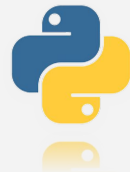


# 스킨케어 추천 프로그램

```
>>> print(현지인(現智人))
```

```
['김지현', '백지은', '손수인', '최정인']
```



## ■ 프로젝트 주제를 선정한 이유

다양한 피부타입과 성별, 연령대에 맞는 스킨케어 아이템을 추천하여 사용자가 쉽고 빠르게 원하는 정보를 얻을 수 있는 서비스를 제공한다.

- ✓ 간단한 선택지를 통해 사용자의 피부타입을 측정하여 추천 기능 제공
- ✓ 연령대별, 피부타입 별 전체 스킨케어 리스트 제공

## ■ 프로젝트 주요 기능 소개

- ✓ 회원가입
- ✓ 로그인 & 로그아웃
- ✓ 메뉴
  1. 내게 맞는 스킨케어 찾기
  2. 피부 타입 별로 보기
  3. 연령대별 추천 아이템
  4. 모든 화장품 리스트 보기
  5. 이전 결과 리스트 보기

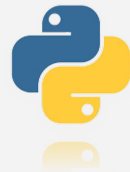
===== J O I N =====
1. 로그인
2. 회원가입
3. 메인 메뉴
4. 프로그램 종료
=====
===== S K I N C A R E   S E R V I C E =====
1. 스킨케어 추천 서비스
2. 연령별 화장품 리스트 보기
3. 피부타입별 화장품 리스트 보기
4. 모든 화장품 리스트 보기
5. 이전 결과 리스트 보기
6. 로그인, 회원가입 메뉴
7. 종료
=====



# 회원가입 및 로그인

```
def check_id(self): # 아이디 유효성 검사
    if self._id not in customerList:
        if len(re.findall("[a-z]+", self._id)) == 0 or len(re.findall("[A-Z]", self._id)) != 0 or len(re.findall("[0-9]", self._id)) == 0:
            print("규격에 맞지않습니다.\n영어소문자와 숫자로 작성해주세요")
            return False
        elif len(self._id) < 6 or len(self._id) > 12:
            print("아이디의 길이는 6자리이상 12자리 이하로 설정해주세요.")
            return False
        else:
            print("올바른 형식입니다.")
            return True
    else:
        print("이미 존재하는 아이디입니다.")
```

- 회원가입 순서 : 아이디 - 비밀번호 - 이름 - 전화번호
- 정보를 입력받고 유효성 검사
  - customerList에 포함여부 검사 (아이디 only)
  - 규격을 정규식표현으로 검사
- 가입된 회원 정보를 customerList에 딕셔너리 타입으로 저장



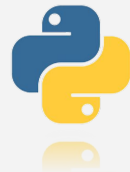
# 로그인 실패시

```
elif pwd == "종료":
    print("메뉴로 돌아갑니다.")
    self.control2 = False
    break
else:
    print("비밀번호를 %d회 잘못 입력 하였습니다." % i)
if i == 3:
    print("")
    print("부정한 접속 시도가 감지되었습니다.")
    print("5초간 시스템 사용이 불가능 합니다.")
    sleep(5)
    print("이제 시스템 사용이 가능합니다.")
    break
break
elif uid == '종료':
    print("메뉴로 돌아갑니다.")
    break
else:
    print("ID가 존재하지 않습니다.")
    print("메뉴로 돌아가시려면 '종료'를 입력해 주세요")
```

번호를 선택해 주세요 ▶ 1  
로그인 합니다.  
아이디를 입력해 주세요 ▶ apple1  
비밀번호 ▶ Apped1  
비밀번호를 1회 잘못 입력 하였습니다.  
비밀번호 ▶ as23fS  
비밀번호를 2회 잘못 입력 하였습니다.  
비밀번호 ▶ vdlnl2XX  
비밀번호를 3회 잘못 입력 하였습니다.

