

Python programming and practice

# ***Lunch menu recommendation***

***projects***

Date : 2023.11.19

Name : 정승현

ID : 221186

# **1. Introduction**

## **1) Background**

These days, college students have budget constraints, and college students always eat at limited restaurants during lunch break, so choosing a lunch menu every day has become such a difficult task. To solve this difficult situation, a program is needed to recommend lunch menus according to the user's budget and cooking needs on the day.

## **2) Project goal**

If college student users enter the food category and budget they want to eat, the goal is to randomly inform them of food and restaurants that fit the food category and budget.

## **3) Differences from existing programs**

Existing programs simply enter the food category that users want to eat and randomly recommend menus, so there is a problem of recommending menus at ridiculous prices or difficult-to-eat menus. We ask users to enter not only the food category they want to eat but also the user's budget, and randomly recommend menus that take into account categories and budgets. And considering college student users who are sensitive to calories, writing calories next to random menus helps them select lunch menus. In this regard, menus can be recommended to users more realistically, and there is a difference from

other programs.

## 2. Design and Implementation

```
# 파일에서 식당 정보를 읽어오는 함수
def read_restaurants_from_file(file_name): # file_name으로 파일을 불러온다.
    try:
        restaurant_list = open(file_name, 'r', encoding='utf8')
        restaurant_data = restaurant_list.read() # restaurant_list를 문자열로 읽어와 변수에 저장한다.
        # 파일에 저장된 텍스트를 문자열로 저장한 restaurant_data를 파이썬 코드로 실행하여 식당 정보를 담고 있는 리스트로 변환시킨다.
        restaurants = eval(restaurant_data)
        return restaurants
    except FileNotFoundError: # 파일을 찾을 수 없을 때를 대비한 예외 처리이다.
        print("파일을 찾을 수 없습니다.")
```

### 1) Function: read\_restaurants\_from\_file(file\_name)

- Functionality: Reads restaurant information from a file and converts it into a list.

Parameters:

- file\_name: Name of the file containing restaurant data.

Process:

- Opens the specified file (file\_name).
- Reads the content from the file and converts it into a string (restaurant\_data).
- Uses eval() to convert the string into a list (restaurants).

Returns:

- 'restaurants': List of restaurant information.

Except FileNotFoundError:

- Handles the case when the specified file is not found.

```
# 사용자에게 식당을 선택하도록 하는 함수
def select_restaurant(restaurants):
    print("선택 가능한 식당 목록:")
    for i, restaurant in enumerate(restaurants): # enumerate 함수를 통해 restaurants 리스트를 반복하면서 각 식당의 인덱스와 식당 정보를 동시에 얻는다.
        print(f"{i + 1}. {restaurant['name']} ({restaurant['cuisine']})") # 반복문을 통해 딕셔너리 형태의 5개의 식당의 이름과 음식 종류를 보여준다.

    while True:
        try:
            choice = int(input(f"원하는 식당을 선택하세요 (1부터 {len(restaurants)}까지): ")) # 사용자에게 원하는 식당 번호를 입력받는다.
            if 1 <= choice <= len(restaurants): # 만약 사용자가 입력한 숫자가 1부터 마지막 식당까지의 숫자에 포함되면
                return restaurants[choice - 1] # 인덱스는 0부터 시작하므로 사용자가 입력한 숫자에 -1을 하면 해당 인덱스의 딕셔너리 형태의 원하는 식당의 정보가 반환된다.
            else: # 만약 1부터 마지막 식당까지의 숫자가 아니라면 다시 선택하도록 한다.
                print(f"잘못된 선택입니다. 1부터 {len(restaurants)}까지의 번호 중 하나를 선택하세요.")
        except ValueError: # 숫자가 아닌 다른 문자를 입력했을 때를 대비한 예외 처리이다.
            print("올바른 숫자를 입력하세요.")
```

## 2) Function: select\_restaurant(restaurants)

Functionality: Allows the user to select a restaurant from a given list.

Parameters:

- restaurants: List of restaurant information.

Process:

- Displays the available restaurants and their cuisine types.
- Accepts user input for selecting a restaurant by its index.

Returns:

- Information of the selected restaurant.

```
# 메뉴를 랜덤하게 선택하는 함수
def select_random_menu(restaurant):
    menu = restaurant['menu'] # restaurant에서 'menu'의 정보들을 변수에 저장한다.
    random_choice = random.choice(menu) # 저장된 'menu'의 식사 메뉴 중 하나를 무작위로 뽑는다.
    return random_choice # 무작위로 뽑힌 식사 메뉴를 반환한다.
```

## 3) Function: select\_random\_menu(restaurants)

Functionality: Randomly selects a menu item from a given restaurant.

