파이썬 포트폴리오

2 - C

20160771 서시진

목차

강 의 계 획 서

2020 1학기	전공		학부	
과 목 명	파이썬프로그래밍(2019009-PC)			

강의실 과 강의시간	:1(3-217),2(3-217),3(3-217)	학점	3
교과분류	이론/실습	시수	3

담당 교수	

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.

강 의 계 획 서

1	[개강일(3/16)]
	교과목 소개 및 강의 계획 1장 파이썬 언어의 개요와 첫 프로그래밍
목표및 내용	• 파이썬 언어란 무엇인지 이해하고 이 언어가 인기 있는 이유를 설명할 수 있다. • 파이썬 개발 도구를 설치해 프로그램을 구현할 수 있다. • 파이썬의 특징과 활용 분야를 설명할 수 있다.
미리읽어오기	교재 1장, 파이썬 개발환경 설치 파이썬 IDLE
과제,시험,기타	도전 프로그래밍
2 주차	[2주]
학습주제	2장 파이썬 프로그래밍을 위한 기초 다지기
목표및 내용	• 파이썬의 재료인 문자열과 수에 대해 이해하고 코드로 구현할 수 있다. • 변수를 이해하고 다양한 대입 연산자를 활용할 수 있다. • 표준 입력으로 문자열을 입력받은 후 원하는 자료로 변환해 활용할 수 있다. • 파이썬 IDLE을 활용할 수 있다.
미리읽어오기	교재 2장 리터럴과 변수의 이해 아나콘다의 주피터 노트북
과제,시험,기타	도전 프로그래밍
3 주차	[3주]
학습주제	3장 일상에서 활용되는 문자열과 논리 연산
목표및 내용	• 문자열에서 문자나 부분 문자열을 반환하는 여러 방법을 구현할 수 있다. • 문자열 객체에 소속된 다양한 메소드를 이해하고 활용할 수 있다. • 논리 값을 이해하고 다양한 연산을 사용해 실생활에서의 표현에 활용할 수 있다. • 아나콘다의 주피터 노트북을 활용할 수 있다.
미리읽어오기	교재 3장 문자열과 논리연산 파이참(pycharm)
과제,시험,기타	도전 프로그래밍
4 주차	[4주]
학습주제	4장 일상생활과 비유되는 조건과 반복
목표및 내용	• 조건에 따라 하나를 결정하는 if문을 구현할 수 있다. • 반복을 수행하는 while문과 for문을 구현할 수 있다. • 임의의 수인 난수를 이해하고 반복을 제어하는 break문과 continue문을 활용할 수 있다. • 파이참(pycharm)을 활용할 수 있다.
미리읽어오기	교재 4장 조건과 반복
과제,시험,기타	도전 프로그래밍

강 의 계 획 서

5	[5주]
	5장 항목의 나열인 리스트와 튜플
목표및 내용	• 다양한 종류의 항목을 쉽게 나열하는 리스트를 구현할 수 있다. • 리스트에서 부분 참조 방법, 이를 이용한 수정, 리스트 연결, 삽입과 삭제 그리고 리스트 컴프리헨션 등을 구현할 수 있다. • 수정할 수 없는 다양한 종류의 항목 나열을 쉽게 처리하는 튜플을 구현할 수 있다.
미리읽어오기	교재 5장 배열과 리스트
과제,시험,기타	도전 프로그래밍
6 주차	**[6주]**
학습주제	6장 키와 값의 나열인 딕셔너리와 중복을 불허하는 집합
목표및 내용	• 키와 값의 쌍인 항목을 관리하는 딕셔너리를 생성하고 수정하는 방법을 이해하고, 다양한 방법으로 딕셔너리를 구현할 수 있다. • 집합의 특징을 이해하고, 합집합 등과 같은 다양한 집합의 연산을 구현할 수 있다. • 내장 함수 zip( )과 enumerate( ), 시퀀스 간의 변환을 이해하고, 구현할 수 있다.
미리읽어오기	교재 6장 집합
과제,시험,기타	도전 프로그래밍
7 주차	**[7주]**
학습주제	7장 특정 기능을 수행하는 사용자 정의 함수와 내장 함수
목표및 내용	• 함수의 내용과 필요성을 이해하고 함수를 직접 정의해 호출할 수 있다. • 인자의 기본 이해와 기본값 지정, 가변 인수와 키워드 인수를 활용할 수 있다. • 간편한 람다 함수와 표준 설치된 내장 함수를 사용할 수 있다.
미리읽어오기	교재 7장 함수의 정의와 호출
과제,시험,기타	도전 프로그래밍
8 주차	**[중간고사]**
학습주제	- 직무수행능력평가 1차(중간고사)
목표및 내용	직무수행능력평가, 서술형 평가
미리읽어오기	교재 1장에서 7장까지
과제,시험,기타	
9 주차	**[9주]**
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	

강 의 계 획 서

10	[10주]
	9장 라이브러리 활용을 위한 모듈과 패키지
목표및 내용	• 표준 모듈을 이해하고 사용자 정의 모듈도 직접 구현해 사용할 수 있다. • 표준 모듈인 turtle을 사용해 기본적인 도형을 그릴 수 있다. • 써드파티 모듈 numpy와 matplotlib 등을 설치해 활용할 수 있다.
미리읽어오기	교재 9장
과제,시험,기타	도전 프로그래밍
11 주차	[11주]
학습주제	10장 그래픽 사용자 인터페이스 Tkinter와 Pygame
목표및 내용	• GUI를 이해하고 GUI 표준 모듈인 Tkinter를 사용해 필요한 위젯을 구성하고 윈도를 생성할 수 있다. • 이벤트 처리를 이해하고 Tkinter에서 이벤트 처리를 구현할 수 있다. • 써드파티 GUI 모듈인 pygame을 설치해 기본적인 윈도를 구현할 수 있다.
미리읽어오기	교재 10장
과제,시험,기타	도전 프로그래밍
12 주차	[12주]
학습주제	11장 실행 오류 및 파일을 다루는 예외 처리와 파일 입출력
목표및 내용	• 예외 처리의 필요성을 이해하고 try except 구문을 사용해 예외를 처리할 수 있다. • 프로그램에서 파일을 생성하는 필요성을 이해하고 필요한 파일을 만들 수 있다. • 이미 생성된 파일에서 내용을 읽어 처리할 수 있다
미리읽어오기	교재 11장
과제,시험,기타	도전 프로그래밍
13 주차	[13주]
학습주제	12장 일상생활의 사물 코딩인 객체지향 프로그래밍
목표및 내용	• 객체와 클래스를 이해하고 필요한 클래스를 정의하고 객체를 만들어 활용할 수 있다. • 클래스 속성과 인스턴스 속성, 정적 메소드와 클래스 메소드를 이해하고 정의할 수 있다. • 상속을 이해하고 부모 클래스와 자식 클래스를 정의할 수 있다. • 추상 메소드와 추상 클래스를 이해하고 정의할 수 있다
미리읽어오기	교재 12장
과제,시험,기타	도전 프로그래밍
14 주차	[14주]
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	

강 의 계 획 서

15	[기말고사]
	직무수행능력평가 2차(기말고사)
목표및 내용	직무수행능력평가, 서술형평가
미리읽어오기	8장에서 13장까지
과제,시험,기타	

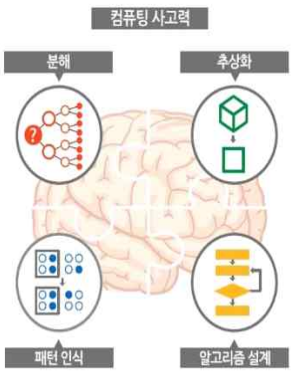
수업지원 안내	장애학생을 위한 별도의 수강 지원을 받을 수 있습니다. 언어가 문제가 되는 학생은 글로 된 과제 안내, 확대문자 시험지 제공 등의 지원을 드립니다.

▶ chapter01

- *파이썬 언어와 컴퓨팅 사고력*

파이썬 언어란 배우기 쉽고 누구나 무료로 사용할 수 있는 오픈 소스(open source) 프로그래밍 언어(programming language)이다. 1991년 네덜란드의 귀도 반 로섬(Guido van Rossum)이 개발했으며, 현재는 비영리 단체인 파이썬 소프트웨어 재단(PSF: Python Software Foundation)이 관리하고 있다. 배우기 쉽고 간결하며 개발 속도가 바르고 강력할 뿐만아니라 라이브러리(library)가 풍부하고 다양한 개발 환경(development environment)을 제공하고 있어 개발자가 쉽고 빠르게 소프트웨어를 개발하는데 도움을 준다. 교육분야 외에 실무에서도 사용이 급증하고 있으며 스택오버플로(stackoverfloow.com)에서 '가장 빠르게 성장하는 프로그래밍 언어'로 선택 되었다. 제4차 산업혁명이란'모든 사물이 연결된 초연결 사회에서 생산되는 빅데이터를 기존 산업과 융합해 인공 지능, 클라우드 등의 첨단 기술로 처리하는 정보·지능화 혁명 시대'라고 할 수 있다. 제4차 산업혁명을 이끌 인재가 갖춰야 할 덕목으로 문제 해결 능력, 창의·융합 사고 능력, 의사소통 능력과 협업능력, 자기 주도 학습능력을 들 수 있다. 미래의 인재는 컴퓨터 과학 원리와 개념을 활용해 자신의 영역과 융합할 수 있는 역량을 갖춰야하며 이러한 능력을 컴퓨팅 사고력(CT : Computational Thinking)이라고 하는데 이는 '컴퓨터 과학의 기본 개념과 원리 및 컴퓨팅 시스템을 활용해 실생활 및 다양한 학문 분야의 문제를 이해하고 창의적 해법을 구현해 적용할 수 있는 능력'으로 정의할 수 있다. 컴퓨팅 사고력 구성요소는 분해 패턴 인식, 추상화, 알고리즘으로 되어있으며 컴퓨팅 사고력을 향상시키는 방법은 직접 프로그래밍을 하는 것이고 이 책에서 제안하는 프로그래밍의 절차는 이해(U: Understand), 설계(D : Design), 구현(I : Implement), 공유(S : Share)의 절차로 하고 있다.

