Enumerated Iteration Using ndenumerate()

Enumeration means mentioning sequence number of somethings one by one.

Sometimes we require corresponding index of the element while iterating, the ndenumerate() method can be used for those usecases.

e.g

1)
```
import numpy as np
arr = np.array([1,2,3])
for idx, x in np.ndenumerate(arr):
    print(idx,x)
```

Output

```
(0,) 1
(1,) 2
(2,) 3
```

2)
```
import numpy as np
arr = np.array([[1,2,3,4],[5,6,7,8]])
for idx, x in np.ndenumerate(arr):
    print(idx,x)
```

Output

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

Numpy Joining Array

1) Join two arrays:

```
import numpy as np
arr1 = np.array([1,2,3])
arr2 = np.array([4,5,6])
arr = np.concatenate((arr1, arr2))
print(arr)     O/P  [1 2 3 4 5 6]
```

2) Join two 2-D arrays along rows (axis=1)

```
import numpy as np
arr1 = np.array([[1,2],[3,4]])
arr2 = np.array([[5,6],[7,8]])
arr = np.concatenate((arr1, arr2), axis=1)
print(arr)
```

Output

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

Joining Arrays Using Stack Functions

```
import numpy as np
arr1 = np.array([1,2,3])
arr2 = np.array([4,5,6])
arr = np.stack((arr1, arr2), axis=1)
print(arr)
```

Output

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

Stacking Along Rows

Numpy provides a helper function: hstack() to stack along rows.

```
    ~~impos~~
import numpy as np
arr1 = np.array([1,2,3])
arr2 = np.array([4,5,6])
arr = np.hstack ((arr1, arr2))
print(arr)
```

Output

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

Stacking Along Columns

Numpy provides a helper function: vstack() to stack along columns

eg:

```
import numpy as np
arr1 = np.array ([1,2,3])
arr2 = np.array ([4,5,6])
arr = np. vstack ((arr1,arr2))
print( arr)
```

Output

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

Stacking Along Height (depth)

Numpy provides a helper function: dstack() to stack along height, which is the same as depth.

```
import numpy as np
arr1 = np.array([1, 2, 3])
arr2 = np.array([4, 5, 6])
arr = np.dstack((arr1, arr2))
print(arr)
```

Output

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