

---

# 파이썬 중간고사 포트폴리오

---

20192614

전지은

---

# 목차

---

- 01 '파이썬'이란?
- 02 '파이썬' 설치 방법
- 03 '파이썬' 연습 예제
- 04 마무리



V

01

'파이썬' 이란?

# 1.개요

파이썬은 1991년에 발표된 인터프리터 방식의 프로그래밍 언어입니다.

창시자는 귀도 반 로섬(Guido van Rossum).

2000년에는 Python 2, 2008년에는 Python 3가 나왔습니다.

문법이 매우 쉬운 까닭에, 초보자들이 처음 프로그래밍을 배울 때 추천되는 언어입니다.

학습용으로도 좋은 언어인 동시에 실사용률과 생산성도 높은 강력한 언어이기도 합니다.

## 2. 특징

1. **무료** – 소프트웨어 재단에서 관리하며 무료로 사용이 가능합니다.
2. **높은 생산성** – c언어가 성능에 초점을 두었다면, 파이썬은 생산성에 초점을 두었습니다.
3. **가독성** – print 함수 한줄 사용으로 출력이 가능하여 간결하고, 가독성이 좋습니다.
4. **쉬운 접착성** – 다양한 언어로 되어있는 라이브러리를 쉽게 붙일 수 있습니다.
5. **풍부한 라이브러리** – 10만개 이상의 라이브러리를 다운로드 할 수 있습니다.
6. **유니코드** – 문자열을 유니코드로 처리하기 때문에, 한글,중국어,영어 모두 문제없이 처리합니다.
7. **동적 타이핑** – 런타임 시에 타입 체크를 하는 동시에 자동으로 메모리 관리를 합니다.

## 3. 활용 분야



### 1.YOUTUBE, GMAIL

웹서버 측 코드가 파이썬으로 구현되어 있습니다.



### 2.DROPBOX

웹서비스에 개발 언어로 사용합니다.



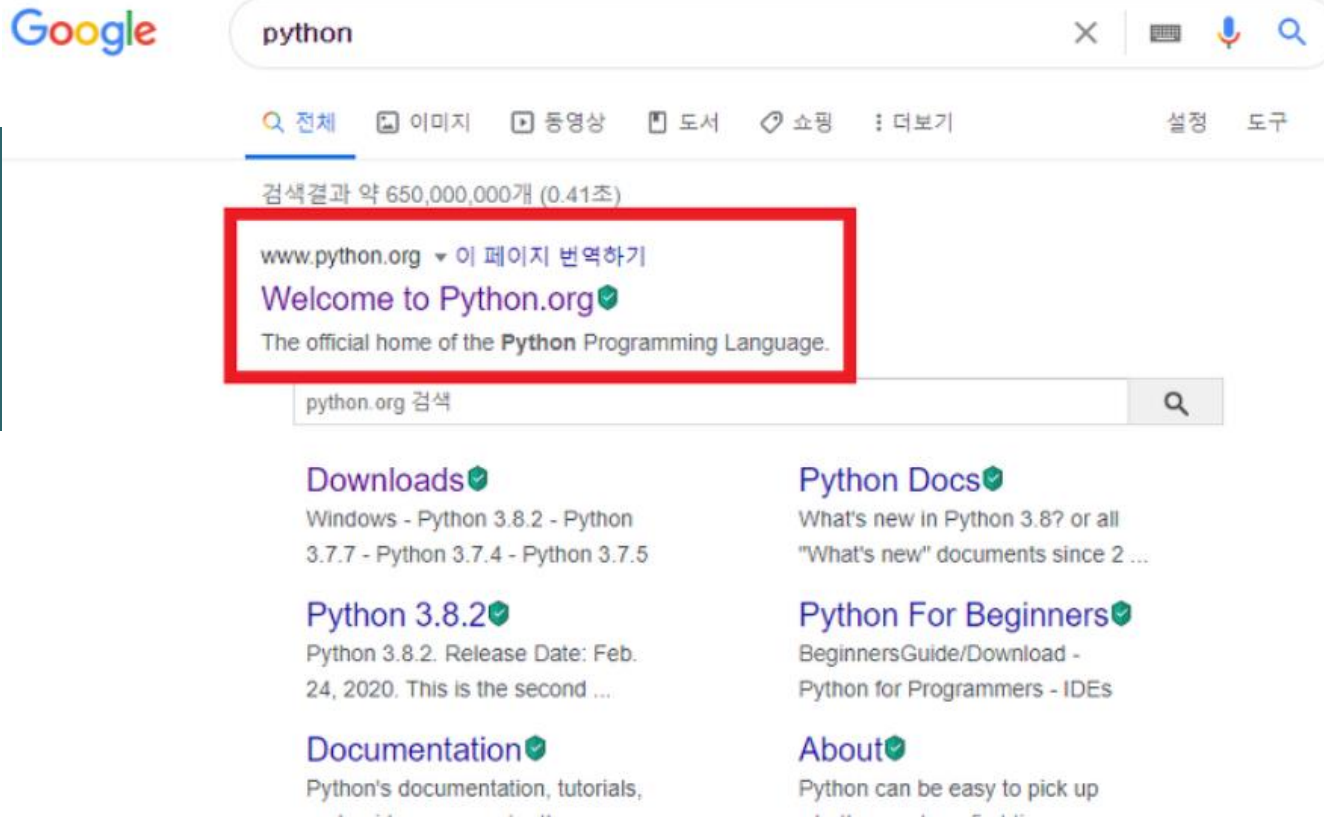
### 3.Machine learning library

인공지능에서 사용되는 대부분의 머신러닝 라이브러리들에서 기본 언어로 사용합니다.