▲ 그림9 컴퓨팅 사고력



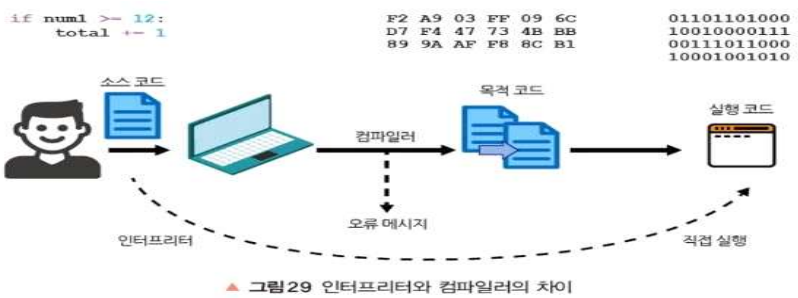
▲ 그림10 영국 컴퓨팅 교육에서 제안한 컴퓨팅 사고

▼ 표1 코딩 절차와 컴퓨팅 사고력 요소

코딩 절차	컴퓨팅 사고력 요소 활용	내용
이해	분해	• 주어진 문제를 이해하고 파악 • 분할: 문제를 좀 더 쉬운 작은 문제들로 분해
설계	컴퓨팅 사고력: 핵심 개념 (분해, 패턴 인식, 추상화, 설계)	• 패턴 인식: 분해된 문제들 사이의 유사성 또는 패턴을 탐색 • 추상화: 문제에서 불필요한 부분은 제거하고 주요 핵심 요소를 추려 내는 과정 • 알고리즘: 프로그래밍 언어인 파이썬에 맞는 입력과 출력, 변수 저장 그리고 절차에 따라 구성
구현	추상화	• 문제 해결을 위해 파이썬으로 코드 개발 • 작성된 코드의 실행과 테스팅(testing, simulation), 디버깅(debugging) 과정을 거치면서 코드를 수정하고 필요하면 다시 설계
공유	협업과 평가 (Collaboration & Evaluation)	• 자신이 구현한 프로그램을 발표하고 다른 학습자의 프로그램과 비교 • 현재의 기능을 향상시키는 방향으로 프로그램 개선 연구 • 교수자의 피드백 및 평가

- *제4차 산업혁명 시대, 모두에게 필요한 파이썬*

파이썬의 특징은 첫째, 간결하고 배우기 쉽고, 둘째 무로이고 생산성이 높으며, 셋째 강력한 라이브러리를 제공하고, 넷째 프로그램과 호환되는 풀 언어(glue language)라는 것이다. 현재 일반 응용 프로그램의 개발뿐만 아니라 제4차 산업혁명 요소 기술인 인공 지능과 빅데이터 처리에 적합한 언어로 인정받아, 교육 및 학술 연구 분야는 물론 데이터과학(data science)과 머신(machine learning) 등 자연어 처리, 음성 인식, 영상 처리, 사물 인터넷, 웹 개발 등의 여러 실무에 적합한 언어로 엄청한 인기를 끌고 있다. 제4차 산업혁명 시대의 핵심 기술인 인공 지능의 구현과 빅데이터 분석 처리 등의 컴퓨팅 과학 실무 분야에 적합하기 때문에 각광을 받고 있다. 파이썬은 인공 지능의 머신 러닝과 딥러닝, 빅데이터 처리를 위한 통계 및 분석 방법의 라이브러리도 풍부하게 제공한다. 파이썬은 베이직 언어와 마찬가지로 인터프리터(해석기) 위에서 실행된는 인터프리트(iterpreted) 방식의 언어다. 자바와 C는 컴파일(compliled) 방식의 언어이다. 인터프리트 방식이란 동시 통역처럼 파이썬의 문장을 한 줄 한 줄마다 즉시 번역해 실행하는 방식이다.



▲ 그림29 인터프리터와 컴파일러의 차이

▶ chapter02

- *주요 내용 정리*

```
#연산자
print("10/4 :", 10/4)
print("10//4 :", 10//4)
print("2**3 :", 2**3)
print("k" * 3)
```

```
10/4 : 2.5
10//4 : 2
2**3 : 8
kkk
```

```
#함수 eval() : 실행 가능한 문자열 연산식 결과 반환
print("eval(""2**3"") :", eval("2**3"))
```

```
eval(2**3) : 8
```

```
#한 번에 여러 자료 대입
a, b = 5, 9
b, a = a, b
print(a, b)
```

```
9 5
```

```
#함수 divmod() : 나누기 몫 연산(//)과 나머지 연산(%) 결과 반환
a, b = 17, 5
c, d = divmod(17, 5)
print(c, d)
```

```
3 2
```

```
#함수 input() : 표준 입력 받기, 함수 int(input()) : 정수로 표준 입력
a = input("정수 입력(문자열) >> ")
b = int(input("정수 입력(정수형) >> "))
print(a, b)
```

```
정수 입력(문자열) >> 5
정수 입력(정수형) >> 3
5 3
```

```
#bin() : 2진수, oct() : 8진수, hex() : 16진수
print("bin(10) :", bin(10))
print("oct(10) :", oct(10))
print("hex(10) :", hex(10))
```

```
bin(10) : 0b1010
oct(10) : 0o12
hex(10) : 0xa
```

```
#p.59_예제 2-15
data = int(input("정수 입력 >> "))
print("2진수 :", bin(data))
print("8진수 :", oct(data))
print("10진수 :", data)
print("16진수 :", hex(data))
```

```
정수 입력 >> 31
2진수 : 0b11111
8진수 : 0o37
10진수 : 31
16진수 : 0x1f
```

```
#int(정수, 진법기수) : int(a, b) => b진수의 문자열 a를 10진수로 변환
print("int('110', 2) :", int("110", 2))
```

```
int('110', 2) : 6
```

```
#p.61_예제 2-16
n = input("16진수 정수 입력 >> ")
data = int(n, 16)

print("2진수 :", bin(data))
```

```
16진수 정수 입력 >> e1
2진수 : 0b11100001
8진수 : 0o341
10진수 : 225
16진수 : 0xe1
```

```
print("8진수 :", oct(data))
print("10진수 :", data)
print("16진수 :", hex(data))
```

- 프로젝트 Lab1 표준 입력한 실수와 연산식의 산술 연산 및 결과 출력

```
#실수형으로 입력받은 뒤 연산자를 통해 계산 결과를 출력하며 함수 eval()을 이용해 문자열 연산식의 결과를 출력
num1 = float(input("첫 번째 수 입력 >> "))
num2 = float(input("두 번째 수 입력 >> "))
print("합:", num1 + num2)
print("차:", num1 - num2)
print("곱하기:", num1 * num2)
print("나누기:", num1 / num2)
str = input("연삭식 입력 >> ")
print("연산식:", str, "결과:", eval(str))
--------------------------------------------------------------------------------
첫 번째 수 입력 >> 3.5
두 번째 수 입력 >> 5.3
합: 8.8
차: -1.7999999999999998
곱하기: 18.55
나누기: 0.6603773584905661
연삭식 입력 >> 3 + 5 * 3
연산식: 3 + 5 * 3 결과: 18
```

- 프로젝트 Lab2 여러 진수를 표준 입력한 수를 여러 진수로 출력

```
#진수와 정수를 입력받은 뒤 함수 int(num, base)를 통해 dat에 10진수로 반환한 뒤
#함수bin(), oct(), hex()를 이용해 출력
base = int(input("입력할 정수의 진수(base)는? "))
num = input(str(base) + "진수 정수 입력 >> ")
data = int(num, base)
print("2진수:", bin(data))
print("8진수:", oct(data))
print("10진수:", data)
print("16진수:", hex(data))
--------------------------------------------------------------------------------
입력할 정수의 진수(base)는? 16
16진수 정수 입력 >> 4f
2진수: 0b1001111
8진수: 0o117
10진수: 79
16진수: 0x4f
```

- 도전 프로그래밍

```
01.
str1 = input("문자열 1: ")
str2 = input("문자열 2: ")
```

```
print(str1, str2)
```
```
--------------------------------------------------------------------------
문자열 1: Python
문자열 2: Program
Python Program
```

02.
```
input = input("차의 속도를 입력(km) >> ")
print(input + "(km)은" , int(input)/1.61, "마일(miles)이다.")
```
```
--------------------------------------------------------------------------
차의 속도를 입력(km) >> 53
53(km)은 32.91925465838509 마일(miles)이다.
```

03.
```
coffee = int(input("아메리카노 몇 개 주문하세요? "))
print("총 가격은", coffee * 3500, "이다.")
```
```
--------------------------------------------------------------------------
아메리카노 몇 개 주문하세요? 3
총 가격은 10500 이다.
```

04.
```
num1 = 40120
num2 = 6378.1 * 2 * 3.141592
print("알려진 지구 둘레:", num1)
print("지구와 같은 원둘레:", num2)
print("차이: " + str(num1 - num2) + "(km)")
```
```
--------------------------------------------------------------------------
알려진 지구 둘레: 40120
지구와 같은 원둘레: 40074.77587040001
차이: 45.22412959999201(km)
```

05.
```
c = int(input("온도 입력 >> "))
print("정확 계산: 섭씨:", c, ", 화씨:", c * (9/5) + 32)
print("약식 계산: 섭씨:", c, ", 화씨:", c * 2 + 30)
print("차이:",(c * (9/5) + 32) - (c * 2 + 30))
```
```
--------------------------------------------------------------------------
온도 입력 >> 29
정확 계산: 섭씨: 29 , 화씨: 84.2
약식 계산: 섭씨: 29 , 화씨: 88
차이: -3.799999999999997
```

06.
```
num1 = int(input("Enter First number: "))
num2 = int(input("Enter Second number: "))
print(num1, "/", num2, "==>",  num1 / num2)
print(num1, "%", num2, "==>",  num1 % num2)
```

```
print(num1, "//", num2, "==>",  num1 // num2)
print(num1, "**", num2, "==>",  num1 ** num2)
```

```
Enter First number: 3
Enter Second number: 8
3 / 8 ==> 0.375
3 % 8 ==> 3
3 // 8 ==> 0
3 ** 8 ==> 6561
```

07.

```
num1 = input("첫 번째 16진수 실수 입력 >> ")
num2 = input("두 번째 16진수 실수 입력 >> ")
print("실수1:", float.fromhex(num1), "실수2:", float.fromhex(num2))
print("합:",float.fromhex(num1) + float.fromhex(num2))
print("차:",float.fromhex(num1) - float.fromhex(num2))
print("곱하기:",float.fromhex(num1) * float.fromhex(num2))
print("나누기:",float.fromhex(num1) / float.fromhex(num2))
```

```
첫 번째 16진수 실수 입력 >> b
두 번째 16진수 실수 입력 >> e.1
실수1: 11.0 실수2: 14.0625
합: 25.0625
차: -3.0625
곱하기: 154.6875
나누기: 0.7822222222222223
```

08.

```
#방법 1
n = int(input("네 자릿수 정수 입력 >> "))
str1 = str(n//1000)
str2 = str(n%1000//100)
str3 = str(n%1000%100//10)
str4 = str(n%10)
print(str4 + str3 + str2 + str1)

#방법 2
num = int(input("네 자릿수 정수 입력 >> "))
a, b = divmod(num, 1000)
c, d = divmod(b, 100)
e, f = divmod(d, 10)
print(str(f) + str(e) + str(c) + str(a))
```

```
네 자릿수 정수 입력 >> 1657
7561
네 자릿수 정수 입력 >> 6548
8456
```

▶ chapter03

- 주요 내용 정리

#슬라이싱, str[start:end:step] : 첨자 start에서 첨자 end-1까지 step간격으로 출력

```
s = "123456789"
print(s[-9:-1])
print(s[-9:4])
print(s[:4])
print(s[2:])
print("s[0:8:2] :", s[0:8:2])
print("s[::-1] :", s[::-1])
```

```
12345678
1234
1234
3456789
s[0:8:2] : 1357
s[::-1] : 987654321
```

#함수 replace(a, b) : a를 b로 바꿈, replace(old, new, count) : old를 new로 count만큼 바꿈 -> 둘 다 문자열 자체가 변경되는 것은 아님

```
s = "파이썬 파이썬 파이썬"
print("s.replace('파이썬', 'HI') :", s.replace("파이썬", "HI"))
print("s.replace('파이썬', 'HI', 2) :", s.replace("파이썬", "HI", 2))
```

```
s.replace('파이썬', 'HI') : HI HI HI
s.replace('파이썬', 'HI', 2) : HI HI 파이썬
```

