# 파이썬 프로그래밍및실습

# Developing a food recommendationfu nction

**Progress Report: 2** 

Date: 23/12/10

Name: 이나경

ID: 231135

### 1. Introduction

### 1) Background

Food is something you have to eat every day, but each time you choose a menu, it takes a lot of time to choose a satisfactory menu. This leads to a decrease inbreakfast preparation time, lunch breaks, and work hours while thinking aboutmeals. We need a program that automatically recommends food to solve the problem of wasting time through eating concerns and to select a more satisfyingmenu.

# 2) Project goal

The goal is to recommend food menus based on food-specific ratings given bycustomers, reducing the time to think about food during meal times and allowingthem to select more satisfying menus.

# 3) Differences from existing programs

Existing programs recommend food lists that are already finished and cannot beedited. It is highly likely that foods that do not suit individual preferences willoften be recommended. We have a difference from the existing program in thatby applying a customization method in which customers directly add menus, words they want to leave, allergies, etc., they store only the information they need to make customized recommendations for that customer.

# 2. Functional Requirement

### 1) Function 1 Menu Recommendation Function

- Explanation: We recommend the food based on the rating given by thecustomer by the food.

# (1) Detailed function 1: Select Category

- Explanation: Allows you to select the category in which you would like to berecommended for food.

### (2) Detailed function 2: Save Memo

- Explanation: Save the words you want to leave on the recommended food, suchas recipes, precautions, and allergies.

### (3) Detailed function 3: Save Preferences

- Explanation: Leave a preference for the recommended food and allow it to bereflected in the new food recommendations.

# 2) Function 2: Menu editing function

- Explanation: Ability to edit items or information on recommended food lists

### (1) Detailed function 1: Add Menu

- Explanation: Add food and leave notes such as preferences.

# (2) Detailed function 2: Delete Menu

- Explanation: Delete foods you don't want to be recommended again.

# **3) Function 3:** File Storage Function

- Explanation: Upload the newly entered information, such as preferences andnotes, to the file.

# 3. Progress

### 1) Implementation of features

### (1) Add Menu

- Input; menus: a dictionary that stores categories and menus, ratings: adictionary that stores menus and menu-specific evaluations, memos: a 2D dictionary that stores menus and memos.

Output; The menu you entered is added to the selected category. In addition, menus and ratings are added to the ratings, and menus and notes are added to the memos and returned.

- Description: First, you get a category input, and if that category is not in the keys of menus, you get it back in. If a category exists, enter the menu and ratingyou want to add and save additional menus and ratings in menus and ratings, respectively.
- What you've learned: Repeat statement, condition question, function, module
- Code Screenshot

```
def add menu(menus, ratings, memos):
   while True: # 알맞은 카테고리한 입력받기
      add cate = input("메뉴 추가를 원하는 카테고리를 선택해 주세요.")
      if add_cate not in menus.keys():
          print("입력한 카테고리가 존재하지 않습니다. ")
      else:
          break
   add menu = input("추가를 원하는 메뉴를 입력해 주세요. ")
   add rat = input("추가한 메뉴에 별점을 남겨 주세요.")
   input memo = input("메모를 남기시겠습니까? (y/n)")
   memo = " "
   if (input_memo == 'y'):
      memo = input("메모를 입력해주세요. ")
   menus[add_cate].add(add_menu)
   ratings[add menu] = add rat
   memos[add menu] = memo
   print("메뉴 추가가 완료되었습니다. ")
```

### (2) Delete Menu

- Input; menus: a dictionary that stores categories and menus, ratings: adictionary that stores menus and menu-specific evaluations, memos: a 2D dictionary that stores menus and memos.

Output: Menus, where the selected menu is deleted, the ratings where the menuand rating are missing. Memo from the selected menu also disappears.

