'uint16', 'uint32', 'uint64', 'uint
8', 'uintc', 'uintp', 'ulonglong', 'unicode', 'unicode_', 'union1d', 'unique',
'unpackbits', 'unravel_index', 'unsignedinteger', 'unwrap', 'ushort', 'vander',
'var', 'vdot', 'vectorize', 'version', 'void', 'void0', 'vsplit', 'vstack', 'wa
rnings', 'where', 'who', 'zeros', 'zeros_like']
```

In [7]:
```
help(np.test)
```

```
Help on PytestTester in module numpy._pytesttester object:

class PytestTester(builtins.object)
 |  PytestTester(module_name)
 |  
 |  Pytest test runner.
 |  
 |  This class is made available in ``numpy.testing``, and a test function
 |  is typically added to a package's __init__.py like so::
 |  
 |    from numpy.testing import PytestTester
 |    test = PytestTester(__name__).test
 |    del PytestTester
 |  
 |  Calling this test function finds and runs all tests associated with the
 |  module and all its sub-modules.
 |  
 |  Attributes
 |  ----------
 |  module_name : str
 |      Full path to the package to test.
 |  
 |  Parameters
 |  ----------
 |  module_name : module name
 |      The name of the module to test.
 |  
 |  Methods defined here:
 |  
 |  __call__(self, label='fast', verbose=1, extra_argv=None, doctests=False, co
verage=False, durations=-1, tests=None)
 |      Run tests for module using pytest.
 |      
 |      Parameters
 |      ----------
 |      label : {'fast', 'full'}, optional
 |          Identifies the tests to run. When set to 'fast', tests decorated
 |          with `pytest.mark.slow` are skipped, when 'full', the slow marker
 |          is ignored.
 |      verbose : int, optional
 |          Verbosity value for test outputs, in the range 1-3. Default is 1.
 |      extra_argv : list, optional
 |          List with any extra arguments to pass to pytests.
 |      doctests : bool, optional
 |          .. note:: Not supported
 |      coverage : bool, optional
 |          If True, report coverage of NumPy code. Default is False.
 |          Requires installation of (pip) pytest-cov.
 |      durations : int, optional
 |          If < 0, do nothing, If 0, report time of all tests, if > 0,
 |          report the time of the slowest `timer` tests. Default is -1.
 |      tests : test or list of tests
 |          Tests to be executed with pytest '--pyargs'
 |      
 |      Returns
 |      -------
 |      result : bool
 |          Return True on success, false otherwise.
 |
 |      Notes
 |      -----
 |      Each NumPy module exposes `test` in its namespace to run all tests for
 |      it. For example, to run all tests for numpy.lib:
 |
 |      >>> np.lib.test() #doctest: +SKIP
 |
 |      Examples
 |      --------
 |      >>> result = np.lib.test() #doctest: +SKIP
 |      ...
 |      1023 passed, 2 skipped, 6 deselected, 1 xfailed in 10.39 seconds
 |      >>> result
 |      True
 |
 |  __init__(self, module_name)
 |      Initialize self.  See help(type(self)) for accurate signature.
 |
 |  ----------------------------------------------------------------------
 |  Data descriptors defined here:
 |
 |  __dict__
 |      dictionary for instance variables (if defined)
 |
 |  __weakref__
 |      list of weak references to the object (if defined)
```

In [8]:
```
# creating 1d array
a = np.array([1,2,3,4])
a
```

Out[8]:
```
array([1, 2, 3, 4])
```

In [9]:
```
a.dtype
```

Out[9]:
```
dtype('int32')
```

In [10]:
```
b = np.array([1,2,"a"])
b
```

Out[10]:
```
array(['1', '2', 'a'], dtype='<U11')
```

In [11]:
```
b.dtype
```

Out[11]:
```
dtype('<U11')
```

```
In [12]: a = np.append(a,[2,3,4])
         a

Out[12]: array([1, 2, 3, 4, 2, 3, 4])
```

```
In [16]: for i in range(1,20,2):
             print(i)

1
3
5
7
9
11
13
15
17
19
```

```
In [15]: range(1,20,1.5)

---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-15-fa8e7873059b> in <module>
----> 1 range(1,20,1.5)

TypeError: 'float' object cannot be interpreted as an integer
```

```
In [18]: np.arange(1,20,1.5)

Out[18]: array([ 1. ,  2.5,  4. ,  5.5,  7. ,  8.5, 10. , 11.5, 13. , 14.5, 16. ,
                17.5, 19. ])
```

```
In [19]: np.linspace(1,20,10)

