Assignment 2 - Numpy Array Operations

This assignment is part of the course "Data Analysis with Python: Zero to Pandas". The objective of this assignment is to develop a solid understanding of Numpy array operations. In this assignment you will:

1. Pick 5 interesting Numpy array functions by going through the documentation: https://numpy.org/doc/stable/reference/routines.html
2. Run and modify this Jupyter notebook to illustrate their usage (some explanation and 3 examples for each function). Use your imagination to come up with interesting and unique examples.
3. Upload this notebook to your Jovian profile using `jovian.commit` and make a submission here: https://jovian.ml/learn/data-analysis-with-python-zero-to-pandas/assignment/assignment-2-numpy-array-operations
4. (Optional) Share your notebook online (on Twitter, LinkedIn, Facebook) and on the community forum thread: https://jovian.ml/forum/t/assignment-2-numpy-array-operations-share-your-work/10575 .
5. (Optional) Check out the notebooks shared by other participants and give feedback & appreciation.

The recommended way to run this notebook is to click the "Run" button at the top of this page, and select "Run on Binder". This will run the notebook on mybinder.org, a free online service for running Jupyter notebooks.

Try to give your notebook a catchy title & subtitle e.g. "All about Numpy array operations", "5 Numpy functions you didn't know you needed", "A beginner's guide to broadcasting in Numpy", "Interesting ways to create Numpy arrays", "Trigonometic functions in Numpy", "How to use Python for Linear Algebra" etc.

NOTE: Remove this block of explanation text before submitting or sharing your notebook online - to make it more presentable.

Title Here

Subtitle Here

Write a short introduction about Numpy and list the chosen functions.

- function 1
- function 2
- function 3
- function 4
- function 5

The recommended way to run this notebook is to click the "Run" button at the top of this page, and select "Run on Binder". This will run the notebook on mybinder.org, a free online service for running Jupyter notebooks.

In [1]:
```
!pip install jovian --upgrade -q
```

In [2]:
```
import jovian
```

In [9]:
```
jovian.commit(project='numpy-array-operations')
```

```
[jovian] Attempting to save notebook..
[jovian] Updating notebook "kirankumarmb0002/numpy-array-operations" on https://jovian.ml/
[jovian] Uploading notebook..
[jovian] Capturing environment..
[jovian] Committed successfully! https://jovian.ml/kirankumarmb0002/numpy-array-operations
```

Out[9]:
```
'https://jovian.ml/kirankumarmb0002/numpy-array-operations'
```

Let's begin by importing Numpy and listing out the functions covered in this notebook.

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

In [8]:
```
# List of functionsndarray.T explained
function1 = np.ndarray.T
function2 = np.atleast_3d()
function3 = np.tile
function4 = np.asanyarray
function5 = np.trim_zeros
```

Function 1 - np.ndarray.T

In [17]:
```
x = np.array([[1.,2.,3.],[4.,5.,6.]])
print(x)
x.T
```

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

Out[17]:
```
array([[1., 4.],
       [2., 5.],
       [3., 6.]])
```

This function will transpose the array.It will take list of same size

In [26]:
```
# Example 2
x = np.array([1.,2.,3.,4.,5.,6.])
print(x)
x.T
```

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

Out[26]:
```
array([1., 2., 3., 4., 5., 6.])
```

in this function we will pass only one list,so it cannot transpose that

In [28]:
```
# Example 3 - breaking (to illustrate when it breaks)
x = np.array([[1.,2.,3.],[4.,5.]])
print(x)
x.T
```

```
[list([1.0, 2.0, 3.0]) list([4.0, 5.0])]
```

```
<ipython-input-28-a817406841cc>:2: VisibleDeprecationWarning: Creating an ndarray from ragged nested sequences (which is a list-or-tuple of lists-or-tuples-or ndarrays with different lengths or shapes) is deprecated. If you meant to do this, you must specify 'dtype=object' when creating the ndarray
  x = np.array([[1.,2.,3.],[4.,5.]])
```

Out[28]:
```
array([list([1.0, 2.0, 3.0]), list([4.0, 5.0])], dtype=object)
```

In this function we are passing two list of diffrent of diffrent size.we can fix it by pass a list of same size

Some closing comments about when to use this function.

In [29]:
```
jovian.commit()
```

```
[jovian] Attempting to save notebook..
[jovian] Updating notebook "kirankumarmb0002/numpy-array-operations" on https://jovian.ml/
[jovian] Uploading notebook..
[jovian] Capturing environment..
[jovian] Committed successfully! https://jovian.ml/kirankumarmb0002/numpy-array-operations
```

Out[29]:
```
'https://jovian.ml/kirankumarmb0002/numpy-array-operations'
```

Function 2

In [33]:
```
# Example 1
np.atleast_3d(4.0,5.0)
```

Out[33]:
```
[array([[[4.]]]), array([[[5.]]])]
```

this function View inputs as arrays with at least three dimensions

In [35]:
```
# Example 2
x = np.arange(3.0)
print(x)
np.atleast_3d(x).shape
```

```
[0. 1. 2.]
```

Out[35]:
```
(1, 3, 1)
```

This function returns the shape.1 is the number of rows,3 is the number of coloumn,1 is the dimension

In [42]:
```
# Example 3 - breaking (to illustrate when it breaks)
x = np.arange(12.0).reshape(3)
print(x)
np.atleast_3d(x).shape
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-42-e7809b0d6166> in <module>
      1 # Example 3 - breaking (to illustrate when it breaks)
----> 2 x = np.arange(12.0).reshape(3)
      3 print(x)
      4 np.atleast_3d(x).shape

ValueError: cannot reshape array of size 12 into shape (3,)
```

In this function,the reshape function has two parameters we passed only one,we can fix it by passing 2 parameters ie (4,3)

this function is used when we require 3d array.

