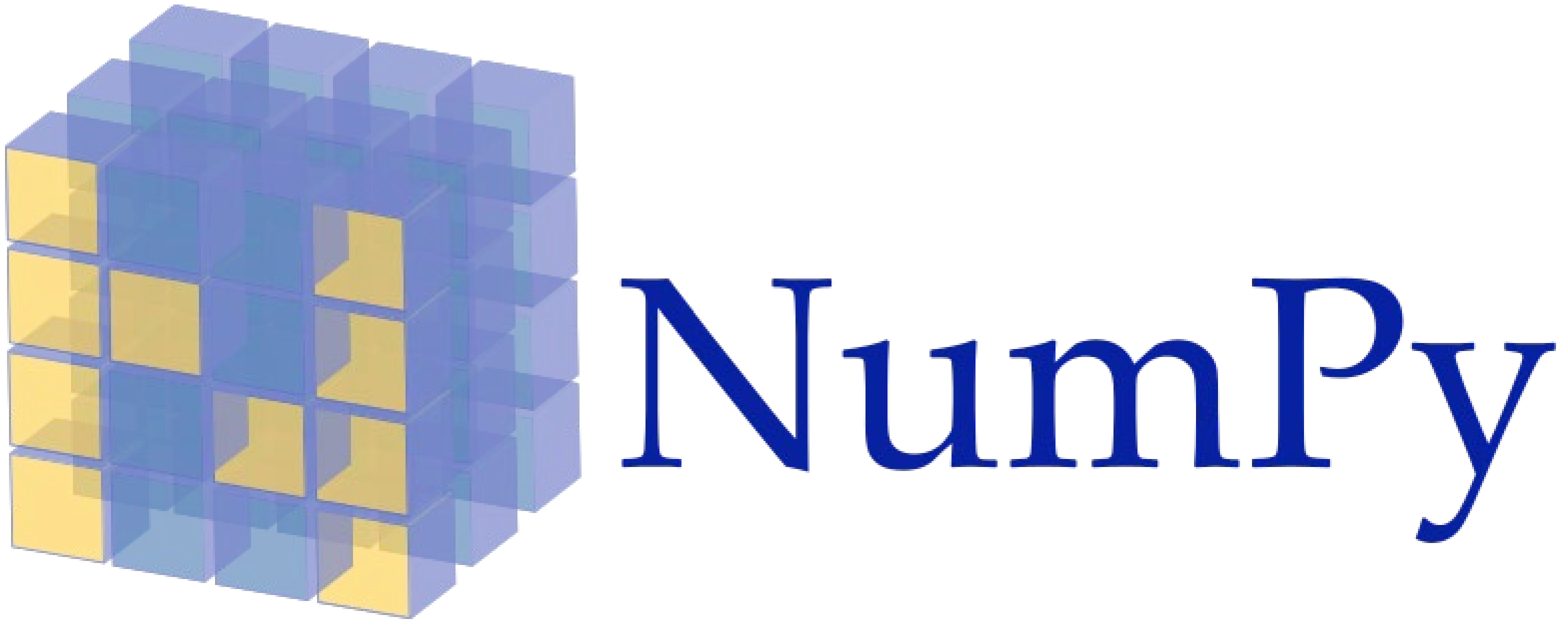
Basic Programming

Lesson 11

Numpy

Numpy

NumPy is the fundamental package for array and matrix computing with Python



Install Numpy library

Install via pip:

- pip install numpy

Import Numpy

```
import numpy as np
```

Initial array

1D:

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

2D:

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

3D:

```
arr = np.array([[[2, 4, 0, 6], [4, 7, 5, 6]],
               [[0, 3, 2, 1], [9, 4, 5, 6]],
               [[5, 8, 6, 4], [1, 4, 6, 8]]]), dtype=int)
```

Initial array with built-in functions

zeros:

```
arr = np.zeros((3, 4), dtype=int)
```

ones:

```
arr = np.ones((2, 3, 4), dtype=int)
```

arange:

```
arr = np.arange(1, 7, 2)
```

full:

```
arr = np.full((2, 3), 5)
```

eye:

```
arr = np.eye(4, dtype=int)
```

random:

```
arr = np.random.random((2, 3))
```

Array information

dtype:

```
print(arr.dtype)
```

shape:

```
print(arr.shape)
```

size:

```
print(arr.size)
```

ndim:

```
print(arr.ndim)
```

Array indexing

Slicing:

```
arr = np.array([[1, 2, 3, 4],
               [5, 6, 7, 8],
               [9, 10, 11, 12]])
print(arr[:2, 1:3])
print(arr[0, 1])
print(arr[1, :])
print(arr[1:2, :])
print(arr[:, 1])
print(arr[:, 1:2])
```

Array indexing

Integer array indexing:

```
arr = np.array([[1, 2, 3, 4],
               [5, 6, 7, 8],
               [9, 10, 11, 12]])
print(arr[[0, 1, 2], [0, 1, 0]])
print(arr[[0, 0], [1, 1]])
```

Array indexing

Boolean array indexing:

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

Array Math

Add:

```
x = np.array([[1, 2], [3, 4]], dtype=np.float64)
y = np.array([[5, 6], [7, 8]], dtype=np.float64)
print(x + y)
print(np.add(x, y))
```

Subtract:

```
x = np.array([[1, 2], [3, 4]], dtype=np.float64)
y = np.array([[5, 6], [7, 8]], dtype=np.float64)
print(x - y)
print(np.subtract(x, y))
```

Array Math

Multiply:

```
x = np.array([[1, 2], [3, 4]], dtype=np.float64)
y = np.array([[5, 6], [7, 8]], dtype=np.float64)
print(x * y)
print(np.multiply(x, y))
```

