



statements

목차

조건문 if

반복문 while

반복문 for

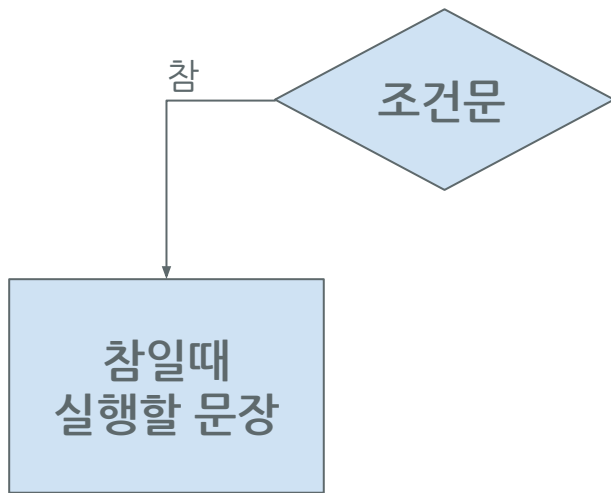
조건문 if

if

if 조건문 조건에 따라 코드를 실행하거나 실행하지 않게 만들고 싶을 때 사용

if 조건문:
조건이 참일 때 실행할 문장

콜론 : 과 들여쓰기 주의할 것



if

```
if True:  
    print('참입니다.')
```

참입니다.

```
if False:  
    print('거짓입니다.')
```

```
x = 100  
y = 200
```

```
if x == y:  
    print('x와 y는 같습니다.')
```

```
if x < y:  
    print('x가 y보다 작습니다.')
```

x가 y보다 작습니다.

```
if x > y:  
    print('x가 y보다 큼니다.')
```

if

pass 아무것도 하지 않음. 프로그램의 골격을 잡아두고 나중에 작성하고자 할 때 입력해두는 용도로 사용.

```
x = 100  
if x > 10:
```

File "<ipython-input-30-27e9f3185226>", line 2

```
if x > 10:
```

^

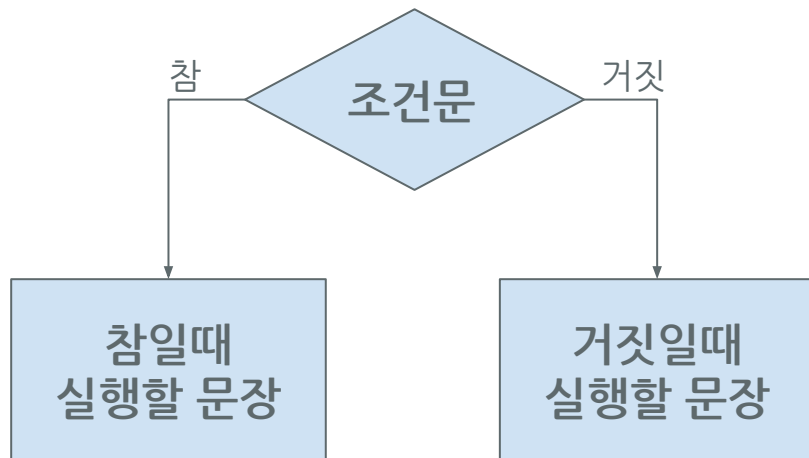
SyntaxError: unexpected EOF while parsing

```
x = 100  
if x > 10:  
    pass
```

if else

```
if 조건문:  
    조건이 참일 때 실행할 문장  
else:  
    조건이 거짓일 때 실행할 문장
```

콜론 : 과 들여쓰기 주의할 것



if else

```
fruits = ['Banana', 'Apple', 'Orange']  
if 'Watermelon' in fruits:  
    print('수박이 있습니다.')  
else:  
    print('수박이 없습니다.')
```

수박이 없습니다.

```
x = 100  
y = 200  
if x < y:  
    print('x가 y보다 작습니다.')  
else:  
    print('x가 y보다 크거나 같습니다.')
```

x가 y보다 작습니다.

```
if x == y:  
    print('x와 y는 같습니다.')  
else:  
    print('x와 y는 다릅니다.')
```

x와 y는 다릅니다.

