

NguyenThaiNguyen_2274802010587_Lab2

February 16, 2025

1 LAB 2: NUMPY, PANDAS, VÀ MATPLOTLIB NÂNG CAO

1.1 1. NUMPY

```
[3]: import numpy as np

x = np.array([1, 2, 3, 4, 5, 6, 7, 8])

x[3]

#lấy từ 3-8
x[2:8]

#Lấy từ 3 - 7
x[2:7]
x[2:-1]

#Lấy 7-8
x[-2:]

# Lấy phần tử 2 và 4
x[[2,4]]
```

```
[3]: array([3, 5])
```

```
[4]: data = [(1, 2), (3, 4), (5, 6)]

mang = np.array(data)
print(mang)

#Lấy phần tử 2
print(mang[0,1])

#Lấy phần tử 6
print(mang[2,1])

#Lấy 3,4,5,6
print(mang[1:, :])
```

```

#Cách khác
print(mang[1:])
print(mang[(1,2), :])
print(mang[-2:])

#Lấy 1,3,5
print(mang[:,0])

#Lấy 1-5
print(mang[(0,2),0])

#Lấy 1 và 5 theo format start:end:step
print(mang[:,2,0])

```

```

[[1 2]
 [3 4]
 [5 6]]
2
6
[[3 4]
 [5 6]]
[[3 4]
 [5 6]]
[[3 4]
 [5 6]]
[[3 4]
 [5 6]]
[1 3 5]
[1 5]
[1 5]

```

```

[5]: x = [
      (1, 2, 3, 4),
      (5, 6, 7, 8),
      (9, 10, 11, 12),
      (13, 14, 15, 16),
    ]
mangg = np.array(x)
# lấy 2 6 10 14
print(mangg[:,1])

#Lấy 7 và 11
print(mangg[1:3,2])

#Lấy 4 7 10
print(mangg[[0,1,2], [3,2,1]])

```

```
#lấy 16 15 14 13
print(mangg[3,:][::-1][:4])
print(mangg[3,::-1])
print(mangg[-1, -1::-1])
```

```
[ 2  6 10 14]
[ 7 11]
[ 4  7 10]
[16 15 14 13]
[16 15 14 13]
[16 15 14 13]
```

```
[6]: import numpy as np
```

```
x = np.array([
    (1, 2, 3),
    (4, 5, 6),
    (7, 8, 9)
])

# Lấy 4, 5, 6
print(x[1,:])

# Lấy 3, 5
print(x[:2, 1])

# Lấy 3, 4
print(x[[0, 1], [2, 0]])

# Lấy 9, 6, 3
print(x[[2, 1, 0], [2, 2, 2]])
print(x[:3,-1][::-1])
print(x[::-1, 2])
```

```
[4 5 6]
[2 5]
[3 4]
[9 6 3]
[9 6 3]
[9 6 3]
```

```
[7]: x = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```

```
# Lấy các phần tử chẵn
mang_chan = [num for num in x if num % 2 == 0]
print("Các phần tử chẵn:", mang_chan)
```

```
Các phần tử chẵn: [2, 4, 6, 8, 10]
```

```
[8]: np.ones((3,3))
```

```
[8]: array([[1., 1., 1.],  
          [1., 1., 1.],  
          [1., 1., 1.]])
```

```
[9]: z = np.arange(3)  
z
```

```
[9]: array([0, 1, 2])
```

```
[10]: np.ones((3,3)) + np.arange(3)
```

```
[10]: array([[1., 2., 3.],  
          [1., 2., 3.],  
          [1., 2., 3.]])
```