#함수 join() : 원하는 문자를 문자열 사이사이에 삽입

```
s = "123456789"
print("'->'.join(s) :", "->".join(s))
```

```
'->'.join(s) : 1->2->3->4->5->6->7->8->9
```

#함수 find(), rfind() index(), rindex() : 찾고자 하는 문자열(문자)의 처음 첨자를 반환, find()는 없다면 ValueError, index()는 없다면 -1 반환

```
s = "123456789"
print("s.find('4') :", s.find("4"))
print("s.index('67') :", s.index("67"))
```

```
s.find('4') : 3
s.index('67') : 5
```

#함수 split() : 문자열 나누기, 인자가 있다면 인자값으로 문자열을 나눔, 인자가 없다면 공백을 기준으로 나눔

```
s = "파이썬 파이썬 파이썬"
print(s.split())
a, b = "100 200".split()
print(a, b)
```

```
['파이썬', '파이썬', '파이썬']
100 200
```

#함수 center(), strip() : 공백 제거

```
print(" HI " .center(15, "-"))
s = "     HI     "
print(s.lstrip())
print(s.rstrip())
print(s.strip())
```

```
------ HI -----
HI
    HI
HI
```

#format을 이용한 출력

```
a, b = 10, 3
print("{0:d} / {1:d} = {2:f}" .format(a, b, a/b))
```

```
10 / 3 = 3.333333
```

#x in s : 문자열 s에 부분 문자열 x가 있다면 Ture, 없다면 False

#x not in s : 문자열 s에 부분 문자열 x가 있다면 False, 없다면 True

```
s = "123456789"
print("345" in s)
print("345" not in s)
```

```
True
False
```

```
#비트 베타적 논리합 ^
#a ^ a == 0, a ^ 0 == a, a ^ 1 == -a
#a ^ b == b ^ a, (a ^ b) ^ c == a ^ (b ^ c)
#(a ^ b) ^ b == a ^ (b ^ b) == a ^ 0 == a
```

- 프로젝트 Lab1 할인율에 따른 할인 가격

```
#My
#정수로 가격을 입력 받은 뒤 one, two, four에 가격에 대한 True or False를 저장
#표를 통해 할인된 가격, 할인율, 할인액을 각각 변수에 저장하여 출력
n = int(input("총 가격(원 가격) 입력 >> "))

one = 10000 <= n < 20000
two = 20000 <= n < 40000
four = n >= 40000

result = ((n * 0.99 * one) +
          (n * 0.98 * two) + (n * 0.96 * four))
rate = ((one * 0.01) +
        (two * 0.02) + (four * 0.04))
rateN = ((n * 0.01 * one) +
         (n * 0.02 * two) + (n * 0.04 * four))

print("원 가격: {} 할인된 가격: {}" .format(n, result))
print("할인율: {} 할인액: {}" .format(rate, rateN))
```

```
#Book
price = int(input("총 가격(원 가격) 입력 >> "))

rate1 = (10000 <= price < 20000) * 0.01
rate2 = (20000 <= price < 40000) * 0.02
rate3 = (price >= 40000) * 0.04
rate = rate1 + rate2 + rate3

discount = price * rate
disPrice = price - discount

print("원 가격:", price, "할인된 가격:", disPrice)
print("할인율:", rate, "할인액:", discount)
```

```
--------------------------------------------------------------------------------
총 가격(원 가격) 입력 >> 15480
원 가격: 15480 할인된 가격: 15325.2
할인율: 0.01 할인액: 154.8
```

- 프로젝트 Lab2 단어를 추출해 회문인지 검사

```
#콤마로 문자를 구분해서 입력받은 뒤 함수 replace()를 통해 공백을 제거한 뒤
#함수 split(",")을 통해 콤마로 문자열을 나누어 변수에 저장한 뒤 슬라이싱을 이용해 회문인지 비교 후 결과 출력
str = input("콤마로 구분된 단어 3개 입력 >> ")
str = str.replace(" ", "")
str1, str2, str3 = str.split(",")

print("단어: {}, 역순: {}, 화문: {}" .format(str1, str1[::-1], str1 == str1[::-1]))
print("단어: {}, 역순: {}, 화문: {}" .format(str2, str2[::-1], str2 == str2[::-1]))
print("단어: {}, 역순: {}, 화문: {}" .format(str3, str3[::-1], str3 == str3[::-1]))
```

```
--------------------------------------------------------------------------------
콤마로 구분된 단어 3개 입력 >> level, hello, Helleh
단어: level, 역순: level, 화문: True
```

```
단어: hello, 역순: olleh, 화문: False
단어: Helleh, 역순: helleH, 화문: False
```

- 도전 프로그래밍

```
01
string = input("문자열 입력 >> ")
print("참조할 첨자: 0 ~", len(string) - 1)
idx = int(input("참조할 첨자 입력 >> "))
print("문자열: {}, 길이: {}" .format(string, len(string)))
print("참조 문자:", string[idx])
--------------------------------------------------------------------------------
문자열 입력 >> Python is a good language!
참조할 첨자: 0 ~ 25
참조할 첨자 입력 >> 5
문자열: Python is a good language!, 길이: 26
참조 문자: n
```

```
02
string = input("다섯 문자 이상의 문자열 입력 >> ")
print("입력 문자열:", string)
print("첫 문자:", string[0])
print("마지막 문자:", string[len(string) - 1])
print("첫 문자를 제외한 부분 문자열:", string[1:])
print("마지막 문자를 제외한 부분 문자열:", string[:len(string) - 1])
print("맨 앞과 뒤의 두 문자씩 제외한 부분 문자열:", string[2:len(string) - 2])
print("문자 하나씩 건너뛴 부분 문자열:", string[::2])
print("역문자열:", string[::-1])
--------------------------------------------------------------------------------
다섯 문자 이상의 문자열 입력 >> Python String Slicing
입력 문자열: Python String Slicing
첫 문자: P
마지막 문자: g
첫 문자를 제외한 부분 문자열: ython String Slicing
마지막 문자를 제외한 부분 문자열: Python String Slicin
맨 앞과 뒤의 두 문자씩 제외한 부분 문자열: thon String Slici
문자 하나씩 건너뛴 부분 문자열: Pto tigSiig
역문자열: gnicilS gnirtS nohtyP
```

```
03
str = "Beautiful is better than ugly."
print(str)
str = str.replace("Beautiful", "Explicit")
print("위 철학을 메소드 replace()를 사용해 다음 철학으로 다시 저장")
print(str.replace("ugly", "implicit"))
--------------------------------------------------------------------------------
Beautiful is better than ugly.
위 철학을 메소드 replace()를 사용해 다음 철학으로 다시 저장
```

```
Explicit is better than implicit.
```

```
04
url = "https://docs.python.org/3/tutorial"
print(url[url.find("ht") : url.find(":/")])
print(url[url.find("do") : url.find("/3")])
print(url[url.rfind("tu") : len(url)])
--------------------------------------------------------------------------------
https
docs.python.org
tutorial
```

```
05
time = input("시각정보(16:30:15) 입력 >> ")
hours, mins, secs = time.split(":")
print("입력 시각 정보:", time)
print(hours + "시 " + mins + "분 " + secs + "초")
--------------------------------------------------------------------------------
시각정보(16:30:15) 입력 >> 14:19:35
입력 시각 정보: 14:19:35
14시 19분 35초
```

```
06
a, b = input("실수 두개 입력 >> ").split()
num1 = float(a)
num2 = float(b)
print("{} > {} 결과: {}" .format(num1, num2, num1 > num2))
print("{} >= {} 결과: {}" .format(num1, num2, num1 >= num2))
print("{} < {} 결과: {}" .format(num1, num2, num1 < num2))
print("{} <= {} 결과: {}" .format(num1, num2, num1 <= num2))
print("{} == {} 결과: {}" .format(num1, num2, num1 == num2))
print("{} != {} 결과: {}" .format(num1, num2, num1 != num2))
--------------------------------------------------------------------------------
실수 두개 입력 >> 5.3 1.6
5.3 > 1.6 결과: True
5.3 >= 1.6 결과: True
5.3 < 1.6 결과: False
5.3 <= 1.6 결과: False
5.3 == 1.6 결과: False
5.3 != 1.6 결과: True
```

```
07
mask = 0xff
print("10진수: {0:3d} 2진수: {1:8b} 8진수: {1:5o} 16진수: {1:3x}" .format(-7, -7 & mask))
print("10진수: {0:3d} 2진수: {1:8b} 8진수: {1:5o} 16진수: {1:3x}" .format(-6, -6 & mask))
print("10진수: {0:3d} 2진수: {1:8b} 8진수: {1:5o} 16진수: {1:3x}" .format(-5, -5 & mask))
print("10진수: {0:3d} 2진수: {1:8b} 8진수: {1:5o} 16진수: {1:3x}" .format(-4, -4 & mask))
```

```
print("10진수: {0:3d} 2진수: {1:8b} 8진수: {1:5o} 16진수: {1:3x}" .format(-3, -3 & mask))
print("10진수: {0:3d} 2진수: {1:8b} 8진수: {1:5o} 16진수: {1:3x}" .format(-2, -2 & mask))
print("10진수: {0:3d} 2진수: {1:8b} 8진수: {1:5o} 16진수: {1:3x}" .format(-1, -1 & mask))
print("10진수: {0:3d} 2진수: {1:8b} 8진수: {1:5o} 16진수: {1:3x}" .format(0, 0 & 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
```

```
08
num = int(input("정수 입력 >> "))
x = int(input("2의 지수승 입력 >> "))
print()
print("정수 {0} >> {1}, 결과: {2}" .format(num, x, num << x))
print("정수 {0} * 2**{1}, 결과: {2}" .format(num, x, num * 2**x))
print("2진수(32비트):{0:32b} 정수: {0}". format(num))
print("2진수(32비트):{0:32b} 정수: {1} << {2}". format(num << x, num, x))
print("2진수(32비트):{0:32b} 정수: {1} * 2**{2}". format(num * 2**x, num, x))
```

```
--------------------------------------------------------------------------------
정수 입력 >> 62
2의 지수승 입력 >> 3

정수 62 >> 3, 결과: 496
정수 62 * 2**3, 결과: 496
2진수(32비트):                                 111110 정수: 62
2진수(32비트):                              111110000 정수: 62 << 3
2진수(32비트):                              111110000 정수: 62 * 2**3
```

▶ chapter04

- 주요 내용 정리

```
#if ... else : 조건문
s = 10
if s > 5:
    print("5보다 큼")
else:
    print("5보다 작음")
```

```
5보다 큼
```

```
#if ... elif : 조건문
n = int(input("자연수 입력 >> "))

if n > 12:
    print("12보다 큼")
elif n > 9:
    print("9보다 큼")
elif n > 6:
    print("6보다 큼")
elif n > 3:
    print("3보다 큼")
else:
    print("0보다 큼")
```

```
자연수 입력 >> 36
12보다 큼
```

```
#for : 반복문
s = 0
for i in range(1, 10):
    s += i
print("1~9 까지 합:", s)

s = 0
for i in range(1, 10, 2):
    s += i
print("1~9 까지 홀수 합:", s)
```

```
1~9 까지 합: 45
1~9 까지 홀수 합: 25
```

```
#while : 반복문
m, s, i = True, 0, 1
while m:
    s += i
    i += 1
    if i == 10:
        m = False
else:
    print("1~9까지 합:", s)
```

```
1~9까지 합: 45
```

```
#p.138_예제 4-12
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
```

```
#모듈 random의 함수 randint(시작, 끝)을 이용해 임의의 정수 얻기
import random
n = random.randint(1,5)
print("1~5 중 랜덤:", n)

from random import randint
n = randint(1, 5)
print("1~5 중 랜덤:", n)
```

```
1~5 중 랜덤: 1
1~5 중 랜덤: 4
```

```
#break : 멈춤, continue : 건너뛰고 진행
s = 1
while True:
    s += 1
    if s == 3:
        break
print(s)

for i in range(1, 10):
    if i == 2:
        continue
    else:
        print(i, end=" ")
```

```
3
1 3 4 5 6 7 8 9
```