V

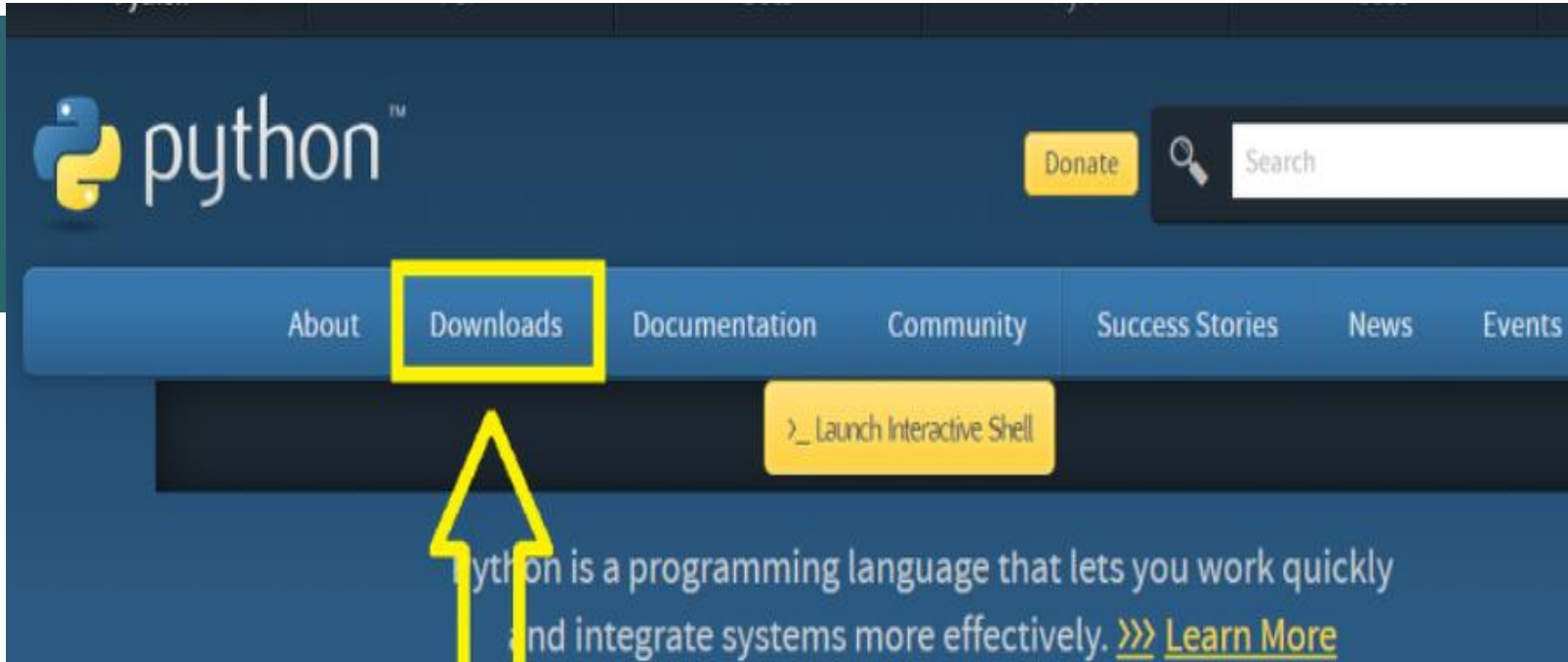
02

'파이썬' 설치 방법



1. 검색창에 파이썬을 검색합니다.

