

# 파이썬 포트폴리오

20160773 -심기현

# 머리말

복학 후에 새로 배우는 프로그래밍 언어인 '파이썬' 과연 잘 해낼 수 있을까? 라는 걱정이 가득한 상태에서 배우게 되었다.

'파이썬'은 4차 산업혁명에서 적합한 프로그래밍언어이고 , 향후 전망이 밝은 언어이다.

학교를 졸업하고 난 후 , 나는 어디에서 어떤 언어로 실무를 하고 있을지는 모르지만 다양한 언어를 배움으로써 미래의 취업의 길에서 한가지 길이 더 생겼다고 생각할 수 있다.

또 다른 갈림길이 생기는 만큼 그 갈림길이 장애물이 없도록 하려면 역시 '파이썬' 이해를 잘해야 된다. 기본문법들을 이해함으로 장애물 없는 갈림길을 만들어보자는 마음가짐입니다.

주된 내용으로는 도전프로그래밍 과제를 통해 이 과제를 해결하기 위해 사용된 함수,문법을 다시한번 복습하고, 과제소스를 다시한번 이해한대로 풀이를 한내용입니다.

# 목차

1. 강의 계획서
2. 파이썬 소개
  - 간략한 파이썬 소개.
  - 장단점
  - 설치방법
  - 실행을 통한 간략한 코딩.
  - 1장 과제
3. 과제를 통한 복습.
  - 1~6장까지 중간고사 기간까지 배웠던과제  
(도전 프로그래밍 학번에맞는 1,3,5,7번)
  - 사용된 문법,메소드 ,소스 풀어설명하기.

2020 학년도 1학기	전공	컴퓨터정보공학과	학부	컴퓨터공학부
과 목 명	파이썬프로그래밍(2019009-PD)			
강의실 과 강의시간	수:6(3-217),7(3-217),8(3-217)		학점	3
교과분류	이론/실습		시수	3
담당 교수	강환수 + 연구실 : 2호관-706 + 전 화 : 02-2610-1941 + E-MAIL : hskang@dongyang.ac.kr + 면담가능기간 : 화요일 13~16			
학과 교육목표				
과목 개요	2010년 이후 파이썬의 폭발적인 인기는 제4차 산업혁명 시대의 도래와도 밀접한 연관성이 있다. 컴퓨팅 사고력은 누구나가 가져야할 역량이며, 인공지능, 빅데이터, 사물인터넷 등의 첨단 정보기술이 제4차 산업혁명 시대의 기술을 이끌고 있다. 제4차 산업혁명 시대를 주도하는 핵심 기술은 데이터과학과 머신러닝, 딥러닝이며, 이러한 분야에 적합한 언어인 파이썬은 매우중요한 언어가 되었다. 본 교과목은 파이썬 프로그래밍의 기초적이고 체계적인 학습을 수행한다. 본 교과목을 통하여 데이터 처리 방법에 대한 효율적인 파이썬 프로그래밍 방법을 학습한다.			
학습목표 및 성취수준	1. 컴퓨팅 사고력의 중요성을 인지하고 4차 산업혁명에서 파이썬 언어의 필요성을 이해할 수 있다. 2. 기본적인 파이썬 문법을 이해하고 데이터 처리를 위한 자료구조를 이해하여 적용할 수 있다. 3. 문제 해결 방법을 위한 알고리즘을 이해하고 데이터 처리에 적용 할 수 있다. 4. 파이썬 프로그램을 이용하여 실무적인 코딩 작업을 할 수 있다.			
	도서명	저자	출판사	비고
주교재	파이썬으로 배우는 누구나 코딩	강환수, 신용현	홍릉과학출판사	
수업시 사용도구	파이썬 기본 도구, 파이참, 아나콘다와 주피터 노트북			
평가방법	중간고사 30%, 기말고사 40%, 과제물 및 퀴즈 10% 출석 20%(학교 규정, 학업성적 처리 지침에 따름)			
수강안내	1. 파이썬의 개발환경을 설치하고 활용할 수 있다. 2. 파이썬의 기본 자료형을 이해하고 조건과 반복 구문을 활용할 수 있다. 3. 파이썬의 주요 자료인 리스트, 튜플, 딕셔너리, 집합을 활용할 수 있다. 4. 파이썬의 표준 라이브러리와 외부 라이브러리를 이해하고 활용할 수 있다. 5. 파이썬으로 객체지향 프로그래밍을 수행할 수 있다.			



<b>1 주차</b>	<b>[개강일(3/16)]</b>
학습주제	교과목 소개 및 강의 계획 1장 파이썬 언어의 개요와 첫 프로그래밍
목표및 내용	<ul style="list-style-type: none"> <li>• 파이썬 언어란 무엇인지 이해하고 이 언어가 인기 있는 이유를 설명할 수 있다.</li> <li>• 파이썬 개발 도구를 설치해 프로그램을 구현할 수 있다.</li> <li>• 파이썬의 특징과 활용 분야를 설명할 수 있다.</li> </ul>
미리읽어오기	교재 1장, 파이썬 개발환경 설치 파이썬 IDLE
과제,시험,기타	도전 프로그래밍
<b>2 주차</b>	<b>[2주]</b>
학습주제	2장 파이썬 프로그래밍을 위한 기초 다지기
목표및 내용	<ul style="list-style-type: none"> <li>• 파이썬의 자료인 문자열과 수에 대해 이해하고 코드로 구현할 수 있다.</li> <li>• 변수를 이해하고 다양한 대입 연산자를 활용할 수 있다.</li> <li>• 표준 입력으로 문자열을 입력받은 후 원하는 자료로 변환해 활용할 수 있다.</li> <li>• 파이썬 IDLE을 활용할 수 있다.</li> </ul>
미리읽어오기	교재 2장 리터럴과 변수의 이해 아나콘다의 주피터 노트북
과제,시험,기타	도전 프로그래밍
<b>3 주차</b>	<b>[3주]</b>
학습주제	3장 일상에서 활용되는 문자열과 논리 연산
목표및 내용	<ul style="list-style-type: none"> <li>• 문자열에서 문자나 부분 문자열을 반환하는 여러 방법을 구현할 수 있다.</li> <li>• 문자열 객체에 소속된 다양한 메소드를 이해하고 활용할 수 있다.</li> <li>• 논리 값을 이해하고 다양한 연산을 사용해 실생활에서의 표현에 활용할 수 있다.</li> <li>• 아나콘다의 주피터 노트북을 활용할 수 있다.</li> </ul>
미리읽어오기	교재 3장 문자열과 논리연산 파이참(pycharm)
과제,시험,기타	도전 프로그래밍
<b>4 주차</b>	<b>[4주]</b>
학습주제	4장 일상생활과 비유되는 조건과 반복
목표및 내용	<ul style="list-style-type: none"> <li>• 조건에 따라 하나를 결정하는 if문을 구현할 수 있다.</li> <li>• 반복을 수행하는 while문과 for문을 구현할 수 있다.</li> <li>• 임의의 수인 난수를 이해하고 반복을 제어하는 break문과 continue문을 활용할 수 있다.</li> <li>• 파이참(pycharm)을 활용할 수 있다.</li> </ul>
미리읽어오기	교재 4장 조건과 반복
과제,시험,기타	도전 프로그래밍

<b>5 주차</b>	<b>[5주]</b>
학습주제	5장 항목의 나열인 리스트와 튜플
목표및 내용	<ul style="list-style-type: none"> <li>• 다양한 종류의 항목을 쉽게 나열하는 리스트를 구현할 수 있다.</li> <li>• 리스트에서 부분 참조 방법, 이를 이용한 수정, 리스트 연결, 삽입과 삭제 그리고 리스트 컴프리헨션 등을 구현할 수 있다.</li> <li>• 수정할 수 없는 다양한 종류의 항목 나열을 쉽게 처리하는 튜플을 구현할 수 있다.</li> </ul>
미리읽어오기	교재 5장 배열과 리스트
과제,시험,기타	도전 프로그래밍
<b>6 주차</b>	<b>[6주]</b>
학습주제	6장 키와 값의 나열인 딕셔너리와 중복을 불허하는 집합
목표및 내용	<ul style="list-style-type: none"> <li>• 키와 값의 쌍인 항목을 관리하는 딕셔너리를 생성하고 수정하는 방법을 이해하고, 다양한 방법으로 딕셔너리를 구현할 수 있다.</li> <li>• 집합의 특징을 이해하고, 합집합 등과 같은 다양한 집합의 연산을 구현할 수 있다.</li> <li>• 내장 함수 zip()과 enumerate(), 시퀀스 간의 변환을 이해하고, 구현할 수 있다.</li> </ul>
미리읽어오기	교재 6장 집합
과제,시험,기타	도전 프로그래밍
<b>7 주차</b>	<b>[7주]</b>
학습주제	7장 특정 기능을 수행하는 사용자 정의 함수와 내장 함수
목표및 내용	<ul style="list-style-type: none"> <li>• 함수의 내용과 필요성을 이해하고 함수를 직접 정의해 호출할 수 있다.</li> <li>• 인자의 기본 이해와 기본값 지정, 가변 인수와 키워드 인수를 활용할 수 있다.</li> <li>• 간편한 람다 함수와 표준 설치된 내장 함수를 사용할 수 있다.</li> </ul>
미리읽어오기	교재 7장 함수의 정의와 호출
과제,시험,기타	도전 프로그래밍
<b>8 주차</b>	<b>[중간고사]</b>
학습주제	- 직무수행능력평가 1차(중간고사)
목표및 내용	직무수행능력평가, 서술형 평가
미리읽어오기	교재 1장에서 7장까지
과제,시험,기타	
<b>9 주차</b>	<b>[9주]</b>
학습주제	8장 조건과 반복, 리스트와 튜플 기반의 미니 프로젝트 I
목표및 내용	8개의 미니 프로젝트를 스스로 생각하고 프로그래밍해 코딩 능력뿐 아니라 문제 해결 능력을 키울 수 있다.
미리읽어오기	교재 8장
과제,시험,기타	

