

پروژه عملی 1: بازی دوز - راهنمای گام به گام

در این بخش، راهنمای گام به گام بازی دوز یا Tic Tac Toe را در اختیار شما قرار می دهیم.

مرور برخی از ابزارهایی که در بخش دست گرمی استفاده کردیم:

دریافت ورودی کاربر:

```
player1 = input("Please pick a marker 'X' or 'O'")
```

توجه نمایید که تابع `input` ورودی کاربر را در قالب رشته برمیگرداند، به همین دلیل برای تبدیل به عدد صحیح از روش زیر برای `cast` استفاده نمایید:

```
position = int(input('Please enter a number'))
```

برای پاک کردن صفحه نمایش میان تعویض نوبت بازی بازیکنان از روش زیر استفاده نمایید:

```
from IPython.display import clear_output  
clear_output()
```

توجه نمایید که دستور `clear_output()` تنها در محیط Jupyter Notebook قابل اجرا می باشد و برای IDE های دیگر از روش زیر استفاده نمایید.:

```
print('\n'*100)
```

دستور بالا خروجی کنسول قبلی را از محدوده نمایش خارج می نماید. حالا بریم سراغ کد:

مرحله 1: تابعی بنویسید که بتواند یک تخته را چاپ کند. تخته خود را به عنوان یک لیست تنظیم کنید، به طوری که هر شاخص 0-9 با یک شماره در پد شماره مطابقت داشته باشد، بنابراین نمایش 3 در 3 تخته را دریافت می کنید.

X		O		X

O		X		O

X		O		X

In [1]:

```
1 from IPython.display import clear_output
2
3 def display_board(board):
4     clear_output()
5
6     print(' | | ')
7     print(f" {board[7]} | {board[8]} | {board[9]} ")
8     print(' | | ')
9
10    print('-----')
11
12    print(' | | ')
13    print(f" {board[4]} | {board[5]} | {board[6]} ")
14    print(' | | ')
15
16    print('-----')
17
18    print(' | | ')
19    print(f" {board[1]} | {board[2]} | {board[3]} ")
20    print(' | | ')
21
```

تست مرحله 1: تابع خود را با ورودی آزمایش زیر از نقشه مورد آزمایش قرار دهید و در صورت نیاز فاصله ها را تنظیم نمایید.

In []:

```
1 test_board = ['#', 'X', 'O', 'X', 'O', 'X', 'O', 'X', 'O', 'X']
2 display_board(test_board)
```

مرحله 2: تابع بنویسید تا ورودی کاربران در قالب یک عدد که مکان علامت X یا O را بیان می کند، دریافت نماید. می توانید از یک حلقه while برای انجام پرسش تا زمان دریافت ورودی صحیح استفاده نمایید.

In [2]:

```
1 def player_input():
2
3     marker = ''
4
5     while not (marker == 'X' or marker == 'O'):
6         marker = input('Player 1: Do you want to be X or O? ').upper()
7
8     if marker == 'X':
9         return ('X', 'O')
10    else:
11        return ('O', 'X')
```

تست گام دوم: تابع را اجرا نمایید تا از دریافت خروجی مورد نظر اطمینان حاصل نمایید.

In []:

```
1 player_input()
```

گام سوم: تابعی بنویسید که لیست علامت های روی برد بازی X یا 0 و مکان علامت گذاری در بازه صفر تا 9 را دریافت کرده و علامت مورد نظر را در مکان مناسب انتخاب شده قرار دهد.

In [3]:

```
1 def place_marker(board, marker, position):  
2  
3     board[position] = marker
```

تست مرحله 3: تابع فوق را با ورودی برد و مکان انتخاب شده آزمایش نمایید.

In []:

```
1 place_marker(test_board, '$', 8)  
2 display_board(test_board)
```

مرحله 4: تابعی بنویسید تا نشانه X یا 0 را بعنوان ورودی دریافت کرده تشخیص دهد که آیا کاربر برنده شده است یا نه

In [4]:

```
1 def win_check(board, mark):  
2  
3     return ((board[7] == mark and board[8] == mark and board[9] == mark) or  
4             (board[4] == mark and board[5] == mark and board[6] == mark) or  
5             (board[1] == mark and board[2] == mark and board[3] == mark) or  
6             (board[7] == mark and board[4] == mark and board[1] == mark) or  
7             (board[8] == mark and board[5] == mark and board[2] == mark) or  
8             (board[9] == mark and board[6] == mark and board[3] == mark) or  
9             (board[7] == mark and board[5] == mark and board[3] == mark) or  
10            (board[9] == mark and board[5] == mark and board[1] == mark))
```

تست مرحله 4: تابع win_check را بر روی برد آزمایش خودمان تست کنید - باید True را برگرداند.

