Swapping NumPy Arrays

While handling arrays it might be required to swap rows for matrix calculation, for this swapping operation is performed.

A) Using index positions

- The index values of columns to be swapped are given as input.
- **Syntax:**

> **arr[:,[frm, to]] = arr[:,[to, frm]]**
>
> OR
>
> **arr[:, [start_index, last_index]] = arr[:, [last_index, start_index]]**

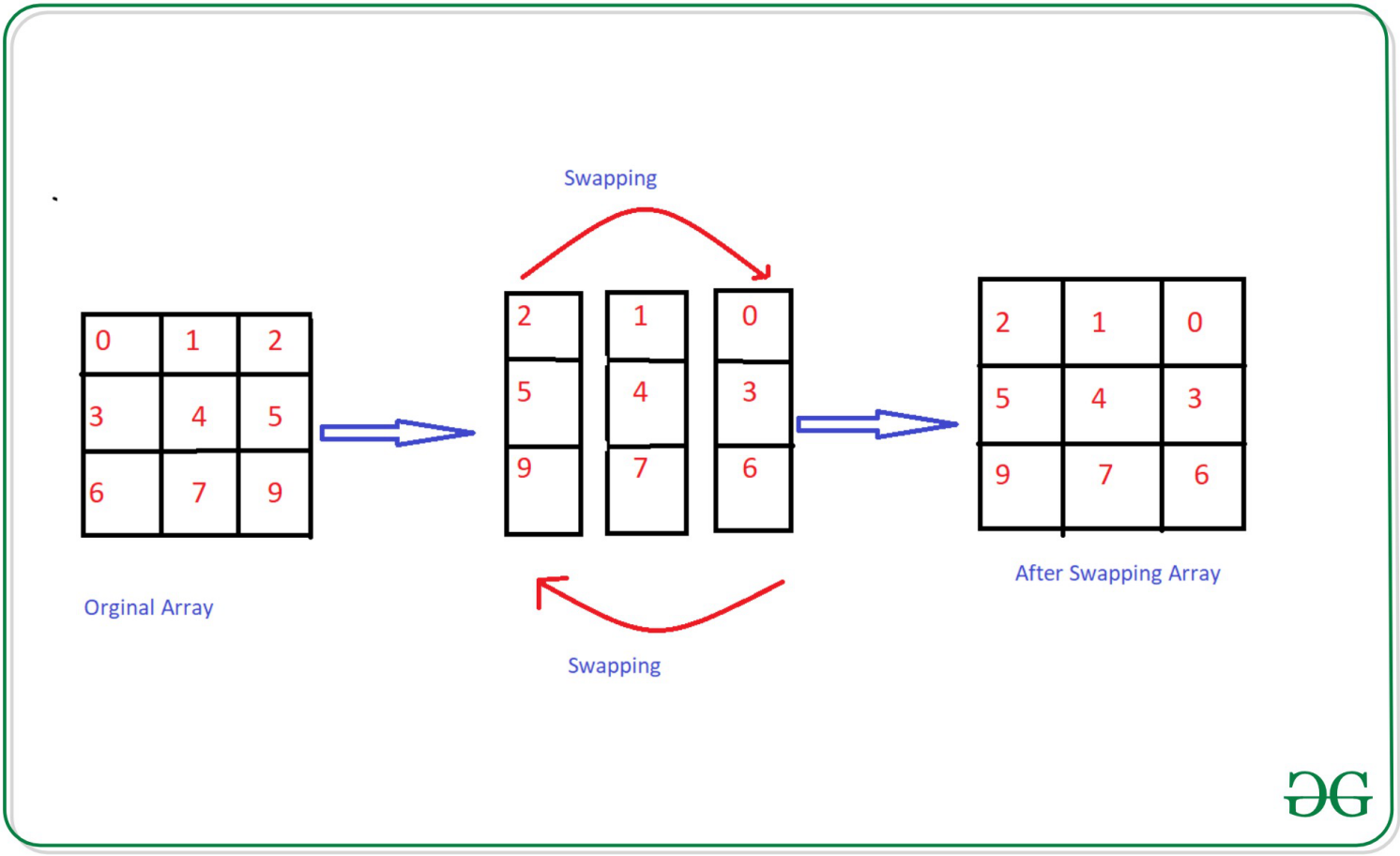
B) Using swapaxes() for multi-dimensional array (3D or higher)