<b>10 주차</b>	<b>[10주]</b>
학습주제	9장 라이브러리 활용을 위한 모듈과 패키지
목표및 내용	<ul style="list-style-type: none"> <li>표준 모듈을 이해하고 사용자 정의 모듈도 직접 구현해 사용할 수 있다.</li> <li>표준 모듈인 turtle을 사용해 기본적인 도형을 그릴 수 있다.</li> <li>썸드파티 모듈 numpy와 matplotlib 등을 설치해 활용할 수 있다.</li> </ul>
미리읽어오기	교재 9장
과제,시험,기타	도전 프로그래밍
<b>11 주차</b>	<b>[11주]</b>
학습주제	10장 그래픽 사용자 인터페이스 Tkinter와 Pygame
목표및 내용	<ul style="list-style-type: none"> <li>GUI를 이해하고 GUI 표준 모듈인 Tkinter를 사용해 필요한 위젯을 구성하고 윈도우를 생성할 수 있다.</li> <li>이벤트 처리를 이해하고 Tkinter에서 이벤트 처리를 구현할 수 있다.</li> <li>썸드파티 GUI 모듈인 pygame을 설치해 기본적인 윈도우를 구현할 수 있다.</li> </ul>
미리읽어오기	교재 10장
과제,시험,기타	도전 프로그래밍
<b>12 주차</b>	<b>[12주]</b>
학습주제	11장 실행 오류 및 파일을 다루는 예외 처리와 파일 입출력
목표및 내용	<ul style="list-style-type: none"> <li>예외 처리의 필요성을 이해하고 try except 구문을 사용해 예외를 처리할 수 있다.</li> <li>프로그램에서 파일을 생성하는 필요성을 이해하고 필요한 파일을 만들 수 있다.</li> <li>이미 생성된 파일에서 내용을 읽어 처리할 수 있다.</li> </ul>
미리읽어오기	교재 11장
과제,시험,기타	도전 프로그래밍
<b>13 주차</b>	<b>[13주]</b>
학습주제	12장 일상생활의 사물 코딩인 객체지향 프로그래밍
목표및 내용	<ul style="list-style-type: none"> <li>객체와 클래스를 이해하고 필요한 클래스를 정의하고 객체를 만들어 활용할 수 있다.</li> <li>클래스 속성과 인스턴스 속성, 정적 메소드와 클래스 메소드를 이해하고 정의할 수 있다.</li> <li>상속을 이해하고 부모 클래스와 자식 클래스를 정의할 수 있다.</li> <li>추상 메소드와 추상 클래스를 이해하고 정의할 수 있다.</li> </ul>
미리읽어오기	교재 12장
과제,시험,기타	도전 프로그래밍
<b>14 주차</b>	<b>[14주]</b>
학습주제	13장 GUI 모듈과 객체지향 기반의 미니 프로젝트 II
목표및 내용	학습한 파이썬 문법 구조와 프로그래밍 기법을 활용해 8개의 미니 프로젝트를 스스로 생각하고 프로그래밍해 코딩 능력뿐 아니라 문제 해결 능력을 키울 수 있다.
미리읽어오기	교재 1장
과제,시험,기타	

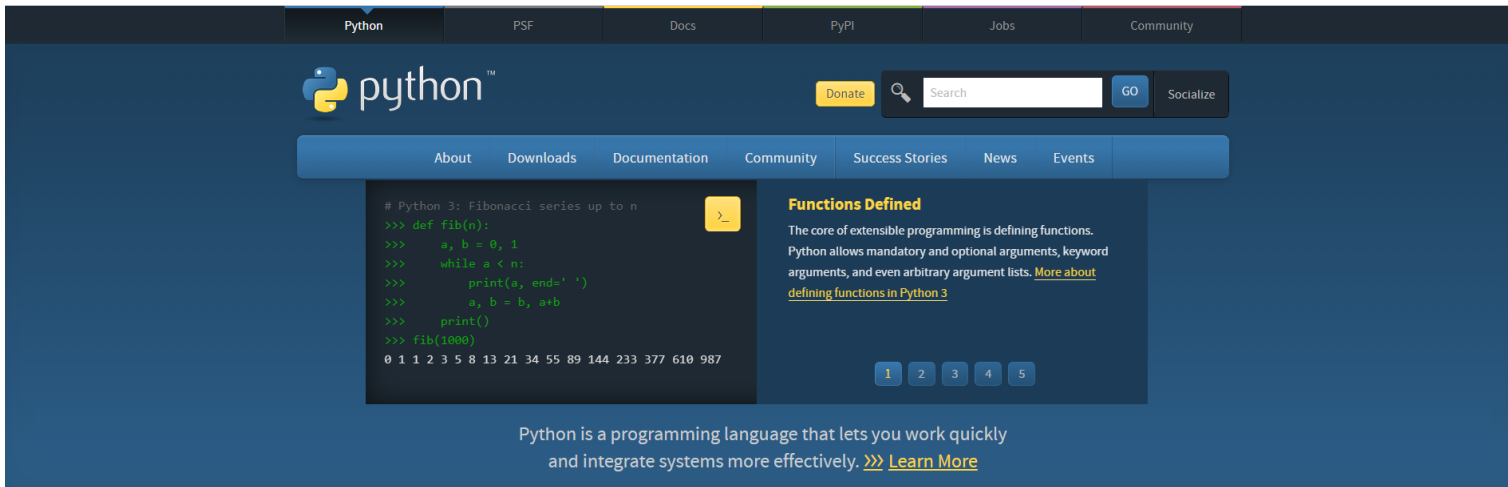


15 주차	[기말고사]
학습주제	직무수행능력평가 2차(기말고사)
목표및 내용	직무수행능력평가, 서술형평가
미리읽어오기	8장에서 13장까지
과제,시험,기타	
수업지원 안내	장애학생을 위한 별도의 수강 지원을 받을 수 있습니다. 언어가 문제가 되는 학생은 글로 된 과제 안내, 확대문자 시험지 제공 등의 지원을 드립니다.



# 파이썬 정의

- 파이썬은 무료로 사용할 수 있는 오픈 소스 프로그래밍 언어이다.
- 1991년 네덜란드의 귀도 반로섬이 개발
- 현재는 비영리 단체인 파이썬 소프트웨어 재단이 관리 하고 있다.
- 4차 산업혁명에서 주의깊게 보고 있는 프로그래밍 언어.



파이썬 홈페이지([www.python.org](http://www.python.org))

