

인공 지능 기본

## 02 파이썬 실전 프로그래밍

주소록 프로그램 만들기

# 학습 목표

가. 하나의 클래스를 생성해 본다.

나. 주소록 프로그램을 프로그래밍하면서 프로그래밍에 대해 이해한다.

# 01 클래스 및 함수

```
class Contact:
```

주소록 클래스

```
def print_contact():
```

주소록 출력

```
def print_menu():
```

메뉴 출력

```
def set_contact():
```

주소록 입력(이름, HP, email, 주소)

```
def delete_contact():
```

주소록 삭제

```
def run():
```

주소록 초기화 및 메뉴 선택에 따른 실행 부분

# STEP 01 - Contact 클래스 만들기

```
class Contact:
    def __init__(self, name, phone_number, e_mail, addr):
        self.name = name
        self.phone_number = phone_number
        self.e_mail = e_mail
        self.addr = addr

    def print_info(self):
        print("Name: ", self.name)
        print("Phone Number: ", self.phone_number)
        print("E-mail: ", self.e_mail)
        print("Address: ", self.addr)
```

01 클래스 객체 생성시의 처음 동작  
을 정의

```
def __init__():
```

02 현재 저장된 내용을 출력

```
def print_info(self):
```

## STEP 02 – Contact 클래스

```
class Contact:
    def __init__(self, name, phone_number, e_mail, addr):
        self.name = name
        self.phone_number = phone_number
        self.e_mail = e_mail
        self.addr = addr

    def print_info(self):
        print("Name: ", self.name)
        print("Phone Number: ", self.phone_number)
        print("E-mail: ", self.e_mail)
        print("Address: ", self.addr)

def run():
    kim = Contact('김일구', '010-8812-1193', 'ilgu.kim@python.com',
'Seoul')
    kim.print_info()

if __name__ == "__main__":
    run()
```

정상적으로 동작하는지  
확인하기 위해, 하나의 정보를  
임의로 지정한다.

이후, Contact 클래스에 해당  
정보를 넘겨준다.

# STEP 03 set\_contact() : 정보 입력

## ▶ 연락처 정보 입력받기

```
class Contact:
    ..

def set_contact():
    name = input("Name: ")
    phone_number = input("Phone Number: ")
    e_mail = input("E-mail: ")
    addr = input("Address: ")
    contact = Contact(name, phone_number, e_mail, addr)
    return contact

def run():
    set_contact()

if __name__ == "__main__":
    run()
```

이름, 전화번호, Email, Address  
를 입력받는다. Contact 클래스  
에 이 정보를 넘겨준다.

## STEP 04 메뉴 출력(print\_menu() )

### ▶ 메뉴를 출력

```
def print_menu():  
    print("1. 연락처 입력")  
    print("2. 연락처 출력")  
    print("3. 연락처 삭제")  
    print("4. 종료")  
    menu = input("메뉴선택: ")  
    return int(menu)
```

## STEP 05 연락처 출력 - print\_contact()

- ▶ 주소록 리스트의 주소록을 보여준다.

```
## 출력해 보기
def print_contact(contact_list):
    for contact in contact_list:
        contact.print_info()
```



# STEP 06 run() 메소드

## ▶ 주소록 초기화 및 메뉴 선택에 따른 실행

```
def run():  
    contact_list = []  
    while 1:  
        menu = print_menu()  
        if menu == 1:  
            contact = set_contact()  
            contact_list.append(contact)  
        elif menu == 2:  
            print_contact(contact_list)  
        elif menu == 4:  
            break
```

print\_menu() : 메뉴 출력

set\_contact() : 정보 입력

contact\_list.append(contact)  
: list에 주소 정보 추가

print\_contact(contact\_list)  
: 정보 확인

## STEP 07 delete\_contact() – 연락처 삭제

- ▶ 주소록 리스트의 이름을 기준으로 삭제한다.

```
def delete_contact(contact_list, name):  
    for i, contact in enumerate(contact_list):  
        if contact.name == name:  
            del contact_list[i]
```