

쇼핑 총 가격, 사이즈 계산기

2218088 한상우

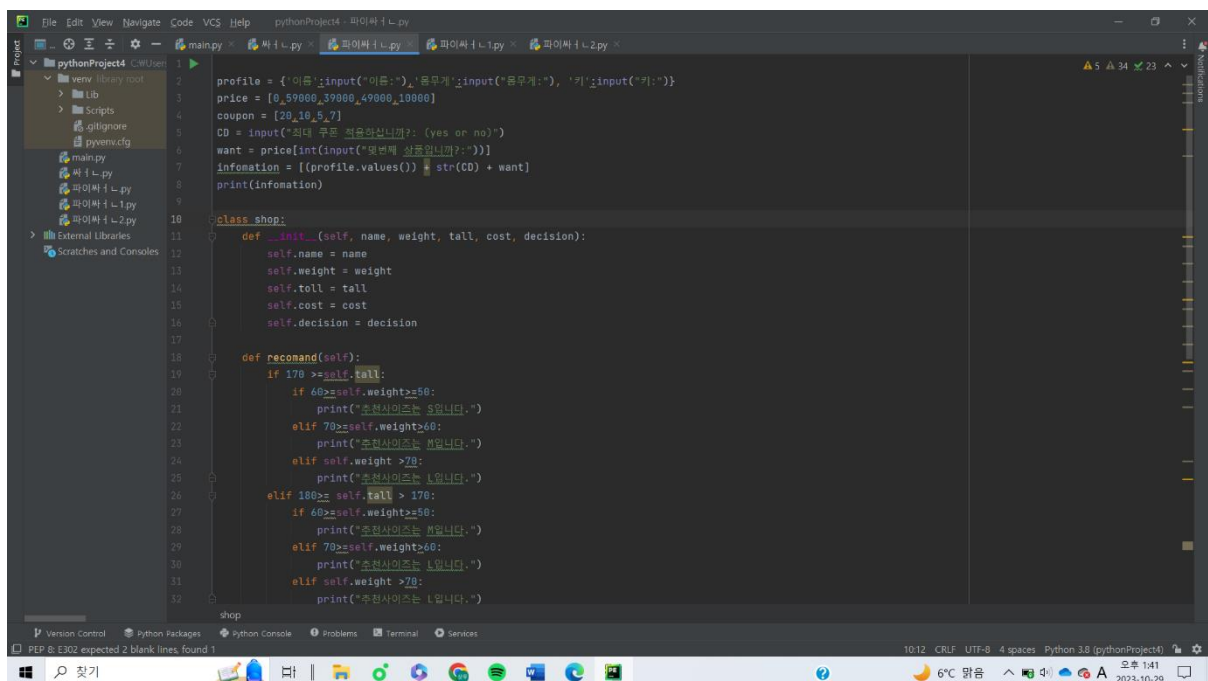
계산기의 목적

- 일상생활에서 쇼핑앱과 쇼핑물을 자주 사용을 한다. 사용 할 때 마다 사이즈를 모르고 가격도 할인가는 잘 모른다. 그래서 이 계산기는 쇼핑앱과 쇼핑사이트에서 마음에 드는 디자인의 옷을 구매할 때 최저가를 계산하고 자신의 사이즈를 찾아줘서 미리 얼마나 사용할지 계산해줌.

계산기 개발 계획

- 프로필 딕셔너리에 자신의 이름 몸무게 키를 집어넣고 쿠폰리스트를 만들어 상품의 이름과 가격을 입력시켜 장바구니에 추가하고 최대 쿠폰의 사용여부를 물어보고 그에 맞게 사이즈와 가격, 남은 쿠폰을 원하는 대로 출력을 시켜줌.

개발 과정 및 후기



```
1 profile = {'이름':input("이름:"), '몸무게':input("몸무게:"), '키':input("키:")}
2 price = [0.59888, 39888, 49888, 19888]
3 coupon = [20, 10, 5, 7]
4 CD = input("최대 쿠폰 적용하시겠습니까? (yes or no):")
5 want = price[int(input("몇번째 상품원하시겠습니까?"))]
6 infomation = [(profile.values()) * str(CD) + want]
7 print(infomation)
8
9
10 class shop:
11     def __init__(self, name, weight, tall, cost, decision):
12         self.name = name
13         self.weight = weight
14         self.tall = tall
15         self.cost = cost
16         self.decision = decision
17
18     def recomand(self):
19         if 170 >= self.tall:
20             if 60 >= self.weight >= 50:
21                 print("추천사이즈는 5입니다.")
22             elif 70 >= self.weight >= 60:
23                 print("추천사이즈는 6입니다.")
24             elif self.weight > 70:
25                 print("추천사이즈는 7입니다.")
26         elif 180 >= self.tall > 170:
27             if 60 >= self.weight >= 50:
28                 print("추천사이즈는 6입니다.")
29             elif 70 >= self.weight >= 60:
30                 print("추천사이즈는 7입니다.")
31             elif self.weight > 70:
32                 print("추천사이즈는 8입니다.")
```