부정한 접속 시도가 감지되었습니다.  
5초간 시스템 사용이 불가능 합니다.  
이제 시스템 사용이 가능합니다.

- 로그인 3회 실패 후 5초간 시스템을 제한
- time의 sleep()함수 사용



# 스킨케어 추천 서비스

```
def choose_sex(self): # 성별고르는 함수명
    print("==" * 11, '스킨케어 추천 서비스', "==" * 11, '\n')
    while True:
        a = input("🟡 성별을 입력해주세요 1.여자 2.남자 ▶ ")
        if a == '1':
            self.gender = '여자'

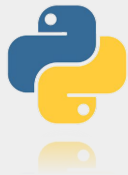
            self.choose_age()
            self.choose_skin()
            customerList.get(self._id)[3] = self.gender
            customerList.get(self._id)[4] = self.old
            customerList.get(self._id)[5] = self.skin
            break

        elif a == '2':
            self.gender = '남자'
            self.choose_age()
            self.choose_skin()
            customerList.get(self._id)[3] = self.gender
            customerList.get(self._id)[4] = self.old
            customerList.get(self._id)[5] = self.skin
            break

    else:
        print("다시 입력해주세요")
    self.survey_summary()
```

```
def choose_age(self): # 나이대를 고르는 함수
    while True:
        b = input("🟡 나이대를 입력해주세요 1.10대 2.20대 3.30대 4.40대+ ▶ ")
        if b == '1':
            self.old = '10대'
            break
        elif b == '2':
            self.old = '20대'
            break
        elif b == '3':
            self.old = '30대'
            break
        elif b == '4':
            self.old = '40대'
            break
        else:
            print("다시 입력해주세요")
```

- 성별(남, 여) 나이대(10~40+) 피부타입(건성, 지성, 민감성)에 대한 설문조사 결과를 받는 함수



# 스킨케어 추천 서비스

```
insert into cosmetic_product values(1, '1025 녹도 토너', 15000, '라운드랩', '10대', '건성', '남자, 여자');
insert into cosmetic_product values(2, '레드 블레미쉬 클리어 수딩크림', 36000, '닥터지', '10대, 30대', '건성', '남자, 여자');
insert into cosmetic_product values(3, '다이버인 저문자 히알루론산 세럼', 22000, '토리든', '10대, 20대, 30대, 40대', '건성', '남자, 여자');
insert into cosmetic_product values(4, '여성초 77% 수딩 토너', 45000, '아누아', '10대', '건성', '남자, 여자');
insert into cosmetic_product values(5, '더 심플 데일리 로션', 28000, '사이닉', '10대', '건성', '남자, 여자');
insert into cosmetic_product values(6, '비자 시카 잠', 23000, '이니스프리', '10대', '건성', '남자, 여자');
insert into cosmetic_product values(7, '티트리 시카 수딩 크림', 16000, '브링그린', '10대', '건성', '남자, 여자');
insert into cosmetic_product values(8, '원더 세라마이드 모찌 토너', 15000, '토니모리', '10대', '건성', '남자, 여자');
insert into cosmetic_product values(9, '녹두 밸런싱 토너', 18000, '비플레인', '20대', '건성', '남자, 여자');
insert into cosmetic_product values(10, '아쿠아 오아시스 토너', 25000, '에스네이처', '20대', '건성', '남자, 여자');
```

- 오라클에 아이템 데이터베이스를 저장

```
def result(self): # 결과 값 제출 함수 (오라클 연동)
    print("\n", "=" * 9, "📦 결과 📦", "=" * 9)
    print()
    if self.gender == '여자' and self.old == '10대' and self.skin == '건성':
        sql1 = "select name, price, company from cosmetic_product where gender like '%여자%' and age like '%10대%' and type like '%건성%'"
        cursor.execute(sql1)
        for row in cursor:
            print(row)
```

- 저장된 데이터를 sql함수에 연동 후 조건에 맞게 출력 (피부타입, 연령대, 전체 목록)

“Life is short, you need Python.”

