

Week 2 Handout

- **Working with variables:** Initialization, mathematical operations, data types, type() function, logical operations
- **String functions:** len(), string.capitalize(), string.upper, string.center(), string.rjust(), string.replace(), string.strip()
- **Containers:-**
 - ➔ **Lists** – Creating a list [a, b, c], Indexing list elements using list[index], appending to list: list.append(), popping from list: list.pop(), listing a range of values: range(), slicing list: list[index1 : index2], looping over lists, enumerate(), list comprehensions
 - ➔ **Dictionaries** - Creating a dictionary {'a' : 1, 'b' : 2}, retrieving values using keys, retrieving using dict.get(), iterating using dict.items(), dictionary comprehensions
 - ➔ **Sets** – Create a set {1, 4, 6, 2, 7}, add to set using set.add(), remove element using set.remove(), length of set using len(), enumerate()
 - ➔ **Tuples** – Create a tuple (2, 3, 4), Indexing in a tuple using tuple[index]
- **Function:** def keyword before function name, returns a value
- **Class:** class keyword before class name, class functions can be called by creating an object
- **Numpy:-**

Importing the library: import numpy as np

Arrays: np.array(): creating an array, np.zeros(): creating an array of zeros, np.ones(): creating an array of ones, np.full(): creating an array of a particular number, np.eye(): creating an identity matrix, np.random.random(): creating an array of randomly initialized values between 0 and 1, np.arange(): range of numbers with a definite step size, array.T: transpose of the array

Array indexing:

- ➔ array[x] or array [x, y] for accessing individual data items, array[starting_index : ending_index+1] for accessing a range in 1 dimensional data
- ➔ array[starting_index1 : ending_index1 + 1, starting_index2 : ending_index2 + 1] for accessing a range in 2 dimensional data and so on.
- ➔ array.shape for getting the dimensions of the numpy array

Numpy math:-

- ➔ np.add(), np.subtract(), np.multiply(), np.divide(), np.sqrt(), np.dot(), np.sum(), np.sin(), np.cos()

Broadcasting:-

- ➔ np.empty_like(), np.tile(), np.reshape()

- **Matplotlib:-**

import matplotlib.pyplot as plt

plt.plot(), plt.title(), plt.legend(), plt.xlabel(), plt.ylabel(), plt.subplot(), plt.show()