

(과제1) a = [2, 3, 4, 5, 6]이 있을 경우, 이 리스트의 순서를 바꾸는 기능을 reverse() 메소드를 사용하지 않고 for-in문과 pop() 메소드를 사용하여 구현하시오(힌트 : 리스트의 원소를 하나하나 순회하면서 pop() 메소드를 호출하시오).

모양 기존 목록 a = [2, 3, 4, 5, 6]에서 요소를 하나씩 선택 pop()하여 제거하면서 새로운 목록에 추가하면 반전리스트를 만들 수 있습니다.

스토어for-in문을 사용하여 리스트를 구성하면서 요소를 리스트에 저장합니다.

구현 코드

파이썬

복사

편집

```
a = [2, 3, 4, 5, 6] # 원본 리스트
reversed_list = [] # 뒤집힌 리스트를 저장할 빈 리스트
for _ in range(len(a)): # 리스트 크기만큼 반복
    reversed_list.append(a.pop()) # pop()으로 마지막 요소 제거 후 새 리스트에 추가
print(reversed_list) # 출력 결과: [6, 5, 4, 3, 2]
```

(과제2) 주어진 튜플에서가장 많이 나타나는 요소를 출력하라.
이때 가장 많이 나타나는 요소가 두 개 이상일 경우 이들 중에서 가장 큰 값을 출력하여라.

collections모듈의 Counter클래스를 사용하여 각 요소를 제거합니다.

최빈값(가장 많이 중요한 값)을 찾습니다.

최빈값이 여러 개일 경우, 선택에서 가장 큰 가치를 선택하세요.

구현 코드

파이썬

복사

편집

```
from collections import Counter
```

```
# 예제 튜플
```

```

t = (4, 2, 4, 3, 2, 5, 4, 5, 5, 5)
# 요소 빈도 계산
counter = Counter(t)
# 가장 많이 등장한 빈도 값 찾기
max_frequency = max(counter.values())
# 빈도 값이 max_frequency인 요소들 중 최댓값 찾기
most_frequent_elements = [key for key, value in
counter.items() if value == max_frequency]
result = max(most_frequent_elements)
print(result) # 출력 결과: 5

```

(과제3-1) 동윤이가 새로 문을 열고 운영하는 커피 가게에는

Americano, Ice Americano, Cappuccino, Caffe Latte, Espresso의 5가지 메뉴가 있으며, 각 메뉴의 가격은 각각 3,000원, 3,500원, 4,000원, 4,500원, 3,600원이다. 이 목록을 menu라는 딕셔너리로 작성하라.

1) 이 menu의 내용을 for key in menu: 구문을 사용하여 다음과 같이 출력하여라.

) 형태로 저장합니다 .메뉴와 가격을 딕셔너리(dict)형태로 저장합니다.

각 메뉴와 가격을 출력하여 하세요 .for key in menu:각 메뉴와 가격을 출력하여 사용하세요.

구현 코드

파이썬

복사

편집

커피 메뉴와 가격을 저장한 딕셔너리

menu = {

```
"Americano": 3000,  
"Ice Americano": 3500,  
"Cappuccino": 4000,  
"Caffe Latte": 4500,  
"Espresso": 3600
```

```
}# for문을 사용하여 출력  
for key in menu:  
    print(f"{key}: {menu[key]}원")
```

출력

yaml

복사

편집

Americano: 3000원

Ice Americano: 3500원

Cappuccino: 4000원

Caffe Latte: 4500원

Espresso: 3600원

(과제3-2) ### 2) 사용자로부터 다음과 같은 주문을 받은 후 주문한 내용이 메뉴에 있는지 구하는 프로그램을 작성하라(힌트: in 연산자를 사용할 것).

메뉴를 딕셔너리로 저장 → 메뉴와 가격 정보를 딕셔너리에 저장
→ 메뉴와 가격 정보를 menu딕셔너리에 저장하세요.

사용자가 입력한 입력 사용자 를 사용하여 주문을 입력합니다 .→
input()을 사용하여 주문을 입력하세요.

다른 방식으로 활용하여 메뉴 확인 → 사용자가 입력한 메뉴 에
존재하는 검사자 .→ 사용자가 입력한 메뉴 menu에 등록된 검사
를 받으세요.

존재하면 가격을 출력합니다.

존재하지 않는 "메뉴에 없습니다" 메시지를 출력합니다.

구현 코드

파이썬

복사

편집

커피 메뉴와 가격을 저장한 딕셔너리

```
menu = {
```

```
    "Americano": 3000,
```

```
    "Ice Americano": 3500,
```

```
    "Cappuccino": 4000,
```

```
    "Caffe Latte": 4500,
```

```
    "Espresso": 3600
```

```
}# 사용자 입력 받기
```

```
order = input("어떤 메뉴를 주문하시겠습니까? ")
```

```
# in 연산자를 사용하여 메뉴 확인
```

```
if order in menu:
```

```
    print(f"{order}의 가격은 {menu[order]}원입니다.")
```

```
else:
```

```
    print("죄송합니다. 해당 메뉴는 없습니다.")
```

작업 구성

평문

복사

편집

어떤 메뉴를 주문하시겠습니까? Cappuccino

Cappuccino의 가격은 4000원입니다.

평문

복사

편집

어떤 메뉴를 주문하시겠습니까? Mocha
죄송합니다. 해당 메뉴는 없습니다.