- 프로젝트 Lab1 1에서 10 사이의 수 맞히기

```
#함수 randint()를 이용해 1~10까지 임의의 수를 r에 저장한 뒤 정수를 입력받고
#반복문 while을 통해 r에 저장된 값과 같은 값을 입력받을 때까지 반복한 뒤 찾은 값을 출력하고 종료
from random import randint

r = randint(1, 10)
n = int(input("1에서 10 사이의 수를 맞히세요 >> "))

more = True
while more:
    if n == r:
```

```
        print("축하한다. {}: 정답이다." .format(n))
        more = False
    elif n > r:
        a = input("{}보다 더 작은 수로 다시 입력 >> " .format(n))
        n = int(a)
    else:
        a = input("{}보다 더 큰 수로 다시 입력 >> ".format(n))
        n = int(a)
else:
    print("종료" .center(30, "*"))
--------------------------------------------------------------------------------
1에서 10 사이의 수를 맞히세요 >> 5
5보다 더 큰 수로 다시 입력 >> 8
8보다 더 큰 수로 다시 입력 >> 9
9보다 더 큰 수로 다시 입력 >> 10
축하한다. 10: 정답이다.
**************종료**************
```

- *프로젝트 Lab2 커피 주문받아 주문 가격 표시*

```
#반복문 while을 통해 종류를 입력받은 뒤 order==0이라면 종료가 되도록 했으며 아니라면 수량을 입력받은 뒤
#조건문 if..elif문을 통해 각 조건별로 출력을 하며 total에 현재 주문가격을 저장하여
#while문 종료 시 총 주문가격을 출력
print("환영합니다. 음료를 선택하세요.")

menu ='''Coffee menu!
    1. 아메리카노        2000
    2. 카페라테          2500
    3. 카푸치노          3000
    4. 캐러멜마키아또    4000
    0. 주문 종료
종류 ? '''

total = 0
while True:
    order = int(input(menu))

    if order == 0:
        print()
        print("주문 종료" .center(30, "*"))
        break

    count = int(input("수량 ? "))
    print()

    if order == 1:
        total += 2000 * count
        print("아메리카노 {}개 주문" .format(count))
```

```
            print("현재 주문 가격: {}원" .format(total))
        elif order == 2:
            total += 2500 * count
            print("카페라테 {}개 주문" .format(count))
            print("현재 주문 가격: {}원" .format(total))
        elif order == 3:
            total += 3000 * count
            print("카푸치노 {}개 주문" .format(count))
            print("현재 주문 가격: {}원" .format(total))
        else:
            total += 4000 * count
            print("캐러멜마키아또 {}개 주문" .format(count))
            print("현재 주문 가격: {}원" .format(total))
        print()

print("총 주문 가격: {}원" .format(total))
print("안녕!" .center(30, "="))
```

```
--------------------------------------------------------------------------------
환영합니다. 음료를 선택하세요.
Coffee menu!
    1. 아메리카노          2000
    2. 카페라테            2500
    3. 카푸치노            3000
    4. 캐러멜마키아또      4000
    0. 주문 종료
종류 ? 2
수량 ? 3

카페라테 3개 주문
현재 주문 가격: 7500원

Coffee menu!
    1. 아메리카노          2000
    2. 카페라테            2500
    3. 카푸치노            3000
    4. 캐러멜마키아또      4000
    0. 주문 종료
종류 ? 3
수량 ? 7

카푸치노 7개 주문
현재 주문 가격: 28500원

Coffee menu!
    1. 아메리카노          2000
    2. 카페라테            2500
    3. 카푸치노            3000
```

```
    4. 캐러멜마키아또     4000
    0. 주문 종료
종류 ? 0

************주문 종료*************
총 주문 가격: 28500원
==============안녕!==============
```

- 프로젝트 Lab3 1개월 달력 출력

```
#My
#최대 일수와 시작 요일을 입력 받은 뒤 요일을 출력하고 day != 0이 아니라면 day만큼 공백을 출력한 뒤
#반복문 for문을 통해 달력을 출력
date = int(input("한 달 최대 일수 입력 >> "))
day = int(input("첫 날 1일의 시작 요일 입력(0=일, 1=월, 2=화, 3=수, 4=목, 5=금, 6=토) >> "))

print("\n일 월 화 수 목 금 토\n")

if day != 0:
    for i in range(day):
        print("    ", end="")

cnt = day
for i in range(1, date + 1):
    print("%2d" %i, end=" ")
    cnt += 1
    if cnt % 7 == 0:
        print()
print()
```

```
#Book
dates = int(input("한 달 최대 일수 입력 >> "))
day = int(input("첫 날 1일의 시작 요일 입력(0=일, 1=월, 2=화, 3=수, 4=목, 5=금, 6=토) >> "))
day %= 7
print("\n", end=" ")
for i in "일월화수목금토":
    print("%s" %i, end=" ")
else:
    print("\n")
```

```
cnt = 0
if day != 0:
    print("    " * day, end="")
    cnt += day
for i in range(1, dates + 1):
    print("%2d" %i, end=" ")
    cnt += 1
    if cnt % 7 == 0:
        print()
else:
    print("\n")
```

```
--------------------------------------------------------------------------------
한 달 최대 일수 입력 >> 31
첫 날 1일의 시작 요일 입력(0=일, 1=월, 2=화, 3=수, 4=목, 5=금, 6=토) >> 3

일 월 화 수 목 금 토

          1  2  3  4
 5  6  7  8  9 10 11
12 13 14 15 16 17 18
19 20 21 22 23 24 25
26 27 28 29 30 31
```

- 도전 프로그래밍

```
01
month = int(input("월 입력 ? "))

tmp = month % 10
if tmp == 0:
```

```
    print("{}월 가을" .format(month))
elif tmp >= 9:
    print("{}월 가을" .format(month))
elif tmp >= 6:
    print("{}월 여름" .format(month))
elif tmp >= 4:
    print("{}월 봄" .format(month))
elif tmp >= 1:
    print("{}월 겨울" .format(month))
```

```
--------------------------------------------------------------------------------
월 입력 ? 12
12월 겨울
```

02

```
import random
for i in range(5):
    n = random.randint(35, 50)
    if n >= 40:
        print("근로 시간 %d 시간, 주급 %d" %(n, (40*12000) + ((n-40)*12000*1.5)))
    else:
        print("근로 시간 %d 시간, 주급 %d" %(n, n * 12000))
```

```
--------------------------------------------------------------------------------
근로 시간 43 시간, 주급 534000
근로 시간 37 시간, 주급 444000
근로 시간 44 시간, 주급 552000
근로 시간 40 시간, 주급 480000
근로 시간 45 시간, 주급 570000
```

03

```
from random import randint

num1 = randint(1, 99)
num2 = randint(1, 99)
num3 = randint(1, 99)
if num1 > num2:
    if num2 > num3:
        print("{0} {1} {2} 중에서 최대: {0}" .format(num1, num2, num3))
    else:
        if num1 > num3:
            print("{0} {1} {2} 중에서 최대: {0}" .format(num1, num2, num3))
        else:
            print("{0} {1} {2} 중에서 최대: {2}" .format(num1, num2, num3))
else:
    if num1 > num3:
        print("{0} {1} {2} 중에서 최대: {1}" .format(num1, num2, num3))
    else:
        if num2 > num3:
```

```
            print("{0} {1} {2} 중에서 최대: {1}" .format(num1, num2, num3))
        else:
            print("{0} {1} {2} 중에서 최대: {2}" .format(num1, num2, num3))
```

```
55 2 86 중에서 최대: 86
```

04

```
h, w = input("당신의 키(cm)와 몸무게(kg)는? >> ").split()
height = float(h)
weight = float(w)
bmi = weight / (height / 100)**2

print("키: {0}(cm), 몸무게: {1}(kg)" .format(height, weight))
if bmi >= 40:
    print("BMI: %.1f 고도 비만"  %bmi)
elif bmi >= 35:
    print("BMI: %.1f 중등도 비만"  %bmi)
elif bmi >= 30:
     print("BMI: %.1f 비만"  %bmi)
elif bmi >= 25:
     print("BMI: %.1f 과체중"  %bmi)
elif bmi >= 18.5:
     print("BMI: %.1f 정상"  %bmi)
elif bmi < 18.5:
    print("BMI: %.1f 저체중"  %bmi)
```

```
당신의 키(cm)와 몸무게(kg)는? >> 198 80
키: 198.0(cm), 몸무게: 80.0(kg)
BMI: 20.4 정상
```

05

```
for i in range(20, 42, 3):
    f1 = i * (9/5) + 32
    f2 = i * 2 + 30
    result = abs(f1 - f2)
    print("섭씨: %d 화씨: %.1f 화씨(약식): %d 차이: %.2f" % (i, f1, f2, result))
```

```
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
```

06

```
from random import randint
more = True

while more:
    num1 = randint(1, 99)
    num2 = randint(1, 99)

    print("{0} * {1} = {2}" .format(num1, num2, num1 * num2))
    print()
    yn = input("계속 y / n ? ")
    if yn == 'n':
        break
```

```
80 * 86 = 6880

계속 y / n ? y
75 * 40 = 3000

계속 y / n ? n
```

```
07
num = int(input("소수(prime number)인지를 판별한 2 이상의 정수 입력 >> "))

check = True
for i in range(2, num):
    if(num % i == 0):
        check = False
        break

if check == True:
    print("정수 {}는 소수이다." .format(num))
else:
    print("정수 {}는 소수가 아니다." .format(num))
```

```
소수(prime number)인지를 판별한 2 이상의 정수 입력 >> 19
정수 19는 소수이다.
```

```
08
from random import randint

rand = randint(1, 100)
print("첫 값은", rand, "이다.")

more = True;
while more:
    string = input("산술 연산의 종류를 입력하세요. >> ")
```

```
    if string == "+":
        num = int(input("두 번째 피연자를 입력하세요. >> "))
        print(rand, "+", num, "=", rand + num)
    elif string == "-":
        num = int(input("두 번째 피연자를 입력하세요. >> "))
        print(rand, "-", num, "=", rand - num)
    elif string == "*":
        num = int(input("두 번째 피연자를 입력하세요. >> "))
        print(rand, "*", num, "=", rand * num)
    elif string == "/":
        num = int(input("두 번째 피연자를 입력하세요. >> "))
        print(rand, "/", num, "=", rand / num)
    else:
        print("종료".center(34, "*"))
        more = False
    print()
```

```
--------------------------------------------------------------------------------
첫 값은 51 이다.
산술 연산의 종류를 입력하세요. >> +
두 번째 피연자를 입력하세요. >> 5
51 + 5 = 56

산술 연산의 종류를 입력하세요. >> *
두 번째 피연자를 입력하세요. >> 3
51 * 3 = 153

산술 연산의 종류를 입력하세요. >> /
두 번째 피연자를 입력하세요. >> 7
51 / 7 = 7.285714285714286

산술 연산의 종류를 입력하세요. >> ?
**********종료**********
```

