

Mini Project #1 เกม

ไพ่ Blackjack

ให้สร้างเกมไพ่ Blackjack

ทดลองเล่นได้ที่ <https://games.washingtonpost.com/games/blackjack/>

เงื่อนไขของเกม Blackjack

1. จำนวนไพ่ไม่จำกัดชุด
2. ไม่มี Joker
3. Jack/Queen/King เท่ากับ 10
4. Ace สามารถนับเป็น 1 หรือ 11 ก็ได้

ให้ใช้ List ต่อไปนี้ แทน A -> K

cards = [11, 2, 3, 4, 5, 6, 7, 8, 9, 10, 10, 10, 10]

```
import random
def deal_card():
    cards = [11, 2, 3, 4, 5, 6, 7, 8, 9, 10, 10, 10, 10]
    card = random.choice(cards)
    return card
```

Requirement

1. เริ่มต้นสุ่มแจกไพ่ user และ computer ข้างละ 2 ใบ

```
def play_game():
    user_cards = []
    computer_cards = []
    for i in range(2):
        user_cards.append(deal_card())
        computer_cards.append(deal_card())
    is_game_over = False
```

2. ตรวจสอบว่าเป็น Blackjack หรือไม่ (A + ไพ่แต้ม 10)

```
while not is_game_over:
    user_score = calculate_score(user_cards)
    computer_score = calculate_score(computer_cards)

def calculate_score(cards):
    if sum(cards) == 21 and len(cards) == 2:
        return 0
    if 11 in cards and sum(cards) > 21:
        cards.remove(11)
        cards.append(1)
    return sum(cards)
```

3. ถ้า computer ได้ Blackjack ผู้เล่นจะแพ้ แม้ว่า user จะได้เหมือนกัน ถ้า computer เป็นหน้าอื่น แต่ user ได้ Blackjack ก็ชนะไปเลย

```
print(compare(user_score, computer_score))

def compare(user_score, computer_score):
    if user_score == computer_score:
        return "Draw 🐼"
    elif computer_score == 0:
        return "Lose, opponent has Blackjack 😱"
    elif user_score == 0:
        return "Win with a Blackjack 😎"
```

4. ให้คำนวณคะแนนของ user และ computer

```
while not is_game_over:
    user_score = calculate_score(user_cards)
    computer_score = calculate_score(computer_cards)

def calculate_score(cards):
    if sum(cards) == 21 and len(cards) == 2:
        return 0
    if 11 in cards and sum(cards) > 21:
        cards.remove(11)
        cards.append(1)
    return sum(cards)
```

5. ถ้าแจกไพ่ A จะนับเป็น 11 แต่ถ้าคะแนนเกิน 21 จะนับเป็น 1

```
def calculate_score(cards):  
    if sum(cards) == 21 and len(cards) == 2:  
        return 0  
    if 11 in cards and sum(cards) > 21:  
        cards.remove(11)  
        cards.append(1)  
    return sum(cards)
```

6. เปิดไพ่ใบแรกของ computer ให้ user เห็น

```
print(f"    Your cards: {user_cards}, current score: {user_score}")  
print(f"    Computer's first card: {computer_cards[0]}")
```

7. เกมจะจบเมื่อ user ได้คะแนน > 21 หรือได้ blackjack

```
if user_score == 0 or computer_score == 0 or user_score > 21:  
    is_game_over = True
```

8. ถามผู้ใช้ว่าจะเอาไพ่เพิ่มหรือไม่

```
while True:  
    user_action = input("Type 'y' to get another card, type 'n' to pass: ")  
    if user_action == "y" or user_action == "Y":  
        user_cards.append(deal_card())  
        break  
    elif user_action == "n" or user_action == "N":  
        is_game_over = True  
        break  
    else:  
        print("Please try again, Type 'y' or 'n' only.")
```

8. เมื่อ user ไม่ต้องการไพ่เพิ่ม computer จะเอาไพ่เพิ่มหากได้คะแนน < 16

```
while computer_score != 0 and computer_score < 16:  
    computer_cards.append(deal_card())  
    computer_score = calculate_score(computer_cards)
```