In []:

```
1 win_check(test_board, 'X')
```

مرحله 5: تابعی بنویسید تا با استفاده از ماژول random یک بازیکن را بعنوان نفر اول انتخاب کرده و یک رشته برای این کار چاپ کند. می توانید از تابع random.randint() استفاده نمایید.

In [5]:

```
1 import random
2
3 def choose_first():
4
5     if random.randint(0, 1) == 0:
6         return 'Player 2'
7     else:
8         return 'Player 1'
```

In []:

```
1 choose_first()
```

مرحله 6: تابعی بنویسید که چک کند آیا فضای خالی در برد بازی در موقعیت مشخص شده، وجود دارد یا نه

In [6]:

```
1 def space_check(board, position):
2
3     return board[position] == ' '
```

مرحله 7: تابعی بنویسید تا بررسی کند آیا برد بازی توسط علامت های بازیکن ها پر شده است یا نه؟ True برای حالت پر و False برای حالت دیگر

In [7]:

```
1 def full_board_check(board):
2
3     for i in range(1, 10):
4         if space_check(board, i):
5             return False
6     return True
```

مرحله 8: تابعی بنویسید تا از کاربر موقعیت حرکت بعدی بازیکن را بپرسد (1-9) و سپس از تابع مرحله 6 برای بررسی خالی بودن موقعیت انتخاب شده استفاده نماید و در صورتیکه موقعیت خالی باشد مقدار موقعیت انتخاب شده را برای ادامه بازی برگرداند.

In [8]:

```
1 def player_choice(board):
2
3     position = '0'
4
5     while position not in ['1', '2', '3', '4', '5', '6', '7', '8', '9'] or not space_check(board, position):
6         position = input('Choose your next position: (1-9) ')
7
8     return int(position)
```

مرحله 9: تابعی بنویسید تا از بازیکن بپرسد که تمایل به انجام بازی مجدد دارد یا نه

In [9]:

```
1 def replay():  
2  
3     return input('Do you want to play again? Enter Yes or No: ').lower().startswith(
```

مرحله 10: حالا به مرحله سخت رسیدیم 😈 سعی کنین تمام منطق بازی را در یک حلقه while با شرط همواره درست قرار دهید.

In [10]:

```
1 print('Welcome to Tic Tac Toe!')
2
3 while True:
4     theBoard = [' '] * 10 # [' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ']
5
6     player1_marker, player2_marker = player_input()
7
8     turn = choose_first()
9
10    print(f"{turn} will go first.")
11
12    play_game = input('Are you ready to play? Enter Yes or No: ')
13
14    if play_game.lower()[0] == 'y':
15        game_on = True
16    else:
17        game_on = False
18
19    while game_on:
20        if turn == 'Player 1':
21
22            display_board(theBoard)
23            position = player_choice(theBoard)
24            place_marker(theBoard, player1_marker, position)
25
26            if win_check(theBoard, player1_marker):
27                display_board(theBoard)
28                print('Congratulations! You have won the GAME!')
29                game_on = False
30            else:
31                if full_board_check(theBoard):
32                    display_board(theBoard)
33                    print('The Game is a DRAW!')
34                    break
35                else:
36                    turn = 'Player 2'
37
38        else:
39            # player2's turn
40            display_board(theBoard)
41            position = player_choice(theBoard)
42            place_marker(theBoard, player2_marker, position)
43
44            if win_check(theBoard, player2_marker):
45                display_board(theBoard)
46                print('Player 2 has WON!')
47                game_on = False
48            else:
49                if full_board_check(theBoard):
50                    display_board(theBoard)
51                    print('The game is a DRAW!')
52                    break
53                else:
54                    turn = 'Player 1'
55
56    if not replay():
57        break
```

58			
0			x

	0		x

0	x		x

Congratulations! You have won the GAME!
Do you want to play again? Enter Yes or No: n

دست مریزاد!