Parameters:

- staurant: Information about a specific restaurant.

Process:

- Retrieves the menu items from the provided restaurant data.
- Randomly selects one menu item.

Returns:

- Information of the randomly selected menu item.

```
# 사용자가 자신의 예산을 입력하면 예산에 맞춰 랜덤하게 메뉴를 추천하는 함수
def recommend_menu_in_budget(restaurant):
    while True:
        budget = int(input("예산을 입력하세요: ")) # 사용자에게 자신의 예산을 입력 받는다.
        if budget <= 0: # 만약 예산이 0보다 작다면 0 이상의 올바른 예산을 다시 입력받는다.
            print("올바른 예산을 입력하세요. (0보다 큰 숫자)")
        elif budget > 0: # 만약 예산이 0보다 크다면
            menu_in_budget = [] # 사용자의 예산 내 메뉴를 저장하는 리스트를 만든다.
            for menu in restaurant['menu']: # 각 식당의 메뉴들을 반복문을 통해 menu에 저장한다.
                if menu['price'] <= budget: # 만약 메뉴의 가격이 사용자의 예산 내라면
                    menu_in_budget.append(menu) # 메뉴를 리스트에 저장한다.
            if not menu_in_budget: # 만약 리스트에 사용자의 예산 내 메뉴가 아무것도 없다면 출력한다.
                print("선택 가능한 메뉴가 없습니다. 더 높은 예산을 입력하세요.")
            else: # 리스트에 사용자의 예산 내 메뉴가 있다면 그 중에서 랜덤하게 메뉴 하나를 출력한다.
                random_menu = random.choice(menu_in_budget)
                return random_menu
            break
        else: # 예산을 입력할 때 숫자가 아니라면 출력한다.
            print('숫자를 입력하세요.')
```

#### 4) Function: recommend\_menu\_in\_budget(restaurant)

Functionality: Recommends a menu item based on user-defined budget.

Parameters:

- restaurant: Information about a specific restaurant.

Process:

- Prompts the user to input their budget.
- Filters menu items within the budget and randomly selects one.

Returns:

- Information of the menu item within the user's budget.

```
# 사용자 리뷰 입력 함수
def write_review():
    review = input("식당에 대한 리뷰를 작성하시겠습니까? (Y/N): ").lower() # 사용자에게 리뷰를 작성할건지에 대해 물어본다.
    if review == "y": # 만약 작성한다면(Y) 리뷰를 작성하고 review_text 변수에 저장한다.
        review_text = input("리뷰를 작성해주세요: ")
        return review_text
    else: # 작성 하지 않으면(N) 그냥 return문을 반환한다.
        return "사용자가 리뷰를 작성하지 않았습니다."
```

## 5) Function: write\_review()

Functionality: Allows the user to write a review for a restaurant.

Process:

- Asks the user if they want to write a review (Y/N).
- If the user chooses to write a review, prompts for the review text.

Returns:

- User-written review text or a default message if no review is provided.

```
restaurant_data = read_restaurants_from_file("list_restaurants.txt") # "list_restaurants.txt" 파일을 read_restaurants_from_file 함수를 통해 불러온다.
if not restaurant_data: # 만약 불러온 파일에 텍스트가 없다면 출력한다.
    print("식당 리스트가 없습니다.")
```

## 6) Function:

- Reads restaurant information from the file "list\_restaurants.txt".

```
selected_restaurant = select_restaurant(restaurant_data) # 파일의 문자를 리스트로 변환 후 저장된 restaurant_data에서 함수를 통해 사용자가 원하는 식당의 정보를 반환한다.
print(f"선택한 식당: {selected_restaurant['name']} ({selected_restaurant['cuisine']})") # 딕셔너리로 저장된 식당 정보에서 식당의 이름과 요리 종류를 출력한다.
```

## 7) Function:

- After converting the characters of the file into a list, information on the restaurant desired by the user is returned through a function from the stored restaurant\_data.