- The swapaxes() **takes a pair of axis numbers and switches the indicated axes to rearrange the data.**
- This function interchanges the two axes of an array.
- **Syntax:**

> **numpy.swapaxes(arr, axis1, axis2)**
>
> 1. arr - Input array whose axes are to be swapped
> 2. axis1 - An int corresponding to the first axis
> 3. axis2 - An int corresponding to the second axis

OR

numpy.swapaxes(a, source, destination)



A) Using index positions

> arr[:, [start_index, last_index]] = arr[:, [last_index, start_index]]

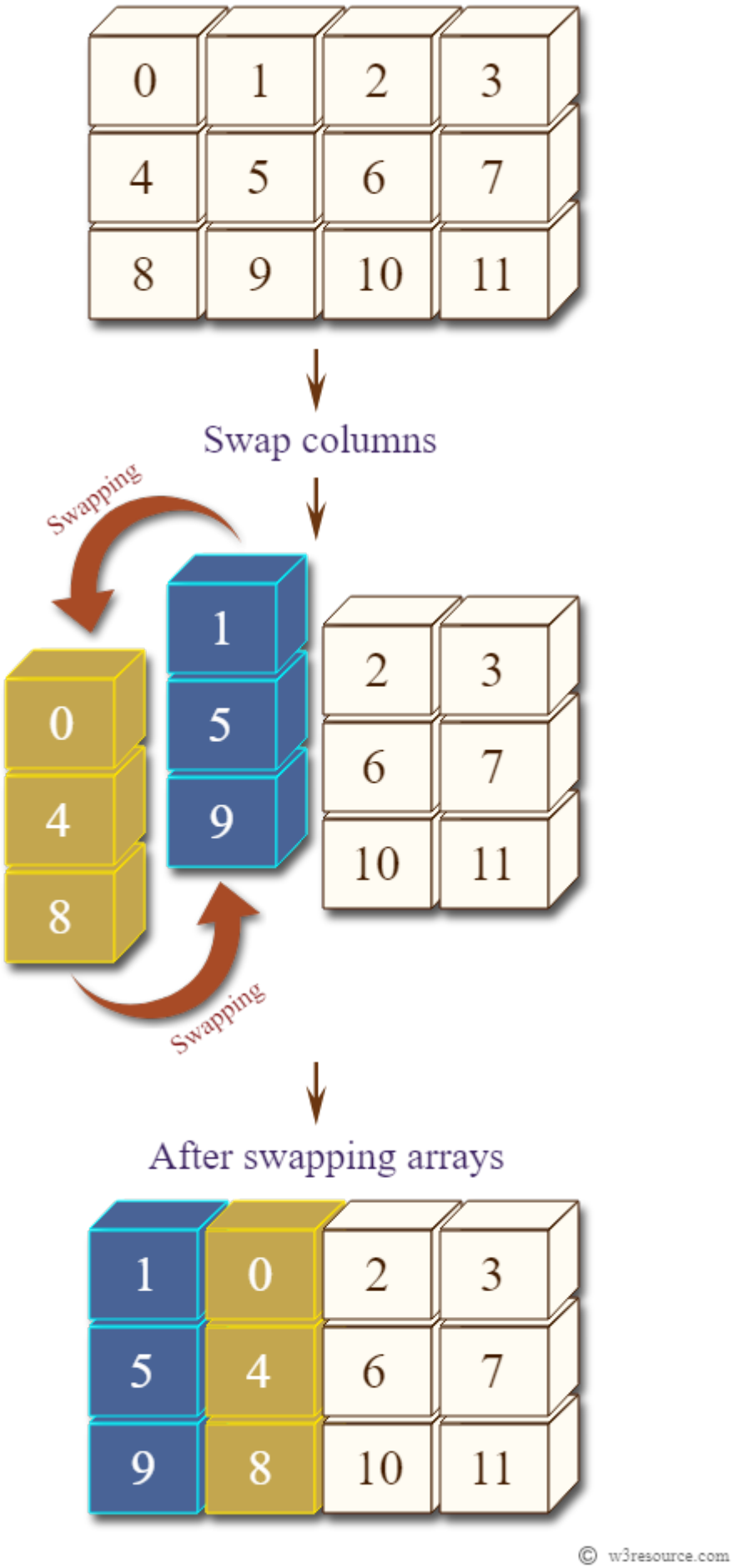
```
# importing Module
import numpy as np


# creating array with shape(4,3)
my_array = np.arange(12).reshape(4, 3)
print("Original array:")
print(my_array)

# swapping the column with index of
# original array
my_array[:, [0, 2]] = my_array[:, [2, 0]]
print("After swapping arrays the last column and first column:")
print(my_array)
```

```
Original array:
[[ 0  1  2]
 [ 3  4  5]
 [ 6  7  8]
 [ 9 10 11]]
After swapping arrays the last column and first column:
[[ 2  1  0]
 [ 5  4  3]
```

```
 [ 8  7  6]
 [11 10  9]]
```

```
import numpy as np

my_array = np.arange(12).reshape(3, 4)
print("Original array:")
print(my_array)
print("\n")

my_array[:,[0, 1]] = my_array[:,[1, 0]]
print("\nAfter swapping arrays:")
print(my_array)
```

```
Original array:
[[ 0  1  2  3]
 [ 4  5  6  7]
 [ 8  9 10 11]]



After swapping arrays:
[[ 1  0  2  3]
 [ 5  4  6  7]
 [ 9  8 10 11]]
```




```
# Importing Module
import numpy as np


# Creating array
```

```
my_array = np.arange(12).reshape(4, 3)
print(my_array)
print("\n")

#Swapping
my_array[:, [0, 1]] = my_array[:, [2, 0]]
print(my_array)
```

```
[[ 0  1  2]
 [ 3  4  5]
 [ 6  7  8]
 [ 9 10 11]]


[[ 2  0  2]
 [ 5  3  5]
 [ 8  6  8]
 [11  9 11]]
```

```
import numpy as np

my_array = np.arange(12).reshape(3, 4)
print("Original array:")
print(my_array)

print("\n")

my_array[:,[2, 3]] = my_array[:,[3, 0]]
print("\nAfter swapping arrays:")
print(my_array)
```

