The Learning Hub

Numpy

Education and Training Solutions 2023

The Learning Hub

What is NumPy?

What is NumPy?

- NumPy comes from Numerical Python.
- NumPy is a Python library.
- NumPy is used to work with arrays.

The Learning Hub

Creating Arrays

Creating Arrays

- Create a NumPy ndarray Object.

```
1  import numpy as np
2  array = np.array([5, 2, 39, 6])
3  print(type(array))
```

```
<class 'numpy.ndarray'>
```

- 0-Dimension Array.

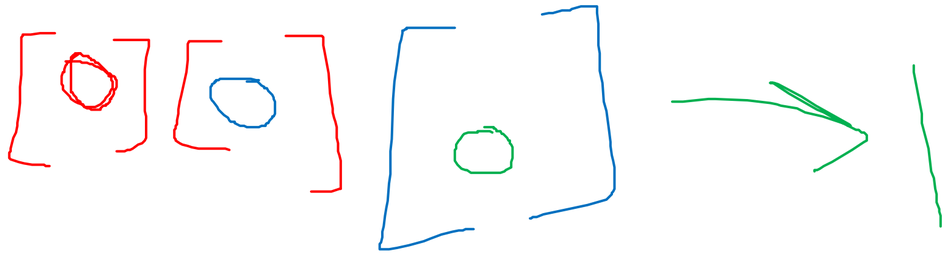
```
1  import numpy as np
2  array2 = np.array(42)
3
4  print(array2)
```

```
42
```

```
1  import numpy as np
2  array3 = np.array([[[1, 2, 3], [4, 5, 6]], [[1, 2, 3], [4, 5, 6]]])
3  print(array3)
```

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

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



- Check Number of Dimensions using ndim..

```
1  import numpy as np
2  array3 = np.array([[[1, 2, 3], [4, 5, 6]], [[1, 2, 3], [4, 5, 6]]])
3  print(array3.ndim)
```

```
3
```

The Learning Hub

Indexing & Slicing

Indexing & Slicing

- Access Array Elements.

```
1 import numpy as np
2 array = np.array([5,2,39,6])
3 print(array[0])
```

```
5
```

- Access 2-Dimension Array.

```
1 import numpy as np
2 array2 = np.array([[5,2,39,6], [60,71,81,93,12]])
3 print('num: ', array2[1][0])
```

```
num:  60
```

Indexing & Slicing

- Access 3-Dimension Array.

```
#import numpy as np
array3 = np.array([[[1,2,3], [4,5,6]], [[1,2,3], [4,5,6]]])
print(array3[0, 1, 2])
```

```
6
```

- Negative Indexing.

```
import numpy as np
array2 = np.array([[5,2,39,6], [60,71,81,93,12]])
print(array2[1][-1])
```

```
12
```

Indexing & Slicing

- Access 3-Dimension Array.

```
#import numpy as np
array3 = np.array([[[1,2,3], [4,5,6]], [[1,2,3], [4,5,6]]])
print(array3[0, 1, 2])
```

```
6
```

- Negative Indexing.

```
import numpy as np
array2 = np.array([[5,2,39,6], [60,71,81,93,12]])
print(array2[1][-1])
```

```
12
```

Indexing & Slicing

- Slicing Array [start : end : step]