```
[11]: z = z.reshape((3,1))  
z
```

```
[11]: array([[0],  
          [1],  
          [2]])
```

```
[12]: # Tạo một mảng numpy có kích thước 150 x 5. Hãy tưởng tượng mảng này chứa 150  
      ↪ mẫu về chiều cao, cân nặng, tuổi, lương, gpa của sinh viên VLU.  
      # Chia mảng 4 cột đầu tiên thành 1 biến có tên là X và cột cuối cùng thành Y  
      # Chia X thành X_train và X_test chứa 70% dữ liệu và chia Y thành Y_train và  
      ↪ Y_test, trong đó Y_train chứa 70% dữ liệu.  
      # Tạo 10 tập dữ liệu không chồng chéo của X_train  
  
      data = np.random.uniform(0,10,(150,5))  
      data.shape  
  
      # Chia dữ liệu thành X và Y  
      X = data[:, :-1] # 4 cột đầu tiên  
      Y = data[:, 4]   # Cột cuối cùng  
  
      X.shape,Y.shape  
  
      dataset_size = X.shape[0]  
      dataset_size  
  
      train_size = int(0.7 * dataset_size)  
      train_size  
  
      x_train = X[:train_size]  
      x_test = X[train_size:]
```

```
x_train.shape, x_test.shape

split = x_train.shape[0]//10
split

for counter in range(0, x_train.shape[0], split):
    print(x_train[counter:counter+split].shape)
```

```
(10, 4)
(10, 4)
(10, 4)
(10, 4)
(10, 4)
(10, 4)
(10, 4)
(10, 4)
(10, 4)
(10, 4)
(5, 4)
```

[13]: *# BTVN:*
input: ma trận có kích thước (150, 5), tách dữ liệu thành 5 phần 4 phần x
→ và 1 phần y, chia x thành train 70% và test 30%
output:
Nhận xét bài trên:

BTVN: Cho ma trận sau = [(99, 99, 99), (99, 99, 99), (99, 99, 99),] giả sử 0 = O và 1 là X nhận đầu vào từ phía X và O luân phiên cho một cặp chỉ số nếu phía x nhập ((0,0)) thì ma trận trở thành: [(X, 99, 99), (99, 99, 99), (99, 99, 99),]

Nếu phía O nhập ((0,0)) thì lỗi và yêu cầu nhập lại và nếu không thì điền vào ma trận

Nếu ai đó có 3 ô liên tiếp thì dừng trò chơi

```
[ ]: def print_board(board):
    for row in board:
        print(row)

def check_winner(board):
    # Kiểm tra hàng ngang
    for row in board:
        if row[0] == row[1] == row[2] and row[0] != 99:
            return row[0]

    # Kiểm tra hàng dọc
    for col in range(3):
```

```

        if board[0][col] == board[1][col] == board[2][col] and board[0][col] != 99:
↪99:            return board[0][col]

    # Kiểm tra đường chéo
    if board[0][0] == board[1][1] == board[2][2] and board[0][0] != 99:
        return board[0][0]
    if board[0][2] == board[1][1] == board[2][0] and board[0][2] != 99:
        return board[0][2]

    return None

def main():
    # Khởi tạo ma trận
    board = [
        [99, 99, 99],
        [99, 99, 99],
        [99, 99, 99]
    ]

    current_player = 'X'

    while True:
        print_board(board)
        print(f"Lượt của người chơi {current_player}")

        # Nhập chỉ số từ người chơi
        try:
            i, j = map(int, input("Nhập cặp chỉ số (i, j): ").split())
        except ValueError:
            print("Vui lòng nhập đúng định dạng (i j).")
            continue

        # Kiểm tra tính hợp lệ của nước đi
        if i < 0 or i > 2 or j < 0 or j > 2:
            print("Chỉ số không hợp lệ. Vui lòng nhập lại.")
            continue

        if board[i][j] != 99:
            print("Ô đã được đánh dấu. Vui lòng nhập lại.")
            continue

        # Cập nhật ma trận
        board[i][j] = current_player

        # Kiểm tra điều kiện thắng
        winner = check_winner(board)

```

```
    if winner:
        print_board(board)
        print(f"Người chơi {winner} đã chiến thắng!")
        break

    # Đổi lượt chơi
    current_player = 'O' if current_player == 'X' else 'X'

if __name__ == "__main__":
    main()
```

[99, 99, 99]

[99, 99, 99]

[99, 99, 99]

Lượt của người chơi X

[]: