

Python tutorial - revisit

Python tutorial videos

Click on the below links to refresh your Python skills.

1. [Array Math](#)
2. [NumPy Array](#)
3. [Functions](#)
4. [Data processing using NumPy and Pandas](#)
5. [Introduction to object oriented programming](#)

Numpy - Introduction

► Run

PYTHON



1

We assume that you already know Python, either from doing the course ZZEN9021 Principles of Programming or something equivalent, so we won't be teaching it to you. This lesson is a refresher on the aspects of Python that are particularly important for this course.

In slide focusses on Numpy. Click on the below links to practise Numpy:

[Practise link 1](#)

[Practise link 2](#)

Now the run the below code.

► Run

PYTHON



```
1 import numpy as np
2
3 #create and print arrays
4
5 a = np.arange(6)                # 1d array
6 print(a, ' a ')
7
8 b = np.arange(12).reshape(4,3)  # 2d array
9 print(b, ' b ')
10
11 c = np.arange(24).reshape(2,3,4) # 3d array
12 print(c, ' c ')
13
```

► Run

PYTHON



```
1 import numpy as np
2
3 a = np.zeros((2,2))    # Create an array of all zeros
4 print(a)              # Prints "[[ 0.  0.]
5                        #           [ 0.  0.]]"
6
7 b = np.ones((1,2))    # Create an array of all ones
8 print(b)              # Prints "[[ 1.  1.]]"
9
10 c = np.full((2,2), 7) # Create a constant array
11 print(c)              # Prints "[[ 7.  7.]
12                       #           [ 7.  7.]]"
13
14 d = np.eye(2)         # Create a 2x2 identity matrix
```

► Run

PYTHON



```
1 import numpy as np
2
3 # Create the following rank 2 array with shape (3, 4)
4 # [[ 1  2  3  4]
5 #   [ 5  6  7  8]
6 #   [ 9 10 11 12]]
7 a = np.array([[1,2,3,4], [5,6,7,8], [9,10,11,12]])
8
9 # Use slicing to pull out the subarray consisting of the first 2 rows
10 # and columns 1 and 2; b is the following array of shape (2, 2):
11 # [[2 3]
12 #   [6 7]]
13 b = a[:2, 1:3]
14
```

```
1 import numpy as np
2
3 # Create the following rank 2 array with shape (3, 4)
4 # [[ 1  2  3  4]
5 #   [ 5  6  7  8]
6 #   [ 9 10 11 12]]
7 a = np.array([[1,2,3,4], [5,6,7,8], [9,10,11,12]])
8
9 # Two ways of accessing the data in the middle row of the array.
10 # Mixing integer indexing with slices yields an array of lower rank,
11 # while using only slices yields an array of the same rank as the
12 # original array:
13 row_r1 = a[1, :]      # Rank 1 view of the second row of a
14 row_r2 = a[1:2, :]    # Rank 2 view of the second row of a
```

Numpy Array Math

Run the below [Python code](#) on an array.

► Run

PYTHON



```
1 import numpy as np
2
3 x = np.array([[1,2],[3,4]], dtype=np.float64)
4 y = np.array([[5,6],[7,8]], dtype=np.float64)
5
6 # Elementwise sum; both produce the array
7 # [[ 6.0  8.0]
8 #  [10.0 12.0]]
9 print(x + y)
10 print(np.add(x, y))
11
12 # Elementwise difference; both produce the array
13 # [[-4.0 -4.0]
14 #  [-4.0 -4.0]]
```

Inner products

Note that unlike MATLAB, `*` is elementwise multiplication, not matrix multiplication. We instead use the `dot` function to compute inner products of vectors, to multiply a vector by a matrix, and to multiply matrices. `dot` is available both as a function in the [Numpy](#) module and as an instance method of array objects.

Run the code.

► Run

PYTHON



```
1 import numpy as np
2
3 x = np.array([[1,2],[3,4]])
4 y = np.array([[5,6],[7,8]])
5
6 v = np.array([9,10])
7 w = np.array([11, 12])
8
9 # Inner product of vectors; both produce 219
10 print(v.dot(w))
11 print(np.dot(v, w))
12
13 # Matrix / vector product; both produce the rank 1 array [29 67]
14 print(x.dot(v))
```

An error occurred.