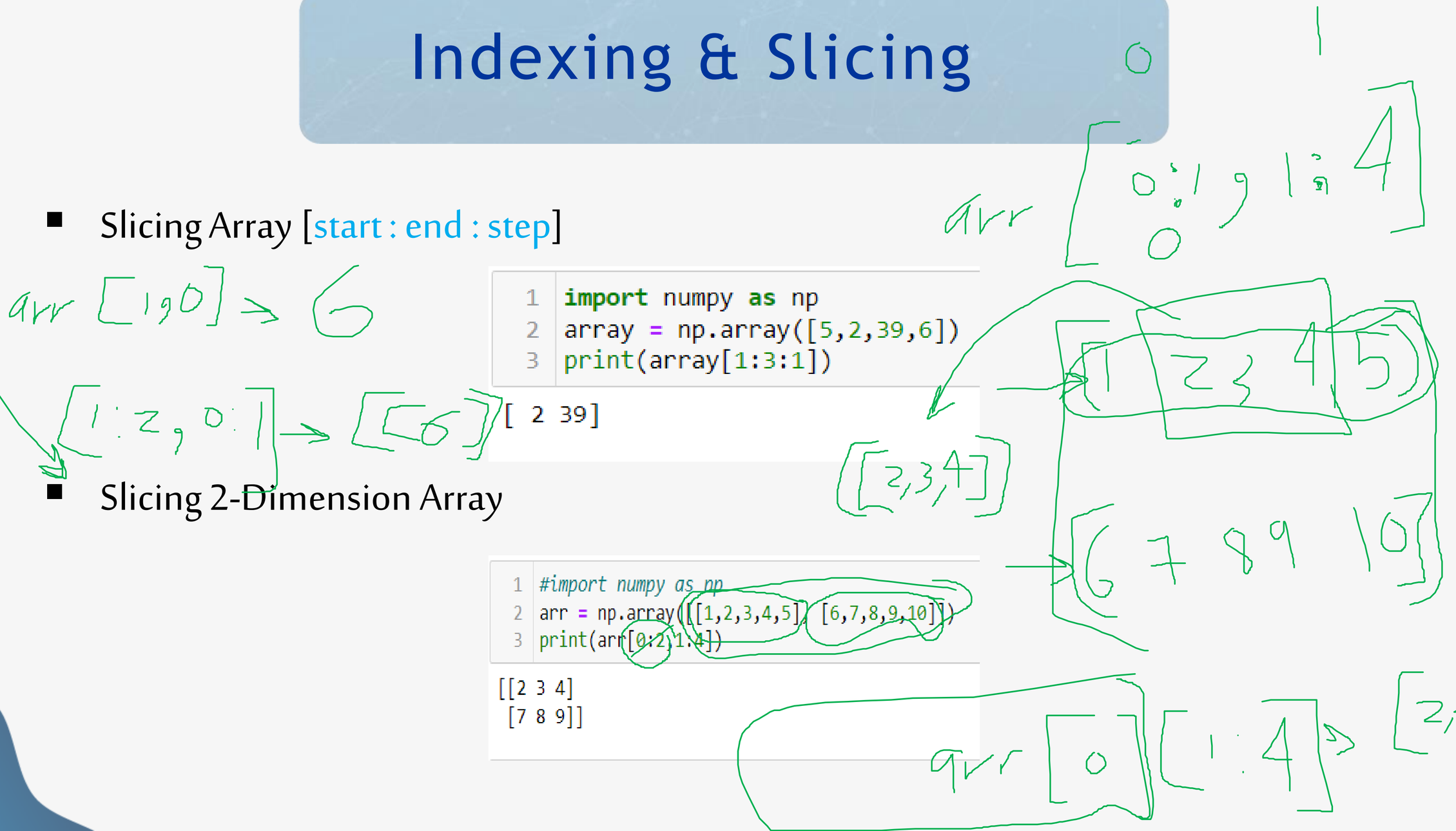
```
1  import numpy as np
2  array = np.array([5,2,39,6])
3  print(array[1:3:1])
```

```
[ 2 39]
```

- Slicing 2-Dimension Array

```
1  #import numpy as np
2  arr = np.array([[1,2,3,4,5], [6,7,8,9,10]])
3  print(arr[0:2,1:4])
```

```
[[2 3 4]
 [7 8 9]]
```



The Learning Hub

Data Types

Data Types

- Data Types in NumPy.

- **i** integer 1,2,3,4,5
- **b** Boolean True,False
- **f** float 1.1,1.2,1.3,1.4,1.5
- **s** string “abcdefg"

Data Types

- Checking the Data type.

```
1 import numpy as np
2 array2 = np.array([5, 2, 39, 6])
3 print(array2.dtype)
```

```
int32
```

```
1 import numpy as np
2 arr = np.array([1.1, 1.2, 1.3])
3 print(arr.dtype)
```

```
float64
```

Data Types

- Converting to another Data Type.

```
1  import numpy as np
2  arr = np.array([1.1,2.1,3.1])
3  newarr = arr.astype(int)
4  print(newarr)
5  print(newarr.dtype)
```

```
[1 2 3]
int32
```

- Array Copy with same Data Type.

```
1  import numpy as np
2  array = np.array([7, 22, 45, 9, 56])
3  coparr = array.copy()
4  array[0] = 11
5  print(array)
6  print(coparr)
```

```
[11 22 45  9 56]
[ 7 22 45  9 56]
```

Data Types

- Shape of an Array.

```
1  import numpy as np
2  sarr = np.array([[1, 2, 38, 41],[50, 68, 7, 8]])
3  print(sarr.shape)
```

```
(2, 4)
```