- The name of the restaurant and the type of dish are output from the restaurant information stored in the dictionary.

```
selected_menu = select_random_menu(selected_restaurant) # 반환된 식당 정보에서 'menu' 안에 해당하는 식사 메뉴 중 하나를 무작위로 반환
print(f"사용자 추천 메뉴: {selected_menu['name']} (가격: {selected_menu['price']}원, 칼로리: {selected_menu['calories']}kcal)" #
```

## 8) Function:

- In the returned restaurant information, one of the meal menus corresponding to the 'menu' is randomly returned and stored in the variable.
- It outputs the name, price, and calories of the randomly selected menu.

```
selected_menu_by_budget = recommend_menu_in_budget(selected_restaurant) # 사용자의 예산에 맞춰 메뉴를 랜덤하게 하나 뽑아 반환되어 변수에 저장한다.
# 무작위로 뽑힌 메뉴의 이름과 가격, 칼로리를 출력한다.
print(f"예산 맞춤 추천 메뉴: {selected_menu_by_budget['name']} (가격: {selected_menu_by_budget['price']}원, 칼로리: {selected_menu_by_budget['calories']}kcal)"
```

## 9) Function:

- According to the user's budget, a menu is randomly selected and returned and stored in a variable.
- It outputs the name, price, and calories of the randomly selected menu.

```
review = write_review() # review 받기
print("리뷰: " + review)
```

## 10) Function:

- It outputs the name, price, and calories of the randomly selected menu.

### 3. Testing

#### 1) When the user selects the desired restaurant and the menu is randomly recommended at that restaurant

선택 가능한 식당 목록:

1. 식당A (중식)
2. 식당B (한식)
3. 식당C (양식)
4. 식당D (일식)
5. 식당F (패스트 푸드)

원하는 식당을 선택하세요 (1부터 5까지): 1

선택한 식당: 식당A (중식)

사용자 추천 메뉴: 짬뽕 (가격: 9000원, 칼로리: 600kcal)

#### 2) When a user writes down his or her budget and the menu is randomly recommended within his or her budget

##### 1) When there is no menu available within the budget

선택 가능한 식당 목록:

1. 식당A (중식)
2. 식당B (한식)
3. 식당C (양식)
4. 식당D (일식)
5. 식당F (패스트 푸드)

원하는 식당을 선택하세요 (1부터 5까지): 1

선택한 식당: 식당A (중식)

사용자 추천 메뉴: 짬뽕 (가격: 9000원, 칼로리: 600kcal)

예산을 입력하세요: 5000

선택 가능한 메뉴가 없습니다. 더 높은 예산을 입력하세요.

##### 2) There's a menu that you can choose within your budget, so when you recommend a random menu



선택 가능한 식당 목록:

1. 식당A (중식)
2. 식당B (한식)
3. 식당C (양식)
4. 식당D (일식)
5. 식당F (패스트 푸드)

원하는 식당을 선택하세요 (1부터 5까지): 1

선택한 식당: 식당A (중식)

사용자 추천 메뉴: 짬뽕 (가격: 9000원, 칼로리: 600kcal)

예산을 입력하세요: 5000

선택 가능한 메뉴가 없습니다. 더 높은 예산을 입력하세요.

예산을 입력하세요: 10000

예산 맞춤 추천 메뉴: 짬뽕 (가격: 9000원, 칼로리: 600kcal)

### 3) When users finish eating and write a review of the restaurant

#### 1) When a user is not writing a review

선택 가능한 식당 목록:

1. 식당A (중식)
2. 식당B (한식)
3. 식당C (양식)
4. 식당D (일식)
5. 식당F (패스트 푸드)

원하는 식당을 선택하세요 (1부터 5까지): 1

선택한 식당: 식당A (중식)

사용자 추천 메뉴: 짬뽕 (가격: 9000원, 칼로리: 600kcal)

예산을 입력하세요: 5000

선택 가능한 메뉴가 없습니다. 더 높은 예산을 입력하세요.

예산을 입력하세요: 10000

예산 맞춤 추천 메뉴: 짬뽕 (가격: 9000원, 칼로리: 600kcal)

식당에 대한 리뷰를 작성하시겠습니까? (Y/N): n

리뷰: 사용자가 리뷰를 작성하지 않았습니다.

#### 2) When a user writes a review

선택 가능한 식당 목록:

1. 식당A (중식)
2. 식당B (한식)
3. 식당C (양식)
4. 식당D (일식)
5. 식당F (패스트 푸드)

원하는 식당을 선택하세요 (1부터 5까지): 1

선택한 식당: 식당A (중식)

사용자 추천 메뉴: 짬뽕 (가격: 9000원, 칼로리: 600kcal)

예산을 입력하세요: 10000

예산 맞춤 추천 메뉴: 짜장면 (가격: 8000원, 칼로리: 500kcal)

식당에 대한 리뷰를 작성하시겠습니까? (Y/N): y

리뷰를 작성해주세요: 음식이 맛있고 빨리 나와서 좋아요!

리뷰: 음식이 맛있고 빨리 나와서 좋아요!