if else

```
age = 15
score = 50

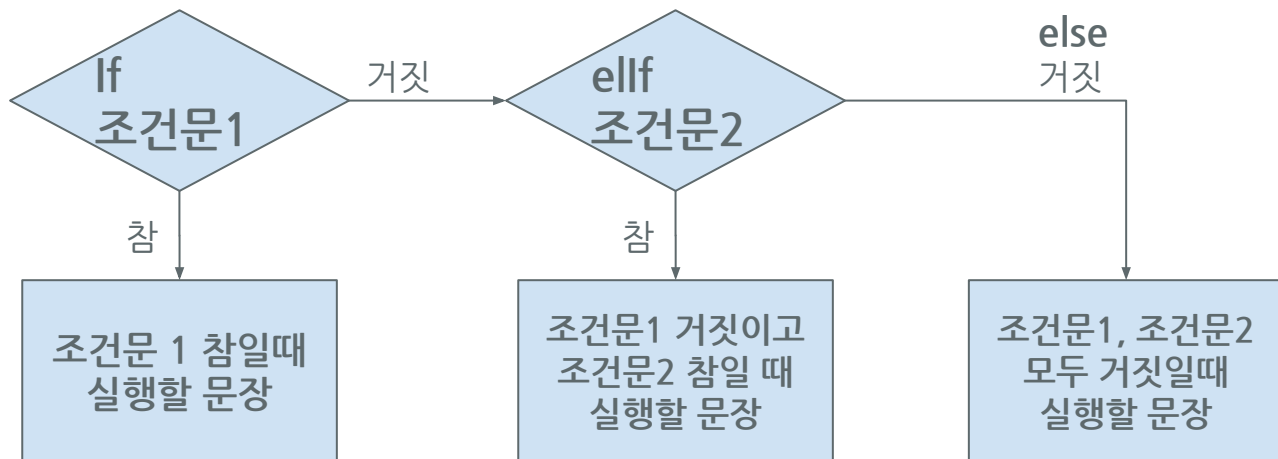
if age < 13:
    if score > 30:
        print('통과!')
    else:
        print('실패!')
else:
    if score > 60:
        print('통과!!')
    else:
        print('실패!!')
```

실패!!

if 문 안에 if 문 중첩 가능

if elif else

```
if 조건문1:  
    실행문장1  
elif 조건문2:  
    실행문장2  
else:  
    실행문장3
```



if elif else

```
x = 200
y = 0
if x == y:
    print('x와 y는 같습니다.')
elif x > y:
    print('x는 y보다 큼니다.')
else:
    print('x는 y보다 작습니다.')
```

x는 y보다 큼니다.

```
score = 60

if score > 90:
    grade = 'A'
elif score > 70:
    grade = 'B'
elif score > 50:
    grade = 'C'
else:
    grade = 'D'
```

grade

'C'

반복문 while

while

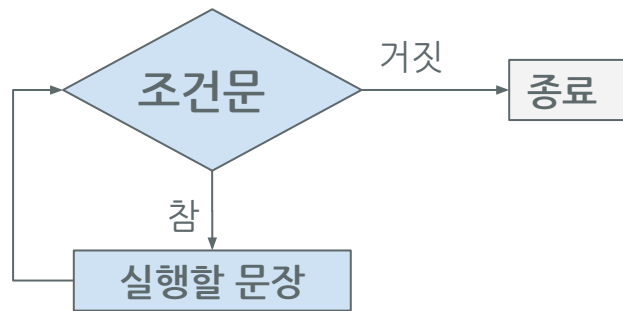
while문 조건문이 참인 동안에 while문 아래의 문장이 반복되서 실행됨

while 조건문:

조건문이 참일 때 실행할 문장1

조건문이 참일 때 실행할 문장2

콜론 : 과 들여쓰기 주의할 것



while

```
x = 0
while x < 5:
    print(x)
    x += 1
```

0
1
2
3
4

x	조건문	조건 판단	실행 문장	while문
0	$0 < 5$	참	0 출력 $x = 1$	반복
1	$1 < 5$	참	1 출력 $x = 2$	반복
2	$2 < 5$	참	2 출력 $x = 3$	반복
3	$3 < 5$	참	3 출력 $x = 4$	반복
4	$4 < 5$	참	4 출력 $x = 5$	반복
5	$5 < 5$	거짓		종료

while

```
numbers = [100, 200, 0, 20, 50]
i = 0
while i < len(numbers):
    print(numbers[i])
    i += 1
```

```
100
200
0
20
50
```