Divide:

```
x = np.array([[1, 2], [3, 4]], dtype=np.float64)
y = np.array([[5, 6], [7, 8]], dtype=np.float64)
print(x / y)
print(np.divide(x, y))
```

Array Math

Square:

```
x = np.array([[1, 2], [3, 4]], dtype=np.float64)
y = np.array([[5, 6], [7, 8]], dtype=np.float64)
print(np.sqrt(x))
```

Dot:

```
x = np.array([[1, 2], [3, 4]], dtype=np.float64)
y = np.array([[5, 6], [7, 8]], dtype=np.float64)
print(x.dot(y))
print(np.dot(x, y))
```

Array Math

Sum:

```
x = np.array([[1, 2], [3, 4]], dtype=np.float64)
y = np.array([[5, 6], [7, 8]], dtype=np.float64)
print(x.sum())
print(np.sum(x))
print(x.sum(axis=0))
print(np.sum(x, axis=0))
print(x.sum(axis=1))
print(np.sum(x, axis=1))
```

Array Math

Max:

```
x = np.array([[1, 2], [3, 4]], dtype=np.float64)
print(x.max())
print(np.max(x))
print(x.max(axis=0))
print(np.max(x, axis=0))
print(x.max(axis=1))
print(np.max(x, axis=1))
```

Array Math

Min:

```
x = np.array([[1, 2], [3, 4]], dtype=np.float64)
print(x.min())
print(np.min(x))
print(x.min(axis=0))
print(np.min(x, axis=0))
print(x.min(axis=1))
print(np.min(x, axis=1))
```

Array Math

Mean:

```
x = np.array([[1, 2], [3, 4]], dtype=np.float64)
print(x.mean())
print(np.mean(x))
print(x.mean(axis=0))
print(np.mean(x, axis=0))
print(x.mean(axis=1))
print(np.mean(x, axis=1))
```

Array Math

Median:

```
x = np.array([[1, 2], [3, 4]], dtype=np.float64)
print(np.median(x))
print(np.median(x, axis=0))
print(np.median(x, axis=1))
```

Transposition:

```
x = np.array([1, 2, 3, 4], dtype=np.float64)
y = np.array([[5, 6], [7, 8]], dtype=np.float64)
print(x.T)
print(y.T)
```

Array Math

Absolute:

```
x = np.array([[10, 2, 50],
             [-3, 0, 9],
             [11, 6, -20]], dtype=np.int64)
print(np.abs(x))
print(np.absolute(x))
```

Other functions

Concatenation:

```
x = np.array([[1, 2], [3, 4]], dtype=np.float64)
y = np.array([[5, 6], [7, 8]], dtype=np.float64)
print(np.concatenate([x, y], axis=0))
print(np.concatenate([x, y], axis=1))
```

Split:

```
x = np.array([10, 2, 50, -3, 0, 9, 11, 6], dtype=np.int64)
print(np.split(x, [2, 5]))
```

Other functions

Sort:

```
x = np.array([[10, 2, 50],
              [-3, 0, 9],
              [11, 6, -20]], dtype=np.int64)
print(np.sort(x, axis=0))
print(np.sort(x, axis=1))
print(np.sort(x, axis=None))
```

Count:

```
x = np.array([10, 2, 50, -3, 0, 9, 11, 6], dtype=np.int64)
print(np.count_nonzero(x))
print(np.count_nonzero(x > 3))
```

Other functions

Sort:

```
x = np.array([[10, 2, 50],
              [-3, 0, 9],
              [11, 6, -20]], dtype=np.int64)
print(np.sort(x, axis=0))
print(np.sort(x, axis=1))
print(np.sort(x, axis=None))
```

Sin, Cosin, Logarit, Reduce, Variance, Standard deviation

Bitwise: and, or, xor, not

Pandas

Pandas

Powerful data structures for data analysis, time series, and statistics

Install Pandas library

Install via pip:

- pip install pandas

Import Pandas

```
import pandas as pd
```

Series

Sort:

```
import pandas as pd


x = pd.Series([1232, 3234, 3250, 2222])
print(x)
print(x.values)
print(type(x.values))
```

Indexing:

```
print(x[2])
print(x[0:3])
```

Initial DataFrame

Create a series:

```
population_dict = {
    'TP.HCM': 8993,
    'Hanoi': 8053,
    'Lam Dong': 1297,
    'Quang Tri': 623
}
population = pd.Series(population_dict)
print(population)
```

Initial DataFrame

Create a series:

```
area_dict = {
    'TP.HCM': 2061,
    'Hanoi': 3359,
    'Lam Dong': 9765,
    'Quang Tri': 4746
}
area = pd.Series(area_dict)
print(area)
```

Initial DataFrame

Create a DataFrame from two dict:

```
vietnam = pd.DataFrame({'Dân số': population, 'Diện tích': area})
print(vietnam)


print(vietnam.index)
print(vietnam.columns)
```

Indexing:

```
print("Dân số TP.HCM: ", vietnam['Dân số']['TP.HCM'])
print("Dân số TP.HCM: ", vietnam['Dân số'][0])
```

Initial DataFrame

Create DataFrame from a series:

```
df = pd.DataFrame(population, columns=['Dân số'])
```

Create DataFrame from a dict:

```
data = [
    {'a': 1, 'b': 2, 'c': 'Hello'},
    {'a': 5, 'b': 3, 'c': 'Python'}
]
df = pd.DataFrame(data)
```

Create DataFrame from a numpy array 2D:

```
df = pd.DataFrame([[4324, 1242], [6788, 7334]], columns=['Pandas', 'NumPy'])
```