

Bt trên lớp

```
import numpy as np
x = np.array([1, 2, 3, 4, 5])

# in ra phần tử thứ 3 của x

print(x[3:])

[3] Python
... [4 5]

data = [[1,2],[3,4],[5,6]]
mang = np.array(data)
print(mang)

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

print(mang[[0,2],0])

#lấy 1 và 5 theo fommat start:end:step

print(x[0:5:4])

[5] Python
... [1 5]
    [1 5]
```

```
print(mang[:,1,0])

[6] Python
... [1 3 5]

y = [
    [1,2,3,4],
    [5,6,7,8],
    [9,10,11,12],
    [13,14,15,16]
]

y = np.array(y)
print(y)

[7] Python
... [[ 1  2  3  4]
      [ 5  6  7  8]
      [ 9 10 11 12]
      [13 14 15 16]]

print(y[0:,1])

[8] Python
... [ 2  6 10 14]
```

▶ ▾

#2 6 10 14
print(y[0:,1])

[9] Python

... [2 6 10 14]

#7 11
print(y[1:3,2])

[10] Python

... [7 11]

#lây 4 7 10
print(y[[0,1,2],[3,2,1]])

[11] Python

... [4 7 10]

▶ ▾

#16 15 14 13
print(y[3,0:])

[12] Python

... [13 14 15 16]

#13
a = np.array([[1,2,3],[4,5,6],[7,8,9]])
print(a)

[13] Python

... [[1 2 3]
[4 5 6]
[7 8 9]]

▶ ▾

#4 5 6
print(a[1,0:])

[14] Python

... [4 5 6]

+ Code

+ Markdown

#2 5
print(a[[0,1],[1,1]])

[15] Python

... [2 5]

#3 4
print(a[[0,1],[2,0]])

[16] Python

... [3 4]

```
#9 6 3

print(a[2,2],a[1,2],a[0,2])

[17] Python
... 9 6 3

c = np.array([1,2,3,4,5,6,7,8,9,10])
# Lấy các phần tử chẵn trong mảng

even_elements = c[c % 2 == 0]
print(even_elements)

[18] Python
... [ 2 4 6 8 10]

mang1 = np.array([[1., 2., 3.], [4., 5., 6.], [7., 8., 9.]])
print("Mảng ban đầu:")
print(mang1)

# In kích thước của mảng
print("Kích thước ban đầu của mảng:")
print(mang1.shape)

# Tạo một mảng mới có kích thước khác
mang = mang1.reshape(3, 3) # Hoặc dùng kích thước hợp lệ khác
print("Mảng sau khi thay đổi hình dạng:")
print(mang)

[19]
```

```
# 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.

import numpy as np
# Tạo mảng numpy có kích thước 150 x 5
data = np.random.rand(150, 5)

print(data)

Python

[[0.47091638 0.43964287 0.85180422 0.04490048 0.17741901]
 [0.9100844 0.15678895 0.9766307 0.5491604 0.03306373]
 [0.26093759 0.25766773 0.55031793 0.82983444 0.75995321]
 [0.45525646 0.67737778 0.84379792 0.54918259 0.49551534]
 [0.5751502 0.33158833 0.56242546 0.35233006 0.04405264]
 [0.90532315 0.42263528 0.52015669 0.68356907 0.00875563]
 [0.32514519 0.90149748 0.62280223 0.96487457 0.21242377]
 [0.36711524 0.2719175 0.70965927 0.98783461 0.04786877]
 [0.28427341 0.04463034 0.54686143 0.6194306 0.61395873]
 [0.85621042 0.54912549 0.59071734 0.23062174 0.75601763]
 [0.45500039 0.41514993 0.82105489 0.37087378 0.20655744]
 [0.73258792 0.94527697 0.98460117 0.83559886 0.40371949]
 [0.47714409 0.64908235 0.4919489 0.79088299 0.92812156]
 [0.83169171 0.93838394 0.66912742 0.47579343 0.20586079]
 [0.97433226 0.72114827 0.63438555 0.32488509 0.19192324]
 [0.99123313 0.30808322 0.55556922 0.01287113 0.25552618]
 [0.40567055 0.0280219 0.94615638 0.25833585 0.76002911]
 [0.67136085 0.47236431 0.94851323 0.86449399 0.97787921]
 [0.35514508 0.75164084 0.28035789 0.60728783 0.03911099]
 [0.41256705 0.92575441 0.03958725 0.59652734 0.97817835]
 [0.60812776 0.3309665 0.03101586 0.24605062 0.77485695]
 [0.8448966 0.82627178 0.69555174 0.58441673 0.77487799]
 [0.89811653 0.56516538 0.36612708 0.17183082 0.40287263]
 [0.27565507 0.3124209 0.8667716 0.40794259 0.03556316]
```

```
# Chia mảng bốn cột đầu tiên thành một biến có tên là x và cột cuối cùng thành y

x = data[:, 0] # Cột đầu tiên
y = data[:, -1] # Cột cuối cùng

print("x:")
print(x)
print("y:")
print(y)
```

