

## Python 기본 문법(3)



### 조건문 : if

- if 조건문 뒤에는 반드시 콜론(:)
  - 바로 아래 문장부터 if문에 속하는 모든 문장에 들여쓰기(indentation)를 해야 함
- 파이썬에는 다른 언어에 있는 switch 문이 존재하지 않음
  - if...elif...elif... 문으로 수행
- pass
  - def 문이나 if 문처럼 코드 블록을 본문으로 갖는 표현에서 본문을 비워 둘 때 사용

```
#조건식
x = 10

print("조건식 1:")
if x % 2 == 1 :
    print(x, ": 홀수")

print("조건식 2:")
if x % 2 == 0 :
    print(x, ": 짝수")
else:
    print(x, ": 홀수")

print("조건식 3:")
if x <= 0 :
    print("양수가 아님")
elif x % 2 == 0 :
    print(x, ": 짝수")
else:
    print(x, ": 홀수")
```

조건식 1:  
조건식 2:  
10 : 짝수  
조건식 3:  
10 : 짝수

## 비교연산자/논리연산자

### 비교 연산자 의미

<code>x &lt; y</code>	x가 y보다 작다
<code>x &gt; y</code>	x가 y보다 크다
<code>x == y</code>	x와 y가 같다
<code>x != y</code>	x와 y가 같지 않다
<code>x &gt;= y</code>	x가 y보다 크거나 같다
<code>x &lt;= y</code>	x가 y보다 작거나 같다

### 논리 연산자 의미

<code>and</code>	A and B : A와 B 모두 True일때 True
<code>or</code>	A or B : A 혹은 B가 True일때 True
<code>not</code>	not A : A가 True면 False, False면 True

```
#비교 연산자
x = 10
y = 20

print("비교 연산자")
print(x, ">", y, ":", x > y)
print(x, "<", y, ":", x < y)
print(x, ">=", y, ":", x >= y)
print(x, "<=", y, ":", x <= y)
print(x, "==", y, ":", x == y)
print(x, "!=", y, ":", x != y)

print("논리 연산자")
if ( x >= 20 and y >= 20 ) : print ("두수는 20보다 크다")
elif ( x >= 20 or y >= 20 ) : print ("두수 중 20보다 큰수가 있다.")
else : print ("두수는 20보다 작다")

비교 연산자
10 > 20 : False
10 < 20 : True
10 >= 20 : False
10 <= 20 : False
10 <= 20 : True
10 <= 20 : False
10 != 20 : True

논리 연산자
두수 중 20보다 큰수가 있다.
```

## in연산자

in	not in
x in 리스트	x not in 리스트
x in 튜플	x not in 튜플
x in 문자열	x not in 문자열

```
#in 연산자
str = "안녕하세요"
print(str, "에 안이란 글자가 있습니다.:", '안' in str)

#리스트
lst = list(range(1,5))
print("리스트에 100이 있습니다.:", 10 in lst)

안녕하세요 에 안이란 글자가 있습니다.: True
리스트에 100이 있습니다.: False
```

## 반복문 - for

- for

- 컬렉션으로부터 하나씩 요소(element)를 가져와, 루프 내의 문장들을 실행

- 컬렉션 : 리스트, Tuple, 문자열 등

```
#for문으로 collection 접근하기
#문자열
str = "안녕하세요"
print("문자열")
for item in str :
    print(item)

#리스트
lst = list(range(1,5))
print("리스트")
for item in lst :
    print(item)

#딕셔너리
dic = {'a':1, 'b':2}
print("딕셔너리")
for item in dic :
    print(item)

for item in dic.items():
    print(item)
```

문자열  
안  
녕  
하  
세  
요  
리스트  
1  
2  
3  
4  
딕셔너리  
b  
a  
( 'b', 2)  
( 'a', 1)



## For문을 이용한 리스트 변형

```
st = ['1', '2', '3', '4']
print("리스트 :", lst)

#for문을 이용한 리스트 항목의 형 변환
lst = [int(item) for item in lst]
print("리스트 :", lst)

#for문을 이용한 리스트 항목추출
lst = [item for item in lst if item % 2 == 0]
print("리스트 :", lst)
```



## 반복문 while

- while

- while 키워드 다음의 조건식이 참일 경우 계속 while 안의 블록을 실행

```
: #for문과 while문  
  
print("for 문")  
for i in range(1, 6) :  
    print(i)  
  
print("while 문")  
i = 0  
while i < 5 :  
    i = i + 1  
    print(i)
```



## break/continue

- break 문

- 반복문 안에서 루프를 빠져나오기 위해 을 사용

- continue문

- 루프 블록의 나머지 문장들을 실행하지 않고 다음 루프로 직접 돌아가게 함

```
#break/continue  
for i in range(1, 100) :  
    if ( i % 2 == 1 ) : continue  
    if ( i > 5 ) : break  
    print(i)
```

2  
4



## 해결문제

- 단을 입력 받아서 해당하는 단의 구구단을 출력하시오.