[Try watching this video on www.youtube.com](#), or enable JavaScript if it is disabled in your browser.

An error occurred.

[Try watching this video on www.youtube.com](#), or enable JavaScript if it is disabled in your browser.

<https://cs231n.github.io/python-numpy-tutorial/>

► Run

PYTHON



```
1 #source: https://cs231n.github.io/python-numpy-tutorial/
2
3 import numpy as np
4
5 x = np.array([[1,2],[3,4]])
6
7 print(np.sum(x)) # Compute sum of all elements; prints "10"
8 print(np.sum(x, axis=0)) # Compute sum of each column; prints "[4 6]"
9 print(np.sum(x, axis=1)) # Compute sum of each row; prints "[3 7]"
10
11 print(' -----')
12
13 x = np.array([[1,2], [3,4]])
14 print(x) # Prints "[[1 2]
```

Basic functions

Basic functions to check conditions (if and else)

Run the python program by activating the terminal and type and enter: `python function.py`

timepass exercise:

update function, `speed_function_super()` so that considers different fine for speed of 100 - 119, 120 - 139 and 140 - 159, based on `speed_function_advanced()`. You can create fine based on your own rules, similar to 80 -100 limit based on `speed_function_advanced()`.

Functions with Jupyter Notebook

Simple function example for speed calculator

Numpy array examples

► Run

PYTHON



```
1
2 import numpy as np
3
4 import random
5
6 def numpy_lists():
7
8     magic = np.random.rand(3,4)
9     magic_three = np.random.rand(3,4,2)
10    # homework, write a function for summing 3D magic
11    magic_one = np.random.rand(10)
12
13    print(magic, ' is magic ')
14
```

► Run

PYTHON



```
1
2 import numpy as np
3 import random
4 import sys # for end or endlne in nested food loops, python 3
5
6
7 def nested_loops():
8
9     print(' nested loops function')
10    https://www.ict.social/python/basics/multidimensional-lists-in-python
11
12    length = 3
13    width = 4
14    height = 3
```

Sum Matrices

Sum elements in a 2D Matrix example given using nested loops (to demonstrate nested loops)

Challenge: implement a function for 3D case using nested loops

Nested loops: 2D and 3D

Example of operations with nested loops

Advanced Numpy methods

This code slide does not have a description

Load and save files

This code slide does not have a description

Mandlebrot example

https://en.wikipedia.org/wiki/Mandelbrot_set

source: #<https://www.codingame.com/playgrounds/2358/how-to-plot-the-mandelbrot-set/mandelbrot-set>

An error occurred.

Try watching this video on www.youtube.com, or enable JavaScript if it is disabled in your browser.

Pandas data processing

► Run

PYTHON



```
1 import numpy as np
2 import pandas as pd
3
4 s = pd.Series([1, 3, 5, np.nan, 6, 8])
5 print(s, ' s series')
6
7 dates = pd.date_range('20130101', periods=6)
8 print(dates, ' dates')
9
```

source: https://pandas.pydata.org/pandas-docs/stable/getting_started/10min.html

► Run

PYTHON



```
1 import numpy as np
2 import pandas as pd
3
4 data = np.array([[ '', 'Col1', 'Col2'], ['Row1', 1, 2], ['Row2', 3, 4]])
5
6 print(data)
7
8 print(pd.DataFrame(data=data[1:,1:], index=data[1:,0], columns=data[0,1:
9
10
```

► Run

PYTHON



```
1 import numpy as np
2 import pandas as pd
3
4 df = pd.DataFrame(np.array([[1, 2, 3], [4, 5, 6]]))
5
6 # Using `iloc[]`
7 print(df.iloc[0][0]) https://pandas.pydata.org/pandas-docs/stable/referenc
8 https://www.shanelynn.ie/select-pandas-dataframe-rows-and-columns-using
9
10 # Using `iat[]`
11 print(df.iat[0,0])
```

source: <https://www.datacamp.com/community/tutorials/pandas-tutorial-dataframe-python>

tutorial: <https://realpython.com/pandas-python-explore-dataset/>

<https://www.geeksforgeeks.org/python-pandas-dataframe/>

<https://www.geeksforgeeks.org/python-pandas-working-with-text-data/>

An error occurred.