짝수만 출력하기

```
numbers = [16, 13, 52, 3, 1, 11, 47, 20, 0, 24, 53]
i = 0
while i < len(numbers):
    if numbers[i] % 2 == 0:
        print(numbers[i])
    i += 1
```

```
16
52
20
0
24
```

while

break

while문에서 강제로 빠져나올 때

```
while True:
    x = input('단어를 입력하세요 ')
    if x == 'end':
        break
    print('입력:', x)
```

단어를 입력하세요

```
while True:
    x = input('단어를 입력하세요 ')
    if x == 'end':
        break
    print('입력:', x)
```

단어를 입력하세요 Hi

입력: Hi

단어를 입력하세요 python

입력: python

단어를 입력하세요 numpy

입력: numpy

단어를 입력하세요 pandas

입력: pandas

단어를 입력하세요 matplotlib

입력: matplotlib

단어를 입력하세요 end

input 함수: 사용자가 입력한 값을 변수에 넣고 싶을 때 사용

while

continue 현재 반복을 생략하고

다음 반복으로 넘어갈 때 사용

```
while True:
    x = input('정수입력하기(종료:q)')
    if x == 'q':
        break
    num = int(x)
    if num % 2 != 0:
        continue
    print(num, 'squared is', num*num)
```

정수입력하기(종료:q)

```
while True:
    x = input('정수입력하기(종료:q)')
    if x == 'q':
        break
    num = int(x)
    if num % 2 != 0:
        continue
    print(num, 'squared is', num*num)
```

정수입력하기(종료:q)10
10 squared is 100
정수입력하기(종료:q)14
14 squared is 196
정수입력하기(종료:q)5
정수입력하기(종료:q)7
정수입력하기(종료:q)1
정수입력하기(종료:q)2
2 squared is 4
정수입력하기(종료:q)q

반복문 for

for

for 데이터의 첫번째 요소부터 마지막 요소까지 차례대로 변수에 대입되어 실행문장1, 실행문장2 등이 실행됨.

```
for 변수 in 데이터:  
    실행문장1  
    실행문장2
```

콜론 : 과 들여쓰기 주의할 것

```
for i in [10, 20, 30, 40]:  
    print(i)
```

10

20

30

40

for

```
scores = [100, 40, 50, 60, 20, 10]
for s in scores:
    if s > 80:
        print(s, '합격')
    else:
        print(s, '불합격')
```

```
100 합격
40 불합격
50 불합격
60 불합격
20 불합격
10 불합격
```

```
for t in 'python':
    print(t)
```

```
p
y
t
h
o
n
```

```
for i in (-10, 0, 10):
    print(i)
```

```
-10
0
10
```

for

break, continue while문의 break, continue와 동일하게 동작

```
list_a = ['a', 'b', 'c', 'd', 'end', 'e', 'f']  
for t in list_a:  
    if t == 'end':  
        break  
    print(t)
```

a
b
c
d

for

break, continue while문의 break, continue와 동일하게 동작

```
scores = [10, 30, 120, 20, 60, 90, 110]
for s in scores:
    if s > 100:
        continue
    print(s, '점입니다. 100점을 위해서', 100-s, '점이 더 필요합니다.')
```

10 점입니다. 100점을 위해서 90 점이 더 필요합니다.
30 점입니다. 100점을 위해서 70 점이 더 필요합니다.
20 점입니다. 100점을 위해서 80 점이 더 필요합니다.
60 점입니다. 100점을 위해서 40 점이 더 필요합니다.
90 점입니다. 100점을 위해서 10 점이 더 필요합니다.

