

[| 커스텀 API | >](#)

클래시 로얄 플레이어 API



주in 챗봇 고수 1:1 채팅

2020.04.07. 16:10 조회 160

댓글 0 URL 복사

<http://htmlcode.dothome.co.kr/royale.php?tag={플레이어태그}>

클래시 로얄 플레이어의 기본 정보를 json형식으로 불러옵니다

예:

<http://htmlcode.dothome.co.kr/royale.php?tag=UYYC8VGL>

(이건 제 프로필입니다)



주in님의 게시글 더보기 >

좋아요 1 댓글 0

공유 | 신고

댓글

댓글알림 ☐**Hibot**

댓글을 남겨보세요



등록

[글쓰기](#)[답글](#)[목록](#)[▲ TOP](#)

| | | |
|-------------------------|------|-------------|
| 브롤 스타즈 플레이어 API [5] | 주in | 2020.04.07. |
| 클래시 로얄 플레이어 API | 주in | 2020.04.07. |
| 실시간 업데이트 되는 뉴스 Json [4] | 세종대왕 | 2020.03.28. |
| 카링 유튜브 연결용 링크 ▶ [15] | 프로그 | 2020.03.28. |

이 카페 인기글

eval 질문

Kiri

♡0💬5

안녕하세요

tomohong

♡1💬3

안녕하세요

KHP

♡0💬6

타 메신저 사용시 무한반복

NaN

♡0💬1

카톡방 나가기

hajuhee01

♡0💬15

kakao.py에 대해서

undefined

♡0💬13

오픈이발 괴롭히기(메봇R)

ERROR

♡0💬7

함수를 입력받아 그래프 그리기

Steve28

♡2💬4

틱택토 (Player vs Bot)

```
1 def board():
2     return [
3         [' ', ' ', ' '],
4         [' ', ' ', ' '],
5         [' ', ' ', ' ']]
6
7 def print_board(board):
8     for row in board:
9         print(' '.join(row))
10
11 def is_full(board):
12     for row in board:
13         if ' ' in row:
14             return False
15     return True
16
17 def is_winner(board, player):
18     for row in board:
19         if row == [player, player, player]:
20             return True
21     for col in range(3):
22         if [board[row][col] for row in range(3)] == [player, player, player]:
23             return True
24     if board[0][0] == board[1][1] == board[2][2] == player:
25         return True
26     if board[0][2] == board[1][1] == board[2][0] == player:
27         return True
28     return False
29
30 def minimax(board, depth, player, maximizing):
31     if is_full(board) or is_winner(board, player):
32         return (None, 0)
33     if maximizing:
34         best_score = -float('inf')
35         for i in range(3):
36             for j in range(3):
37                 if board[i][j] == ' ':
38                     board[i][j] = player
39                     score, _ = minimax(board, depth + 1, 'O', False)
40                     board[i][j] = ' '
41                     best_score = max(score, best_score)
42         return (best_score, None)
43     else:
44         best_score = float('inf')
45         for i in range(3):
46             for j in range(3):
47                 if board[i][j] == ' ':
48                     board[i][j] = 'O'
49                     score, _ = minimax(board, depth + 1, player, True)
50                     board[i][j] = ' '
51                     best_score = min(score, best_score)
52         return (best_score, None)
53
54 def main():
55     board = board()
56     while True:
57         print_board(board)
58         player = 'X'
59         i, j = None, None
60         while i is None or j is None:
61             i, j = input("Player X's move (row col): ").split()
62             if i not in range(3) or j not in range(3) or board[i][j] != ' ':
63                 print("Invalid move. Try again.")
64                 continue
65             board[i][j] = player
66         if is_winner(board, player):
67             print(f"Player {player} wins!")
68             break
69         if is_full(board):
70             print("Game is a draw!")
71             break
72         player = 'O'
73         i, j = None, None
74         while i is None or j is None:
75             i, j = input("Player O's move (row col): ").split()
76             if i not in range(3) or j not in range(3) or board[i][j] != ' ':
77                 print("Invalid move. Try again.")
78                 continue
79             board[i][j] = player
80         if is_winner(board, player):
81             print(f"Player {player} wins!")
82             break
83         if is_full(board):
84             print("Game is a draw!")
85             break
86         player = 'X'
87
88 if __name__ == '__main__':
89     main()
```

틱택토 (Player vs Bot)