Try watching this video on www.youtube.com, or enable JavaScript if it is disabled in your browser.

Data cleaning with Pandas

External tutorials here:

1. <https://www.kaggle.com/regivm/data-cleaning-and-eda-tutorial>
2. <https://realpython.com/python-data-cleaning-numpy-pandas/>

An error occurred.

Try watching this video on www.youtube.com, or enable JavaScript if it is disabled in your browser.

An error occurred.

Try watching this video on www.youtube.com, or enable JavaScript if it is disabled in your browser.

notebook for above video: <https://github.com/KeithGalli/pandas>

Object oriented programming

Classes

https://www.w3schools.com/python/python_classes.asp

▶ Run

PYTHON



```
1 # source: https://www.csdojo.io/class
2
3 class Robot:
4     def __init__(self, name, color, weight):
5         self.name = name
6         self.color = color
7         self.weight = weight
8
9     def introduce_self(self):
10        print("My name is " + self.name)
11
12 # r1 = Robot()
13 # r1.name = "Tom"
14 # r1.color = "red"
```

An error occurred.

[Try watching this video on www.youtube.com](https://www.youtube.com), or enable JavaScript if it is disabled in your browser.

Inheritance

https://www.w3schools.com/python/python_inheritance.asp

► Run

PYTHON

```
1
2 # source: https://www.csdojo.io/class2
3
4 class Robot:
5     def __init__(self, n, c, w):
6         self.name = n
7         self.color = c
8         self.weight = w
9
10    def introduce_self(self):
11        print("My name is " + self.name)
12
13
14 class Person:
```

An error occurred.

[Try watching this video on www.youtube.com](#), or enable JavaScript if it is disabled in your browser.

some other examples:

<https://www.programiz.com/python-programming/object-oriented-programming>

<https://python.swaroopch.com/oop.html>

Exercise 1: Body mass index

Develop a Body mass index program using either Python or R.

Use the information from here: <http://healthyweight.health.gov.au/wps/portal/Home/get-started/are-you-a-healthy-weight/bmi/>

Ensure that you use functions to do the following:

1. Ask information from user including name and age and calculate BMI
2. Save the BMI in a file in summary.txt
3. Read the summary.txt and compute the mean and std of BMI for the users.
4. Try out any visualization of your summary.txt, using bar plots etc.

Exercise 2: BMI using Classes

Using object oriented programming, implement BMI program in Exercise 1 using Classs. You can use either Python or R for this exercise.

References

1. <https://www.datacamp.com/community/tutorials/r-objects-and-classes>
2. <https://realpython.com/python3-object-oriented-programming/>

Exercise 3: Visualise data

See the Iris dataset attached in the workspace below. More information about the dataset is here: https://en.wikipedia.org/wiki/Iris_flower_data_set and the original source is here: <https://archive.ics.uci.edu/ml/datasets/iris>

Use either functions or methods in a class to do the following:

1. Use simple find and replace to convert the class labels to 1, 2, and 3 in the dataset.
2. Read the data and report mean and standard deviation for each column in the features (4 features)
3. Report the class distribution (i. e number of instances for each class)
4. Show histogram for each feature. Note you need to use a single function/method that outputs the histogram with a given filename. eg. feature1.png which is given as a parameter to the function. A for loop should be used to call the function/method
5. Split data into a train and test test. Use 60 percent data in the training and test set which is assigned i. randomly ii. assigned by first 60 percent as train and rest as test.
6. Use previous functions to report the mean and standard deviation of the train and test set and class distribution and also the histograms for each feature.
7. Create another subset of the train and test set where only 1 feature selected by the user makes the dataset with the class.
8. Create a subset of the dataset where you consider only instances that feature class 1 or 2, so that you treat this problem as a binary classification problem later, i.e save it as binary_istrain.txt and binary_istest.txt. Carry out the stats and visuals in Step 6 for this dataset.
9. Can you normalise the input features between [0 and 1] ? Write code that can do so and save normalised versions.