21]

Python

```
.. x:
[0.47091638 0.9100844 0.26093759 0.45525646 0.5751502 0.90532315
0.32514519 0.36711524 0.28427341 0.85621042 0.45500039 0.73258792
0.47714409 0.83169171 0.97433226 0.99123313 0.40567055 0.67136085
0.35514508 0.41256705 0.60812776 0.8448966 0.89811653 0.27565507
0.25174563 0.4611337 0.86082533 0.22638225 0.82905924 0.13767906
0.78248611 0.17355627 0.64156549 0.09481472 0.37025279 0.33433307
0.24631472 0.80666017 0.30814938 0.11934509 0.73219283 0.66100169
0.00424541 0.07540143 0.77975921 0.97683199 0.60855991 0.16935153
0.02671456 0.46148389 0.99652888 0.21468793 0.6917841 0.65415329
0.41582222 0.90595852 0.4075714 0.27301005 0.92229072 0.8079264
0.63658933 0.58039055 0.67010046 0.30579491 0.08689065 0.54815432
0.88988533 0.10337602 0.96114185 0.07459746 0.51305529 0.92489468
0.88244544 0.61749808 0.64140406 0.81207586 0.95481886 0.67777461
0.28180072 0.13320258 0.13952484 0.46860242 0.23980272 0.75302351
0.0680853 0.85530837 0.07471818 0.43930451 0.2434368 0.06406201
0.48229246 0.71312165 0.33310982 0.93820046 0.12745993 0.59448069
0.75465804 0.21389982 0.39750637 0.38030605 0.59332812 0.48240973
0.15815002 0.02991805 0.56048083 0.40308909 0.05373219 0.7757118
0.09878552 0.95302812 0.89362815 0.00754426 0.48274687 0.58451767
0.06809484 0.01348385 0.79306774 0.82692875 0.9622028 0.49473601
0.05940927 0.56297664 0.19748545 0.86258207 0.50141572 0.48276969]
```

```
▷ ▾ # 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.

from sklearn.model_selection import train_test_split

# Chia dữ liệu thành X_train, X_test, y_train, y_test
X_train, X_test, y_train, y_test = train_test_split(data[:, :-1], data[:, -1], test_size=0.3, random_state=42)

print("X_train:")
print(X_train)
print("X_test:")
print(X_test)
print("y_train:")
```

[]