Out[19]: array([ 1.        ,  3.11111111,  5.22222222,  7.33333333,  9.44444444,
                11.55555556, 13.66666667, 15.77777778, 17.88888889, 20.        ])
```

```
In [20]: np.full(10,"apssdc")

Out[20]: array(['apssdc', 'apssdc', 'apssdc', 'apssdc', 'apssdc', 'apssdc',
                'apssdc', 'apssdc', 'apssdc', 'apssdc'], dtype='<U6')
```

```
In [21]: np.ones(10)

Out[21]: array([1., 1., 1., 1., 1., 1., 1., 1., 1., 1.])
```

```
In [23]: np.zeros(10,dtype=int)

Out[23]: array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0])
```

```
In [25]: np.eye(1)

Out[25]: array([[1.]])

In [26]: np.eye(2)

Out[26]: array([[1., 0.],
                [0., 1.]])

In [27]: a

Out[27]: array([1, 2, 3, 4, 2, 3, 4])

In [28]: a.ndim

Out[28]: 1

In [29]: # 2d array

In [30]: b = np.array([[1,2,3],[4,5,6]]) # number of rows and columns
         b

Out[30]: array([[1, 2, 3],
                [4, 5, 6]])

In [31]: [[1,2,3],[4,5,6]]

Out[31]: [[1, 2, 3], [4, 5, 6]]

In [32]: b.ndim

Out[32]: 2

In [33]: len(b)

Out[33]: 2

In [34]: b.size

Out[34]: 6

In [35]: # 3d array

In [38]: c = np.array([[[1,2,3],[5,6,7]],[[3,4,5],[7,8,9]]])
In [39]: c# position,rows,columns

Out[39]: array([[[1, 2, 3],
                 [5, 6, 7]],

                [[3, 4, 5],
                 [7, 8, 9]]])

In [ ]: [[[1,2,3],[5,6,7]],
         [[3,4,5],[7,8,9]]]

In [40]: c.ndim

Out[40]: 3

In [41]: # 1d array

In [42]: a

Out[42]: array([1, 2, 3, 4, 2, 3, 4])

In [43]: a[5]

Out[43]: 3

In [44]: a[-1]

Out[44]: 4

In [45]: a[-4]

Out[45]: 4

In [46]: a[1:5]

Out[46]: array([2, 3, 4, 2])

In [47]: a[1:5:2]

Out[47]: array([2, 4])

In [48]: # 2d array

In [49]: b

Out[49]: array([[1, 2, 3],
                [4, 5, 6]])
```

```
In [50]: b[0,0]
```

```
Out[50]: 1
```

```
In [51]: b[0]
```

```
Out[51]: array([1, 2, 3])
```

```
In [52]: # 3d array
```

```
In [53]: c
```

```
Out[53]: array([[[1, 2, 3],
                 [5, 6, 7]],

                [[3, 4, 5],
                 [7, 8, 9]]])
```

```
In [54]: c[1,1,1]
```

```
Out[54]: 8
```

```
In [55]: # reshaping an array
```

```
In [56]: a = np.array(np.arange(1,11))
```

```
In [57]: a
```

```
Out[57]: array([ 1,  2,  3,  4,  5,  6,  7,  8,  9, 10])
```

```
In [58]: a.shape
```

```
Out[58]: (10,)
```

```
In [60]: b1 = a.reshape(5,2)
         b1
```

```
Out[60]: array([[ 1,  2],
                [ 3,  4],
                [ 5,  6],
                [ 7,  8],
                [ 9, 10]])
```

```
In [62]: b1.ndim
```

```
Out[62]: 2
```

In [63]:
```
a.reshape(2,5)
```
Out[63]:
```
array([[ 1,  2,  3,  4,  5],
       [ 6,  7,  8,  9, 10]])
```

In [65]:
```
a.reshape(2,6)
```
```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-65-35ce22445ebd> in <module>
----> 1 a.reshape(2,6)

ValueError: cannot reshape array of size 10 into shape (2,6)
```

In [66]:
```
a
```
Out[66]:
```
array([ 1,  2,  3,  4,  5,  6,  7,  8,  9, 10])
```

In [67]:
```
a.reshape(1,2,5)
```
Out[67]:
```
array([[[ 1,  2,  3,  4,  5],
        [ 6,  7,  8,  9, 10]]])
```

In [68]:
```
a.reshape(1,5,2)
```
Out[68]:
```
array([[[ 1,  2],
        [ 3,  4],
        [ 5,  6],
        [ 7,  8],
        [ 9, 10]]])
```

In [69]:
```
a.reshape(2,1,5)
```
Out[69]:
```
array([[[ 1,  2,  3,  4,  5]],