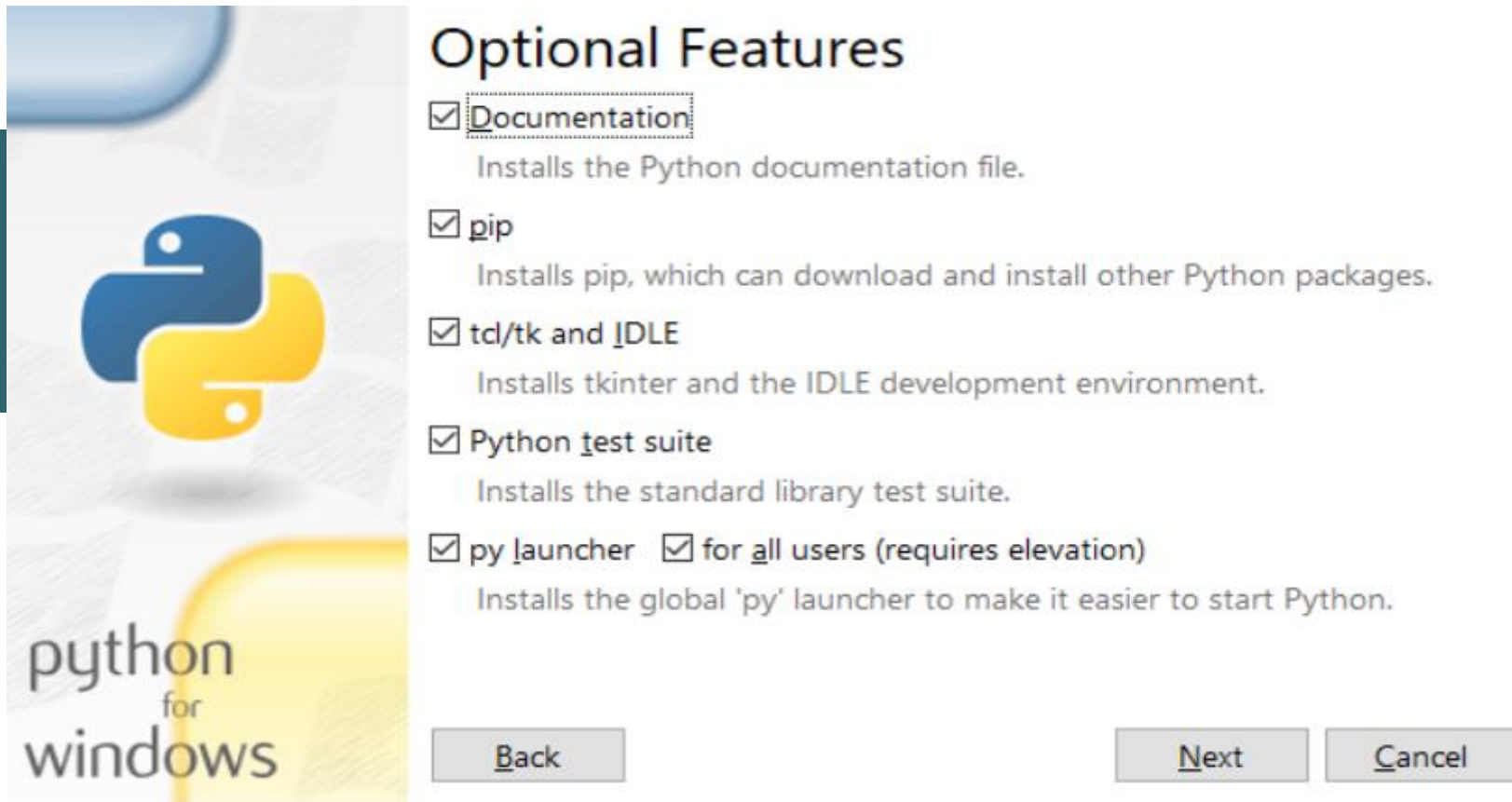




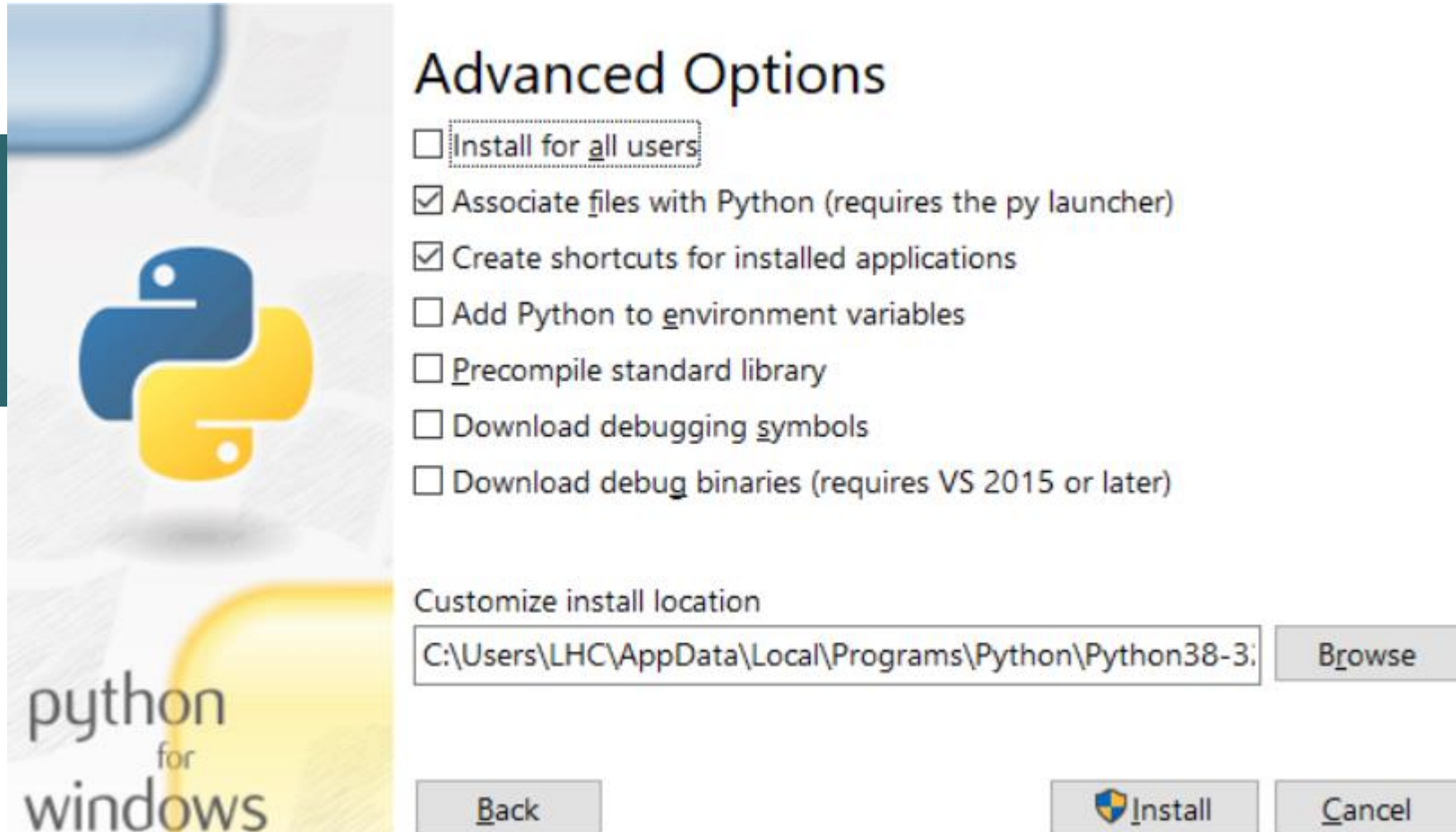
2. 홈페이지에서 원하는 버전을 찾아 다운로드를 합니다.



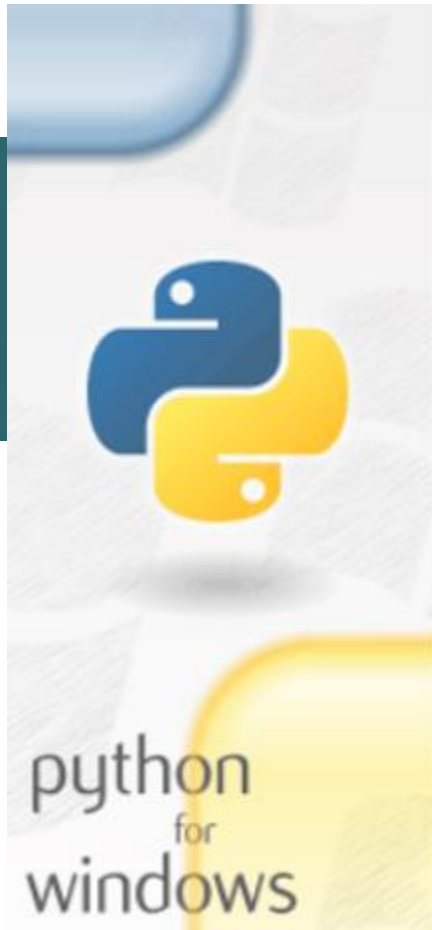
3. 고급설정을 묻는 창이 뜨면, 원하는 옵션을 선택합니다.



4. 설치 시에 어떠한 기능을 설치할지 선택하는 창이 나옵니다.  
원하는 기능을 체크하고 넘어갑니다.