The Learning Hub

Arrays Reshape

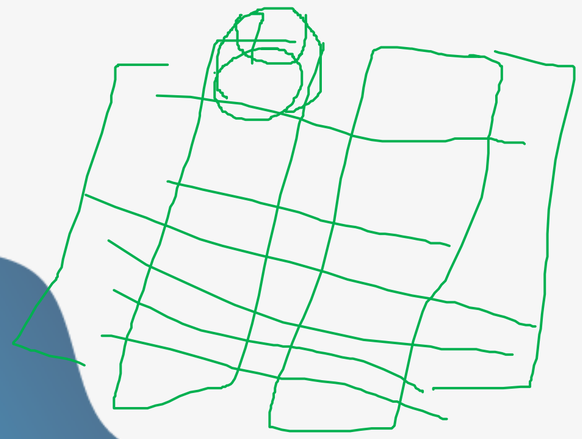
Arrays Reshape

- Reshape from 1-Dimension to 2-Dimension.

```
rarr = np.array([50, 68, 7, 8, 5, 7, 8, 87, 99, 72, 63, 12])
nearr = rarr.reshape(4,3)
nearr
```

```
array([[50, 68,  7],
       [ 8,  5,  7],
       [ 8, 87, 99],
       [72, 63, 12]])
```

- Unknown Dimension



```
#Pass -1 as the value, NumPy will calculate the correct dimension.
#import numpy as np
array = np.array([10, 2, 39, 4, 5, 6, 7, 81])
narr = array.reshape(2,2,-1)
narr
```

```
array([[[10,  2],
        [39,  4]],

       [[ 5,  6],
        [ 7, 81]]])
```

Arrays Reshape

- **Flattening the arrays.**

```
1  import numpy as np
2  arr1 = np.array([[1, 2, 3], [4, 5, 6]])
3  flaarr = arr1.reshape(-1)
4  flaarr
```

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

The Learning Hub

Arrays Join

Arrays Join

- Joining Arrays using concatenate() function.

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

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

```
1  import numpy as np
2  arr1 = np.array([[1, 2], [3, 4]])
3  arr2 = np.array([[5, 6], [7, 8]])
4  arr = np.concatenate((arr1, arr2), axis=0)
5  print(arr)
```

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

Arrays Join

- **Joining Arrays Using Stack Function.**

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

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

Arrays Join

- Joining Arrays Using hstack Function(stack along rows).

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

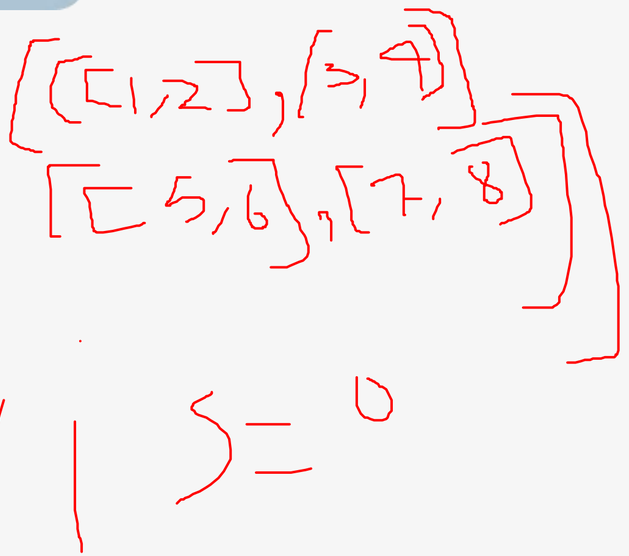
```
[1 2 3 4 5 6]
```

Arrays Join

- **Joining Arrays Using vstack Function (stack along columns).**

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

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



Arrays Join

- Joining Arrays Using dstack Function (stack along height).

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

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

Arrays Search

- **To search an array, use the where() method.**

➢ Find the indexes where the value is 3

```
1  import numpy as np
2  arr = np.array([1, 2, 3, 4, 5, 3, 3])
3  x = np.where(arr >= 3)
4  print(x)
```

```
(array([2, 5, 6], dtype=int64),)
```

Arrays Search

- **To search an array, use the where() method.**

➢ Find the indexes where the values are even

```
1  import numpy as np
2  arr = np.array([1, 2, 3, 4, 5, 3, 3])
3  x = np.where(arr%2 == 0)
4  print(x)
```

```
(array([1, 3], dtype=int64),)
```

References

www.w3schools.com. (n.d.). W3Schools Free Online Web Tutorials.

[online] Available at: http://W3schools.com. [Accessed 29 Sep. 2022].

THANK YOU