       [[ 6,  7,  8,  9, 10]]])
```

In [70]:
```
a.reshape(-1,2)# -1 is unknown,2 is known
```
Out[70]:
```
array([[ 1,  2],
       [ 3,  4],
       [ 5,  6],
       [ 7,  8],
       [ 9, 10]])
```

In [71]:
```
a
```
Out[71]:
```
array([ 1,  2,  3,  4,  5,  6,  7,  8,  9, 10])
```

```
In [72]: a.reshape(5,-1)

Out[72]: array([[ 1,  2],
                [ 3,  4],
                [ 5,  6],
                [ 7,  8],
                [ 9, 10]])
```

```
In [73]: a.reshape(-1,-1)

---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-73-aa43799da6cd> in <module>
----> 1 a.reshape(-1,-1)

ValueError: can only specify one unknown dimension
```

```
In [74]: a.reshape(6,-1)

---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-74-155e25d74457> in <module>
----> 1 a.reshape(6,-1)

ValueError: cannot reshape array of size 10 into shape (6,newaxis)
```

```
In [75]: # concatenation
```

```
In [78]: a1  = np.array([1,2,3])
         a2 = np.array([5,6,7])
         np.concatenate((a1,a2))

Out[78]: array([1, 2, 3, 5, 6, 7])
```

```
In [79]: b

Out[79]: array([[1, 2, 3],
                [4, 5, 6]])
```

```
In [80]: b1 = np.array([[3,4,5],[7,8,9]])
         b1

Out[80]: array([[3, 4, 5],
                [7, 8, 9]])
```

```
In [82]: np.concatenate((b,b1),axis=0) # axis = 0 , columns

Out[82]: array([[1, 2, 3],
                [4, 5, 6],
                [3, 4, 5],
                [7, 8, 9]])

In [83]: np.concatenate((b,b1),axis=1) # axis =1 , rows

Out[83]: array([[1, 2, 3, 3, 4, 5],
                [4, 5, 6, 7, 8, 9]])

In [84]: b

Out[84]: array([[1, 2, 3],
                [4, 5, 6]])

In [85]: np.min(b)

Out[85]: 1

In [86]: np.max(b)

Out[86]: 6

In [87]: np.mean(a)

Out[87]: 5.5

In [88]: np.median(b)

Out[88]: 3.5

In [89]: np.argmin(a)# index value of minimum element

Out[89]: 0

In [90]: np.average(b)

Out[90]: 3.5

In [91]: np.var(b)

Out[91]: 2.9166666666666665

In [92]: np.std(b)

Out[92]: 1.707825127659933
```

```
In [93]: np.sum(b)

Out[93]: 21

In [94]: np.cumsum(b)

Out[94]: array([ 1,  3,  6, 10, 15, 21], dtype=int32)

In [95]: b

Out[95]: array([[1, 2, 3],
                [4, 5, 6]])

In [96]: np.min(b,axis=1)

Out[96]: array([1, 4])

In [97]: np.min(b,axis=0)

Out[97]: array([1, 2, 3])

In [98]: np.argmin(b,axis=1)

Out[98]: array([0, 0], dtype=int64)

In [99]: np.argmax(b,axis=1)

Out[99]: array([2, 2], dtype=int64)

In [100]: np.log(a)

Out[100]: array([0.        , 0.69314718, 1.09861229, 1.38629436, 1.60943791,
                 1.79175947, 1.94591015, 2.07944154, 2.19722458, 2.30258509])

In [101]: np.log2(a)

Out[101]: array([0.        , 1.        , 1.5849625 , 2.        , 2.32192809,
                 2.5849625 , 2.80735492, 3.        , 3.169925  , 3.32192809])

In [102]: np.log10(a)

Out[102]: array([0.        , 0.30103   , 0.47712125, 0.60205999, 0.69897   ,
                 0.77815125, 0.84509804, 0.90308999, 0.95424251, 1.        ])

In [103]: np.exp(a)

Out[103]: array([2.71828183e+00, 7.38905610e+00, 2.00855369e+01, 5.45981500e+01,
                 1.48413159e+02, 4.03428793e+02, 1.09663316e+03, 2.98095799e+03,
                 8.10308393e+03, 2.20264658e+04])
```

```
In [104]: # stacking--arranging elements in proper order
          # horizontal stacking
          # vertical stacking
```

