

## BÀI ÔN TẬP KIẾN THỨC TOÁN VÀ LẬP TRÌNH

Phần 1: Các đại lượng và cấu trúc dữ liệu tương ứng trong Python (sử dụng thư viện Numpy)

Đại lượng Toán học	Mã lệnh Python
<b>Đại lượng vô hướng – tensor hạng 0</b> $x, y, z \in R$ Ví dụ: $x = 5, y = -12, z = 3.1417$	$x = 5$ $y = -12$ $z = 3.1417$
<b>Đại lượng có hướng – vector – tensor hạng 1</b> Tổng quát: $x, y \in R^n, n \in N$ Ví dụ: $x, y \in R^4$ $x = [x_0, x_1, x_2, x_3] = [2, -1, 3, 6]$ $x = [x_0; x_1; x_2; x_3] = \begin{bmatrix} 2 \\ -1 \\ 3 \\ 6 \end{bmatrix}$	Sử dụng thư viện numpy Tạo vector (thực chất là cấu trúc ndarray với $n = 1$ ) $x = \text{np.array}([2, -1, 3, 6])$  Kiểm tra số chiều $\text{print}(x.\text{ndim})$  Kiểm tra kích thước $\text{print}(x.\text{shape})$  <i>Lưu ý:</i> Với numpy, vector (tensor 1 chiều) không phân biệt là vector hàng hay vector cột. Khi thực hiện chuyển vị vector đều có kết quả như nhau.
<b>Ma trận – tensor hạng 2</b> Tổng quát $X \in R^{m \times n}$ Ví dụ: $X \in R^{2 \times 3}$ $X = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$  Lưu ý: Một vector cột có thể thể hiện là một ma trận $R^{m \times 1}$ , khi chuyển vị vector cột này sẽ thành vector hàng tương ứng với ma trận $R^{1 \times m}$ .	Tạo ma trận (thực chất là cấu trúc ndarray với $n = 2$ ) Cách 1: $X = \text{np.array}([[1, 2, 3], [4, 5, 6]])$ $\text{print}(X)$  Cách 2: $X = \text{np.array}([1, 2, 3, 4, 5, 6]).\text{reshape}((2, 3))$ $\text{print}(X)$  Kiểm tra số chiều và kích thước của ma trận $\text{print}(X.\text{ndim})$ $\text{print}(X.\text{shape})$
<b>Tensor hạng 3</b>	Tạo tensor hạng 3 (thực chất là cấu trúc ndarray với $n = 3$ )

<p>Tổng quát</p> $X \in R^{h \times w \times c}$ <p>Ví dụ:</p> $X \in R^{2 \times 2 \times 3}$ $X = \begin{bmatrix} \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \\ \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix} \\ \begin{bmatrix} 9 & 10 \\ 11 & 12 \end{bmatrix} \end{bmatrix}$ <p>Trong thực tế, biểu diễn (hay mô tả) Toán học của một bức ảnh màu chính là 1 tensor hạng 3 - <math>R^{h \times w \times c}</math> với <math>h</math> là chiều cao bức ảnh, <math>w</math> là chiều rộng bức ảnh và <math>c</math> là số kênh màu của bức ảnh.</p>	<p>Cách 1:</p> <pre>X = np.array([[[1, 2], [3, 4]], [[5, 6], [7, 8]], [[9, 10], [11, 12]]]) print(X)</pre> <p>Cách 2:</p> <pre>X = np.array([1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]).reshape((3, 2, 2)) print(X)</pre> <p>Kiểm tra số chiều và kích thước</p> <pre>print(X.ndim) print(X.shape)</pre> <p><b>Đọc 1 bức ảnh màu, và in nội dung (tensor hạng 3) ra màn hình</b></p> <p><b>Cách 1: sử dụng thư viện matplotlib</b></p> <pre>from matplotlib import pyplot as plt img = plt.imread('D:/data/spring.jpg') print(img.ndim) print(img.shape)</pre> <p>Tham khảo tài liệu kỹ thuật: <a href="#">link</a>.</p> <p><b>Cách 2: sử dụng thư viện scikit-image</b></p> <pre>from skimage import io img = io.imread('D:/data/spring.jpg') print(img.ndim) print(img.shape)</pre> <p>Tham khảo tài liệu kỹ thuật: <a href="#">link</a>. (Có thể sử dụng tài liệu này để học cách truy xuất camera của máy tính; cách đọc dữ liệu từ tập tin video)</p> <p><b>Cách 3: sử dụng thư viện opencv</b></p> <pre>import cv2 img = cv2.imread('D:/data/spring.jpg') print(img.ndim) print(img.shape)</pre> <p>Tham khảo tài liệu hướng dẫn: <a href="#">link</a> (Có phần hướng dẫn cài đặt)</p>
---	---

	Lưu ý: cả 3 cách này đều trả về dữ liệu là 1 ndarray của numpy. Do vậy, các phép toán xây dựng trên numpy hoàn toàn có thể thực hiện với dữ liệu ảnh thu được từ 3 thư viện này.
<b>Câu hỏi mở rộng: Theo bạn biểu diễn (hay mô tả) Toán học của đoạn phim (video) là gì?</b> Gợi ý: Tensor hạng 4, hãy xác định ý nghĩa mỗi chỉ số trong tensor hạng 4 này cho phù hợp.	Hãy viết chương trình đọc 1 đoạn video vào cấu trúc ndarray của numpy