In [44]:
```
jovian.commit()
```

```
[jovian] Attempting to save notebook..
[jovian] Updating notebook "kirankumarmb0002/numpy-array-operations" on https://jovian.ml/
[jovian] Uploading notebook..
[jovian] Capturing environment..
[jovian] Committed successfully! https://jovian.ml/kirankumarmb0002/numpy-array-operations
```

Out[44]:
```
'https://jovian.ml/kirankumarmb0002/numpy-array-operations'
```

Function 3 -np.tile((A, reps)

Constructing an array by repeating A the number of times given by reps.

In [45]:
```
# Example 1
a = np.array([0, 1, 2])
np.tile(a, 2)
```

Out[45]:
```
array([0, 1, 2, 0, 1, 2])
```

This function is used to repeat the list element for a particular time

In [46]:
```
# Example 2
a = np.array([0, 1, 2])
np.tile(a, (2, 2))
```

Out[46]:
```
array([[0, 1, 2, 0, 1, 2],
       [0, 1, 2, 0, 1, 2]])
```

This function is used to repeat the list element for a particular time & repeats the list again

In [58]:
```
# Example 3
b = np.array([[1, 2], [3., 4,5.]])
np.tile(b, (2,3))
```

```
<ipython-input-58-f2606485eef7>:2: VisibleDeprecationWarning: Creating an ndarray from ragged nested sequences (which is a list-or-tuple of lists-or-tuples-or ndarrays with different lengths or shapes) is deprecated. If you meant to do this, you must specify 'dtype=object' when creating the ndarray
  b = np.array([[1, 2], [3., 4,5.]])
```

Out[58]:
```
array([[list([1, 2]), list([3.0, 4, 5.0]), list([1, 2]),
        list([3.0, 4, 5.0]), list([1, 2]), list([3.0, 4, 5.0])],
       [list([1, 2]), list([3.0, 4, 5.0]), list([1, 2]),
        list([3.0, 4, 5.0]), list([1, 2]), list([3.0, 4, 5.0])]],
      dtype=object)
```

in this function the array should have same number of elements in the list

Some closing comments about when to use this function.

In [59]:
```
jovian.commit()
```

```
[jovian] Attempting to save notebook..
[jovian] Updating notebook "kirankumarmb0002/numpy-array-operations" on https://jovian.ml/
[jovian] Uploading notebook..
[jovian] Capturing environment..
[jovian] Committed successfully! https://jovian.ml/kirankumarmb0002/numpy-array-operations
```

Out[59]:
```
'https://jovian.ml/kirankumarmb0002/numpy-array-operations'
```

Function 4 - numpy.asanyarray(a, dtype=None, order=None)

Convert the input to an ndarray, but pass ndarray subclasses through.

In [61]:
```
# Example 1 - working
a = [1, 2]
np.asanyarray(a)
```

Out[61]:
```
array([1, 2])
```

This function will return the the list to an ndarray

In [62]:
```
# Example 2
a = np.array([(1.0, 2), (3.0, 4)], dtype='f4,i4').view(np.recarray)
np.asanyarray(a) is a
```

Out[62]:
```
True
```

This function has parameters of dtype which is float

In [66]:
```
# Example 3
a = np.array([(1, 2), (3,5, 4)], dtype='float').view(np.recarray)
np.asanyarray(a) is a
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
TypeError: float() argument must be a string or a number, not 'tuple'

The above exception was the direct cause of the following exception:

ValueError                                Traceback (most recent call last)
<ipython-input-66-f885554a82cc> in <module>
      1 # Example 3
----> 2 a = np.array([(1, 2), (3,5, 4)], dtype='float').view(np.recarray)
      3 np.asanyarray(a) is a

ValueError: setting an array element with a sequence.
```

the error is that the elements are not of the same size

Some closing comments about when to use this function.

In [68]:
```
jovian.commit()
```

```
[jovian] Attempting to save notebook..
[jovian] Updating notebook "kirankumarmb0002/numpy-array-operations" on https://jovian.ml/
[jovian] Uploading notebook..
[jovian] Capturing environment..
[jovian] Committed successfully! https://jovian.ml/kirankumarmb0002/numpy-array-operations
```

Out[68]:
```
'https://jovian.ml/kirankumarmb0002/numpy-array-operations'
```

Function 5 -np.trim_zeros(filt, trim='fb')

Trim the leading and/or trailing zeros from a 1-D array or sequence.

In [69]:
```
# Example 1
a = np.array((0, 0, 0, 1, 2, 3, 0, 2, 1, 0))
np.trim_zeros(a)
```

Out[69]:
```
array([1, 2, 3, 0, 2, 1])
```

Explanation about example

In [70]:
```
# Example 2 - working
np.trim_zeros([0, 1, 2, 0])
```

Out[70]:
```
[1, 2]
```

Explanation about example

In [74]:
```
# Example 3 - breaking (to illustrate when it breaks)
np.trim_zeros(a,'a','b')
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-74-e4a36732b528> in <module>
      1 # Example 3 - breaking (to illustrate when it breaks)
----> 2 np.trim_zeros(a,'a','b')

<__array_function__ internals> in trim_zeros(*args, **kwargs)

TypeError: _trim_zeros() takes from 1 to 2 positional arguments but 3 were given
```