▶ chapter05

- 주요 내용 정리

```
#리스트 : 각 항목이 같은 자료형일 필요 없음, 항목 순서는 의미가 있음, 중복 가능
a = ['abc', 3, 'ax4gx2', '!']
print(a)
```

```
['abc', 3, 'ax4gx2', '!']
```

```
#빈 리스트 생성(a = [], a = list()), 항목 추가(a.append() : 항목 가장 뒤에 추가)
a = []
b = list()
print("리스트 a :", a, "리스트 b :", b)
a.append(1)
b.append(2)
print("리스트 a :", a, "리스트 b :", b)
```

```
리스트 a : [] 리스트 b : []
리스트 a : [1] 리스트 b : [2]
```

```
#p.164_예제 5-4
pl = ["C", "C++", "Python", "Java"]
print(pl[0])
print(pl[2])
print()

for i in range(len(pl)):
    print(pl[i])
print()

for i in pl:
    print(i)
print()
```

```
C
Python

C
C++
Python
Java

C
C++
Python
Java
```

```
#from random import choice : 리스트에서 랜덤으로 하나를 선택
from random import choice
pl = ["C", "C++", "Python", "Java"]
print(choice(pl))
```

```
C++
```

```
#리스트 항목 수정
#count() : 몇 개 있는지 카운트, index() : 첨자의 위치 반환(동일한 값이 있다면 첫 번째 첨자 반환, 없다면 오류)
top = ["BTS", "볼사", "BTS", "블랙핑크", "태연", "BTS"]
print("count('BTS') :", top.count("BTS"))
print("index('BTS') :", top.index("BTS"))
top[1] = "잔나비" #리스트에 추가
print(top)
```

```
count('BTS') : 3
index('BTS') : 0
['BTS', '잔나비', 'BTS', '블랙핑크', '태연', 'BTS']
```

```
#p.169_예제 5-7 (중첩 리스트)
animal = [["사자", "코끼리", "호랑이"], "조류", "어류"]
print(animal)

for s in animal:
```

```
    print(s)
print()

bird = ["독수리", "참새", "까치"]
fish = ["갈치", "붕어", "고등어"]
animal[1:] = [bird, fish]
print(animal)

for lst in animal:
    for item in lst:
        print(item, end=" ")
    print()
```

```
[['사자', '코끼리', '호랑이'], '조류', '어류']
['사자', '코끼리', '호랑이']
조류
어류

[['사자', '코끼리', '호랑이'], ['독수리', '참새', '까치'], ['갈치', '붕어', '고등어']]
사자 코끼리 호랑이
독수리 참새 까치
갈치 붕어 고등어
```

```
#p.174_예제 5-8
#리스트 부분 참조인 슬라이싱 : 리스트[start:stop:step] => 첨자start에서 첨자stop-1까지 step간격으로 출력
wlist = ["밥", "삶", "길", "죽", "꿈", "차", "떡", "복", "말"]
print("wlist[:] =", wlist[::])
print("wlist[::] =", wlist[::])
print("wlist[::-1] =", wlist[::-1])
print(wlist[::3])
print(wlist[1::3])
print(wlist[2::3])
print(wlist[::-2])
print(wlist[::-3])
print(wlist[-8:8])
print(wlist[-2:-9:-3])
```

```
wlist[:] = ['밥', '삶', '길', '죽', '꿈', '차', '떡', '복', '말']
wlist[::] = ['밥', '삶', '길', '죽', '꿈', '차', '떡', '복', '말']
wlist[::-1] = ['말', '복', '떡', '차', '꿈', '죽', '길', '삶', '밥']
['밥', '죽', '떡']
['삶', '꿈', '복']
['길', '차', '말']
['말', '떡', '꿈', '길', '밥']
['말', '차', '길']
['삶', '길', '죽', '꿈', '차', '떡', '복']
['복', '꿈', '삶']
```

```
#리스트 부분 수정
sport = ["풋살", "족구", "비치사커", "아구", "농구", "배구"]
sport[0:3] = ["축구"]
print(sport)
other = ["미식축구", "골프", "축구"]
sport[1:3] = other[0:1]
print(sport)
```

```
['축구', '아구', '농구', '배구']
['축구', '미식축구', '배구']
```

```
#리스트 항복 삽입과 삭제
#insert(첨자, 항목), remove(항목), pop(첨자) : 지정된 첨자의 항목을 삭제하고 반환
#pop() : 마지막 첨자를 삭제하고 반환
kpop = ["BTS", "블핑", "볼사", "장범준", "BTS", "잔나비", "BTS"]
print(kpop)
kpop.insert(1, "블핑")
print(kpop)
kpop.remove("장범준")
print(kpop)
kpop.pop(kpop.index("BTS"))
print(kpop)
kpop.pop()
```

```
['BTS', '블핑', '볼사', '장범준', 'BTS', '잔나비', 'BTS']
['BTS', '블핑', '블핑', '볼사', '장범준', 'BTS', '잔나비', 'BTS']
['BTS', '블핑', '블핑', '볼사', 'BTS', '잔나비', 'BTS']
['블핑', '블핑', '볼사', 'BTS', '잔나비', 'BTS']
['블핑', '블핑', '볼사', 'BTS', '잔나비']
```

```
print(kpop, "\n")

#del : 항목 삭제, 리스트 일부분 삭제, 리스트 삭제(메모리에서 제거하므로 참조하려하면 오류발생)
kpop = ["BTS", "블핑", "볼사", "장범준", "BTS", "잔나비", "BTS"]
print(kpop)
del kpop[2]
print("del kpop[2] :", kpop)
del kpop[0:2]
print("del kpop[0:2] :", kpop, "\n")
```

```
['BTS', '블핑', '볼사', '장범준', 'BTS', '잔나비', 'BTS']
del kpop[2] : ['BTS', '블핑', '장범준', 'BTS', '잔나비', 'BTS']
del kpop[0:2] : ['장범준', 'BTS', '잔나비', 'BTS']
```

```
#clear() : 리스트의 모든 항목이 삭제되며 출력 시 빈 리스트로 출력
kpop = ["BTS", "블핑", "볼사", "장범준", "BTS", "잔나비", "BTS"]
print(kpop)
kpop.clear()
print("kpop.clear() :", kpop, "\n")
```

```
['BTS', '블핑', '볼사', '장범준', 'BTS', '잔나비', 'BTS']
kpop.clear() : []
```

```
#리스트에 리스트 추가 : extend(), 리스트 연결하기 : +, 리스트와 정수와 *로 곱하면 지정된 정수만큼 반복
d1 = [1, 2, 3]
d2 = [4, 5]
d1.extend(d2)
print(d1)
print(d1 + d2)
print(d2 * 2)
```

```
[1, 2, 3, 4, 5]
[1, 2, 3, 4, 5, 4, 5]
[4, 5, 4, 5]
```

```
#리스트 항목 순서 뒤집기 : reverse(), 리스트 항목 정렬 sort()
#내장 함수 sorted()를 이용한 리스트 항목 정렬(내장 함수 sorted() 사용 시 새로운 리스트를 반환함)
s = list("74239740")
s.reverse()
print("s.reverse() :", s, "\n")

s.sort()
print("s.sort() :", s) #오름차순
s.sort(reverse = True)
print("s.sort(reverse=True) :", s, "\n") #내림차순

print("s :", s)
print("sorted(s) :", sorted(s)) #오름차순
print("sorted(s, reverse=True) :", sorted(s, reverse=True)) #내림차순
```

```
s.reverse() : ['0', '4', '7', '9', '3', '2', '4', '7']

s.sort() : ['0', '2', '3', '4', '4', '7', '7', '9']
s.sort(reverse=True) : ['9', '7', '7', '4', '4', '3', '2', '0']

s : ['9', '7', '7', '4', '4', '3', '2', '0']
sorted(s) : ['0', '2', '3', '4', '4', '7', '7', '9']
sorted(s, reverse=True) : ['9', '7', '7', '4', '4', '3', '2', '0']
```

```
#p184_예제 5-11 (리스트컴프리헨션)
#for으로 리스트 생성
a = []
for i in range(10):
    a.append(i)
print(a)

#컴프리헨션으로 리스트 생성
```

```
b = [i for i in range(10)]
print(b)

#for문으로 리스트 생성
a = []
for i in range(1, 10, 2):
    a.append(i*i)
print(a)

#컴프리헨션으로 리스트 생성
b = [i*i for i in range(1, 10) if i%2==1]
print(b)
```

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

```
#얕은 복사 : f2와 f1은 하나의 같은 리스트
f1 = [1, 2, 3, 4]
f2 = f1
f2.pop()
print("f1 :", f1)
print("f2 :", f2)
```

```
f1 : [1, 2, 3]
f2 : [1, 2, 3]
```

```
#깊은 복사 : 슬라이스[:], copy(), list()
#슬라이스[:]
f1 = [1, 2, 3, 4]
f2 = f1[:]
f2.pop()
print("f1 :", f1)
print("f2 :", f2)
print()

#copy()
f1 = [1, 2, 3, 4]
f2 = f1.copy()
f2.pop()
print("f1 :", f1)
print("f2 :", f2)
print()

#list()
f1 = [1, 2, 3, 4]
f2 = list(f1)
f2.pop()
print("f1 :", f1)
print("f2 :", f2)
```

```
f1 : [1, 2, 3, 4]
f2 : [1, 2, 3]

f1 : [1, 2, 3, 4]
f2 : [1, 2, 3]

f1 : [1, 2, 3, 4]
f2 : [1, 2, 3]
```

```
#문장 is : 피연산자 두 개가 동일한 메모리를 사용한다면 Ture, 아니면 False
f1 = [1, 2, 3, 4]
f2 = f1
```

```
print("f2 = f1 :", f1 is f2)
f3 = f1[:]
print("f3 = fl[:] :", f1 is f3)
```

```
f2 = f1 : True
f3 = fl[:] : False
```

```
#튜플 : 리스트와 달리 항목의 순서나 내용의 수정이 불가능
#항목에 제한 없음, 괄호 사이에 기술하지만 괄호 생략 가능
#튜플 생성(항목이 1개인 튜플 생성 시 마지막에 반드시 콤마를 붙여야함) : (), tuple()
a = ()
print(a)
b = tuple()
print(b)
c = 3,
print(c)
```

```
()
()
(3,)
```

```
#튜플 참조와 출력 : 리스트와 마찬가지로 tuple[index], tuple[start:stop:step]
#튜플 연결 + 연산자, 반복 * 연산자
na = '대한민국', '뉴질랜드', '캐나다'
print(na[2])

k = ('BTS', '블핑')
print(k + na)
print(k * 2)
```

```
캐나다
('BTS', '블핑', '대한민국', '뉴질랜드', '캐나다')
('BTS', '블핑', 'BTS', '블핑')
```

```
#내장 함수 sorted()를 이용하여 정렬, 정렬을 하게 되면 새로운 리스트로 반환
n = 1,5,98,4,6,5,9
print("n:", n)
print("sorted(n):", sorted(n))
print("sorted(n, reverse = True):", sorted(n, reverse = True))
print("n:", n)
```

```
n: (1, 5, 98, 4, 6, 5, 9)
sorted(n): [1, 4, 5, 5, 6, 9, 98]
sorted(n, reverse = True): [98, 9, 6, 5, 5, 4, 1]
n: (1, 5, 98, 4, 6, 5, 9)
```

```
#del tuple : 튜플 삭제, 메모리에서 제거되므로 참조하면 오류 발생
n = 1,5,98,4,6,5,9
del n
print(n)
print()
```

```
Traceback (most recent call last):
  File "D:\PythonProject\Chapter005.py", line 291, in <module>
    print(n)
NameError: name 'n' is not defined
```