9. เปรียบเทียบคะแนนระหว่าง user กับ computer ว่าใครชนะ แพ หรือเสมอ

```
print(compare(user_score, computer_score))
```

```
def compare(user_score, computer_score):
    if user_score == computer_score:
        return "Draw 🤝"
    elif computer_score == 0:
        return "Lose, opponent has Blackjack 🙄"
    elif user_score == 0:
        return "Win with a Blackjack 😎"
    elif user_score > 21:
        return "You went over. You lose 😭"
    elif computer_score > 21:
        return "Opponent went over. You win 😁"
    elif user_score > computer_score:
        return "You win 😊"
    else:
        return "You lose 😞"
```

10. เมื่อจบ ให้แสดงหน้าไพ่ของทั้ง 2 ข้าง

```
print(f"    Your final hand: {user_cards}, final score: {user_score}")
print(f"    Computer's final hand: {computer_cards}, final score: {computer_score}")
```

11. ถามว่าจะเล่นต่อหรือไม่

```
while True:
    start_game = input("Do you want to play a game of Blackjack? Type 'y' or 'n': ")
    if start_game == "y" or start_game == "Y":
        play_game()
    elif start_game == "n" or start_game == "N":
        break
    else:
        print("Please try again, Type 'y' or 'n' only.")
```

แนะนำฟังก์ชัน

1. deal_card() สำหรับแจกไพ่
2. calculate_score(cards) คำนวณคะแนน
3. compare(user_score, computer_score)

แนะนำส่วน main

1. แจก user และ computer 2 ใบ โดยใช้ deal_card() และแสดงไพ่ของ computer 1 ใบ
2. ทุกครั้งที่แจกไพ่ ให้ตรวจสอบคะแนน โดยใช้ calculate_score() ถ้า computer หรือ user ได้ blackjack หรือคะแนน > 21 เกมจบ
3. ถ้าเกมยังไม่จบ ถาม user ต้องการไพ่เพิ่มหรือไม่ ถ้าต้องการให้เรียก deal_card() เพื่อเพิ่มไพ่ใน user list แต่ถ้าไม่ก็จบเกม
4. เมื่อผู้ใช้เสร็จ คอมพิวเตอร์เพิ่มไพ่ให้ตัวเอง ถ้าจนกว่าจะ > 16
5. ตัดสินว่าใครชนะ
6. ถามว่าจะกลับไปเล่นใหม่หรือไม่

ตัวอย่างการทำงาน

Do you want to play a game of Blackjack? Type 'y' or 'n': y

Your cards: [10, 10], current score: 20

Computer's first card: 5

Type 'y' to get another card, type 'n' to pass: n

Your final hand: [10, 10], final score: 20

Computer's final hand: [5, 4, 9], final score: 18

You win

Do you want to play a game of Blackjack? Type 'y' or 'n': y

Your cards: [2, 7], current score: 9

Computer's first card: 8

Type 'y' to get another card, type 'n' to pass: y

Your cards: [2, 7, 10], current score: 19

Computer's first card: 8

Type 'y' to get another card, type 'n' to pass: n Your

final hand: [2, 7, 10], final score: 19

Computer's final hand: [8, 11], final score: 19

Draw

Do you want to play a game of Blackjack? Type 'y' or 'n':

คะแนน

1) ตรวจสอบผลการทำงานว่าถูกต้องในทุกเงื่อนไขที่กำหนด

Rubric		
น้อย (1)	ปานกลาง (2)	ดี (3)
ผลการทำงานไม่ถูกต้องเลย	ผลการทำงานถูกต้องบางส่วน	ผลการทำงานถูกต้องทั้งหมด

2) รูปแบบการเขียนโปรแกรมมีความเหมาะสม

Rubric		
น้อย (1)	ปานกลาง (2)	ดี (3)
การกำหนดตัวแปร ไม่มีความหมาย ใน 1 บรรทัด > 1 คำสั่ง	การกำหนดตัวแปรมีความหมาย ใน 1 บรรทัด = 1 คำสั่ง การเลือกโครงสร้างโปรแกรมยังไม่ เหมาะสม	การกำหนดตัวแปรมีความหมาย ใน 1 บรรทัด = 1 คำสั่ง การเลือกโครงสร้างโปรแกรมมีความ เหมาะสมดี