5. 원하는 경로를 선택합니다.



## Setup was successful

Special thanks to Mark Hammond, without whose years of freely shared Windows expertise, Python for Windows would still be Python for DOS.

New to Python? Start with the [online tutorial](#) and [documentation](#).

See [what's new](#) in this release.

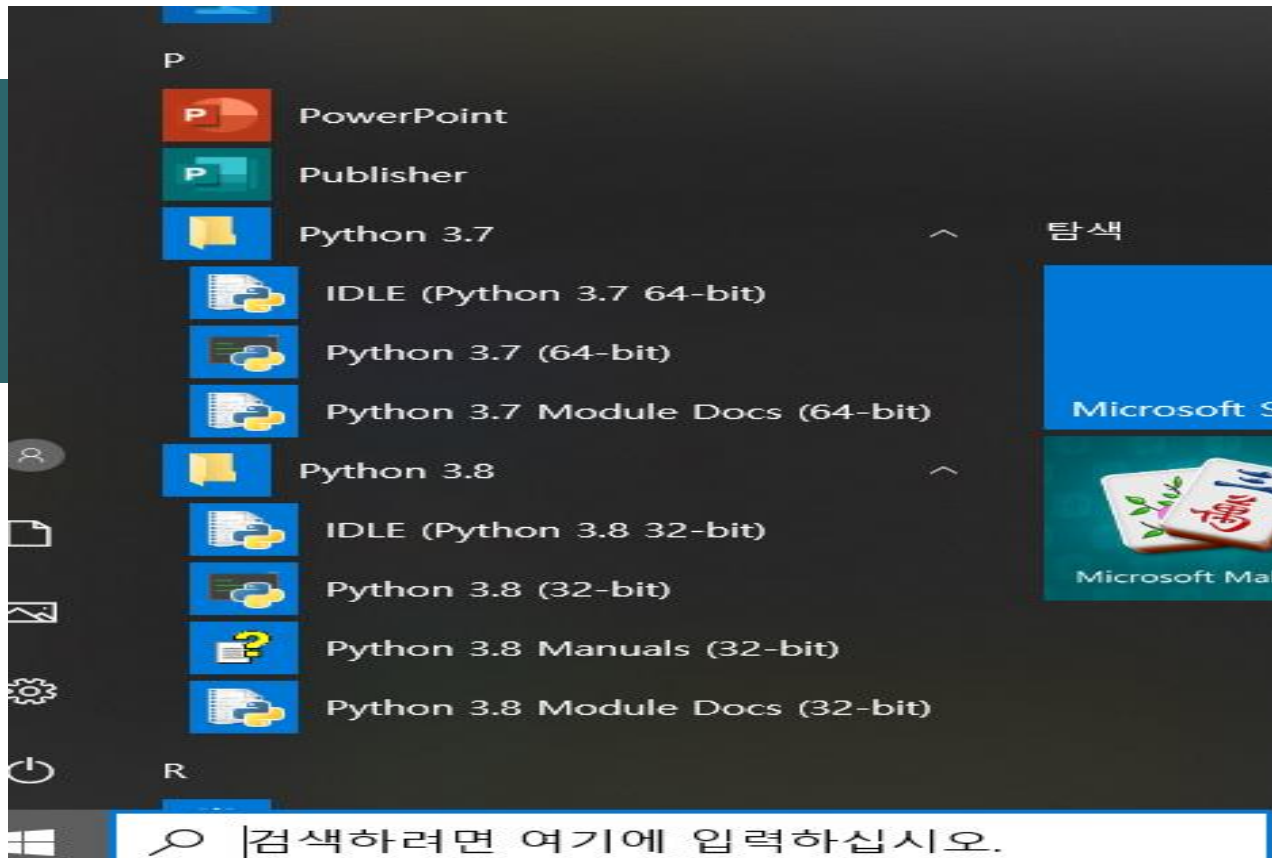


### Disable path length limit

Changes your machine configuration to allow programs, including Python, to bypass the 260 character "MAX\_PATH" limitation.

Close

설치가 완료되었습니다!



설치된 모습을 확인하실 수 있습니다.

V

03

'파이썬' 연습 예제

## 02-01. 다양한 자료: 문자열과 수

```
''' 01-02 comments.py  
    2019 3. kang hwan soo '''
```

```
print('# 이후는 주석')  
print('string: "python"')  
print("number: 1 5 3.14")  
print("string: 'python'")
```

```
Python 3.8.2 (tags/v3.8  
tel)] on win32  
Type "help", "copyright"  
>>>
```

```
===== RE  
# 이후는 주석  
string: "python"  
number: 1 5 3.14  
string: 'python'  
>>>
```



## 02-01. 다양한 자료: 문자열과 수

```
print(1, 2, -5, 3.14, 2.71828)
print('Hi', 'Python!')
```

```
print('23000원은', '5000원 ?개', '1000원 ?개')
print('5000원', 23000//5000, '개')
print('1000원', (23000%5000)//1000, '개')
```

```
1 2 -5 3.14 2.71828
```

```
Hi Python!
```

```
23000원은 5000원 ?개 1000원 ?개
```

```
5000원 4 개
```

```
1000원 3 개
```

```
>>>
```

## 02-02. 변수와 키워드, 대입 연산자

```
distance = 384400
unit = 10000
manUnit, remainder = divmod(distance, unit)
print('지구에서 달까지의 거리:', manUnit, '만', remainder, '킬로미터')
```

```
>>>
===== RESTART: C:/Users/8210
지구에서 달까지의 거리: 38 만 4400 킬로미터
>>>
```

## 02-03. 자료의 표준 입력과 자료 변환 함수

```
age = input('나이는?')  
print('실제 나이는', int(age) - 1)
```

```
-----  
나이는?23  
실제 나이는 22  
>>>
```

## 02-03. 자료의 표준 입력과 자료 변환 함수

```
invar = input('16진수 정수 입력>>')
data = int(invar,16)
print('2진수:', bin(data))
print('8진수:', oct(data))
print('10진수:', data)
print('16진수:', hex(data))
```

```
16진수 정수 입력>>32
2진수: 0b110010
8진수: 0o62
10진수: 50
16진수: 0x32
>>>
```