- Description: Enter the category with the menu you want to delete first, andproceed with the deletion only if the category exists in the keys of the menu. Similarly, configure the deletion to proceed only when an incoming menu exists.
- What you've learned: Repeat statement, conditional statement, function, module
- Code Screenshot

```
def del menu(menus, ratings, memos):
   while True: # 카테고리 존재하지 않는 경우 삭제.
      del cate = input("삭제를 원하는 메뉴가 있는 카테고리를 선택해 주세요.")
      if del cate not in menus.keys():
          print("선택한 카테고리가 존재하지 않습니다. 다시 선택해 주세요.")
          continue
      else:
          break
   print(*menus[del cate])
   while True:
      del menu = input("삭제를 원하는 메뉴를 선택해 주세요.")
      if del menu not in menus[del cate]:
          print("해당 카테고리에 존재하지 않는 음식입니다. ")
          continue
      else:
          break
   menus[del cate].remove(del menu)
   del ratings[del menu]
   del memos[del menu]
   print("삭제가 완료되었습니다.")
```

### (3) File Storage Function

- Input; menus: a dictionary that stores categories and menus, ratings: adictionary that stores menus and menu-specific evaluations, memos: a 2D dictionary that stores menus and memos.

Output: Save to a file reflecting the facts you have added or deleted. Returns the file where the results are stored.

- Description: Convert menus to a list of tubules, and then to DataFrame with columns Category and Menu. Convert each element in the list to a separate row in the expode, and set drop = True to delete the previous index value.

Other dictriaries also proceed to DataFrame in a similar way, merge them into merges and write them to the csv file as pandas.

- What you've learned: pandas, dictionary
- Code Screenshot

```
# menus, ratings, memos를 DataFrame으로 변환
menus_df = pd.DataFrame(list(menus.items()), columns=['Category', 'Menu']).explode('Menu').reset_index(drop=True)
ratings_df = pd.DataFrame(list(ratings.items()), columns=['Menu', 'Ratings'])
memos_df = pd.DataFrame(list(memos.items()), columns=['Menu', 'Memo'])

# menus, ratings, memos를 병합
result_df = pd.merge(menus_df, ratings_df, on='Menu')
result_df = pd.merge(result_df, memos_df, on='Menu')
# 결과를 csv 파일로 저장
result_df.to_csv('menus.csv', index=False)
```

# 2) Test Results

### (1) Add Menu

- Description: Test to see if the menu and horoscope have been added as normal.
- Test Results Screenshot

```
배뉴들 수가입니다.
메뉴: 한식 씨개 중식 양식 일식 간편식 기타
메뉴: 한식 씨개 중식 양식 일식 간편식 기타
메뉴 추가를 원하는 메뉴를 입력해 주세요. 6dd
추가를 원하는 메뉴를 입력해 주세요. 8dd
추가를 원하는 메뉴를 입력해 주세요. 8dd
추가를 원하는 메뉴를 입력해 주세요. 8
메뉴 추가가 만료되었습니다.
메뉴 추가가 완료되었습니다.
Whather (*물고기: '오징어 두루지기: '닭볶음', '쌤밥', '비뱀밥', '생선구이', '낚지볶음', '게장', '백길비', 'ddd']
watings (*물고기: '9', '오징어 두루지기: '9', '닭볶음': '9', '쌤밥': '9', '쌤밥': '9', '생선구이': '9', '낚지볶음': '9', '게장': '9', '역강비': '9', '김치찌
계': '9', '숙두부찌개: '9', '모징어 두루치기: '9', '닭볶음': '9', '생법': '9', '청국장': '9', '걸비탕': '9', '추어탕': '9', '성계탕': '9', '청장면': '8', '평생': '9', '점망면': '9', '대망면': '9', '점망면': '9', '대망면': '9', '점망면': '9', '전망면': '9', '전망면': '9', '전망면': '9', '주제비': '9', '결국수': '9', '아구찜': '9', '맛타이': '9', '커데': '9', '편당면': '9', '수제비': '9', '결국수': '9', '다구찜': '9', '되라면': '9', '원대': '9', '점망면': '9', '성대': '9', '점망면': '9', '전망면': '9', '전망면': '9', '수제비': '9', '점국수': '9', '되라면': '9', '원대': '9', '원대': '9', '성대': '9', '점망면': '9', '전망면': '9', '현': '9', '전망면':
```

