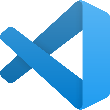
**CS506 Programming for Computing**

**HOP06B – Getting Started with NumPy**

06/06/2020 Created by Apiwat Chuaphan

11/08/2020 Revised by Kim Nguyen

Center for Information Assurance (CIAE) @City University of Seattle (CityU)

  A close up of a sign

Description automatically generated

**Before You Start**

* The directory path shown in screenshots may be different from yours.
* Some steps might not be explained in the tutorial.  If you are not sure what to do:
  + Consult the resources listed below.
  + If you cannot solve the problem after a few tries, ask a TA for help.

**Learning Outcomes**

* Understand array object in NumPy
* Understand array manipulation using NumPy’s functions

**Resources**

* NumPy Documentation: <https://numpy.org/doc/stable/>
* Stanford University. (2020). CS231n: Convolutional Neural Networks for Visual Recognition: NumPy.

**NumPy**

Which stands for Numerical Python, is an open source and the core library for scientific computing in Python. It provides a high-performance multidimensional array object, and tools for working with these arrays.

NumPy is memory efficiency, meaning it can handle the vast amount of data more accessible than any other library. We’ll cover the following in this module:

* Datatypes
* Arrays

Before we begin, let’s install NumPy module if you have not done that. Run this command in terminal:

>>> pip install numpy

Check the version to make sure we have it installed, run:

>>> pip list | grep numpy

Alright, we have it. Let’s begin!

1. In Visual Studio Code, open the private repository generated when you accepted the HOP06 assignment (If you cannot find that repository in your machine, you might have not cloned the repo, if so, please do before proceeding).

**Datatypes**

The most object defined in NumPy is an N-dimensional array type called **ndarray**, describing the collection of items of the same type. An element in ndarray is an object of data-type object called **dtype**.

A **dtype** described in the following aspects:

* Type of the data (integer, float, Python object, etc.)
* Size of the data (how many bytes is in e.g. the integer)
* Byte order of the data (little-endian or big-endian)
* Structured data type (*e.g.*, describing an array item consisting of an integer and a float)

Open Jupyter Notebook:

1. Under module folder, create a new file called **datatypes.ipynb** and simply click on the file to open notebook.
2. Type the following to see how Numpy says about data types. Run selected cell to see result for that cell.

A screenshot of a cell phone

Description automatically generated

1. We also can define a structured data type. For example, we will define a data type named student with the 3 fields: name as string, age as integer, points as float.

A screenshot of a cell phone

Description automatically generated

1. Let’s apply to ndarray object.

A screenshot of a cell phone

Description automatically generated

As you can see here, “*student2withlongname”* has exceeded the string length, only 10 characters acceptance was defined.

**Arrays**

A numpy array is a grid of values similar to Python lists, but all of the same type, and is indexed by a tuple of nonnegative integers. The **rank** and **shape** are the number of dimensions and a tuple of integers of the size of the array, respectively.

1. Under module folder, create a new file called **arrays.ipynb** and simply click on the file to open notebook.
2. Type the following to create numpy arrays. Accessing items in array is similar to Python list.

A screenshot of a cell phone

Description automatically generated

1. Stay in same file, type to following functions from numpy to generate arrays in a different way. You will see how it can make your life easier with array.

A screenshot of a cell phone

Description automatically generated

1. **Indexing and slicing** in NumPy array is similar to Lists in Python. Try the following into a new cell.

Screen of a cell phone

Description automatically generated

1. In NumPy, you can do integer indexing which helps in selecting any arbitrary item in an array using the data from another array like the example below. Add this to new cell.

Screen of a cell phone

Description automatically generated