HEX to BIN

```

일요일 기러기
일요일 기러기
일요일 기러기 일요일 기러기 일요일 기러기
일일기
일기
기기일일
기러기 일요일
>>>

```

## 03-02. 문자열 관련 메소드

```
m,n,x,y=input('4개의 수 입력>>').split()
a,b,c,d = float(m), float(n), float(x), float(y)
print('입력값:',a,b,c,d)
sum = a+b+c+d
print('합:',sum, '평균:', sum/4)
print('최대:', max(a,b,c,d), '최소:',min(a,b,c,d))
```

```
4개의 수 입력>>3.7 5.8 9 2.5
입력값: 3.7 5.8 9.0 2.5
합: 21.0 평균: 5.25
최대: 9.0 최소: 2.5
>>>
```

## 03-03. 논리 자료와 다양한 연산

```
usage = float(input('가정의 전기 사용량(kwh)은>>'))
less200 = usage <=200
less400 = 200 < usage <=400
greater400 = 400 < usage

base = 730*less200 + 1260*less400 +6060*greater400
print('전기 사용량(kw): %d, 기본요금(원): %d' % (usage,base))
```

```
----- RLST8NT: C:\Users\702
가정의 전기 사용량(kwh)은>>180
전기 사용량(kw): 180, 기본요금(원): 730
>>>
```

## 03-03. 논리 자료와 다양한 연산

```
orgpwd = int(input('ID로 사용할 여덟자리의 정수를 입력하세요>>'))
keymask = 27182818
encpwd = orgpwd ^ keymask
print('입력한 ID: %d'% encpwd)
print('암호화해 저장된 ID: %d' % encpwd)
inpwd = int(input('로그인할 ID를 입력하세요>>'))
result = encpwd ^ keymask
print('로그인 성공: {}'.format(inpwd == result))
```

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



[illegible]

## 04-01. 조건에 따른 선택 if .... else

```
grade= float(input('1학기 평균 평점은?'))
if 3.8 <=grade:
    print('축하합니다! 장학금 지금 대상자이다.')
print('당신의 1학기 평균 평점 %.2f이다'%(grade))
```

```
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb
tel)] on win32
Type "help", "copyright", "credits" or
>>>
==== RESTART: C:\Users\82104\Desktop\2
1학기 평균 평점은?3.9
축하합니다! 장학금 지금 대상자이다.
당신의 1학기 평균 평점 3.90이다
>>>
```

## 04-01. 조건에 따른 선택 if .... else

```
from time import localtime
hour = localtime().tm_hour
mnt = localtime().tm_min

if hour < 10:
    print('지금시각: %d시 %d분, 조조 할인 된다.' % (hour, mnt))
else:
    print('지금시각: %d시 %d분, 조조 할인 안된다.' % (hour, mnt))
```

```
Python 3.7.4 (tags/v3.7.4:e09359112e,
(AMD64)] on win32
Type "help", "copyright", "credits" o
>>>
```

```
RESTART: C:\Users\W82104\Desktop\2020
```

```
지금시각: 0시 44분, 조조 할인 된다.
>>>
```

## 04-01. 조건에 따른 선택 if .... else

```
n = int(input('정수입력>>'))  
if n%2 == 0:  
    print('%d은 짝수다.'% n)  
else:  
    print('%d은 홀수다.'% n)
```

```
Python 3.7.4 (tags/v3.7.4:09000000, Sep 16 2019, [AMD64]) on win32  
Type "help", "copyright", "credits()" or "quit()" for more  
>>>  
== RESTART: C:\Users\W82104\Desktop>>>  
정수입력>>163  
163은 홀수다.  
>>>
```

## 04-01. 조건에 따른 선택 if .... else

```
category = int(input('원하는 음료는? 1. 커피 2. 주스'))

if category == 1:
    menu = int(input('번호 선택? 1. 아메리카노 2. 카페라테 3. 카푸치노'))
    if menu == 1:
        print('1. 아메리카노 선택')
    elif menu == 2:
        print('2. 카페라테 선택')
    elif menu == 3:
        print('3. 카푸치노 선택')
else:
    menu = int(input('번호 선택? 1. 키위주스 2. 토마토주스 3. 오렌지주스'))
    if menu == 1:
        print('1. 키위주스 선택')
    elif menu == 2:
        print('2. 토마토주스 선택')
    elif menu == 3:
        print('3. 오렌지주스 선택')
```

```
Python 3.7.4 (tags/v3.7.4:e09359112e, Jul 8 2019, 20:
(AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for
>>>
== RESTART: C:\Users\82104\Desktop\2020 python code\ch
원하는 음료는? 1. 커피 2. 주스1
번호 선택? 1. 아메리카노 2. 카페라테 3. 카푸치노2
2. 카페라테 선택
>>>
>>>
```

## 04-02. 반복을 제어하는 for문과 while 문

```
n = input("10진수의 한 자릿수 입력 >>")
print('두 자릿수 정수에서 최소 한 자릿수가 %s인 정수 찾기: ' %n)
print(' 결과 '.center(50, '='))

for i in range(10,100):
    snum = str(i)
    if n in snum:
        print(i, end= ' ')
```

```
Python 3.7.4 (tags/v3.7.4:e09359112e, Jul  8 2019, 20:3
(AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for
>>>
=== RESTART: C:\Users\82104\Desktop\2020 python code\cd
10진수의 한 자릿수 입력 >>9
두 자릿수 정수에서 최소 한 자릿수가 9인 정수 찾기:
===== 결과 =====
19 29 39 49 59 69 79 89 90 91 92 93 94 95 96 97 98 99
>>>
```