# 파이썬 언어의 장단점

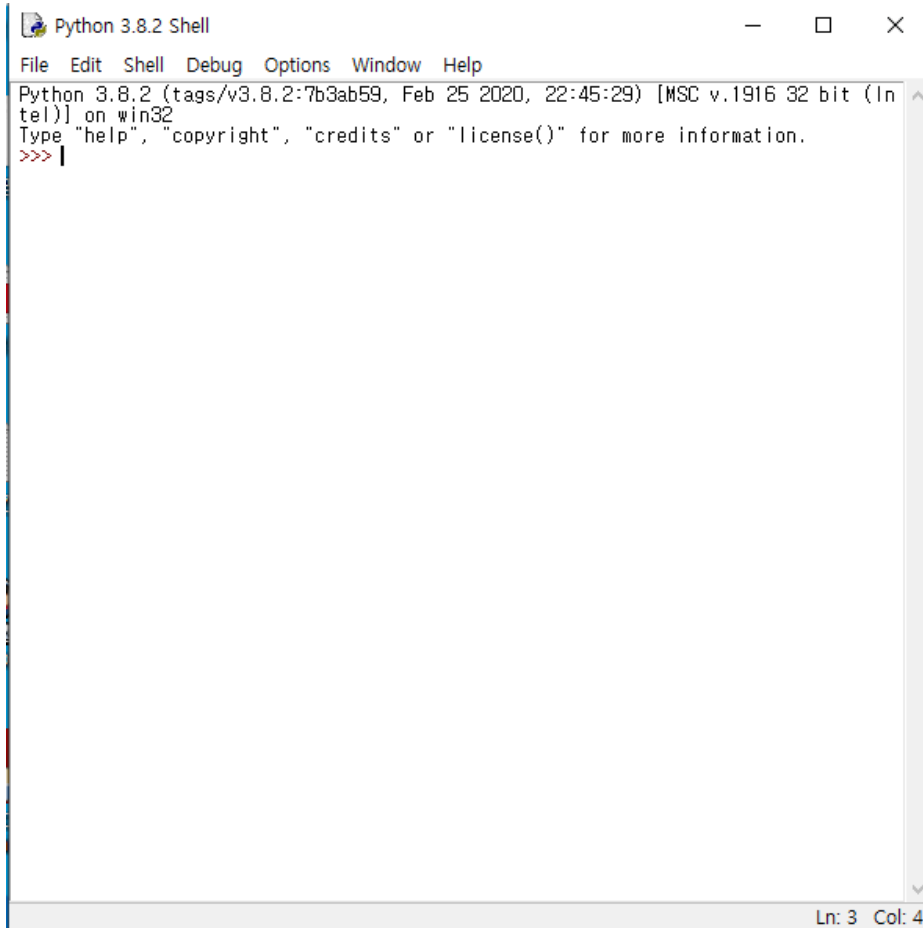
## 장점

1. 다양한 라이브러리들이 있다. (파이썬 커뮤니티에 표준 라이브러리 외에도 다양한 라이브러리들을 공유하고 있어서, 원하는 기능을 가진 라이브러리를 찾아서 다운로드 가능. 오픈소스의 이점)
2. 자료형을 써주지 않아도 된다. (ex int phone , char gender)  
숫자는 그냥 숫자만 기입하고, 문자열은 작은따옴표, 큰따옴표로 구분한다.

## 단점

실행속도가 느리다.

# 파이썬 실행.



```
Python 3.8.2 Shell
File Edit Shell Debug Options Window Help
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:45:29) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> |
```

설치 후 파이썬 IDLE (파이썬셸) 실행화면. 여기서 파이썬의 코딩을 할 수 있다.

자바를 배울때 이클립스를 사용했던것처럼, 파이썬셸도 일종의 코딩을 하기 위한 툴이다.

# 간단한 코딩



```
Python 3.8.2 Shell
File Edit Shell Debug Options Window Help
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:45:29) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> print('첫번째 파이썬 코딩!')
첫번째 파이썬 코딩!
>>> |
```

첫번째 코딩.

print문을 통해 원하는 문구 출력!  
다른 프로그래밍 언어와 달리

세미콜론이 붙지 않는다.

다른 프로그래밍 툴처럼 실행을 위해  
Run을 클릭하는 등의 과정이 없이  
Enter키를 통해 바로바로 출력이  
가능하다.

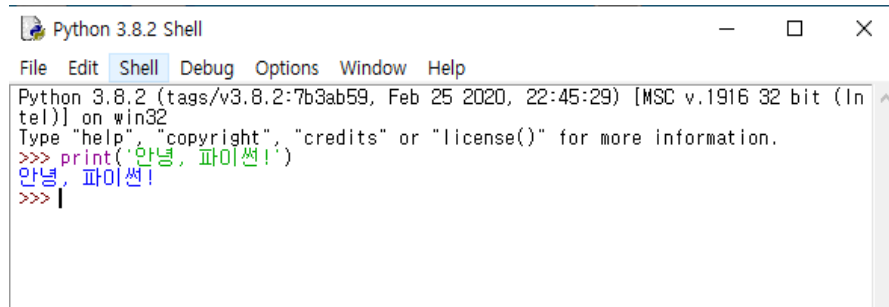
물론 새 파일을 만들어서 코딩을  
할 경우에는, 메뉴탭에서  
[Run]- [Run module]을 통해  
실행가능.

파이썬은 프로그램의 코드가  
한줄씩 순서대로  
해석되고 실행되기를 반복하는  
인터프리트 방식의 언어이다.

# 1장 도전 프로그래밍 - 과제

1.파이썬 IDLE에서 다음을 출력하는 코드를 작성하시오.

- 안녕, 파이썬!

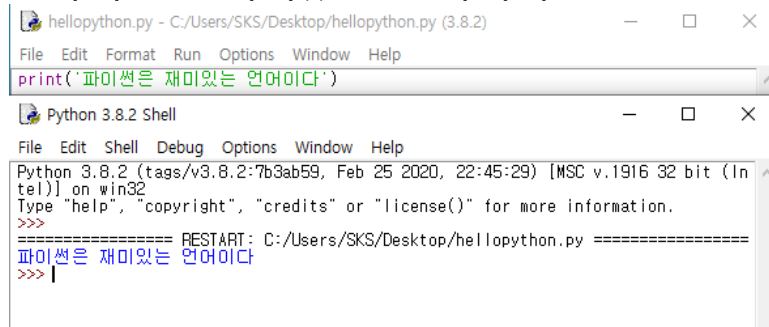


```
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:45:29) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> print('안녕, 파이썬!')
안녕, 파이썬!
>>>
```

2. 파이썬 쉘에서 다음을 출력하는 프로그램을 지정된 파일에 저장해 실행하시오.

-파일명 : hellopython.py

- 파이썬은 재미있는 언어이다.



```
hellopython.py - C:/Users/SKS/Desktop/hellopython.py (3.8.2)
File Edit Format Run Options Window Help
print('파이썬은 재미있는 언어이다')

Python 3.8.2 Shell
File Edit Shell Debug Options Window Help
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:45:29) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/SKS/Desktop/hellopython.py =====
파이썬은 재미있는 언어이다
>>>
```

print()함수는 c언어로 치면 printf와 같이 출력을 위해 사용된다.

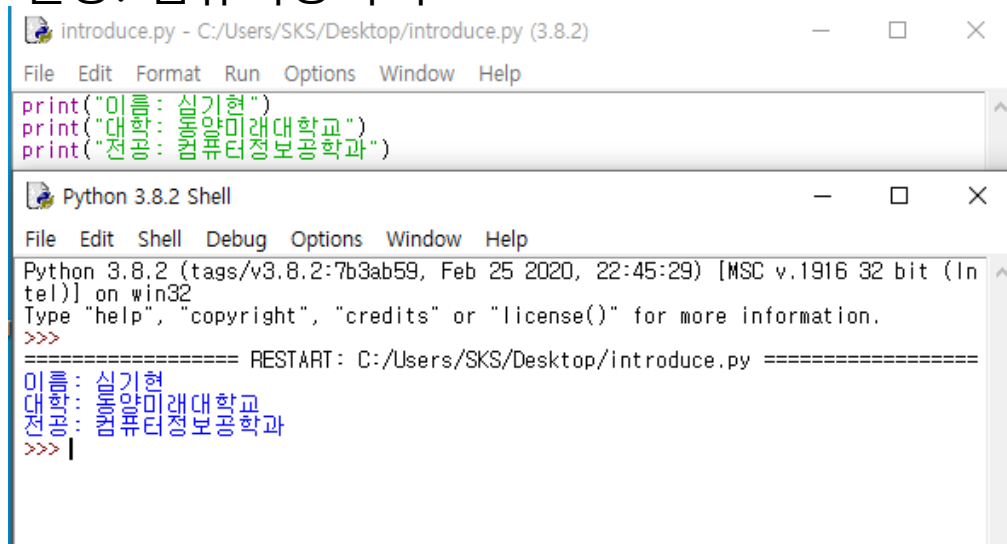
3. 파이썬 셸에서 자신을 소개하는 프로그램을 지정된 파일에 저장해 실행하시오.

-파일 introduce.py

- 이름 : 홍길동 (예시내용이므로 나의 정보로 변경.)

- 대학: 한국대학교

- 전공: 컴퓨터공학과



The screenshot displays two windows from a Python IDE. The top window, titled 'introduce.py - C:/Users/SKS/Desktop/introduce.py (3.8.2)', contains the following Python code:

```
print("이름: 심기현")
print("대학: 서울과학기술대학교")
print("전공: 컴퓨터정보공학과")
```

The bottom window, titled 'Python 3.8.2 Shell', shows the output of running the script. It includes the standard Python version and architecture information, followed by a restart command and the printed output of the script:

```
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:45:29) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/SKS/Desktop/introduce.py =====
이름: 심기현
대학: 서울과학기술대학교
전공: 컴퓨터정보공학과
>>> |
```

## 2장 중요코딩.

```
#2-15 코딩 10진수 정수를 입력받아 2진수, 8진수,  
#10진수, 16진수로 출력하기  
data = int(input('정수 입력 >> '))  
  
print('2진수:', bin(data))  
print('8진수:', oct(data))  
print('10진수:', data)  
print('16진수:', hex(data))  
  
#함수 int()를 사용해서 문자열을 int로 변환.
```



```
정수 입력 >> 25  
2진수: 0b11001  
8진수: 0o31  
10진수: 25  
16진수: 0x19  
>>> |
```

중간고사 코딩으로 나올 법한 진수변환 코딩.

bin()함수는 매개변수로 받은 값을 2진수로 변환해준다.

Oct()함수는 매개변수로 받은 값을 8진수로 변환해준다.