딕셔너리의 밸류값과 리스트와 문자열을 합치는데 안된다는걸 알았음.

```
1 print("안녕하세요!")
2 profile = {'이름': input("이름: "), '몸무게': float(input("몸무게: ")), '키': float(input("키: "))}
3 price = {'가디건': 59000, '셔츠': 39000, '청바지': 49000, '면티': 10000}
4 coupon = [20, 10, 5, 7]
5 CD = input("최대 쿠폰 적용과산니까? (yes or no): ")
6 want = price[input("어떤 상품원시겠습니까?")]
7
8
9
10 class Shop:
11     def __init__(self, name, weight, tall, cost, decision):
12         self.name = name
13         self.weight = weight
14         self.tall = tall
15         self.cost = cost
16         self.decision = decision
17
18     def recommend(self):
19         if self.tall <= 170:
20             if 50 <= self.weight <= 60:
21                 print("초청사이즈는 S입니다.")
22             elif 60 < self.weight <= 70:
23                 print("초청사이즈는 M입니다.")
24             else:
25                 print("초청사이즈는 L입니다.")
26         elif 170 < self.tall <= 180:
27             if 50 <= self.weight <= 60:
28                 print("초청사이즈는 M입니다.")
29             elif 60 < self.weight <= 70:
30                 print("초청사이즈는 L입니다.")
31             else:
32                 print("초청사이즈는 XL입니다.")
```

```
29
30 elif 60 < self.weight <= 70:
31     print("초청사이즈는 L입니다.")
32 else:
33     print("초청사이즈는 XL입니다.")
34
35 if 50 <= self.weight <= 60:
36     print("초청사이즈는 M입니다.")
37 elif 60 < self.weight <= 70:
38     print("초청사이즈는 L입니다.")
39 else:
40     print("초청사이즈는 XL입니다.")
41
42 def bill(self):
43     if 'yes' in self.decision:
44         total = want - (want * max(coupon) / 100)
45         print("총가격은", total, "입니다.")
46         max_coupon = max(coupon)
47         coupon.remove(max_coupon)
48
49     elif 'no' in self.decision:
50         print(want)
51
52 customer = Shop(profile['이름'], profile['몸무게'], profile['키'], want, CD)
53 customer.recommend()
54 customer.bill()
55 print(coupon)
```

그래서 나중에 클래스를 쓸 때 합치는 방법으로 바꿈 그 치만 상품을 여러가지 담을 수 없다는 것을 알게됨

```

1 print("안녕하세요?")
2 profile = {'이름': input("이름: "), '몸무게': float(input("몸무게: ")), '키': float(input("키: "))}
3 products = []
4 products_name = []
5 coupon = [20, 15, 10, 5]
6
7 while True:
8     product_name = input("어떤 상품을입니까? (끝내려면 'end'을 입력하세요): ")
9     if product_name == 'end':
10         break
11     price = float(input(f"{product_name}의 가격: "))
12     products.append({'이름': product_name, '가격': price})
13     products_name.append(str(product_name))
14
15 CD = input("최대 쿠폰 적용하시겠습니까? (yes or no): ")
16
17 class Shop:
18     def __init__(self, name, weight, tall, cart, decision):
19         self.name = name
20         self.weight = weight
21         self.tall = tall
22         self.cart = cart
23         self.decision = decision
24
25     def recommend(self):
26         if self.tall <= 170:
27             if 50 <= self.weight <= 60:
28                 print("추천사이즈는 S입니다.")
29             elif 60 < self.weight <= 70:
30                 print("추천사이즈는 M입니다.")
31             else:
32                 print("추천사이즈는 L입니다.")
33         elif 170 < self.tall <= 180:
34             print("추천사이즈는 XL입니다.")
35         else:
36             print("추천사이즈는 XXL입니다.")
37
38 Shop = Shop("가디건", 70, 175, [], "yes")
39 Shop.recommend()

```

그래서 물건과 가격을 입력 저장하는 식의 장바구니를 구현함.

```

Run: 파이썬 1-2
> 안녕하세요?
이름: 가디건
몸무게: 70
키: 175
어떤 상품입니까? (끝내려면 'end'을 입력하세요): 가디건
가디건의 가격: 70000
어떤 상품입니까? (끝내려면 'end'을 입력하세요): 셔츠
셔츠의 가격: 10000
어떤 상품입니까? (끝내려면 'end'을 입력하세요): end
최대 쿠폰 적용하시겠습니까? (yes or no): yes
추천사이즈는 M입니다.
총 가격은 71850.0원 입니다.
총 구매 상품은 ['가디건', '셔츠']이고 남은 쿠폰은[10, 5]입니다.

Process finished with exit code 0

```

결과로 원하는 총 구매 내역과 남은 쿠폰, 사이즈, 가격의 대한 정보가 나옴

느낀점

- 만들고 보니 옷으로 국한되지 않고 좀 더 잘 다듬으면 모든 쇼핑몰에서 사용 할 수 있을 것 같았음.
- 그동안 코드를 짜면 처음부터 다시 시작했는데 일정부분만 바꾸고 기존에 있는걸 쓰면 더 쉽고 도움이 된다는 걸 깨달음.
- 다 만들고 그전에 오류와 안되는 부분을 따로 공부하여 보다 짧고 간결한 코딩을 할 수 있음.

계산기 효과

일일이 쿠폰사용과 사이즈를 비교 분석하지 않아도 최종 가격과 사이즈만 도출되어 미리 얼마나 사용할지를 알아내는 쇼핑가능

ex) 쇼핑 > 쿠폰, 사이즈 비교, 장바구니 리스트 > 최종가격

계산기 사용 후

쇼핑 > 최종가격