Python

```
... X train:
[[0.46860242 0.61324679 0.82157838 0.17872565]
[0.56379701 0.65209355 0.19931419 0.13133299]
[0.44418058 0.43249722 0.18921606 0.66722304]
[0.81207586 0.36215409 0.18759399 0.86383284]
[0.95302812 0.04810211 0.83382293 0.37527734]
[0.75465804 0.61441264 0.32932215 0.20043894]
[0.40308909 0.74028274 0.74664172 0.73374356]
[0.88988533 0.16570457 0.55415464 0.94955081]
[0.47091638 0.43964287 0.85188422 0.04490048]
[0.19748545 0.18534965 0.22256319 0.11600889]
[0.10337602 0.19509573 0.31339671 0.59206022]
[0.82905924 0.26124867 0.00140377 0.91831544]
[0.73219283 0.70435272 0.00220519 0.22598732]
[0.77975921 0.83467429 0.1876622 0.50316664]
[0.63658933 0.1086675 0.64205821 0.66317501]
[0.86258207 0.86304174 0.67499846 0.47479045]
[0.25174563 0.67874209 0.27290669 0.15177943]
[0.4611337 0.68154901 0.24065875 0.96335682]]
```

```
from sklearn.model_selection import KFold

kf = KFold(n_splits=10, shuffle=True, random_state=42)

# Tạo các tập dữ liệu không chồng chéo của X_train
for train_index, val_index in kf.split(X_train):
    X_train_fold, X_val_fold = X_train[train_index], X_train[val_index]

    print("X_train_fold:")
    print(X_train_fold)
    print("X_val_fold:")
    print(X_val_fold)

... X_train_fold:
[[0.56379701 0.65209355 0.19931419 0.13133299]
 [0.44418058 0.43249722 0.18921606 0.66722304]
 [0.81207586 0.36215409 0.18759399 0.86383284]
 [0.95302812 0.04810211 0.83382293 0.37527734]
 [0.75465804 0.61441264 0.32932215 0.20043894]
 [0.40308909 0.74028274 0.74664172 0.73374356]
 [0.88988533 0.16570457 0.55415464 0.94955081]
 [0.47091638 0.43964287 0.85180422 0.04490048]
 [0.19748545 0.18534965 0.22256319 0.11600889]
 [0.82905924 0.26124867 0.00140377 0.91831544]
 [0.73219283 0.70435272 0.00220519 0.22598732]
 [0.77975921 0.83467429 0.1876622 0.50316664]
 [0.63658933 0.1086675 0.64205821 0.66317501]
 [0.86258207 0.86304174 0.67499846 0.47479045]
 [0.25174563 0.67874209 0.27298669 0.15177943]
 [0.4611337 0.68154901 0.24065875 0.96335682]
 [0.12745993 0.19410228 0.06988851 0.24171767]
 [0.11934509 0.45277187 0.23976647 0.59927733]
 [0.50000000 0.50000000 0.50000000 0.50000000]]
```

Bt home

```
import numpy as np
board = np.full((3, 3), 99, dtype=object)
def print_board(board):
    for row in board:
        print(" ".join(str(cell) if cell not in ['X', 'O'] else cell for cell in row))

def check_winner(board):
    for row in board:
        if np.all(row == 'X') or np.all(row == 'O'):
            return row[0]

    for col in board.T: # board.T là ma trận chuyển vị
        if np.all(col == 'X') or np.all(col == 'O'):
            return col[0]

    if np.all(np.diag(board) == 'X') or np.all(np.diag(board) == 'O'):
        return board[0, 0]

    if np.all(np.diag(np.fliplr(board)) == 'X') or np.all(np.diag(np.fliplr(board)) == 'O'):
        return board[0, -1]

    return None

def play_game():
    board = np.full((3, 3), 99, dtype=object)
    players = ['X', 'O']
    turn = 0 # X đi trước

    while True:
        print_board(board)
        print(f"Lượt của {players[turn]}:")

        try:
            row, col = map(int, input("Nhập vị trí (row, col): ").split())
```

```

    return None

def play_game():
    board = np.full((3, 3), 99, dtype=object)
    players = ['X', 'O']
    turn = 0 # X đi trước

    while True:
        print_board(board)
        print(f"Lượt của {players[turn]}:")

        try:
            row, col = map(int, input("Nhập vị trí (row, col): ").split())

            if board[row, col] in ['X', 'O']: # Kiểm tra ô trống
                print("Vị trí đã có, hãy nhập lại!")
                continue

            board[row, col] = players[turn]

            winner = check_winner(board)
            if winner:
                print_board(board)
                print(f"Người chơi {winner} thắng!")
                break

            turn = 1 - turn # Chuyển lượt cho người chơi còn lại
        except Exception:
            print("Lỗi nhập liệu, hãy nhập lại!")
            continue

if __name__ == "__main__":
    play_game()

```

"trống": Unknown word. cSpell
View Problem (Alt+... Quick Fix... (Ctrl...

```

99 99 99
99 99 99
99 99 99
Lượt của X:
X 99 99
99 99 99
99 99 99
Lượt của O:
Lỗi nhập liệu, hãy nhập lại!
X 99 99
99 99 99
99 99 99
Lượt của O:
X 99 99
O 99 99
99 99 99
Lượt của X:
Vị trí đã có, hãy nhập lại!
X 99 99
O 99 99
99 99 99
Lượt của X:
X 99 99
O X 99
99 99 99
...
X 99 99
O X 99
O 99 X
Người chơi X thắng!

```

Link git

https://github.com/Huysdfghf/sohoa_hk2/tree/main/lab2