

**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT HƯNG YÊN**  
**KHOA CÔNG NGHỆ THÔNG TIN**

----- oOo -----



# **BÀI TẬP THỰC HÀNH**

# **LẬP TRÌNH PYTHON NÂNG CAO**

**TRÌNH ĐỘ: ĐẠI HỌC CHÍNH QUY**

**NGÀNH ĐÀO TẠO: KHOA HỌC MÁY TÍNH**

**Hưng Yên – Tháng 11 năm 2021**

## BÀI THỰC HÀNH SỐ 1: XỬ LÝ DỮ LIỆU MẢNG VỚI NUMPY

### A. MỤC TIÊU BÀI THỰC HÀNH

Sau bài thực hành này sinh viên có thể:

- Hiểu được cách tạo mảng dữ liệu trong NumPy
- Biết cách truy xuất các phần tử của mảng dữ liệu trong NumPy
- Sử dụng thành thạo các hàm tính toán trong NumPy
- Vận dụng các kỹ thuật xử lý mảng trong NumPy để giải quyết các bài toán thực tế trong khoa học dữ liệu

### B. ĐIỀU KIỆN THỰC HÀNH

Với đặc thù của môn Lập trình Python nâng cao, mục này sẽ liệt kê một số công cụ sử dụng để làm bài thực hành. Trong bài thực hành này, sinh viên cần kiểm tra và chắc chắn các phần mềm sau trên máy tính còn hoạt động tốt:

1. Anacoda phiên bản 3.0 trở lên
2. Jupyter Notebooks

### C. TÀI NGUYÊN THAM CHIẾU

Để hoàn thành tốt bài thực hành này sinh viên nên tham khảo các tài nguyên sau:

STT	Tên tài nguyên	Mô tả tài nguyên
1	Practice 01.doc	Tài liệu hướng dẫn thực hành bài số 1
2	Lesson 01 – Working with Numpy.pdf	Slide bài giảng về xử lý dữ liệu mảng với NumPy

### D. YÊU CẦU BÀI THỰC HÀNH

**Bài 1.** Thực hiện các thao tác cơ bản với NumPy

- a) Tạo một mảng đa chiều **a** với nhận các giá trị: [1, 2, 3, 4, 5], [6, 7, 8, 9, 10]
- b) In ra giá trị các thuộc tính: ndim, size, shape
- c) Tạo và in ra mảng **b** bằng cách chuyển mảng **a** với thành 5 hàng và 2 cột
- d) Truy xuất mảng **a** lấy về các phần tử sau:

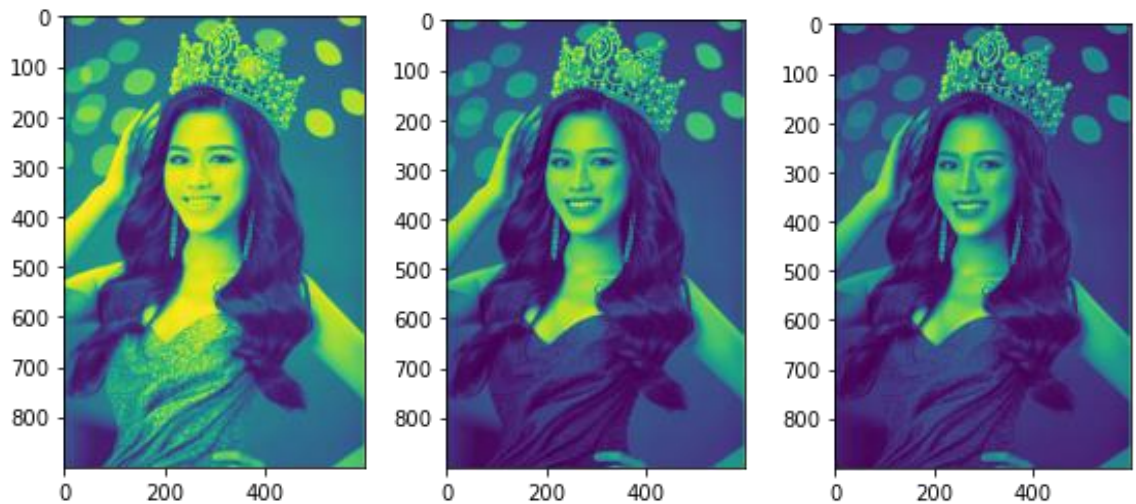
- Giá trị “8” trong mảng **a**
  - Lấy về toàn bộ các giá trị của hàng đầu tiên
  - Lấy về giá trị 3 phần tử đầu tiên của mỗi hàng trong mảng **a**
- [1, 2, 3], [6, 7, 8]

**Bài 2.** Cho một bức ảnh màu có kích thước 600x900, hãy thực hiện một số yêu cầu sau:

- Đọc dữ liệu hình ảnh vào biến **a**
- Kiểm tra kiểu dữ liệu của biến **a**
- In ra giá trị các thuộc tính của **a**: ndim, size, shape
- In ra màn hình giá trị của **a** và hiển thị ra màn hình mức ảnh thông qua **a**



- e) In ra màn hình giá trị của **a** theo các kênh màu khác nhau (**R,G,B**) và hiển thị ra màn hình mức ảnh theo từng kênh màu