- *프로젝트 Lab1 스포츠 종목과 팀원 수를 리스트로 적절히 출력*

```
#리스트 sports, num을 만든 뒤 첫 번째는 리스트 각각을 참조해서 출력, 두 번째는 중첩 리스트로 만든 뒤 출력,
#세 번째는 컴프리헨션을 이용해 중첩리스트를 만든 뒤 출력
sports = ["축구", "야구", "농구", "배구"]
num = [11, 9, 5, 6]

print(sports)
print(num)
for i in range(len(sports)):
    print("%s: %d명" %(sports[i], num[i]), end=" ")
print(); print()
```

```
l = [sports, num]
print(l)
for i in range(len(l[0])):
    print("%s: %d명" %(l[0][i], l[1][i]), end=" ")
print(); print()

l = [[sports[i], num[i]] for i in range(len(sports))]
print(l)
for i in l:
    print("%s: %d명" %(i[0], i[1]), end=" ")
```

```
--------------------------------------------------------------------------------
['축구', '야구', '농구', '배구']
[11, 9, 5, 6]
축구: 11명 야구: 9명 농구: 5명 배구: 6명

[['축구', '야구', '농구', '배구'], [11, 9, 5, 6]]
축구: 11명 야구: 9명 농구: 5명 배구: 6명

[['축구', 11], ['야구', 9], ['농구', 5], ['배구', 6]]
축구: 11명 야구: 9명 농구: 5명 배구: 6명
```

- 프로젝트 Lab2 햄버거 콤보 번호를 받아 주문 가격 표시

```
#반복문 while을 통해 주문 번호와 수량을 입력받은 뒤 공백여부를 확인하여 각각 주문 번호와 수량을 저장한 뒤
#조건문 if..elif를 통해 계산 결과를 출력하고 주문 번호가 0이라면 while문 종료 후 총 주문가격을 출력
menu = ("주문종료", "올인원팩", "투게더팩", "트리오팩", "패밀리팩")
price = (0, 6000, 7000, 8000, 100000)

msg = "주문할 콤보 번호와 수량을 계속 입력하세요!"
for i in range(len(menu)):
    msg += "\n\t %d %s" %(i, menu[i])
    if i != 0:
        msg += "%5d 원" %(price[i])
msg += "\n >> "

total = 0
more = True
while more:
    instr = input(msg)

    if instr.count(' ') > 0:
        order, cnt = instr.split()
        cnt = int(cnt)
    else:
        order = instr

    order = int(order)
```

```
    if order == 0:
        print()
        print("주문종료" .center(20, "*"))
        more = False
    elif 1 <= order <= 4:
        print("%s, %d개 주문" %(menu[order], cnt))
        sub = price[order] * cnt
        total += sub
        print("%s 주문 가격 %d, 총 가격 %d\n" %(menu[order], sub, total))
    else:
        print("모르겠어요. 다시 주문하세요.")
else:
    print("총 주문 가격 %d 원" %total)
    print("주문을 마치겠습니다.")
    print("안녕" .center(20, "*"))
```

```
--------------------------------------------------------------------------------
주문할 콤보 번호와 수량을 계속 입력하세요!
         0 주문종료
         1 올인원팩 6000 원
         2 투게더팩 7000 원
         3 트리오팩 8000 원
         4 패밀리팩100000 원
 >> 1 1
올인원팩, 1개 주문
올인원팩 주문 가격 6000, 총 가격 6000

주문할 콤보 번호와 수량을 계속 입력하세요!
         0 주문종료
         1 올인원팩 6000 원
         2 투게더팩 7000 원
         3 트리오팩 8000 원
         4 패밀리팩100000 원
 >> 3 2
트리오팩, 2개 주문
트리오팩 주문 가격 16000, 총 가격 22000

주문할 콤보 번호와 수량을 계속 입력하세요!
         0 주문종료
         1 올인원팩 6000 원
         2 투게더팩 7000 원
         3 트리오팩 8000 원
         4 패밀리팩100000 원
 >> 0

********주문종료********
총 주문 가격 22000 원
```

```
주문을 마치겠습니다.
*********안녕*********
```

- 도전 프로그래밍

```
01
from random import randint

list=[]
for i in range(10):
    n = randint(1, 99)
    list.append(n)

print("리스트", list)

list.sort()
print("정렬 리스트", list)

list.sort(reverse=True)
print("역순 리스트", list)
--------------------------------------------------------------------------
리스트 [68, 69, 47, 63, 91, 49, 35, 74, 66, 12]
정렬 리스트 [12, 35, 47, 49, 63, 66, 68, 69, 74, 91]
역순 리스트 [91, 74, 69, 68, 66, 63, 49, 47, 35, 12]
```

```
02
korean = ("정렬", "초보자", "내포", "사전")
english = ("sorting", "novice", "comprehension", "dictionary")

s = input("찾을 단어 입력 ? ")

check = False
for i in range(len(korean)):
    if s == korean[i]:
        check = True

if check == True:
    print(english[korean.index(s)])
else:
    print("단어가 없습니다.")
--------------------------------------------------------------------------
찾을 단어 입력 ? 초보자
novice
```

```
03
print("+ 012345678901")
print("  HelloPython!")
print("- 210987654321")
```

```
lis = list("HelloPython!")

more = True
while more:
    s = input("슬라이스[?:?:?] 3개 입력 >> ")
    a, b, c = s.split()
    start = int(a)
    stop = int(b)
    step = int(c)

    if start == 0 and stop == 0 and step == 0:
        more = False
    else:
        print(lis[start:stop:step])
else:
    print("종료" .center(30, "*"))
```

```
--------------------------------------------------------------------------------
+ 012345678901
  HelloPython!
- 210987654321
슬라이스[?:?:?] 3개 입력 >> 1 2 3
['e']
슬라이스[?:?:?] 3개 입력 >> 3 10 2
['l', 'P', 't', 'o']
슬라이스[?:?:?] 3개 입력 >> 0 0 0
**************종료**************
```

```
04
data = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]

rsum = []
csum = []
for i in range(3):
    r = 0
    for j in range(3):
        r += data[j][i]
    rsum.append(r)

for i in data:
    c = 0
    for j in i:
        c += j
    csum.append(c)

print("각 행의 합:", csum)
print("각 열의 합:", rsum)
```

```
--------------------------------------------------------------------------------
각 행의 합: [6, 15, 24]
각 열의 합: [12, 15, 18]
```

```
05
sports = ["축구", "야구", "농구", "배구"]
num = [11, 9, 5, 6]

print("메소드 insert()로 팀원 수 삽입")
sports.insert(1, num[0])
sports.insert(3, num[1])
sports.insert(5, num[2])
sports.insert(7, num[3])
print(sports, "\n")

print("메소드 insert()로 ''삽입")
for i in range(len(num)):
    sports.remove(num[i])
    sports.insert(i * 2 + 1, '')
print(sports, "\n")

print("슬라이스 sports[1::2]에 num을 대입")
sports[1::2] = num[:]
print(sports, "\n")
--------------------------------------------------------------------------------
메소드 insert()로 팀원 수 삽입
['축구', 11, '야구', 9, '농구', 5, '배구', 6]

메소드 insert()로 ''삽입
['축구', '', '야구', '', '농구', '', '배구', '']

슬라이스 sports[1::2]에 num을 대입
['축구', 11, '야구', 9, '농구', 5, '배구', 6]
```

```
06
m = [[1, 2], [3, 4], [5, 6], [7, 8]]

print("원 행렬(m) 출력:")
for i in m:
    for j in i:
        print(j, end=" ")
    print()
print()

print("전치 행렬 출력:")
for i in range(len(m[0])):
    for j in range(len(m)):
```

```
            print(m[j][i], end=" ")
        print()
```

```
원 행렬(m) 출력:
1 2
3 4
5 6
7 8

전치 행렬 출력:
1 3 5 7
2 4 6 8
```

```
07
m = [[1, 2], [3, 4], [5, 6], [7, 8]]
transpose = [[row[i] for row in m] for i in range(len(m[0]))]

print("트랜스포즈를 컴프리헨션으로 만들어 그대로 출력")
print(transpose, "\n")

print("트랜스포즈를 for문으로 출력")
for i in transpose:
    for j in i:
        print(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
```

```
08
from random import randint

l = []
for i in range(10):
    n = randint(1, 99)
    l.append(n)

t = tuple(l)
print("리스트:", l)
print("튜플:", t)
l = sorted(t)
print("튜플 정렬된 리스트:", l)
print()
```

```
total = 0
for i in l:
    total += i

print("합: %d, 항목수: %d" %(total, len(l)))
print("최대: %d, 최소: %d, 평균: %.2f" %(l[9], l[0], total/len(l)))
```

```
--------------------------------------------------------------------------------
리스트: [36, 78, 46, 64, 43, 71, 82, 64, 91, 44]
튜플: (36, 78, 46, 64, 43, 71, 82, 64, 91, 44)
튜플 정렬된 리스트: [36, 43, 44, 46, 64, 64, 71, 78, 82, 91]

합: 619, 항목수: 10
최대: 91, 최소: 36, 평균: 61.90
```