## 04-02. 반복을 제어하는 for문과 while 문

```
menu = '''버거, 콤보 번호로 주문하세요!
0. 주문 종료
1. 올인원팩
2. 루게더팩
3. 트리오팩
4. 패밀리팩
>> '''
```

```
more = True
while more:
    order = int(input(menu))
    if order == 1:
        print('%s 주문' % '올인원팩')
    elif order == 2:
        print('%s 주문' % '루게더팩')
    elif order == 3:
        print('%s 주문' % '트리오팩')
    elif order == 4:
        print('%s 주문' % '패밀리팩')
    elif order == 0:
        print('주문 종료'.center(20, '*'))
        more = False
    else:
        print('모르겠어요.')
```

```
Python 3.7.4 (tags/v3.7.4:e09359112e, Jul 8 2019, 20:34:20) [MSC v.1916 64 b
(AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
```

```
>>>
```

```
= RESTART: C:\Users\#82104\Desktop\2020 python code\ch04\04-11burgerorder.py =
버거, 콤보 번호로 주문하세요!
```

```
0. 주문 종료
1. 올인원팩
2. 루게더팩
3. 트리오팩
4. 패밀리팩
>> 4
```

```
패밀리팩 주문
버거, 콤보 번호로 주문하세요!
```

```
0. 주문 종료
1. 올인원팩
2. 루게더팩
3. 트리오팩
4. 패밀리팩
>> 0
```

```
*****주문 종료*****
```

```
주문을 마치겠습니다.
```

```
>>>
```

## 04-02. 반복을 제어하는 for문과 while 문

```
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
>>>
```



## 04-03. 임의의 수인 난수와 반복을 제어하는 break, continue문

```

winnumber = 11, 17, 28, 30, 33, 35
print(' 모의 로또 당첨 번호 '.center(28, '='))
print(winnumber)
print()
print(' 내 번호 확인 '.center(30, '-'))
cnt = 0
import random
for i in range(6):
    n = random.randint(1, 45)
    if n in winnumber:
        print(n, 'O ', end = ' ')
        cnt += 1
    else:
        print(n, 'X ', end = ' ')

print()
print(cnt, '개 맞음')

```

```

Python 3.7.4 (tags/v3.7.4:e09359112e,
(AMD64)] on win32
Type "help", "copyright", "credits" or
>>>
==== RESTART: C:\Users\82104\Desktop\2
===== 모의 로또 당첨 번호 =====
(11, 17, 28, 30, 33, 35)

----- 내 번호 확인 -----
38 X  13 X  2 X  26 X  22 X  26 X
0 개 맞음
>>>

```

## 04-03. 임의의 수인 난수와 반복을 제어하는 break, continue문

```

days = ['monday', 'tuesday', 'wednesday']

while True:
    user = input('월,화,수 중 하나 영어 단어 입력 >>')
    if user not in days:
        print('잘못 입력했어요!')
        continue
    print('입력: %s, 철자가 맞습니다.' % user)
    break

print(' 종료 '.center(15, '*'))

```

```

Python 3.7.4 (tags/v3.7.4:e09359112e, Jul
(AMD64)] on win32
Type "help", "copyright", "credits" or "!"
>>>
= RESTART: C:\Users\W82104\Desktop\2020 py
월,화,수 중 하나 영어 단어 입력 >>수
잘못 입력했어요!
월,화,수 중 하나 영어 단어 입력 >>monday
입력: monday, 철자가 맞습니다.
***** 종료 *****
>>>

```

## 05-01. 여러 자료 값을 편리하게 처리하는 리스트

```
goods = []  
for i in range(3):  
    item = input('구입할 품목은 ? ')  
    goods.append(item)  
    print(goods)  
print('길이: %d' % len(goods))
```

```
Python 3.7.4 (tags/v3.7.4:e09359112e  
(AMD64)) on win32  
Type "help", "copyright", "credits"  
>>>  
==== RESTART: C:\Users\W82104\Desktop  
구입할 품목은 ? 책상  
['책상']  
구입할 품목은 ? 의자  
['책상', '의자']  
구입할 품목은 ? 파이썬 교재  
['책상', '의자', '파이썬 교재']  
길이: 3  
>>>
```

## 05-01. 여러 자료 값을 편리하게 처리하는 리스트

```
rsp = ['가위', '바위', '보']
for i in range(len(rsp)):
    print(rsp[i], end=' ')
print()

from random import choice
print('컴퓨터의 가위 바위 보 5개')
for i in range(5):
    print(choice(rsp))
```

```
Python 3.7.4 (tags/v3.7.4:e093
(AMD64)] on win32
Type "help", "copyright", "cre
>>>
=== RESTART: C:\Users\82104\De
가위 바위 보
컴퓨터의 가위 바위 보 5개
바위
바위
보
가위
가위
>>>
```

## 05-01. 여러 자료 값을 편리하게 처리하는 리스트

```

food = ['짜장면', '짬뽕', '우동', '울면']
print(food)
# 탕수육 주문 추가
food.append('탕수육')
print(food)
# 짬뽕을 굴짬뽕으로 주문 변경
food[1] = '굴짬뽕'
print(food)

# 우동을 물만두로 주문 변경
food[food.index('우동')] = '물만두'
print(food)

```

```

Python 3.7.4 (tags/v3.7.4:e09359112e, Jul 8 2019,
(AMD64)) on win32
Type "help", "copyright", "credits" or "license()"
>>>
== RESTART: C:\Users\82104\Desktop\2020 python cod
['짜장면', '짬뽕', '우동', '울면']
['짜장면', '굴짬뽕', '우동', '울면', '탕수육']
['짜장면', '굴짬뽕', '우동', '울면', '탕수육']
['짜장면', '굴짬뽕', '물만두', '울면', '탕수육']
>>>

```

## 05-01. 여러 자료 값을 편리하게 처리하는 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()
print()

```

```

Python 3.7.4 (tags/v3.7.4:e09359112e, Jul 8 2019, 20:34:20) [MSC v.1916 64 bit
(AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
RESTART: C:\Users\#82104\Desktop\#2020 python code\ch05\05-07\nestedanimallist.py
[['사자', '코끼리', '호랑이'], '조류', '어류']
[['사자', '코끼리', '호랑이']
조류
어류

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

>>>

```

## 05-02. 리스트의 부분 참조와 항목의 삽입과 삭제

```

item = ['연필', '볼펜']
# 현재 학용품 품목 출력
print(item)

# 연필 1개와 볼펜 세 자루 삽입
item.insert(1, 2)
item.insert(3, 3)
# 맨 뒤에 지우개, 1개 삽입
item.insert(4, '지우개')
item.insert(5, 1)
# 현재 학용품 품목 출력
print(item)