Hex()함수는 매개변수로 받은 값을 16진수로 변환해준다.

여기서는 data가 입력된 값이 들어있는데, input함수앞에 int함수를 사용해서 입력된 값을 int 정수형으로 형변환을 해준 것이다. 그리하여 얻은 정수를 각각 진수로 변환하는 함수들을 통해서 변환된 값을 출력해주는 것이다.

# 2장 도전 프로그래밍 - 과제

1. 두 문자열을 표준 입력으로 받아 한 줄에 출력하는 프로그램을 작성하시오.

-문자열 1 : python

-문자열 2 : 언어

```
print("첫번째 문자열 입력")
str1 = input()
print("두번째 문자열 입력")
str2 = input()
print(str1, str2)
```



```
첫번째 문자열 입력
python
두번째 문자열 입력
언어
python 언어
>>> |
```

3. 아메리카노를 주문받아 총 가격을 출력하는 프로그램을 작성하시오.

-아메리카노의 가격은 3500원

-아메리카노 몇 개 주문하세요? 7

-총 가격은 24500이다.

```
print("아메리카노 몇 개 주문하세요?")
price = 3500
cnt = int(input())

total_price = price * cnt
print(total_price)
```



```
아메리카노 몇 개 주문하세요?
7
24500
>>> |
```



### 1번 풀이

print()문으로 문자열을 입력해달라고 출력을 해주고

Input()함수를 이용하여 값을 입력해준다. 이때, 이 내용을 str1,str2에 각각 저장하고 print문에서 콤마(,)로 구분하여 두개의 문자열을 한줄에 출력해줬다.

### 3번 풀이:

3번은 아메리카노를 주문하여 , 갯수에대해서 합계를 출력해준다. 그러기 위해서 필요한 아메리카노의 가격, 입력한 개수를 저장하는 변수가 필요하다. 입력한 개수는 input()함수를 이용해 입력받은 값을 변수에 저장해준다.

입력한 값이 저장된 변수 cnt와 아메리카노의 가격인 price를 곱하여서 총 주문 금액을 출력한다.  $total\_price = price * cnt$  . 마지막으로 print문으로 total\_price를 출력해주면 원하는 결과가 출력된다.

5. 다음 조건을 참고해 섭씨 온도를 입력받아 화씨 온도로 변환하는 프로그램을 작성하시오.

-다음 식을 사용해 정확한 계산을 수행하고 ( $F=c * 9/5 + 32$ )

-약식 계산은  $f=c * 2 + 30$ 으로 계산해 차이도 출력

```
#다음 조건을 참고해 섭씨 온도를 입력받아  
#화씨 온도로 변환하는 프로그램을 작성하시오.  
  
#약식 계산은 f= c * 2 + 30으로 계산해 차이도 출력  
c=input('온도 입력 >> ')  
f1=int(c) * 9/5 + 32    #정확계산 화씨온도 구하기  
print("정확 계산: 섭씨:",c,"화씨:",f1)  
f2=int(c) * 2+ 30      #약식계산 화씨온도 구하기  
print("약식 계산: 섭씨:",c,"화씨:",f2)  
print("차이 :",f1-f2)
```



```
온도 입력 >> 38  
정확 계산: 섭씨: 38 화씨: 100.4  
약식 계산: 섭씨: 38 화씨: 106  
차이 : -5.599999999999994  
>>> |
```

온도를 입력받기 위해 input()함수를 통해 입력한 값을 c에 저장한다. 저장된 값이 문자열이므로 int()로 형변환을 처리해주어, 정수형태의 문자열을 정수형으로 변환해주어 정확계산식을 이용해 값을 저장해준다. 약식계산도 정해진 식으로 처리해주고, 이때도 문자열로 입력받은 값이라서 형변환을 통해서 c를 정수형으로 변환해서 계산을 해준다.

7. 다음에서 설명하는 함수 float.fromhex(str)를 이해하고 두 16진수 실수를 입력받아 사칙연산을 수행하는 프로그램을 작성하시오.

-실수 float에 속한 함수(정확한 명칭은 메소드라고 함)float.fromhex(str)는 16진수 형태의 문자열 str을 실수로 변환하는 함수

-즉 float.from('f')는 15, float.from('e.1')은 14.0625

```
#다음에서 설명하는 함수 float.fromhex(str)를 이해하고
#두 16진수 실수를 입력받아 사칙연산을 수행하는 프로그램을 작성하시오.
one=input('첫 번째 16진수 실수 입력\n')
two=input('두 번째 16진수 실수 입력\n')

print("실수1:",float.fromhex(one),"실수2:",float.fromhex(two)) #입력받은 수 출력

sum=float.fromhex(one) + float.fromhex(two)
minus=float.fromhex(one) - float.fromhex(two)
multiply=float.fromhex(one) * float.fromhex(two)
division=float.fromhex(one) / float.fromhex(two)

print("합:",sum)
print("차:",minus)
print("곱하기:",multiply)
print("나누기:",division)
```



```
첫 번째 16진수 실수 입력
f
두 번째 16진수 실수 입력
e.1
실수1: 15.0 실수2: 14.0625
합: 29.0625
차: 0.9375
곱하기: 210.9375
나누기: 1.0666666666666667
>>>
```

float.fromhex(str) . 하나하나씩 따져보면 float는 실수형의자료형을 뜻하고, fromhex는 매개변수(문자열)를 hex() 즉 16진수형태의 문자열을 실수로 변환해주는 함수이다. 각각 입력된 16진수 실수들을 사칙연산을 통해서 그 값을 출력해주는것이다. float.fromhex(str)함수설명이 나와있어서 생각보다 쉽게 처리할 수 있었던 문제이다.

# 3장 중요코딩.

```
#3-12 비트 배타적 논리합으로 ID 암호화
orgPwd = int(input('ID로 사용할 여덟자리의 정수를 입력하세요 >>'))
keyMask = 27182818 #키로 사용할 정수 하나를 저장
encPwd = orgPwd ^ keyMask #ID를 암호화시켜 저장
print('입력한 ID: %d' % orgPwd)
print('암호화해 저장된 ID: %d' % encPwd)
inPwd = int(input('로그인할 ID를 입력하세요 >>'))
result = encPwd ^ keyMask #키로 암호화 된 것을 복호화
print('로그인 성공: {}'.format(inPwd == result))
```



```
ID로 사용할 여덟자리의 정수를 입력하세요 >>87652877
입력한 ID: 87652877
암호화해 저장된 ID: 78101743
로그인할 ID를 입력하세요 >>87652877
로그인 성공: True
>>> |
```

비트 배타적 논리합 연산자 ^을 사용하여

encPwd = orgPwd(ID값) ^ (논리합 연산자) keyMask(키)

논리합 연산자를 사용하여 암호화 해준다.

result = encPwd(암호화된 값) ^ keyMask (키)

암호화된 값이 저장된 encPwd를 다시 논리합 연산자를 사용해서 복호화해준다.

이를 통해서 마지막 출력문구에 format이 사용되었는데

Print('어서오세요 {}'.format('사람이름')) 이런식으로 사용한다.

앞에서 정의된 문자열에서 {}대괄호가 들어간 부분에 format안에 들어있는 매개변수를 넣어주는것이다.

Format안에 정의된 inPwd == result는 해당하는 결과를 true,false로 반환해준다. 현재 결과화면에서는 로그인입력ID와 복호화된 ID가 같으므로 True를 반환한다.

# 3장 도전 프로그래밍 - 과제

1.한 줄의 문자열을 표준 입력으로 받아 문자열의 소속 문자를 참조할 범위를 출력하고 다시 첨자하나를 입력받아 문자열의 전체와 길이 그리고 참조 문자를 출력하는 프로그램을 작성하시오.

```
str = input("문자열 입력 :")
n=len(str)
print("참조할 첨자",n-n,"~",n-1)
a=input("참조할 첨자 입력 :")
```



```
문자열 입력 :Python is a good language!
참조할 첨자 0 ~ 25
참조할 문자열 : Python is a good language! , 길이: 26
참조 문자: g
>>> |
```

```
print("문자열:",str," , 길이:",n)
print("참조 문자:",str[int(a)])
```

변수에 표준입력함수 input()을 통해 문자열을 입력해준다.

그리고 len함수를 통해 표준입력으로 입력한 문자열의 길이를 얻어온다.

참조할 첨자를 입력받아서 참조문자를 출력해야 하므로 , 첨자를 입력받는다.

그리고 출력을 하기위해 입력받은 첨자가 정수형태의 문자열 이므로, int함수를 통해서 형변환을 하여 str[12]와 같이 입력되게 해주어서 참조된 값을 알 수 있다.

3.파이썬의 다음 첫 번째 철학을 저장한 후, 메소드 replace()를 사용해 두 번째 철학으로 다시 저장해 출력하는 프로그램을 작성하시오.