### (2) Delete Menu

- Description: Test to see if the menu and horoscope have been successfully deleted.
- Test Results Screenshot

```
매뉴를 삭제합니다.
한식 찌개 중식 앙식 일식 간편식 기타
삭제를 찾아는 데뉴가 있는 카테고리를 선택해 주세요. 한식
불고기 오징어 두루지기 닭볶음 쌍밥 비빔밥 생선구이 낚지볶음 개장 딱갈비
삭제를 찾아는 데뉴가 있는 카테고리를 선택해 주세요. 한식
불고기 오징어 두루지기 닭볶음 쌍밥 비빔밥 생선구이 낚지볶음 개장 딱갈비
삭제를 찾아는 데뉴를 선택해 주세요. 불고기
삭제가 완료되었습니다.
menus ['오징어 두루지기': '0', '닭볶음': '0', '썸밥': '0', '네ㅂ밥': '0', '생선구이': '0', '낚지볶음': '0', '개장': '0', '딱길비': '0', '김지찌개': '0', '순두
부찌개': '0', '김정찌개': '0', '닭볶음': '0', '쌈밥': '0', '병국장': '0', '갈비팅': '0', '추어팅': '0', '상계팅': '0', '짜장면': '0', '짬뽕': '0', '볶음밥
': '0', '당수육: '0', '마파두부: '0', '양장피': '0', '쌍풍기': '0', '워리기': '0', '고추잡채': '0', '양내계티': '0', '봉골레': '0', '작임파스타': '0',
'피자': '0', '항박스테이크': '0', '김조또': '0', '스테이크': '0', '행비커': '0', '시지 샐러드': '0', '숙밥': '0', '국맨': '0', '닷티기': '0', '댓다': '0', '작맨': '0', '닷티기': '0', '댓다': '0', '작매': '0', '댓다': '0', '딱보기리': '0', '됐다': '0', '목대기': '0', '뭐리': '0', '작매': '0', '댓다': '0', '건다': '0', '
```

# (3) File Storage Function

- Description: Test to ensure that the changes are stored in the file properly.
- Test Results Screenshot

```
할 일을 선택해주세요. (1~4)1
메뉴를 추가합니다.
메뉴 추가를 원하는 카테고리를 선택해 주세요. 한식
추가를 원하는 메뉴를 입력해 주세요. ddd
추가한 메뉴에 별점을 남겨 주세요. 3
메모를 남기시겠습니까? (y/n)n
메뉴 추가가 완료되었습니다.

1. 메뉴 추가
2. 메뉴 삭제
3. 메뉴 추천
4. 종료
할 일을 선택해주세요. (1~4)2
메뉴를 삭제합니다.
삭제를 원하는 메뉴가 있는 카테고리를 선택해 주세요. 기타
월남쌈 아구찜 찜닭 쌀국수 칼국수 카레 수제비 팟타이 닭갈비
삭제를 원하는 메뉴를 선택해 주세요. 닭갈비
삭제를 원하는 메뉴를 선택해 주세요. 닭갈비
삭제를 원하는 메뉴를 선택해 주세요. 닭갈비
```

```
Category,Menu,Ratings,Memo
한식,떡갈비,0,
한식,닭볶음,0,
한식,낚지볶음,0,d
한식,ddd,3,
```

```
간편식,편의점도시락,0,
기타,월남쌈,0,
기타,아구찜,0,
기타,찜닭,0,
기타,쌀국수,0,
기타,칼국수,0,
기타,카레,0,
기타,카레,0,
기타,핫타이,0,
```

# 4. Changes in Comparison to the Plan

No changes made

# 5. Schedule

# - 진행 상황 표기

Work		11/10	11/17	11/24	12/1	12/8	12/15	12/22
Menu Recom mendat ion	Select catego ry					>		
	Save Memo				Ong	Ongoing		
Menu editing Functi on	Save Prefere nces Add Menu	Comp	letion					>
	Delete Menu		Comp	letion				
File Storag e Functi on					Comp	oletion		