# 연필 두 자루 삭제
print(item.pop(1)) # 첨자 1 항목 삭제
item.remove('연필') # 항목 연필 항목 삭제
del item[2:] # 지우개와 수 품목 삭제

# 현재 학용품 품목 출력
print(item)

```

```

Python 3.7.4 (tags/v3.7.4:e09359112e
(AMD64)) on win32
Type "help", "copyright", "credits"
>>>
RESTART: C:\Users\W82104\Desktop\W202
['연필', '볼펜']
['연필', 2, '볼펜', 3, '지우개', 1]
2
['볼펜', 3]
>>>

```

## 05-02. 리스트의 부분 참조와 항목의 삽입과 삭제

```
# for문으로 리스트 생성
```

```
a = []
```

```
for i in range(10):  
    a.append(i)
```

```
print(a)
```

```
# 컴프리헨션으로 리스트 생성
```

```
seq = [i for i in range(10)]
```

```
print(seq)
```

```
# for문으로 리스트 생성
```

```
s = []
```

```
for i in range(10):  
    if i%2 == 1:  
        s.append(i**2)
```

```
print(s)
```

```
# 컴프리헨션으로 리스트 생성
```

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

```
Python 3.7.4 (tags/v3.7.4:e0935f6, Oct 8 2019, [AMD64]) on win32  
Type "help", "copyright", "credits()" or "quit()" for more  
>>>
```

```
RESTART: C:\Users\W82104\Desktop>
```

```
[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]
```

```
>>>
```



## 05-03. 항목의 순서나 내용을 수정할 수 없는 튜플

```
singer = ('BTS', '볼빨간사춘기', 'BTS', '블랙핑크', '태연')
song = ('작은 것들을 위한 시', '나만, 봄', '소우주', 'Kill This Love', '사계')
print(singer)
print(song)

print(singer.count('BTS'))
print(singer.index('볼빨간사춘기'))
print(singer.index('BTS'))
print()

for _ in range(len(singer)):
    print('%s: %s' % (singer[_], song[_]))
```

```
Python 3.7.4 (tags/v3.7.4:e09359112e, Jul 8 2019, 20:34:20) [MSC v.191
(AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information
>>>
== RESTART: C:\Users\B2104\Desktop\2020 python code\ch05\05-12kpo tuple
('BTS', '볼빨간사춘기', 'BTS', '블랙핑크', '태연')
('작은 것들을 위한 시', '나만, 봄', '소우주', 'Kill This Love', '사계')
2
1
0

BTS: 작은 것들을 위한 시
볼빨간사춘기: 나만, 봄
BTS: 소우주
블랙핑크: Kill This Love
태연: 사계
>>>
```

## 05-03. 항목의 순서나 내용을 수정할 수 없는 튜플

```
day1 = ('monday', 'tuesday', 'wednesday')
day2 = ('thursday', 'friday', 'saturday')
# ('sunday')로 하면 튜플이 아니고 문자열
day3 = ('sunday', )
```

```
day = day1 + day2 + day3
print(type(day))
print(day)
```

```
day = day1 + day2 + day3 * 3
print(day)
```

```
Python 3.7.4 (tags/v3.7.4:e09359112e, Jul 8 2019, 20:34:20) [MSC v.1916 64 bit
(AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
=== RESTART: C:\Users\#82104\Desktop\2020 python code\ch05\05-13daytuple.py ===
<class 'tuple'>
('monday', 'tuesday', 'wednesday', 'thursday', 'friday', 'saturday', 'sunday')
('monday', 'tuesday', 'wednesday', 'thursday', 'friday', 'saturday', 'sunday', '
sunday', 'sunday')
>>>
```

## 06-01. 키와 값인 쌍의 나열인 딕셔너리

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

```
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:45:29) [M  
tel)] on win32  
Type "help", "copyright", "credits" or "license()" for more  
>>>  
=== RESTART: C:\Users\#82104\Desktop\2020 python code\ch06\06  
{'잔나비': 5, '트와이스': 9, '블랙핑크': 4, '방탄소년단': 7}  
<class 'dict'>  
>>>
```

## 06-01. 키와 값인 쌍의 나열인 딕셔너리

```
month={1: 'January', 2: 'February', 3: 'March', 4: 'April'}
month[5]='May'
month[6]='June'
month[7]='July'
month[8]='August'
month[9]='September'
print(month)
print()
```

```
from random import randint
#임의로 5번의 월 단어 출력
for i in range(5):
    r=randint(1,9)
    print('%d: %s' % (r, month[r]))
```

```
Python 3.7.4 (tags/v3.7.4:e09359112e, Jul 8 2019, 20:34:20) [MSC v.1916 64 bit
(AMD64)] on win32
```

```
Type "help", "copyright", "credits" or "license()" for more information.
```

```
>>>
```

```
RESTART: C:\Users\#82104\Desktop\2020 python code\ch06\06-04monthdictionary.py
{1: 'January', 2: 'February', 3: 'March', 4: 'April', 5: 'May', 6: 'June', 7: 'J
uly', 8: 'August', 9: 'September'}
```

```
1: January
```

```
6: June
```

```
7: July
```

```
3: March
```

```
8: August
```

```
>>>
```

## 06-01. 키와 값인 쌍의 나열인 딕셔너리

```

season={'봄': 'spring', '여름': 'summer', '가을': 'autumn', '겨울': 'winter'}
print(season.keys())
print(season.items())
print(season.values())

#메소드 keys()로 항목 순회
for key in season.keys():
    print('%s %s' % (key, season[key]))

for item in season.items():
    print('{} {} '.format(item[0], item[1]), end=' ')
print()
#메소드 items()의 반환 값인 튜플을 한 변수에 저장한 경우, 항목 순회 2
for item in season.items():
    print('{} {} '.format(*item), end=' ')
print()

```

```

Python 3.7.4 (tags/v3.7.4:09359112e, Jul 8 2019, 20:34:20) [MSC v.1916 64 bit
(AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
== RESTART: C:\Users\#82104\Desktop\#2020 python code\ch06\06-05seasondict.py ==
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
봄 spring 여름 summer 가을 autumn 겨울 winter
>>>

```

## 06-01. 키와 값인 쌍의 나열인 딕셔너리