– 단은 2단에서 9단까지만  
입력 그 외 입력되면 종료  
하고 그렇지 않을 경우 계  
속 입력

단을 입력하세요6

.... 6 단 ....  
6 x 1 = 6  
6 x 2 = 12  
6 x 3 = 18  
6 x 4 = 24  
6 x 5 = 30  
6 x 6 = 36  
6 x 7 = 42  
6 x 8 = 48  
6 x 9 = 54

단을 입력하세요2

.... 2 단 ....  
2 x 1 = 2  
2 x 2 = 4  
2 x 3 = 6  
2 x 4 = 8  
2 x 5 = 10  
2 x 6 = 12  
2 x 7 = 14  
2 x 8 = 16  
2 x 9 = 18

Co  
단을 입력하세요99  
입력오류



## 해결문제

- 다음은 문재인 대통령의 연설문 일부 입  
니다.

– 국민, 대한민국, 대통령이란 단어 중에서  
2회 이상 나온 단어를 출력하시오.

국민 여러분의 위대한 선택에 머리 숙여 깊이 감사드립니다. 저는 오  
늘 대한민국 제19대 대통령으로서 새로운 대한민국을 향해 첫걸음을  
내딛습니다. 지금 제 두 어깨는 국민 여러분으로부터 부여받은 막중  
한 소명감으로 무겁습니다. 지금 제 가슴은 한번도 경험하지 못한 나  
라를 만들겠다는 열정으로 뜨겁습니다. 그리고 지금 제 머리는 통합  
과 공존의 새로운 세상을 열어갈 청사진으로 가득 차 있습니다.

Come with 80  
PNU!



# 파일 생성

## • 파일 생성

파일 객체 = open(파일 이름, 파일 열기 모드)

파일열기모드	설명
r	읽기모드 - 파일을 읽기만 할 때 사용
w	쓰기모드 - 파일에 내용을 쓸 때 사용
a	추가모드 - 파일의 마지막에 새로운 내용을 추가 시킬 때 사용

## • 파일 닫기

파일 객체.close()

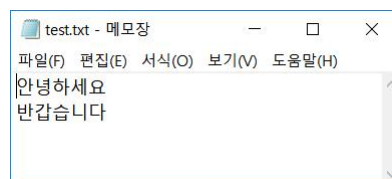


# 파일 출력

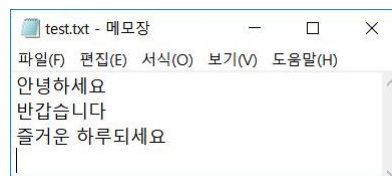
## • 파일출력함수 : write()

– 파일열기모드 : w

```
f = open("test.txt", "w")
inData = input("기록 내용을 입력하세요 ")
f.write(inData+"\n")
inData = input("기록 내용을 입력하세요 ")
f.write(inData)
f.close()
```



```
f = open("test.txt", "a")
inData = input("기록 내용을 입력하세요 ")
f.write(inData+"\n")
f.close()
```



## 파일 입력

- 파일입력함수

- read(): 파일의 내용 전체를 문자열로 리턴
- readline(): 파일의 첫 번째 줄을 읽어 출력하는 경우
- readlines(): 파일의 모든 라인을 읽어서 각각의 줄을 요소로 갖는 리스트로 리턴

```
inData = f.readline()
print(inData)
```

```
while True:
    inData = f.readline()
    if not inData : break
    print(inData)
```

```
inData = f.readlines()
for line in inData :
    print(line)
```

```
inData = f.read()
print(inData)
```



## 해결문제

- 성적 파일을 읽어서 합계 점수를 구하여 결과 파일로 저장하시오.



## python 내장함수

- **sum(), max(), min()**

- 입력으로 받은 리스트나 튜플의 모든 요소의 합, 최대, 최소값을 리턴하는 함수

- **zip()**

- 동일한 개수로 이루어진 자료형을 묶어주는 역할을 하는 함수
- Dictionary의 {key: value} 중 value 값으로 최대값과 최소값, 그리고 정렬을 하는데 사용

```
list1 = ['a', 'b', 'c']
list2 = [1, 3, 2]

#합계
print(sum(list2))
#최대
print(max(list2))
#최소
print(min(list2))
#정렬
print(sorted(list2))

#zip으로 리스트 만들기
list3 = list(zip(list1, list2))
print(list3)

#zip으로 딕셔너리 만들기
dict1 = dict(zip(list1, list2))
print(dict1)

#zip으로 최대값/최소값 찾기
print(max(zip(dict1.values(), dict1.keys())))
print(min(zip(dict1.values(), dict1.keys())))

#zip으로 정렬
print(sorted(zip(dict1.values(), dict1.keys())))
```

## 해결문제

- 주어진 dust201809.txt는 2018년 09월 지역별 미세먼지 대기오염이다.
  - 대기오염 평균을 구하시오.



## 해결문제

- 도별미세먼지.csv 파일을 읽어서 딕셔너리를 만들고 평균과 최대값을 가지는 도를 찾으시오.