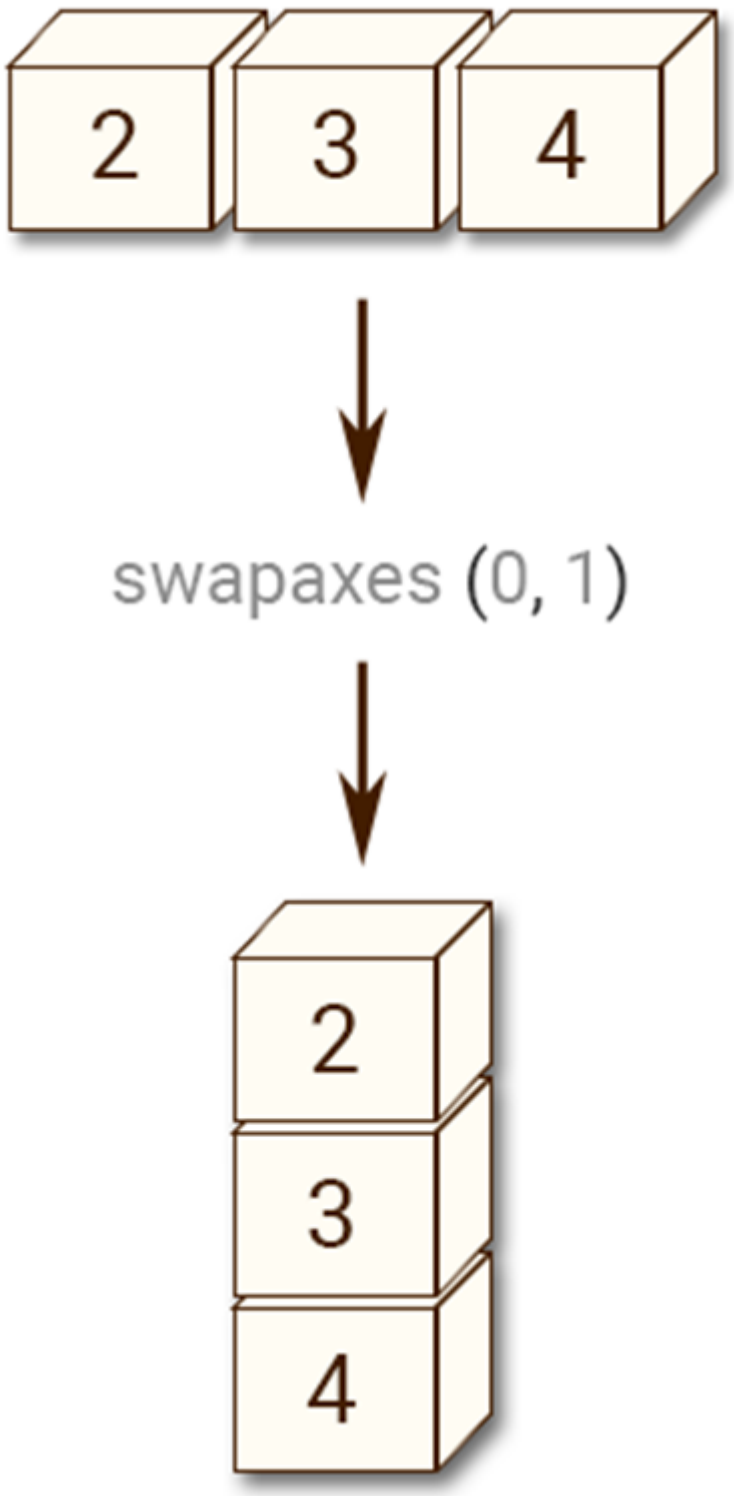
```
Original array:
[[ 0  1  2  3]
 [ 4  5  6  7]
 [ 8  9 10 11]]



After swapping arrays:
[[ 0  1  3  0]
 [ 4  5  7  4]
 [ 8  9 11  8]]
```