```
color=dict(검은색='black', 흰색='white', 녹색='green', 파란색='blue')
print(color)
```

```
#항목 조회
print(color.get('녹색'))
print(color.get('노란색'))
print()
```

```
#항목 추가
color['노란색']='yellow'
print(color)
print()
```

```
#항목 삭제
c='흰색'
print('삭제: %s %s' % (c, color.pop('흰색')))
print(color)
c='빨간색'
print('삭제: %s %s' % (c, color.pop(c, '없어요')))
```

```
print('임의 삭제: {} '.format(color.popitem()))
print('임의 삭제 후: {} '.format(color))
```

```
c='검은색'
del color[c]
print('{} 삭제 후: {}'.format(c, color))
```

```
#모든 항목 삭제
color.clear()
print(color)
```

```
Python 3.7.4 (tags/v3.7.4:e09359112e, Jul 8 2019, 20:34:20) [MSC v.1916 64 bit
(AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
== RESTART: C:\Users\82104\Desktop\2020 python code\ch06\06-color\dict.py ==
{'검은색': 'black', '흰색': 'white', '녹색': 'green', '파란색': 'blue'}
green
None

{'검은색': 'black', '흰색': 'white', '녹색': 'green', '파란색': 'blue', '노란색':
'yellow'}

삭제: 흰색 white
{'검은색': 'black', '녹색': 'green', '파란색': 'blue', '노란색': 'yellow'}
삭제: 빨간색 없어요
임의 삭제: ('노란색', 'yellow')
임의 삭제 후: {'검은색': 'black', '녹색': 'green', '파란색': 'blue'}
검은색 삭제 후: {'녹색': 'green', '파란색': 'blue'}
{}
>>>
== RESTART: C:\Users\82104\Desktop\2020 python code\ch06\06-color\dict.py ==
{'검은색': 'black', '흰색': 'white', '녹색': 'green', '파란색': 'blue'}
green
None

{'검은색': 'black', '흰색': 'white', '녹색': 'green', '파란색': 'blue', '노란색':
'yellow'}

삭제: 흰색 white
{'검은색': 'black', '녹색': 'green', '파란색': 'blue', '노란색': 'yellow'}
삭제: 빨간색 없어요
임의 삭제: ('노란색', 'yellow')
임의 삭제 후: {'검은색': 'black', '녹색': 'green', '파란색': 'blue'}
검은색 삭제 후: {'녹색': 'green', '파란색': 'blue'}
{}
>>>
```

## 06-02. 중복과 순서가 없는 집합

```
planets=set('해달별')
fruits=set(['감','귤'])
nuts={'밤','잣'}
things=({'밤','잣'}, ('감','귤'), '해달')
#things=[['밤','잣'], ('감','귤'), '해달'] #오류 발생

print(planets)
print(fruits)
print(nuts)
print(things)
```

```
Python 3.7.4 (tags/v3.7.4:e09359112e, Ju
(AMD64)] on win32
Type "help", "copyright", "credits" or "
>>>
== RESTART: C:\Users\82104\Desktop\2020
{'해','별','달'}
{'감','귤'}
{'잣','밤'}
{'해달', ('밤','잣'), ('감','귤')}
>>>
```

## 06-02. 중복과 순서가 없는 집합

```
daysA={'월', '화', '수', '목'}
daysB=set(['수', '목', '금', '토', '일'])
weekends=set(['토', '일'])

alldays=daysA | daysB
print(alldays)

workdays=alldays-weekends
print(workdays)

print(daysA & daysB)
print(daysA.symmetric_difference(daysB))
```

```
Python 3.7.4 (tags/v3.7.4:e09359112e, Jul 8
(AMD64)) on win32
Type "help", "copyright", "credits" or "licer
>>>
=== RESTART: C:\Users\82104\Desktop\2020 pyth
{'월', '금', '목', '토', '수', '월', '화'}
{'일', '목', '수', '월', '화'}
{'목', '수'}
{'일', '금', '월', '화', '토'}
>>>
```



## 06-03. 내장함수 zip()과 enumerate(), 시퀀스 간의 변환

```
#구기 종목 리스트
sports=['축구', '야구', '농구', '배구']
#위 종목에 대응하는 팀원 수를 항목으로 구성
num=[11, 9, 5, 6]
print(sports)
print(num)
print()
```

```
print('함수 zip():')
for s, i in zip(sports, num):
    print('%s: %d명' % (s, i), end=' ')
print()
for tp in zip(sports, num):
    print('{}: {}명'.format(*tp), end=' ')
print(); print()
```

```
#dict()와 zip() 함수로 종목의 이름을 키, 인원수를 값으로 저장
print('함수 dict(zip()):')
sportsnum=dict(zip(sports, num))
print(sportsnum)
```

```
Python 3.7.4 (tags/v3.7.4:e09359112e, Jul 8 2019; AMD64) on win32
Type "help", "copyright", "credits" or "license()"
>>>
= RESTART: C:\Users\W82104\Desktop\2020 python 3
['축구', '야구', '농구', '배구']
[11, 9, 5, 6]
```

```
함수 zip():
축구: 11명 야구: 9명 농구: 5명 배구: 6명
축구: 11명 야구: 9명 농구: 5명 배구: 6명
```

```
함수 dict(zip()):
{'축구': 11, '야구': 9, '농구': 5, '배구': 6}
>>>
```

V

04

마무리

## 포트폴리오를 마무리하며....

파이썬은 이번 학기에 처음 배우는 언어인데 간단하고 활용도가 높다고 해서 꼭 한번 배우고 싶었다. 이전에 배운 언어들은 모두 세미콜론을 써야 실행이 가능했는데 파이썬은 쓰지않아서 신기했다.

처음에는 어색했지만 변수도 선언하지않고, 다른 동작없이 print문 한줄로 출력이 가능해서 다른 언어들보다 적응하기 쉬웠다.

포트폴리오는 처음 작성해보는데,

시작할 때는 너무 막막했지만 막상 하다보니까 지금까지 배운 내용들이 머릿속에 차곡차곡 정리되는 것 같아 보람이 느껴진다.

지금은 중간고사 포트폴리오에 불과하지만 나중에 취업을 위한 포트폴리오 작성에서 이 때의 경험이 많은 도움이 될 것 같다.



Thank you !