NSSA 220
Task Automation with Interpreted Languages

NumPy

Instructor: Dr. Fahed Jubair

RIT DUBAI

NumPy

- NumPy is a popular python library for working with arrays
- The array object in NumPy is called ndarray
- NumPy arrays are several times faster than built-in Python lists
- In addition to arrays, NumPy offers many mathematical functions for numerical analysis and linear algebra
- Slides Reference: https://www.w3schools.com/python/default.asp

Installing NumPy

- First, we need to install PIP: python package manager
- Assuming python3 is already installed, to install PIP, execute the following command (skip if already installed):

  sudo apt install python3-pip

- To install numpy, execute the command:

  pip3 install numpy

Creating An Array

```
import numpy as np

arr = np.array([1, 2, 3, 4, 5])

print(arr)

print(type(arr))
```

Numpy is usually imported with the alias np

- array function is used to create an array
- The input can be a list, or a tuple

```
// output

[1 2 3 4 5]
<class 'numpy.ndarray'>
```

Multi-Dimensional Arrays

```
import numpy as np

a = np.array(42)
b = np.array([1, 2, 3, 4, 5])
c = np.array([[1, 2, 3], [4, 5, 6]])
d = np.array([[[1, 2, 3], [4, 5, 6]], [[1, 2, 3], [4, 5, 6]]])

print(a.ndim)     # 0-D array
print(b.ndim)     # 1-D array
print(c.ndim)     # 2-D array
print(d.ndim)     # 3-D array
```

```
// output
0
1
2
3
```

Array Shape

```
import numpy as np

a = np.array(42)
b = np.array([1, 2, 3, 4, 5])
c = np.array([[1, 2, 3], [4, 5, 6]])
d = np.array([[[1, 2, 3], [4, 5, 6]], [[1, 2, 3], [4, 5, 6]]])

print(a.shape)
print(b.shape)
print(c.shape)
print(d.shape)
```

```
// output
()
(5,)
(2, 3)
(2, 2, 3)
```

Array Indexing

```
import numpy as np

a = np.array(42)
b = np.array([1, 2, 3, 4, 5])
c = np.array([[1, 2, 3], [4, 5, 6]])
d = np.array([[[1, 2, 3], [4, 5, 6]], [[1, 2, 3], [4, 5, 6]]])

print(a)              # prints 42
print(b[0])           # prints 1
print(b[2])           # prints 3
print(c[0][0])        # prints 1
print(c[1][2])        # prints 6
print(c[1])           # prints [4, 5, 6]
print(d[0][0][0])     # prints 1
print(d[1][0][2])     # prints 3
print(d[1][1])        # prints [4 5 6]
print(d[1])           # prints [[1 2 3]
                      #         [4 5 6]]
```

Array Slicing

```
import numpy as np

arr = np.array([1, 2, 3, 4, 5])

print(arr[1:3])
print(arr[2:])
print(arr[:3])
print(arr[0:-2])
print(arr[0:3:2])
```

```
// output

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

Array Data Type

```python
import numpy as np

arr1 = np.array([1, 2, 3, 4, 5])
arr2 = np.array(['apple', 'kiwi', 'orange'])
arr3 = np.array([1, 2, 3, 4, 5], dtype='U')
arr4 = np.array(['3', '5', '7'], dtype='i4')

print(arr1.dtype)
print(arr2.dtype)
print(arr3.dtype)
print(arr4.dtype)
```

```
// output
int64
<U6
<U1
int32
```

Array Reshape

```
import numpy as np

arr = np.array([1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12])

newarr = arr.reshape(4, 3)

print(newarr)
```

```
// output

[[ 1  2  3]
 [ 4  5  6]
 [ 7  8  9]
 [10 11 12]]
```

Array Flattening

```
import numpy as np

arr = np.array([[1, 2, 3], [4, 5, 6]])

newarr = arr.reshape(-1)

print(newarr)
```

```
// output

[1 2 3 4 5 6]
```

Array Iterating

```
import numpy as np

arr = np.array([[1, 2, 3], [4, 5, 6]])

for x in arr:
  print(x)

for x in arr:
  for y in x:
    print(y)
```

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

```
1
2
3
4
5
6
```

Array Concatenation

```
import numpy as np

a = np.array([1, 2, 3, 4, 5, 6])

b = np.array([7, 8, 9])

c = np.concatenate((a, b))

print(c)
print(type(c))
```

```
// output

[1 2 3 4 5 6 7 8 9]
<class 'numpy.ndarray'>
```

Array Filtering

```
import numpy as np

arr = np.array([41, 42, 43, 44])

x = [True, False, True, False]

newarr = arr[x]

print(newarr)
```

```
// output

[41 43]
```

Array Filtering (Another Example)

```
import numpy as np

arr = np.array([10, 4, 3, 12, 5, 9])

a = arr[arr > 5]
print(a)                                # prints [10 12 9]

b = arr[arr % 2 == 0]
print(b)                                # prints [10 4 12]

c = arr[(arr > 4) & (arr < 10)]
print(c)                                # prints [5 9]
```

NumPy Arithmetic

```
import numpy as np

arr1 = np.array([15, 8, 9, 10, 12, 7])
arr2 = np.array([7, 4, 4, 8, 11, 25])

a = np.add(arr1, arr2)
b = np.subtract(arr1, arr2)
c = np.multiply(arr1, arr2)
d = np.divide(arr1, arr2)
e = np.round(d,2)
f = np.sum([arr1,arr2])

print(a) # prints [22 12 13 18 23 32]
print(b) # prints [8 4 5 2 1 -18]
print(c) # prints [105 32 36 80 132 175]
print(d) # prints [2.14285714 2. 2.25 1.25 1.09090909 0.28 ]
print(e) # prints [2.14 2. 2.25 1.25 1.09 0.28]
print(f) # prints 120
```

Exercise 1

- NumPy has a random number generation module, called `random`
- Using NumPy `random` module, write a Python program that creates the following:
  - A 1D array that contains 50 random integers between 0 and 99
  - A 2D array with size = (10,20) that contains random real numbers between 0 and 99
- Print both arrays

Exercise 2

- Let us day you have the following Python code

```
import numpy as np

arr1 = np.array([15, 8, 9, 10, 12, 7])
arr2 = np.array([7, 4, 4, 8, 11, 25])
```

- Complete the above program by adding code lines that do the following:
  - Compute and print the dot product between arr1 and arr2
  - Create and print a 2D array that has arr1 as the first row, and has arr2 as the second row