- Beautiful is better than ugly.
- Explicit is better than implicit.

```
str = "Beautiful is better than ugly."  
print(str)  
print("위 철학을 메소드 replace()를 사용해 다음 철학으로 다시 저장")  
change=str.replace("Beautiful","Explicit")  
change2=change.replace("ugly","implicit")  
print(change2)
```



```
Beautiful is better than ugly.  
위 철학을 메소드 replace()를 사용해 다음 철학으로 다시 저장  
Explicit is better than implicit.  
~~~
```

replace()함수를 이용해 replace('a','b') a의 내용을 b로 바꿔주는 함수이므로,  
1차적으로 change에서 Beautiful을 implicit으로 , 2번째로 change2에서 change에서 변한내용  
을 가져온채로 replace를 하여 바뀐정보로 저장해준다.

문자열은 수정될수 없는 특징이 있어서 str변수가 수정이 된게 아니므로, 각각 replace한값을  
다른 변수에 넣어서 값을 저장해서 출력했다.

5. 문자열의 split()메소드를 사용해 '14:21:45'와 같은 시각 정보를 표준 입력으로 받아 입력된 문자열을 출력하고 다시 시,분,초로 출력하는 프로그램을 작성하시오.

-입력 문자열 13:40:15 , '13시 40분 15초'로 출력

-hours,mins,secs=time.split(':')활용

```
sigak=input("시각 정보(16:30:15 형식) 입력 ")
hours,mins,secs = sigak.split(':')

print("입력 시각 정보:",sigak)
print(hours,"시",mins,"분",secs,"초")
```



```
시각 정보(16:30:15 형식) 입력 13:40:15
입력 시각 정보: 13:40:15
13 시 40 분 15 초
```

sigak이라는 변수에 input함수를 통해 입력을 받은 값을 저장해준다.

그리고 문제에서 주어진 hours,mins,secs =sigak.split(':') 사용.

split은 문자열을 매개변수를 기준으로 나눠준다. 그렇게 되면

hours= '13' , mins = '40' , secs = '15' 이렇게 값이 저장된다.

시 ,분, 초로 출력을 하라고 하였으니 print문으로 각각 저장된 값들을 넣어주고, 뒤에 "시" "분" "초"에 해당하는 문자열을 넣어주어서 결과에 맞게 출력해준다.

7. -7에서 0까지 10진수와 2진수, 8진수, 16진수를 다음과 같이 정형화해 출력하는 프로그램을 작성하시오.

- 2진수는 8비트의 정보 표현으로 출력
- 8진수와 16진수는 2진수의 비트 표현에서 8진수와 16진수로 출력

```
a = -7
mask = 0xff

print('10진수:{ } 2진수: { :8b} 8진수:{ :o} 16진수:{ :x} '.format(a, a&mask, a&mask, a&mask))
a = a+1
print('10진수:{ } 2진수: { :8b} 8진수:{ :o} 16진수:{ :x} '.format(a, a&mask, a&mask, a&mask))
a = a+1
print('10진수:{ } 2진수: { :8b} 8진수:{ :o} 16진수:{ :x} '.format(a, a&mask, a&mask, a&mask))
a = a+1
print('10진수:{ } 2진수: { :8b} 8진수:{ :o} 16진수:{ :x} '.format(a, a&mask, a&mask, a&mask))
a = a+1
print('10진수:{ } 2진수: { :8b} 8진수:{ :o} 16진수:{ :x} '.format(a, a&mask, a&mask, a&mask))
a = a+1
print('10진수:{ } 2진수: { :8b} 8진수:{ :o} 16진수:{ :x} '.format(a, a&mask, a&mask, a&mask))
a = a+1
print('10진수:{ } 2진수: { :8b} 8진수:{ :o} 16진수:{ :x} '.format(a, a&mask, a&mask, a&mask))
a = a+1
print('10진수:{ } 2진수: { :8b} 8진수:{ :o} 16진수:{ :x} '.format(a, a&mask, a&mask, a&mask))
a = a+1
print('10진수:{ } 2진수: { :8b} 8진수:{ :o} 16진수:{ :x} '.format(a, a&mask, a&mask, a&mask))
```



```
10진수:-7 2진수: 11111001 8진수:371 16진수:f9
10진수:-6 2진수: 11111010 8진수:372 16진수:fa
10진수:-5 2진수: 11111011 8진수:373 16진수:fb
10진수:-4 2진수: 11111100 8진수:374 16진수:fc
10진수:-3 2진수: 11111101 8진수:375 16진수:fd
10진수:-2 2진수: 11111110 8진수:376 16진수:fe
10진수:-1 2진수: 11111111 8진수:377 16진수:ff
10진수:0 2진수:      0 8진수:0 16진수:0
>>> |
```

개인적으로 어려웠던 문제다. 비트 연산의 관한 문제에서는 한없이 작아지는것 같다. 교재 102쪽에서 &연산자와 해당하는 예시를 통해 기본적인 사용법을 익히고, 출력방식으로 104쪽에 내용을 통해 이해했다.



# 4장 중요코딩.

```
#전형적인 구구단 출력
for i in range(2,10):
    for j in range(1,10):
        print('%d * %d = %2d' % (i,j,i*j),end= ' ')
    print()
```



```
2 * 1 = 2 2 * 2 = 4 2 * 3 = 6 2 * 4 = 8 2 * 5 = 10 2 * 6 = 12 2 * 7 = 14 2 * 8 = 16 2 * 9 = 18
3 * 1 = 3 3 * 2 = 6 3 * 3 = 9 3 * 4 = 12 3 * 5 = 15 3 * 6 = 18 3 * 7 = 21 3 * 8 = 24 3 * 9 = 27
4 * 1 = 4 4 * 2 = 8 4 * 3 = 12 4 * 4 = 16 4 * 5 = 20 4 * 6 = 24 4 * 7 = 28 4 * 8 = 32 4 * 9 = 36
5 * 1 = 5 5 * 2 = 10 5 * 3 = 15 5 * 4 = 20 5 * 5 = 25 5 * 6 = 30 5 * 7 = 35 5 * 8 = 40 5 * 9 = 45
6 * 1 = 6 6 * 2 = 12 6 * 3 = 18 6 * 4 = 24 6 * 5 = 30 6 * 6 = 36 6 * 7 = 42 6 * 8 = 48 6 * 9 = 54
7 * 1 = 7 7 * 2 = 14 7 * 3 = 21 7 * 4 = 28 7 * 5 = 35 7 * 6 = 42 7 * 7 = 49 7 * 8 = 56 7 * 9 = 63
8 * 1 = 8 8 * 2 = 16 8 * 3 = 24 8 * 4 = 32 8 * 5 = 40 8 * 6 = 48 8 * 7 = 56 8 * 8 = 64 8 * 9 = 72
9 * 1 = 9 9 * 2 = 18 9 * 3 = 27 9 * 4 = 36 9 * 5 = 45 9 * 6 = 54 9 * 7 = 63 9 * 8 = 72 9 * 9 = 81
```

4장에서는 반복문과 조건문을 배운다. 이 문제도 중간고사에 나올법한 코딩으로 설명해주셨다.

구구단의 경우 for문을 이해하기 가장 좋은 역할을 하는것 같다.

먼저 for문의 사용방법이 우리가 알던 방법과 다르다.

for 변수 in(키워드) 변수범위: (콜론)

쉽게 이해하면 i변수를 range범위 (n에서 n-1)까지로 지정을 해준다고 생각하면 된다.

# 4장 도전 프로그래밍 - 과제

1. 다음을 참고해 월(month)을 표준 입력으로 받아 계절을 출력하는 프로그램을 작성하시오.

- 봄: 4, 5월 여름: 6, 7, 8월 가을: 9, 10월 겨울: 11, 12, 1, 2, 3월

```
month=int(input('월을 입력하세요'))
if month<=3:
    print("{}월 {}".format(month,'겨울'))
elif month<=5:
    print("{}월 {}".format(month,'봄'))
elif month<=8:
    print("{}월 {}".format(month,'여름'))
elif month<=10:
    print("{}월 {}".format(month,'가을'))
elif month<13:
    print("{}월 {}".format(month,'겨울'))
else:
    print("월 입력이 잘못되었습니다 확인해주세요.")
```



월을 입력하세요 4  
4월 봄  
~~~

월을 입력하세요 9  
9월 가을  
>>>

월을 입력하세요 7  
7월 여름  
~~~

월을 입력하세요 12  
12월 겨울

월을 입력하세요 13  
월 입력이 잘못되었습니다 확인해주세요.  
...

Input함수를 이용해 입력받은값을 통해서 조건문 if를 쓰는 문제이다.

기본적으로 다른 프로그래밍 언어와 달리 if문을 사용할때 if 조건문 콜론(:)이 사용하는 문법이다. 그리고 else if로 그렇지 않은경우 다른 조건을 수행하게 했었는데 파이썬에서는 se가빠진 elif를 통해서 else if와 같은 역할을 수행한다.

# 4장 도전 프로그래밍 - 과제

3. 다음을 참고해 1에서 99까지의 난수인 임의의 정수 3개를 대상으로 가장 큰 정수를 출력하는 프로그램을 작성하시오.

- from random import randint