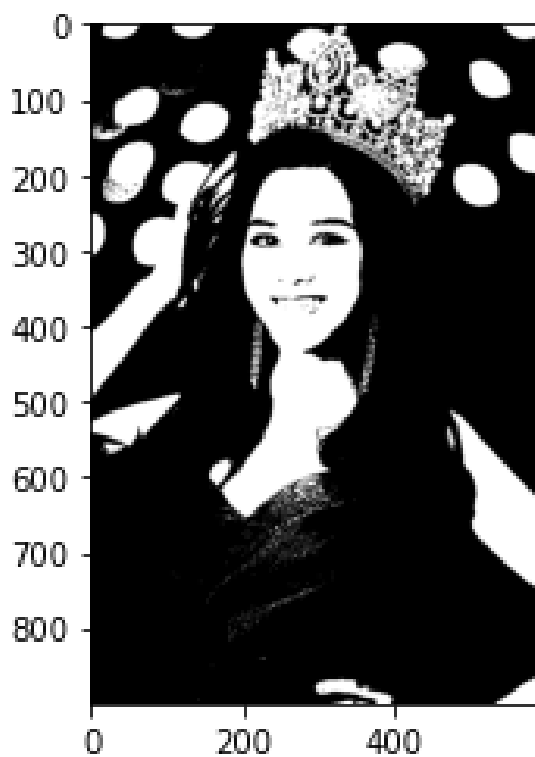
- f) In ra màn hình một phần giá trị của **a** cho phần ảnh 600x600 từ trên xuống; hiển thị phần ảnh tương ứng lên màn hình



g) Chuyển bức ảnh màu đã cho thành ảnh xám



h) Chuyển bức ảnh màu đã cho thành ảnh đen trắng



### Bài 3: Mã hóa One-hot

Trong học máy, nhiều dữ liệu hạng mục (categorical data) được chuyển về dạng số bằng cách mã hóa one-hot. Trong cách mã hóa này, một “từ điển” cần được xây dựng chứa tất cả các giá trị khả dĩ của từng dữ liệu hạng mục. Sau đó mỗi giá trị hạng mục sẽ được mã hóa bằng một vector nhị phân với toàn bộ các phần tử bằng 0 trừ một phần tử bằng 1 tương ứng với vị trí của giá trị hạng mục đó trong từ điển.

Ví dụ, trong bài toán dùng AI để nhận biết các đối tượng, nếu ta có dữ liệu một cột là "People", "Mobile", "Car" thì ta thực hiện các bước sau:

- Xây dựng từ điển. Trong trường hợp này ta có thể xây dựng từ điển là ["Car", "Mobile", "People"]
- Sau khi xây dựng được từ điển ta cần lưu lại chỉ số của từng hạng mục trong từ điển. Với từ điển như trên, chỉ số tương ứng là " Car": 0, " Mobile": 1, " People": 2.
- Cuối cùng, ta mã hóa các giá trị ban đầu như sau:

Giá trị	Mã one-hot
Car	[0, 0, 1]
Mobile	[0, 1, 0]
People	[1, 0, 0]

Bạn hãy sử dụng NumPy để chuyển mảng dữ liệu D chứa các từ sau thành mã one-hot như mô tả ở trên. Biết D = ["Car","Mobile", "Car","People", "Car", "People","Mobile"].

Kết quả như sau:

```
[[1.  0.  0.]
 [0.  1.  0.]
 [1.  0.  0.]
 [0.  0.  1.]
 [1.  0.  0.]
 [0.  0.  1.]
 [0.  1.  0.]
```

### E. HƯỚNG DẪN THỰC HIỆN

Sinh viên tạo tệp **Practice01\_HoVaTen.ipynb** trên Jupyter Notebook và thực hiện viết mã lệnh để giải quyết các bài tập thực hành.

**Bài 1:**

- Sử dụng kiến thức tạo mảng 2 chiều bằng NumPy
- Sử dụng kỹ thuật Slicing trong NumPy

**Bài 2:**

- Đọc dữ liệu hình ảnh vào biến **a**
  - Cài đặt thư viện matplotlib trong Anacoda
  - Sử dụng thư viện matplotlib để đọc ảnh và hiển thị ảnh
    - **import matplotlib.image as img**
    - **import matplotlib.pyplot as pl**
- Kiểm tra kiểu dữ liệu của biến **a**
  - Sinh viên tự làm
- In ra giá trị các thuộc tính của **a**: ndim, size, shape
  - Sinh viên tự làm
- In ra màn hình giá trị của **a** và hiển thị ra màn hình bức ảnh thông qua **a**
  - Sử dụng **print** để in
  - Sử dụng **pl.imshow(a)**
- In ra màn hình giá trị của **a** theo các kênh màu khác nhau (**R,G,B**) và hiển thị ra màn hình mức ảnh theo từng kênh màu
  - Sinh viên tự làm
- In ra màn hình một phần giá trị của **a** cho phần ảnh 600x600 từ trên xuống; hiển thị phần ảnh tương ứng lên màn hình
  - Sinh viên tự làm
- Chuyển bức ảnh màu đã cho thành ảnh xám
  - Tạo mảng dữ liệu **b** để chứa ảnh xám sử dụng công thức  **$b = 0.2989 * R + 0.5870 * G + 0.1140 * B$**

Trong đó R, G, B là mảng các giá trị điểm ảnh theo từng kênh màu (chiều)

  - Hiển thị ảnh xám: **pl.imshow(b,cmap='gray')**
- Chuyển bức ảnh màu đã cho thành ảnh đen trắng
  - Giá trị điểm ảnh trắng: 255
  - Giá trị điểm ảnh đen: 0

- Các giá trị điểm ảnh giao động trong đoạn [0-255]
- Sử dụng phương thức **where** của NumPy để thay thế các giá trị điểm ảnh

**Bài 3:**

- Sử dụng hàm **unique** trong NumPy để lấy về các từ trong từ điển (các từ duy nhất)
- Sử dụng hàm **eye** trong NumPy để chuyển dữ liệu về one-hot vector