▶ chapter06

- 주요 내용 정리

```
#딕셔너리 : 키와 값의 쌍인 항목의 나열
#dct = {<key>: <value>, <key>: <value>, <key>: <value>....}
#항목 순서는 의미가 없으며 키는 중복 불가능, 키는 수정 할 수 없지만 값은 수정 가능, 값은 키로 참조
#딕셔너리 키는 수정 불가능한 객체는 모두 가능(정수, 실수 등)
#수정 불가능한 튜플은 딕셔너리 키로 사용가능 하지만 수정 가능한 리스트는 딕셔너리 키로 사용하지 못함

mycar = {"brand": "현대", "model": "제네시스", "year": 2016}
```

```
#빈 딕셔너리 생성 : {}, dict()
a = {}
b = dict()
b['first'] = 1
b['second'] = 2
print(a, b)
```

```
{} {'first': 1, 'second': 2}
```

```
#리스트 또는 튜플롤 구성된 키-값을 인자로 사용 : a = dict([]), a = dict(())
a = dict([[1, 2], [3, 4], [5, 6]])
print(a)
a = dict([(1, 2), (3, 4), (5, 6)])
print(a)
a = dict(([1, 2], [3, 4], [5, 6]))
print(a)
a = dict(((1, 2), (3, 4), (5, 6)))
print(a)
```

```
{1: 2, 3: 4, 5: 6}
{1: 2, 3: 4, 5: 6}
{1: 2, 3: 4, 5: 6}
{1: 2, 3: 4, 5: 6}
```

```
#p.212_예제 6-3
bts1 = {'그룹명': '방탄소년단', '인원수': 7, '리더': '김남준'}
bts1['소속사'] = '빅히트 엔터테인먼트'
bts2 = dict((('그룹명', '방탄소년단'), ('인원수', 7)))
bts3 = dict([['리더', '김남준'], ['소속사', '빅히트 엔터테인먼트']])

print(bts1)
print(bts2)
print(bts3)

#키가 단순 문자열이라면 키=값으로 나열 가능
bts = dict(그룹명='방탄소년단', 인원수=7, 리더='김남준', 소속사='빅히트 엔터테인먼트')

bts['구성원'] = ['RM', '진', '슈가', '제이홉', '지민', '뷔', '정국']
print(bts)
print(bts['구성원'])

{'그룹명': '방탄소년단', '인원수': 7, '리더': '김남준', '소속사': '빅히트 엔터테인먼트'}
{'그룹명': '방탄소년단', '인원수': 7}
```

```
{'리더': '김남준', '소속사': '빅히트 엔터테인먼트'}
{'그룹명': '방탄소년단', '인원수': 7, '리더': '김남준', '소속사': '빅히트 엔터테인먼트', '구성원': ['RM', '진', '슈가', '
제이홉', '지민', '뷔', '정국']}
['RM', '진', '슈가', '제이홉', '지민', '뷔', '정국']
```

```
#p.214_예제 6-4
m = {1: 'January', 2: 'Frebraury', 3: 'March', 4:'April'}
m[5] = 'May'
m[6] = 'June'
m[7] = 'July'
m[8] = 'August'
m[9] = 'September'
print(m, '\n')

from random import randint
for i in range(5):
    r = randint(1, 9)
    print("%d: %s" %(r, m[r]))
```

```
{1: 'January', 2: 'Frebraury', 3: 'March', 4: 'April', 5: 'May', 6: 'June', 7: 'July', 8: 'August', 9: 'September'}

7: July
4: April
8: August
9: September
6: June
```

```
#keys() : 키로만 구성된 리스트를 반환
d = dict(월='monday', 화='tuseday', 수='wendesday', 목='thursday')
print(d.keys(), '\n')

for i in d.keys():
    print("%s요일 %s" %(i, d[i]))
```

```
dict_keys(['월', '화', '수', '목'])

월요일 monday
화요일 tuseday
수요일 wendesday
목요일 thursday
```

```
#items() : (키, 쌍) 쌍의 튜플이 들어 있는 리스트를 반환
d = dict(월='monday', 화='tuseday',
         수='wendesday', 목='thursday')
print(d.items(), '\n')

for i in d.items():
    print("%s요일 %s" %(i[0], i[1]))
print()
```

```
dict_items([('월', 'monday'), ('화', 'tuseday'), ('수', 'wendesday'), ('목', 'thursday')])
월요일 monday
화요일 tuseday
수요일 wendesday
목요일 thursday
```

```
#p.217_예제 6-5
se = {'봄': 'spring', '여름': 'summer','가을': 'autumn', '겨울': 'winter'}
print(se.keys())
print(se.items())
print(se.values())

for i in se.keys():
    print("%s %s" %(i, se[i]))

for i in se.items():
    print("%s %s" %(i[0], i[1]), end="\t")
```

```
dict_keys(['봄', '여름', '가을', '겨울'])
dict_items([('봄', 'spring'), ('여름', 'summer'), ('가을', 'autumn'), ('겨울', 'winter')])
dict_values(['spring', 'summer', 'autumn', 'winter'])
봄 spring
여름 summer
가을 autumn
겨울 winter
봄 spring	여름 summer	가을 autumn	겨울 winter
```

```
#format(*i) : 자동으로 튜플이 나누어 출력
se = {'봄': 'spring', '여름': 'summer', '가을': 'autumn', '겨울': 'winter'}
for i in se.items():
    print("{} {}" .format(*i), end="\t")
print()
```

```
봄 spring	여름 summer	가을 autumn	겨울 winter
```

```
#키로 조회하는 메소드 get(키, 키가 없을 때 반환 값), 키가 없을 때 반환 값을 지정하지 않는다면 None
c = {'대한민국':'부산', '뉴질랜드':'웰링톤', '캐나다':'몬트리올'}
print(c.get('대한민국'))
print(c.get('미국', '없음'))
print(c.get('미국'))
```

```
부산
없음
None
```

```
#문장 del로 딕셔너리 항목 삭제, 모든 항목을 제거하는 메소드 clear()
c = {'대한민국':'부산', '뉴질랜드':'웰링톤', '캐나다':'몬트리올'}
del c['뉴질랜드']
print(c)
c = {'대한민국':'부산', '뉴질랜드':'웰링톤', '캐나다':'몬트리올'}
c.clear()
print(c)
```

```
{'대한민국': '부산', '캐나다': '몬트리올'}
{}
```

```
#update() : 딕셔너리 결합, 인자 딕셔너리에 원 딕셔너리와 동일한 키가 있다면 인자 딕셔너리의 값으로 대체
k = {'Samsumg Elec.':40000, 'Daum KAKAO':80000}
u = {'MS':150, 'Apple':180}
k.update(u)
print(k)
u.update({'MS':400})
print(u)
```

```
{'Samsumg Elec.': 40000, 'Daum KAKAO': 80000, 'MS': 150, 'Apple': 180}
{'MS': 400, 'Apple': 180}
```

```
#집합 : 중복되는 요소가 없으며 순서가 없는 원소의 모임(원소는 수정이 불가능한 것이어야 하며 중복을 허용하지 않고 순서는 의미가 없다.)
a = {1, 2, 3}
b = {(1, 2), (3, 4)}
print(a)
print(b)
```

```
#함수 set() : 집합 생성, 인자로 리스트 또는 튜플 자체를 사용 가능
# a = set([1, 2, [3, 4]]) -> unhashble 오류
# b = {[1, 2]} -> unhashble 오류
a = set()
print(a)
b = set('0123456')
print(b)
c = set([1, 2, 3, 3])
print(c)
d = set((3, 2, 1, 3))
print(d)
```

```
set()
{'3', '2', '0', '6', '1', '4', '5'}
{1, 2, 3}
{1, 2, 3}
```

```
#함수 add() : 원소 추가, remove(원소), discard(원소), pop() : 항목 삭제, clear() : 모든 원소 삭제
a = {1, 2, 3, 4, 5, 6}
a.add(8)
print(a)
a.remove(2)#remove() : 삭제하려는 원소가 없다면 keyerror
print(a)
a.discard(2)#discard() : 삭제하려는 원소가 없더라도 오류 발생 X
print(a)
a.pop()#pop() : 임의의 원소 삭제
print(a)
a.clear()
print(a)
```

```
{1, 2, 3, 4, 5, 6, 8}
{1, 3, 4, 5, 6, 8}
{1, 3, 4, 5, 6, 8}
{3, 4, 5, 6, 8}
set()
```

```
#합집합(|, union(), update()), 교집합(&, intersection()), 차집합(-, difference()), 여집합(^, symmetric_difference())
#합집합
a = {4, 6, 8, 10, 12}
b = {3, 6, 9, 12}
print('a|b\t\t\t\t', a|b)
a.union(b)#a자체가 수정되지 않음
print('a.union(b)\t\t\t', a.union(b))
a.update(b)#a자체가 수정
print('a.update(b)\t\t\t', a, '\n')

#교집합
a = {4, 6, 8, 10, 12}
b = {3, 6, 9, 12}
print('a&b\t\t\t\t', a&b)
a.intersection(b)#a자체가 수정되지 않음
print('a.intersection(b)\t\t', a.intersection(b), '\n')

#차집합(교환법칙 성립 X)
a = {4, 6, 8, 10, 12}
b = {3, 6, 9, 12}
print('a-b\t\t\t\t', a-b)
a.difference(b)#a자체가 수정되지 않음
print('a.difference(b)\t\t\t', a.difference(b), '\n')

#여집합
a = {4, 6, 8, 10, 12}
b = {3, 6, 9, 12}
print('a^b\t\t\t\t', a^b)
a.symmetric_difference(b)#a자체가 수정되지 않음
print('a.symmetric_difference(b)\t', a.symmetric_difference(b), '\n')
```

```
a|b                         {3, 4, 6, 8, 9, 10, 12}
a.union(b)                  {3, 4, 6, 8, 9, 10, 12}
a.update(b)                 {3, 4, 6, 8, 9, 10, 12}

a&b                         {12, 6}
a.intersection(b)           {12, 6}

a-b                         {8, 10, 4}
a.difference(b)             {8, 10, 4}

a^b                         {3, 4, 8, 9, 10}
a.symmetric_difference(b)   {3, 4, 8, 9, 10}
```

```
#합집합(|=), 교집합(&=), 차집합(-=), 여집합(^=) + _update()
#합집합 : a |= b, a.update(b)
```

```
a = {4, 6, 8, 10, 12}
b = {3, 6, 9, 12}
a.update(b)
print('a.update(b)\t', a)

#교집합 : a &= b, a.intersection_update(b)
a = {4, 6, 8, 10, 12}
b = {3, 6, 9, 12}
a &= b
print('a &= b\t', a)

#차집합(교환법칙 성립 X) : a -= b, a.difference_update(b)
a = {4, 6, 8, 10, 12}
b = {3, 6, 9, 12}
a -= b
print('a -= b\t', a)

#여집합 : a ^= b, a.symmetric_difference_update(b)
a = {4, 6, 8, 10, 12}
b = {3, 6, 9, 12}
a ^= b
print('a ^= b\t', a)
```

```
a.update(b)     {3, 4, 6, 8, 9, 10, 12}
a &= b          {12, 6}
a -= b          {4, 8, 10}
a ^= b          {3, 4, 8, 9, 10}
```

```
#내장함수 zip() : 여러 개의 튜플 항목 시퀀스를 각각의 리스트로 묶어 주는 역할, 자료형은 zip이고 간단히 리스트나 튜플로 변환 가능
a = ['ftp', 'telnet', 'smtp', 'dns']
b = (20, 23, 25, 53)
print(list(zip(a, b)))
print(tuple(zip(a, b)))
print(list(zip('ab', a, b)))
print(dict(zip(a, b)))
```

```
[('ftp', 20), ('telnet', 23), ('smtp', 25), ('dns', 53)]
(('ftp', 20), ('telnet', 23), ('smtp', 25), ('dns', 53))
[('a', 'ftp', 20), ('b', 'telnet', 23)]
{'ftp': 20, 'telnet': 23, 'smtp': 25, 'dns': 53}
```

```
#내장함수 enumerate() : 첨자를 자동으로 만들어 첨자와 값과의 쌍인 튜플을 만들어 주는 내장함수
#첨자는 자동으로 0부터 시작하며, 키워드 인자 start를 이용해 시작 첨자를 바꿀 수 있다
#for문 사용 시 용이
a = ['ftp', 'telnet', 'smtp', 'dns']
print(list(enumerate(a)))
print(list(enumerate(a, start=1)))
print()

for s in enumerate(a):
    print('lst[{}]: {}'.format(s[0], s[1]))
    print('lst[{}]: {}'.format(*s))
print()

for i, name in enumerate(a):
    print('lst[{}]: {}'.format(i, name))
```

```
[(0, 'ftp'), (1, 'telnet'), (2, 'smtp'), (3, 'dns')]
[(1, 'ftp'), (2, 'telnet'), (3, 'smtp'), (4, 'dns')]

lst[0]: ftp         lst[0]: ftp
lst[1]: telnet      lst[1]: telnet
lst[2]: smtp        lst[2]: smtp
lst[3]: dns         lst[3]: dns

lst[0]: ftp
lst[1]: telnet
lst[2]: smtp
lst[3]: dns
```

```
#시퀀스 간의 변환
#튜플 -> 리스트 반환
a = (20, 23, 25, 53)
print(a)
print(list(a), '\n')

#리스트 -> 튜플 반환
a = ['ftp', 'telnet', 'smtp', 'dns']
print(a)
print(tuple(a), '\n')

#리스트 -> 집합 반환
a = ['ftp', 'telnet', 'smtp', 'dns', 'dns']
print(a)
print(set(a), '\n')

#딕셔너리 -> 리스트 or 튜플 or 집합 반환 : 항목이 키로만 구성되어 반환
a = dict(일월='소나무', 이월='매화', 삼월='벗꽃', 사월='등나무')
print(a)
print(list(a))
print(tuple(a))
print(set(a))
```

```
(20, 23, 25, 53)
[20, 23, 25, 53]

['ftp', 'telnet', 'smtp', 'dns']
('ftp', 'telnet', 'smtp', 'dns')

['ftp', 'telnet', 'smtp', 'dns', 'dns']
{'dns', 'telnet', 'ftp', 'smtp'}

{'일월': '소나무', '이월': '매화', '삼월': '벗꽃', '사월': '등나무'}
['일월', '이월', '삼월', '사월']
('일월', '이월', '삼월', '사월')
{'사월', '삼월', '일월', '이월'}
```