B) **Using swapaxes()**

```
import numpy as np
a = np.array([[2,3,4]])
print(a)
print("\n")

print(np.swapaxes(a,0,1)) #same as transpose
```

```
[[2 3 4]]


[[2]
```

```
 [3]
 [4]]
```




```
import numpy as np

x = np.array([[1,2,3]])
print(x)
print("\n")

y = np.swapaxes(x,0,1) #same as transpose
print(y)
```

```
[[1 2 3]]


[[1]
```

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

```
import numpy as np

x = np.array([[1,2,3],[20,30,40],[100,200,300]])
print(x)
print("\n")

y = np.swapaxes(x,0,1) #same as transpose
print(y)
```

```
[[  1   2   3]
 [ 20  30  40]
 [100 200 300]]


[[  1  20 100]
 [  2  30 200]
 [  3  40 300]]
```

```
x = np.array([[[1,2,3],[20,30,40]],[[5,55,555],[9,9,9]]])
print(x)
print("\n")

y = np.swapaxes(x,0,1) #same as transpose
print(y)
print("\n")

#is same as
m = np.swapaxes(x,1,0)
print(m)
print("\n")
```

```
[[[  1   2   3]
  [ 20  30  40]]

 [[  5  55 555]
  [  9   9   9]]]


[[[  1   2   3]
  [  5  55 555]]

 [[ 20  30  40]
  [  9   9   9]]]


[[[  1   2   3]
  [  5  55 555]]

 [[ 20  30  40]
  [  9   9   9]]]
```

```
x = np.array([[[1,2,3],[20,30,40]],[[5,55,555],[9,9,9]]])
```

```
print(x)
print("\n")

z = np.swapaxes(x,0,2)
print(z)
print("\n")

#is same as
m = np.swapaxes(x,0,2)
print(m)
print("\n")
```

```
[[[  1   2   3]
  [ 20  30  40]]

 [[  5  55 555]
  [  9   9   9]]]


[[[  1   5]
  [ 20   9]]

 [[  2  55]
  [ 30   9]]

 [[  3 555]
  [ 40   9]]]


[[[  1   5]
  [ 20   9]]

 [[  2  55]
  [ 30   9]]

 [[  3 555]
  [ 40   9]]]
```

```
x = np.array([[[0,1],[2,3]],[[4,5],[6,7]]])
print(x)
print("\n")

y = np.swapaxes(x,0,1)
print(y)
print("\n")

z = np.swapaxes(x,0,2)
print(z)
print("\n")
```

```
[[[0 1]
  [4 5]]

 [[2 3]
  [6 7]]]


[[[0 4]
```

```
       [2 6]]

      [[1 5]
       [3 7]]]


     [[[0 2]
       [1 3]]

      [[4 6]
       [5 7]]]


     [[[0 2]
       [1 3]]

      [[4 6]
       [5 7]]]
```

```
print(a)
print('\n')
print(np.swapaxes(a, 2, 1))
print('\n')

#is same as

print(np.swapaxes(a, 1, 2))
```

```
     [[[0 1]
       [2 3]]

      [[4 5]
       [6 7]]]


     [[[0 2]
       [1 3]]

      [[4 6]
       [5 7]]]