3) การเลือกใช้ Data Structure และ Algorithm เพื่อแก้ไขปัญหา หาค่า ความเหมาะสมกับโจทย์ที่ได้รับ

Rubric		
น้อย (1)	ปานกลาง (2)	ดี (3)
การเลือก data structure ไม่มีความ เหมาะสมกับลักษณะปัญหา	การเลือก data structure มีความเหมาะสม กับลักษณะปัญหา แต่การเลือกใช้ algorithm ยังไม่เหมาะสม	การเลือก data structure มีความ เหมาะสมกับลักษณะปัญหา และการเลือกใช้ algorithm มีความ เหมาะสมดี

หมายเหตุ

เกมให้เลือกอีก 1 แบบคือ เกมทายตัวเลข 4 หลัก โดยให้คํ อมพิวเตอรสุ่มตัวเลข 1 หลัก 4 ตัว
เมื่อผู้ทายทายให้ บอกว่า ถูกตัวเลข กี่ตัว และถูกตำแหน่งกี่ตัว โดยให้ทายได้จำกัดจำนวน

Code ทั้งหมด

```
import random
def deal_card():
    cards = [11, 2, 3, 4, 5, 6, 7, 8, 9, 10, 10, 10, 10]
    card = random.choice(cards)
    return card
```

```
def calculate_score(cards):
    if sum(cards) == 21 and len(cards) == 2:
        return 0
    if 11 in cards and sum(cards) > 21:
        cards.remove(11)
        cards.append(1)
    return sum(cards)
```

```
def compare(user_score, computer_score):
    if user_score == computer_score:
        return "Draw 🤝"
    elif computer_score == 0:
        return "Lose, opponent has Blackjack 😱"
    elif user_score == 0:
        return "Win with a Blackjack 😎"
    elif user_score > 21:
        return "You went over. You lose 😭"
    elif computer_score > 21:
        return "Opponent went over. You win 😁"
    elif user_score > computer_score:
        return "You win 😊"
    else:
        return "You lose 😞"
```

```

def play_game():
    user_cards = []
    computer_cards = []
    for i in range(2):
        user_cards.append(deal_card())
        computer_cards.append(deal_card())
    is_game_over = False
    while not is_game_over:
        user_score = calculate_score(user_cards)
        computer_score = calculate_score(computer_cards)
        print(f"    Your cards: {user_cards}, current score: {user_score}")
        print(f"    Computer's first card: {computer_cards[0]}")
        if user_score == 0 or computer_score == 0 or user_score > 21:
            is_game_over = True
        else:
            while True:
                user_action = input("Type 'y' to get another card, type 'n' to pass: ")
                if user_action == "y" or user_action == "Y":
                    user_cards.append(deal_card())
                    break
                elif user_action == "n" or user_action == "N":
                    is_game_over = True
                    break
                else:
                    print("Please try again, Type 'y' or 'n' only.")
    while computer_score != 0 and computer_score < 16:
        computer_cards.append(deal_card())
        computer_score = calculate_score(computer_cards)
    print(f"    Your final hand: {user_cards}, final score: {user_score}")
    print(f"    Computer's final hand: {computer_cards}, final score: {computer_score}")
    print(compare(user_score, computer_score))

```

```

while computer_score != 0 and computer_score < 16:
    computer_cards.append(deal_card())
    computer_score = calculate_score(computer_cards)
print(f"    Your final hand: {user_cards}, final score: {user_score}")
print(f"    Computer's final hand: {computer_cards}, final score: {computer_score}")
print(compare(user_score, computer_score))

```

```

while True:
    start_game = input("Do you want to play a game of Blackjack? Type 'y' or 'n': ")
    if start_game == "y" or start_game == "Y":
        play_game()
    elif start_game == "n" or start_game == "N":
        break
    else:
        print("Please try again, Type 'y' or 'n' only.")

```