```
from random import randint

a=randint(1,100)
b=randint(1,100)
c=randint(1,100)

print("{} {} {}중에서 최대 : {}".format(a,b,c,max(a,b,c)))

num1=randint(1,100)
num2=randint(1,100)
num3=randint(1,100)
|
if num1>num2:
    if num1>num3:
        max=num1

elif num2>num3:
    if num2>num1:
        max=num2
else:
    max=num3

print("{} {} {}중에서 최대 : {}".format(num1,num2,num3,max))
```



```
63 37 72중에서 최대 : 72
18 54 79중에서 최대 : 79
>>>
```

참고한 내용이 바로 랜덤함수를 사용하기 위해 import하는 문장이다.

그리하여 사용한 randint함수로 임의의정수를 받아와서 각 변수에 저장해줬다. 첫번째 방법은 조건문을 사용하지 않고 max함수를 이용해서 최대 최소값을 바로 출력한 방법이고, 두번째는 4장에서 배운 조건문을 이용하여 최대값을 max에 저장해주는 방법으로 해서 2가지 방법으로 풀어 보았다.

# 4장 도전 프로그래밍 - 과제

5. 다음을 참고해 섭씨 온도 20도에서 41도까지 3도씩 증가하면서 화씨로 변환해 출력하는 프로그램을 작성하시오.

-섭씨온도에서 화씨 온도로의 변환 공식은 오른쪽 공식을 사용 ( $F = C * 9/5 + 32$ )

-약식 계산으로(섭씨 온도 \* 2 + 30)으로 계산

-정식 계산과 약식 계산의 차이를 내장 함수 abs(인자)로 출력

```
for i in range(20, 42, 3):  
    h = i * 9/5 + 32 #화씨  
    h1 = i * 2 + 30 #화씨(약식)  
    c = h - h1 #화씨와 화씨약식의 차이  
    print("섭씨: {} 화씨: {}, 화씨(약식):{} 차이: {:.2f}".format(i, h, h1, abs(c)))
```



```
섭씨 : 20 화씨 : 68.0, 화씨(약식) : 70 차이 : 2.00  
섭씨 : 23 화씨 : 73.4, 화씨(약식) : 76 차이 : 2.60  
섭씨 : 26 화씨 : 78.8, 화씨(약식) : 82 차이 : 3.20  
섭씨 : 29 화씨 : 84.2, 화씨(약식) : 88 차이 : 3.80  
섭씨 : 32 화씨 : 89.6, 화씨(약식) : 94 차이 : 4.40  
섭씨 : 35 화씨 : 95.0, 화씨(약식) : 100 차이 : 5.00  
섭씨 : 38 화씨 : 100.4, 화씨(약식) : 106 차이 : 5.60  
섭씨 : 41 화씨 : 105.8, 화씨(약식) : 112 차이 : 6.20
```

4장에서 배우게된 파이썬에서의 반복문 사용. 문제를 보면 3도씩 증가를 하고, 화씨로 변환해 출력을 해야한다. 그러므로 중복되는 내용을 수행하는것이므로 반복문을 쓰는 문제이다.

기본적인 파이썬의 반복문 사용은 for 변수 in 시퀀스 : (콜론)이고 반복으로 실행되는 문장은 반드시 들여쓰기를 사용해서 작성해야지 반복문이 실행이된다. 이때 시퀀스에 range함수가 사용되었는데, range함수는 범위를 나타낸다. 흔히 우리가 사용했던 for(int i=0; i<=10; i++)의 문법처럼 밑줄친 부분에 속한 내용이라고 생각하면 이해가 편하다. 물론 range함수는 첫번째 매개변수가 n 시작이고 두번째 매개변수는 n-1까지의 범위이다. 세번째 매개변수로 간격을 지정해주어서 문제에서 제시한 3도씩 증가한다는 내용에 맞게 수행된다.

여기서 in은 파이썬의 키워드로 멤버의소속을 알수있는 멤버십연산을 해준다. for문에서는 편하게 변수에 레인지 범위를 넣어주는방식으로 이해를 하니 더 쉽게 이해 할 수 있었다.

그리고 내장함수 abs는 인자 즉 매개변수로 받아온 값의 절대값을 출력해 주는 함수이다.

# 4장 도전 프로그래밍 - 과제

7. 정수 1개를 표준 입력으로 받아 소수인지를 판별하는 프로그램을 작성하시오.

- 2이상의 정수를 입력해 입력한 정수의 판별.

```
num = int(input("소수(prime number)인지를 판별한 2 이상의 정수 입력"))
check = 0

for i in range(1,num+1):
    if num%i==0:
        check=check+1

if check==2:
    print("정수 {}는 소수이다.".format(num))
else:
    print("정수 {}는 소수가아니다.".format(num))
```



소수(prime number)인지를 판별한 2 이상의 정수 입력11  
정수 11는 소수이다.

소수(prime number)인지를 판별한 2 이상의 정수 입력4  
정수 4는 소수가아니다.

Prime number 즉 소수는, 1과 자기자신으로만 나누어지는 수를 뜻한다.

그러므로 input으로 입력받은 정수를 반복문을 통해 1부터 자기자신까지 나누어서 나머지가 0이되면, check 변수에 +1을 해주었다. 이로 인해 check가 2가 되면 1,자기자신으로만 나누어지는 수이므로 소수인게 판별이되고, 그렇지 않을 경우에는 소수가 아니게 되는 것이다. Range함수는 n부터 n-1까지의 내용이므로 자기자신까지 나누기 위해 num+1을 해주었다.

# 5장 중요코딩.

#5-11 간단한 리스트 컴프리헨션

#for문으로 리스트 생성

```
a = []  
for i in range(10):  
    a.append(i)  
print("for문:", a)
```

#컴프리헨션으로 리스트 생성

```
seq = [i for i in range(10)]  
print("컴프리헨션:", seq)
```

#for문으로 리스트 생성

```
s = []  
for i in range(10):  
    if i%2 == 1:  
        s.append(i**2)  
print("for문", s)
```

#컴프리헨션으로 리스트 생성

```
squares = [i**2 for i in range(10) if i%2 == 1]  
print("컴프리헨션:", squares)
```



```
for문: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]  
컴프리헨션: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]  
for문 [1, 9, 25, 49, 81]  
컴프리헨션: [1, 9, 25, 49, 81]  
...
```

리스트를 만드는 방법중에 하나인 컴프리헨션을 이용한 방법이다.

컴프리헨션방법은 for문으로 만들때보다 더 간결하게 리스트를 만들어주는 방법으로 사용법은 변수(리스트) = [항목연산식 for 항목 in 시퀀스(범위) if(조건식이 있다면) ] 이러한 방식으로 만들어진다. For문을 사용해서 만든 리스트에서 하나씩 append로 추가해주던 구문이 없어도 알아서 추가가 되는 것이다. 두번째코드로 만든리스트를 컴프리헨션으로 만들면

squares = [i\*\*2 (항목연산식 어떤걸추가할지) for i in range(10) if i%2 ==1]

소스를 보면 for문에서 작동하던내용을 다 리스트[]괄호안에 넣어준 내용이다. 코드의 길이만 봐도 컴프리헨션 방식이 훨씬 간결하다는걸 알 수 있고, 현재 흔히 하고 있는 실수중에 하나인 for문과 if문사용시 붙이는 콜론을 붙이지 않아도 되니, 실수도 많이 줄어 들 수 있을것 같다.

```
inta = 1
tupa = 1,
print(type(tupa))
print(type(inta))
```



```
<class 'tuple'>
<class 'int'>
```

튜플: 리스트와 같이 항목의 나열인 시퀀스이다.

리스트와의 차이점은 항목의 순서, 내용(값)을 수정할 수 없다는 것이다.

위의 있는 예제는 교수님께서 꼭 주의하라고 하셨던 부분이다.

항목이 1개인 튜플을 만들 경우 ()괄호를 생략해서 만들 수 있다.

단, 내용뒤의 (, ) 콤마를 붙여야지만 튜플이 되고, 붙이지 않으면 튜플이 아닌 입력한 자료형을 따라가게 된다.

type함수를 통해 확인을 해보면 확실히 알 수 있다. 물론 type말고 그냥 변수를 출력해봐도 알 수 있지만, inta는 값뒤에 콤마를 붙이지않아서 정수형이 되었고 tupa는 값뒤에 콤마를 붙여서 '튜플'로 만들어 진것을 알 수 있다.

# 5장 도전 프로그래밍 - 과제

1. 1에서 99까지의 난수 10개로 리스트를 만든 후 리스트와 정렬된 리스트 그리고 내림차순으로 정렬된 역순 리스트를 출력하는 프로그램을 작성하시오.