```
In [105]: b
```

```
Out[105]: array([[1, 2, 3],
                 [4, 5, 6]])
```

```
In [106]: np.hstack(b)
```

```
Out[106]: array([1, 2, 3, 4, 5, 6])
```

```
In [107]: np.vstack(b)
```

```
Out[107]: array([[1, 2, 3],
                 [4, 5, 6]])
```

```
In [108]: a
```

```
Out[108]: array([ 1,  2,  3,  4,  5,  6,  7,  8,  9, 10])
```

```
In [109]: np.vstack(a)
```

```
Out[109]: array([[ 1],
                 [ 2],
                 [ 3],
                 [ 4],
                 [ 5],
                 [ 6],
                 [ 7],
                 [ 8],
                 [ 9],
                 [10]])
```

```
In [110]: np.sqrt(a)
```

```
Out[110]: array([1.        , 1.41421356, 1.73205081, 2.        , 2.23606798,
                 2.44948974, 2.64575131, 2.82842712, 3.        , 3.16227766])
```

```
In [111]: np.remainder(b,4)
```

```
Out[111]: array([[1, 2, 3],
                 [0, 1, 2]], dtype=int32)
```

```
In [112]: b
```

```
Out[112]: array([[1, 2, 3],
                 [4, 5, 6]])
```

```
In [114]: np.divide(b,4)

Out[114]: array([[0.25, 0.5 , 0.75],
                 [1.  , 1.25, 1.5 ]])

In [115]: np.multiply(b,4)

Out[115]: array([[ 4,  8, 12],
                 [16, 20, 24]])

In [116]: b

Out[116]: array([[1, 2, 3],
                 [4, 5, 6]])

In [117]: np.power(b,6)

Out[117]: array([[    1,    64,   729],
                 [ 4096, 15625, 46656]], dtype=int32)

In [118]: # random methods

In [120]: np.random.random()# it returns value between 0 and 1

Out[120]: 0.28249288641301584

In [122]: np.random.random(10)

Out[122]: array([0.79879588, 0.45521457, 0.72762685, 0.3498936 , 0.91684483,
                 0.13318862, 0.87397832, 0.93346432, 0.90604471, 0.45249893])

In [124]: np.random.random((2,10))

Out[124]: array([[0.43184262, 0.85083435, 0.89706833, 0.83312331, 0.62963933,
                  0.48446593, 0.81363275, 0.67828732, 0.23872463, 0.88312302],
                 [0.05004922, 0.0656812 , 0.69485668, 0.36095224, 0.74189182,
                  0.52343557, 0.99422892, 0.47225979, 0.73841332, 0.74184444]])

In [125]: np.random.random((2,5,2))

Out[125]: array([[[0.79533524, 0.8482311 ],
                  [0.3497417 , 0.90025455],
                  [0.30666866, 0.90552364],
                  [0.1720088 , 0.60916876],
                  [0.27597381, 0.57880213]],

                 [[0.56367574, 0.11786503],
                  [0.03122588, 0.01190475],
                  [0.81277031, 0.96643305],
                  [0.87442441, 0.6313523 ],
                  [0.89666142, 0.78594727]]])
```

```
In [130]: np.random.randint(10)

Out[130]: 2

In [131]: np.random.randint(10,40)

Out[131]: 15
```

Filtering

```
In [132]: marks = np.array([55,34,23,56,78,90,100,46])
          marks

Out[132]: array([ 55,  34,  23,  56,  78,  90, 100,  46])

In [133]: marks>35

Out[133]: array([ True, False, False,  True,  True,  True,  True,  True])

In [134]: marks[marks>35]

Out[134]: array([ 55,  56,  78,  90, 100,  46])

In [135]: x = np.arange(50,101)

In [136]: x

Out[136]: array([ 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])

In [138]: x[x%2==0]

Out[138]: array([ 50,  52,  54,  56,  58,  60,  62,  64,  66,  68,  70,  72,  74,
                  76,  78,  80,  82,  84,  86,  88,  90,  92,  94,  96,  98, 100])

In [140]: x[(x>60) & (x<80)]

Out[140]: array([61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77,
                 78, 79])

In [141]: x[((x>60) & (x<80)) & (x%2==0)]

Out[141]: array([62, 64, 66, 68, 70, 72, 74, 76, 78])

In [ ]: # |--->or
        # ~---->negation
```

```
In [144]: a = [1,2,3,4,5]# +4-->4,6,7,8,9
          b = []
          for i in a:
              b.append(i+4)
          print(b)

          [5, 6, 7, 8, 9]

In [145]: x

Out[145]: array([ 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])

In [146]: x+4

Out[146]: array([ 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, 101, 102, 103, 104])

In [ ]:
```