     [[[0 2]
       [1 3]]

      [[4 6]
       [5 7]]]
```

```
# It creates a 3 dimensional ndarray
import numpy as np
a = np.arange(16).reshape(2,2,2,2)

print(a)
print('\n')
# now swap numbers between axis 0 (along depth) and axis 2 (along width)

print(np.swapaxes(a, 0, 0))
print('\n')
print(np.swapaxes(a, 0, 1))
print('\n')
```

```
print(np.swapaxes(a, 0, 2))
print('\n')
print(np.swapaxes(a, 0, 3))
```

```
[[12 13]
  [14 15]]]]


[[[[ 0  1]
   [ 2  3]]

  [[ 4  5]
   [ 6  7]]]


 [[[ 8  9]
   [10 11]]

  [[12 13]
   [14 15]]]]


[[[[ 0  1]
   [ 2  3]]

  [[ 8  9]
   [10 11]]]


 [[[ 4  5]
   [ 6  7]]

  [[12 13]
   [14 15]]]]


[[[[ 0  1]
   [ 8  9]]

  [[ 4  5]
   [12 13]]]


 [[[ 2  3]
   [10 11]]

  [[ 6  7]
   [14 15]]]]


[[[[ 0  8]
   [ 2 10]]

  [[ 4 12]
   [ 6 14]]]


 [[[ 1  9]
   [ 3 11]]

  [[ 5 13]
   [ 7 15]]]]
```

.

.

```
import numpy as np
a = np.array([[[1,2],[3,4]],[[5,6],[7,8]]])
print(a)
print("\n")

print(np.swapaxes(a,0,1))
print("\n")

print(np.swapaxes(a,1,2))
```

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

 [[5 6]
  [7 8]]]


[[[1 2]
  [5 6]]

 [[3 4]
  [7 8]]]


[[[1 3]
  [2 4]]

 [[5 7]
  [6 8]]]
```

```
 # It creates a 3 dimensional ndarray
import numpy as np
a = np.arange(8).reshape(2,2,2)
print('The original array:' )
print(a)
print('\n')

# now swap numbers between axis 0 (along depth) and axis 2 (along width)
print('The array after applying the swapaxes function:')
print(np.swapaxes(a, 0, 1))
```

```
The original array:
[[[0 1]
  [2 3]]

 [[4 5]
  [6 7]]]


The array after applying the swapaxes function:
[[[0 1]
  [4 5]]

 [[2 3]
  [6 7]]]
```

```
x = np.array([[[0,1],[2,3]],[[4,5],[6,7]]])
print(x)
print("\n")

y= np.swapaxes(x,0,2)
print(y)
```

```
[[[0 1]
  [2 3]]

 [[4 5]
  [6 7]]]


[[[0 4]
  [2 6]]

 [[1 5]
  [3 7]]]
```

```
x = np.array([[[0,1],[2,3]],[[4,5],[6,7]]])
print(x)
print("\n")

y = np.swapaxes(x,0,1)
print(y)
print("\n")

z = np.swapaxes(x,0,2)
print(z)
print("\n")
```

```
[[[0 1]
  [2 3]]

 [[4 5]
  [6 7]]]


[[[0 1]
  [4 5]]

 [[2 3]
  [6 7]]]


[[[0 4]
  [2 6]]

 [[1 5]
  [3 7]]]
```

✓ 0s completed at 14:53