```
from random import randint
```

```
numlist = []
```

```
for i in range(10):  
    num = randint(1,100)  
    numlist.append(num)
```

```
print("리스트:", numlist)
```

```
numlist.sort()
```

```
print("정렬된 리스트:", numlist)
```

```
numlist.sort(reverse=True)
```

```
print("역순 리스트:", numlist)
```



리스트: [45, 66, 97, 13, 47, 33, 32, 65, 25, 22]

정렬된 리스트: [13, 22, 25, 32, 33, 45, 47, 65, 66, 97]

역순 리스트: [97, 66, 65, 47, 45, 33, 32, 25, 22, 13]

...

리스트 : 파이썬의 핵심이라고 불리는 자료형이다. 리스트는 쉽게 말하면 항목의 나열이다. 출력시에는 콤마로 구분하여 각각 저장된 값들을 구분할 수 있고, 문자열을 항목참조했던것처럼 리스트도 항목참조가 가능하고 리스트에 저장된 값은 수정이 가능하다.

이 문제에서 원하는 10개의 난수를 발생시키기 위해 randint를 사용하였고, 이 값들을 리스트에 저장하기 위해 비어있는 리스트 numlist를 생성하고, 리스트의 값들은 반복문을 통해 append 함수로 리스트에 각각 값을 추가해주었다.

그리고 리스트를 출력하여 저장된 값들을 보여주었고, 각각 정렬, 정렬된 역순리스트를 출력하기 위해 리스트의 sort()메소드를 이용해 오름차순으로 정렬을 해주고, 그 이후 역순으로 출력하기 위해 sort()매개변수로 reverse=True를 넣어주어 역순으로 출력해주었다. 정렬로는 sort()메소드 말고, sorted()내장함수를 이용해서도 가능하다.

사용방법으로는 정렬된리스트를 저장하기 위한 새로운 변수 newlist = sorted(정렬할리스트)방식으로 사용하면 오름차순이 되고 sorted(정렬할리스트,reverse=True)를 사용하면 역순출력이 된다.



# 5장 도전 프로그래밍 - 과제

3.문자열 'HelloPython!'으로 리스트를 만들어 부분 리스트 참조인 슬라이스를 위한 3개의 정수를 표준 입력으로 받은 후 슬라이스 결과를 출력하는 프로그램을 작성하시오.

- 계속 입출력을 반복하여, 000을 입력하면 종료
- 슬라이스를 위해 첨자를 문자열의 위아래로 출력

```
slist = list('HelloPython!')

print("+ 0 1 2 3 4 5 6 7 8 9 10 11")
print(slist)
print("- 12 11 10 9 8 7 6 5 4 3 2 1")

more = True
while more:
    a,b,c=(input("슬라이스[?:?:?] 3개 입력 >>")).split()
    if int(a)==0 and int(b)==0 and int(c)==0:
        more =False
    else:
        print(slist[int(a):int(b):int(c)])

else:
    print('종료'.center(15, '*'))
```



```
+ 0 1 2 3 4 5 6 7 8 9 10 11
['H', 'e', 'l', 'l', 'o', 'P', 'y', 't', 'h', 'o', 'n', '!']
- 12 11 10 9 8 7 6 5 4 3 2 1
슬라이스[?:?:?] 3개 입력 >>3 10 2
['l', 'P', 't', 'o']
슬라이스[?:?:?] 3개 입력 >>0 0 0
*****종료*****
```

리스트에 문자열 helloPython! 저장.

각첨자를 프린트함수로 보여줌.

그리고 while문(반복)을 통해서 무한루프를 돌림.

단 , 프로그램에서 말한 000을 입력하면 종료가

되야하므로 if문(조건문)을 이용해서

입력된 a,b,c의 값이 각각 0일 경우에는 와일문을 돌리는 조건인 more의 값을 False로 바꿔서 반복문을 빠져나오게 만들었다.

이 문제는 a,b,c를 int()함수로 형변환해서 input()함수로 값을 받아오려고 하니 오류가 발생했다.

이 부분은 검색을 해보니 , map이라는 함수를 사용하여 해결할 수 있다고는 했으나, 아직 다루지 않은 함수여서 그냥 입력받은값을 int()로 따로 바꿔주었다.

# 5장 도전 프로그래밍 - 과제

5.다음 리스트 sports와 num을 활용해 스포츠 종목과 팀원 수가 번갈아 나오는 리스트를 만든 후 같이 출력하는 프로그램을 작성하시오.

```
sports = ['축구', '야구', '농구', '배구']
```

```
num = [11, 9, 5, 6]
```

-리스트 sports에 insert()메소드로 직접 팀원 수를 삽입

-리스트 sports의 홀수 참조에 빈 문자 ""을 insert()메소드로 삽입

-위 결과 리스트에서 슬라이스 sports[1::2]에 num을 대입

이 문제는 뭔가 이해를 잘 못 한 문제 같다. 제 생각을 말하자면, insert메소드를 통해 삽입이 되면 리스트에 그 항목이 들어갑니다. 그래서 처음에 해당하는 조건을 수행하면 sports는 처음에 나온 실행결과처럼 번갈아서 출력이 된 상태가 되었는데, 이 상태에서 홀수 참조에 빈문자 ""을 삽입한다면 2번째 결과화면처럼 나오지 않고, 각 홀수번째 첨자위치에 ex) '축구', ", 11 이런식으로 놓여지는게 아닌가 생각합니다. 그래서 저는 sports1을 새로 만들어서 결과화면처럼 나오게 해주었습니다.

마지막 결과화면에서는 슬라이스를 이용해서 ""가 저장된 sports1의 처음부터 끝까지 2칸의 간격을 둔 곳에 num리스트의 값을 처음부터 끝까지 넣어줌으로 결과화면처럼 나오게 해주었습니다.

```
sports = ['축구', '야구', '농구', '배구']  
sports1 = ['축구', '야구', '농구', '배구']
```

```
num=[11,9,5,6]
```

```
sports.insert(1,num[0])  
sports.insert(3,num[1])  
sports.insert(5,num[2])  
sports.insert(7,num[3])
```


```
print("메소드 insert()로 팀원 수 삽입\n",sports)
```

```
sports1.insert(1,'')  
sports1.insert(3,'')  
sports1.insert(5,'')  
sports1.insert(7,'')
```

```
print("메소드 insert()로 팀원 수 삽입\n",sports1)
```

```
sports1[1::2]=num[:]
```

```
print("슬라이스 sports[1::2]에 num을 대입\n",sports1)
```



```
메소드 insert()로 팀원 수 삽입  
['축구', 11, '야구', 9, '농구', 5, '배구', 6]  
메소드 insert()로 팀원 수 삽입  
['축구', '', '야구', '', '농구', '', '배구', '']  
슬라이스 sports[1::2]에 num을 대입  
['축구', 11, '야구', 9, '농구', 5, '배구', 6]
```

# 5장 도전 프로그래밍 - 과제

7.위와 동일한 중첩된 리스트에서 리스트 컴프리헨션을 활용해 행과 열이 바뀐 형태의 리스트를 새로 만들고, 이 변환된 리스트를 다음과 같이 출력하는 프로그램을 작성하시오.

```
M=[[1,2],[3,4],[5,6],[7,8]]
```

다음 리스트 컴프리헨션을 활용

```
transpose = [[row[i] for row in m] for i in range(len(m[0]))]  
print("트랜스포즈를 컴프리헨션으로 만들어 그대로 출력")
```

```
m= [[1,2],[3,4],[5,6],[7,8]]
```

```
transpose = [[row[i] for row in m] for i in range(len(m[0]))]
```

```
print(transpose)
```

```
print("트랜스포즈를 for문으로 출력")
```

```
for i in range(len(transpose)):  
    for j in range(len(m)):  
        print(int(transpose[i][j]),end = ' ')  
    print()
```



```
트랜스포즈를 컴프리헨션으로 만들어 그대로 출력  
[[1, 3, 5, 7], [2, 4, 6, 8]]  
트랜스포즈를 for문으로 출력  
1 3 5 7  
2 4 6 8
```

5장에서 중요하다고 하신 컴프리헨션 사용문제이다. 리스트의 항목이 리스트로 구성된 m은 현재 4의 길이를 가지고 있다. Transpose에서 m[0]을 범위로 잡으면, m[0] = [1,2]으로 2개의 항목이라 길이가 2가 된다. 전체적으로 transpose는 길이가 2인 리스트가 되고, 그 안의 항목들은 m의 길이 4 즉 transpose는 [[4개항목],[4개항목]] 이러한 형태로 리스트의 총 길이는 2이고 리스트의 각 항목은 길이가 4개인 리스트가 만들어진다. 그 리하여 이를 출력하려면 리스트를 참조해야한다. 이때 이차원배열을 떠올려서 각 항목들을 참조했다. 그리고 출력되는 항목들을 결과화면처럼 보여주기 위해서, int메소드를 사용해 정수형으로 출력이되게 형변환 해주었다.그리고 end = ' '를 이용해 출력된 숫자 다음에 공백으로 띄워주었고, m의길이 만큼 출력을 하고, 첫번째 포문에 print()를 이용해 줄을 바꿔서 출력해주었다. 리스트도 중첩되었고, 반복문도 중첩이 되었다보니, 헷갈리는 부분이 많았다. 아마 문제에서 리스트컴프리헨션을 알려주지 않았다면, 제대로 풀었는지 의문이 든다.

# 7번문제 풀어써보기.

7번문제 활용.