- *프로젝트 Lab1 철수와 영희의 가위바위보 게임*

```
#My
#딕셔너리를 통해 가위바위보의 승패를 저장, 튜플을 통해 가위, 바위, 보를 저장
#choice를 통해 임의로 선택하고 if문을 통해 승패를 출력
v = {'가위': '보오', '바위':'가위', '보오':'주먹'}
s = ('가위', '바위', '보오')

print('*' * 20)
print('철수\t영희\t승자')
print('*' * 20)

from random import choice
for i in range(10):
    a, b = choice(s), choice(s)
    if a == b:
        print('%s\t%s\t%s' %(a, b, '비김'))
    else:
        if v[a] == b:
            print('%s\t%s\t%s' %(a, b, '철수'))
        else:
            print('%s\t%s\t%s' %(a, b, '영희'))
----------------------------------------------------------------------
********************
```

```
#Book
from random import choice
dcs = {'가위': '보오', '바위':'가위', '보오':'주먹'}
rsp = ('가위', '바위', '보오')
tit = ['비김', '철수', '영희', '승자']

print('*' * 17)
print('{:4} {:4} {:4}'.format(tit[1], tit[2], tit[3]))
print('*' * 17)
```

```
for _ in range(10):
    c, y = choice(rsp), choice(rsp)
    print('{:4} {:4}'.format(c, y), end=' ')
    if c == y:
        idx = 0
    elif dcs[c] == y:
        idx = 1
    else:
        idx = 2
    print('{:4}'.format(tit[idx]))
```

```
철수     영희     승자
**********
보오     바위     영희
보오     가위     영희
가위     보오     철수
보오     바위     영희
바위     보오     영희
보오     바위     영희
보오     바위     영희
가위     바위     영희
보오     바위     영희
가위     바위     영희
```

- 프로젝트 Lab2 딕셔너리로 만드는 K-pop 차트

```
#내장 함수 zip()을 이용해 리스트를 생성해서 출력하고 내장 함수 enumerate를 이용해 딕셔너리 생성 후 출력
singer = ['BTS', '볼빨간사춘기', 'BTS', '블랙핑크', '태연', 'BTS']
song = ['작은 것들을 위한 시', '나만 봄', '소우주', 'Kill This Love', '사계']

kpop = list(zip(singer, song))
print(kpop)

kpchart = dict(enumerate(kpop, start=1))
print(kpchart)
print()

import pprint
pprint.pprint(kpchart)
--------------------------------------------------------------------------------
[('BTS', '작은 것들을 위한 시'), ('볼빨간사춘기', '나만 봄'), ('BTS', '소우주'), ('블랙핑크', 'Kill This Love'), ('태연', '사계')]
{1: ('BTS', '작은 것들을 위한 시'), 2: ('볼빨간사춘기', '나만 봄'), 3: ('BTS', '소우주'), 4: ('블랙핑크', 'Kill This Love'), 5: ('태연', '사계')}

{1: ('BTS', '작은 것들을 위한 시'),
 2: ('볼빨간사춘기', '나만 봄'),
 3: ('BTS', '소우주'),
 4: ('블랙핑크', 'Kill This Love'),
 5: ('태연', '사계')}
```

- 도전 프로그래밍

```
01
# d = dict([('이름', '김영희'), ('전화번호', '010-3017-4468'), ('성별', '여자'), ('나이', 22), ('대학교', '한국대학교')])
d = dict(이름='김영희', 전화번호='010-3017-4468', 성별='여자', 나이='22', 대학교='한국대학교')

for i in d.keys():
    print('%s: %s' %(i, d[i]))
```

```
--------------------------------------------------------------------------------
이름: 김영희
전화번호: 010-3017-4468
성별: 여자
나이: 22
대학교: 한국대학교
```

02

```
d = dict(삼성에스디에스=242000, 삼성전자=47000, 엔씨소프트=52600, 핸디소프트=5120, 골프존=215000, 기아=56300)

print(d)
while True:
    print()
    name = input('주식 이름 ? ')
    result = d.get(name, '주식 이름이 없습니다.')

    if result != '주식 이름이 없습니다.':
        print('%s: %d' %(name, d[name]))
    else:
        print(result)
        break
--------------------------------------------------------------------------------
{'삼성에스디에스': 242000, '삼성전자': 47000, '엔씨소프트': 52600, '핸디소프트': 5120, '골프존': 215000, '기아': 56300}

주식 이름 ? 골프존
골프존: 215000

주식 이름 ? 삼성
주식 이름이 없습니다.
```

03

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

for i in books.keys():
    print('책 이름: %s\t저자: ' %i, end=' ')
    for j in books[i]:
        print(j, end=' ')
    print()
--------------------------------------------------------------------------------
책 이름: 파이썬 개론      저자:  강환수
책 이름: Perfect C       저자:  강환수 이동규
책 이름: 컴퓨터 개론      저자:  강환수 조진형 신용현
```

04

```
from random import choice
```

```
d = {'가위': '보오', '바위': '가위', '보오': '바위'}
rsp = ('가위', '바위', '보오')

print('*' * 17)
print('{:4} {:4} {:4}'.format('철수', '영희', '승자'))
print('*' * 17)

cv, yv, v = 0, 0, 0
for i in range(20):
    c, y = choice(rsp), choice(rsp)

    print('{:4} {:4}'.format(c, y), end=' ')

    if c == y:
        v += 1
        print('{:4} {}'.format('비김', v))
    elif d[c] == y:
        cv += 1
        print('{:4} {}'.format('철수', cv))
    else:
        yv += 1
        print('{:4} {}'.format('영희', yv))
print()
print('총 게임 회수: {} 비긴 회수: {}'.format(20, v))
print('철수 승률: {:.2f}'.format(float(cv)/float(20 - v)))
print('영희 승률: {:.2f}'.format(float(yv)/float(20 - v)))
```

```
*****************
철수   영희   승자
*****************
가위   바위   영희   1
가위   가위   비김   1
바위   보오   영희   2
바위   바위   비김   2
보오   가위   영희   3
바위   보오   영희   4
바위   바위   비김   3
가위   보오   철수   1
가위   보오   철수   2
가위   바위   영희   5
바위   바위   비김   4
가위   바위   영희   6
보오   바위   철수   3
바위   보오   영희   7
가위   보오   철수   4
보오   가위   영희   8
보오   가위   영희   9
```

```
가위    보오    철수    5
바위    가위    철수    6
보오    가위    영희    10

총 게임 회수: 20 비긴 회수: 4
철수 승률: 0.38
영희 승률: 0.62
```

05

```
fruits = ['apple', 'banana', 'grapes', 'pear']
prices = (1000, 500, 1200, 1500)

print(dict(zip(fruits, prices)))
print()

d = list(zip(fruits, prices))
d = dict(enumerate(d, start=1))

for i in d.keys():
    print(i, end=' ')
    cnt = 0
    for j in d[i]:
        cnt += 1
        if cnt == 2:
            print('가격: %s' %j, end=' ')
        else:
            print(j, end=' ')
    print()
```

```
{'apple': 1000, 'banana': 500, 'grapes': 1200, 'pear': 1500}

1 apple 가격: 1000
2 banana 가격: 500
3 grapes 가격: 1200
4 pear 가격: 1500
```

06

```
from random import sample

A = set(sample(list(range(1, 21)), 5))
B = set(sample(list(range(1, 21)), 5))

print('A =', A)
print('B =', B)
print()

print('A | B =', A | B)
```

```
print('A & B =', A & B)
print('A - B =', A - B)
print('A ^ B =', A ^ B)
--------------------------------------------------------------------------------
A = {1, 2, 9, 10, 12}
B = {3, 5, 12, 16, 19}

A | B = {1, 2, 3, 5, 9, 10, 12, 16, 19}
A & B = {12}
A - B = {1, 2, 10, 9}
A ^ B = {1, 2, 3, 5, 9, 10, 16, 19}
```

▶ 마무리 및 소감

- *마무리*

chapter01에서는 파이썬이 무엇인지 그리고 파이썬의 장점 및 단점, 파이썬이 필요한 이유 및 어느 분야에서 사용하는지, 얼마만큼의 발전 가능성이 있는지 등에 대한 내용을 통해 파이썬이라는 프로그래밍 언어에 좀 더 친숙하게 다가갈 수 있었다고 생각합니다. chapter02 ~ 06은 주요 내용 정리, 프로젝트 Lab, 도전 프로그래밍으로 나눠서 정리하였습니다. 주요 내용 정리 부분은 chapter 별로 공부를 한 뒤 주요하게 다룬다고 생각되는 부분을 실습하고 실행한 결과를 첨부하였으며 프로젝트 Lab 부분은 기본적으로 스스로 문제를 해결하려고 했으며 그 후 책의 코드와 비교하며 다른 풀이법이 있는지 공부를 했고 도전 프로그래밍 부분은 지금까지 배운 것을 바탕으로 각 chapter 별로 잘 공부했는지 확인하고 복습하기 위한 것이라고 생각하며 풀었습니다.

- *소감*

이번 과제를 통해 지금까지 배운 내용에 대해 다시 한번 더 정리하는 시간을 가질 수 있어서 좋았습니다. 정리하는 과정에서 내가 이해하지 못한 부분과 어려워하는 부분을 확인하고 다시 공부하는 과정에서 실습과 복습을 통해 개념을 잡을 수 있었고 이해가 되었던 부분도 실습을 통해 복습할 수 있는 기회가 되었다고 생각합니다. 아직도 조금은 어색하지만 파이썬만의 장점인 좀 더 직관적이고 간결하고 쉽게 접근하는 점과 풍부한 라이브러리를 제공하는 점 등을 통해 막연하게 파이썬은 좋은 프로그래밍 언어이고 현재 많은 실무 분야에서 사용 중이라는 얘기를 들어도 그렇게 와 닿지는 않았지만 이번 포트폴리오를 만들게 되면서 왜 전망이 좋은 프로그래밍 언어이고 교육 분야뿐만 아니라 실무 분야에서 활용되고 있는지 알게 되었습니다.

끝까지 읽어주셔서
감사합니다.