```
m = [[1,2],[3,4],[5,6],[7,8]]
```

1

```
transpose = []  
for i in range(len(m[0])):  
    transpose.append([row[i] for row in m])  
  
print("for문과 리스트컴프리헨션:", transpose)
```

```
tr2=[]  
for i in range(len(m[0])):  
    b = []  
    for row in m:  
        b.append(row[i])  
    tr2.append(b)  
  
print("이중 포문:", tr2)
```

2



```
for문과 리스트컴프리헨션: [[1, 3, 5, 7], [2, 4, 6, 8]]  
이중 포문: [[1, 3, 5, 7], [2, 4, 6, 8]]  
>>>
```

7번문제에서 사용한 리스트 컴프리헨션을 각각 풀어서 써보았다.

1번방법은 컴프리헨션을 사용하지만, 문제에서는 컴프리헨션 안에 for의 내용이 2개가 들어있던것을 하나는 for문을 통해 나타내서 풀어 쓴것이다. Transpose에 중첩된 리스트형태로 넣어주기 위해 컴프리헨션을 []로 묶어서 삽입해주면 중첩된 형태로 나타나게 된다.

2번방법은 컴프리헨션을 사용하지 않은채, 이중포문을 이용하여 각각의 값을 넣어준 방법이다. 이렇게 풀어쓰면서 느낀것은 역시 컴프리헨션 방법이 코드의 길이도 간결하고 보기 좋다는것이다. 2번방법에서는 중첩된 리스트로 값을 넣어줘야 하는것을 2개의 리스트를 생성해서 값을 넣어주는 방식으로 해결했다. 리스트에 리스트의 값을 삽입하는것이므로 중첩된 리스트의 형태로 나타나게된다.

# 6장 중요코딩.

```
groupnumber = {'잔나비':5, '트와이스':9, '블랙핑크':4, '방탄소년단':7}  
print(groupnumber)  
print(type(groupnumber))
```

```
{'잔나비': 5, '트와이스': 9, '블랙핑크': 4, '방탄소년단': 7}  
<class 'dict'>
```

6장에서는 새로운 자료구조인 딕셔너리를 배웠다.

딕셔너리는 키와 값의 쌍을 항목으로 관리하는 자료구조로써 dict타입이 바로 딕셔너리이다.

사용법은 grade = {'c언어':'C' , 'python':'B'}

와 같이 {}로 묶여있는 형태이고, 안에 구성된 항목에서 'python'은 키, B는 밸류값이 된다.

dict에서도 항목을 참조하는 방법이 있다. 바로 키값을 이용해서 값을 알아오는 방법인데

print(groupnumber['잔나비']) '잔나비'라는 키값으로 참조하면, 키값의 해당하는 밸류값인 5가 나온다. 이와 같이 키를 통해 밸류값을 얻어올수 있는데, 밸류값은 수정이 가능하지만 밸류값을 알려줄 수 있는 키값은 수정이 불가능하다. 그러므로 수정이 불가능한 튜플은 키값으로 사용할 수 있고, 리스트는 키로 사용할수가 없다.

위의 방식 말고도 딕셔너리 메소드 get을 이용해 얻을수 있다.

print(groupnumber.get('트와이스'))처럼 사용하며, get('키')로 해당하는 밸류인 9를 참조할 수 있다.

매개변수로 get('키',키가없을때 반환값)형태로도 사용이 가능하며, 두번째 매개변수 키가없을때 반환값을 따로 지정하지 않으면 None이라고 아무것도 없다는것을 반환해준다.

리스트는 [] 대괄호 , 튜플은 () 소괄호 , 딕셔너리는 중괄호로 묶인다.

# 6장 도전 프로그래밍 - 과제

1. 다음 개인 정보를 딕셔너리에 저장하고 출력하는 프로그램을 작성하시오.

키 - 이름,전화번호,성별,나이,대학교

```
mydict = {'이름': '심기현', '전화번호': '010-9159-4749', '성별': '남자',  
          '나이': 24, '대학교': '동양미래대학교'}
```

```
for dic in mydict.items():  
    print('{:}: {}'.format(*dic), end = ' ')  
    print()
```



```
이름: 심기현  
전화번호: 010-9159-4749  
성별: 남자  
나이: 24  
대학교: 동양미래대학교  
...
```

문제에서 원하는 대로 딕셔너리에 값들을 저장하였고, items메소드를 이용해 (키,밸류)의 튜플 형태를 받아오고 format메소드로 {} 안에 각각 값을 넣어주었다. 이때 dic앞에 \* 에스터리스크를 사용함으로 튜플을 한개씩 나눠주지 않고도 자동으로 알아서 튜플에서 항목이 2개면 2개로 나눠주어 괄호안에 매칭시켜주어서 결과와 같이 화면이 나오게된다.

# 6장 도전 프로그래밍 - 과제

3. 다음 책에 대한 정보를 저장한 딕셔너리에 대해 다음과 같이 출력하는 프로그램을 작성하시오.

```
books = {'파이썬 개론': ['강환수'],  
         'Perfect C': ['강환수', '이동규'],  
         '컴퓨터 개론': ['강환수', '조진형', '신용현']}
```

```
for key in books.keys():  
    print('책 이름:%s, 저자:%s'%(key, books[key]))
```



```
책 이름:파이썬 개론, 저자:['강환수']  
책 이름:Perfect C, 저자:['강환수', '이동규']  
책 이름:컴퓨터 개론, 저자:['강환수', '조진형', '신용현']
```

처음 시도 했을때, 출력이 책에 나와있는 형태처럼 되지 않았다. 책에서는 저자에 표시되는 내용이 리스트문자열로 된 value값이 '[' 대괄호와, 따옴표 없이 value값만 출력이 된 상태였다.

나는 왜 출력형태를 고칠수가 없는지 생각을 해봐도 도통 해결이 되지 않는 상태다.

# 6장 도전 프로그래밍 - 과제

```
books = {'파이썬 개론': ['강환수'],  
         'Perfect C': ['강환수', '이동규'],  
         '컴퓨터 개론': ['강환수', '조진형', '신용현']}
```

```
|  
a1 = books['파이썬 개론']  
a2 = books['Perfect C']  
a3 = books['컴퓨터 개론']
```

```
for key in books.keys():  
    print('책이름: {}, '.format(key), end = ' ')  
    if key=='파이썬 개론':  
        print('저자:', a1[0])  
    elif key=='Perfect C':  
        print('저자:', a2[0], a2[1])  
    elif key=='컴퓨터 개론':  
        print('저자:', a3[0], a3[1], a3[2])
```

책이름: 파이썬 개론, 저자: 강환수  
책이름: Perfect C, 저자: 강환수 이동규  
책이름: 컴퓨터 개론, 저자: 강환수 조진형 신용현  
~~~

어찌저찌해서 실행결과에 맞게 출력을 했지만, 더 좋은 방법으로 해결하는 방법이 있을 것 같긴 하다.

내가 해결한 방법은 a1, a2, a3에 각각 books의 키를 이용해 value값을 참조하는 방법으로 밸류값을 저장한 후, 그 값들이 리스트형태로 되어있으니, 리스트에서 항목을 참조했던 첨자를 이용해서 결과를 출력한 방법이다. 분명 이것보다 더 간단하게 하는 방법이 존재할 것 같다.

하지만 이게 현재의 능력으로는 최선 인것 같다.



# 마치며..

첫 시작 당시에는 되게 설레기만 하였던 프로그래밍이 끝에 와서는 너무 버거워졌다. 한문제를 풀때 마다 이해가 안되는 요소들이 증가하고, 왜 이렇게 됐지? 라는 생각을 해도 통 해결이 되지 않는다. 물론 이 부분은 파이썬 과목에서만 그런게 아니라 현재 모든 과목에서 이렇게 느끼고 있다.

군휴학을 하기 전 1학년 당시에는 그래도 지금보다는 훨씬 더 프로그래밍 즉 코딩을 하는데에 자신감도 있었고, 문제 해결에 대해서도 나름 잘했었다. 허나, 복학을 한 지금은 머리가 백지가 된 것 같다. 무언가 한 과목을 좀 더 깊게 해보고 싶으면, 다른과목의 과제가 계속해서 밀린다. 제출기간내에 과제를 제출해야되는 불안감에 결국 이해를 못했던 부분은 다시 미뤄두고, 과제를 해결하기 바쁘다. 과제 해결도 빠르게 해결하지 못하고 3시간 4시간씩 점점 걸리는 시간이 늘어나고 있다. 군휴학 기간동안 악화된 허리건강상태도 과제를 해결하느라 장시간 앉아 있다 보니 몸 상태도 더 안좋아졌다. 무엇인가를 해결하려고 할 때 마다 점점 의욕이 없어지고, 몸도 안좋아지고 하다보니 요즘은 내가 현재 선택한 전공과 맞지 않는 사람이라는 생각도 많이 들게 된다.

포트폴리오의 내용이 전체적으로 부실하지만, 저로써는 현재 공부했던 내용이 도전프로그래밍 과제를 제외하고는 없기 때문에, 거짓된 내용으로 쓰고 싶지 않았습니다. 긴 글 읽어주셔서 감사합니다.