내장함수

range for문과 자주 사용되는 함수. 숫자 리스트를 만들어줌.

range(끝)

```
list(range(7))
```

```
[0, 1, 2, 3, 4, 5, 6]
```

range(시작, 끝)

```
list(range(-3, 4))
```

```
[-3, -2, -1, 0, 1, 2, 3]
```

range(시작, 끝, 간격)

```
list(range(100, 200, 20))
```

```
[100, 120, 140, 160, 180]
```

내장함수

range

```
for i in range(2, 10):  
    for j in range(1, 10):  
        print('{:2}'.format(i*j), end=' ')  
    print('')
```

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

내장함수

`zip(iter1, iter2, ...)` 여러 시퀀스를 병렬로 순회

```
names = ['YJ', 'SG', 'YM', 'JG', 'SJ', 'HD']  
scores = [10, 20, 50, 20, 30, 40]
```

```
for n, s in zip(names, scores):  
    print(n, ': ', s)
```

```
YJ : 10  
SG : 20  
YM : 50  
JG : 20  
SJ : 30  
HD : 40
```

Appendix

파일 읽고 쓰기

open 파일을 열 때 사용하는 함수.

파일을 읽고, 쓰기 전에 파일을 열어야 함.

open(파일경로, 모드)

파일 읽고 쓰기

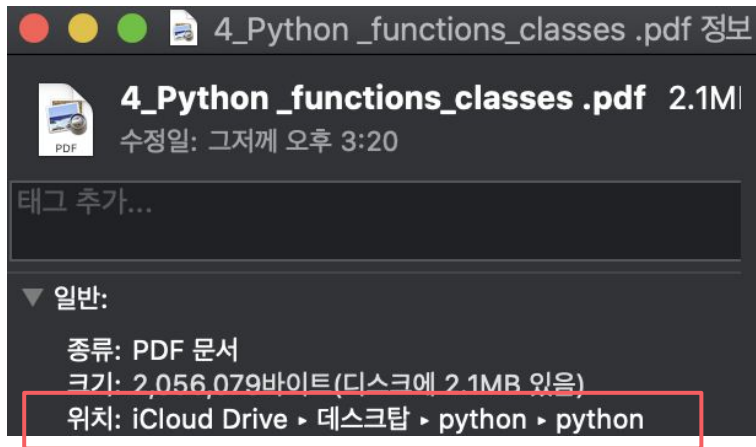
open(**파일경로**, 모드)

- **절대경로** 파일이 가지고 있는 고유한 경로. 최초의 시작점으로 경유한 경로를 전부 기입하는 방식
- **상대경로** 현재 위치를 기준으로 한 해당 파일의 경로

파일 읽고 쓰기

open(**파일경로**, 모드)

- **절대경로** 파일이 가지고 있는 고유한 경로. 최초의 시작점으로 경유한 경로를 전부 기입하는 방식



/Users/park/Desktop/python/python/4_Python_functions_classes .pdf

파일 읽고 쓰기

open(**파일경로**, 모드)

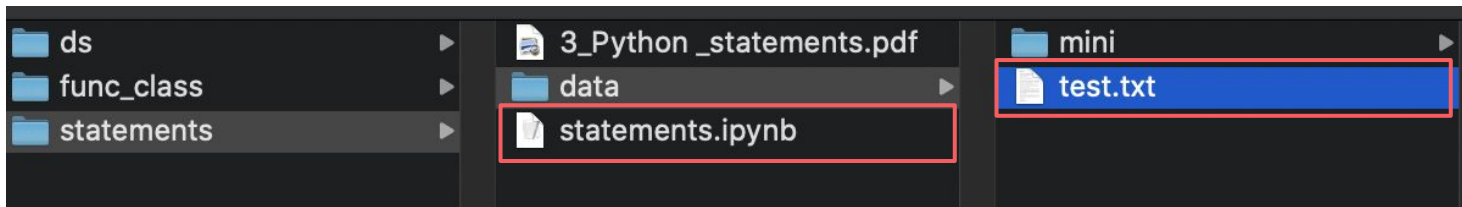
- **상대경로** 현재 위치를 기준으로 한 해당 파일의 경로

현재 위치	./
현재 위치의 상위 폴더	../

파일 읽고 쓰기

open(파일경로, 모드)

- 상대경로 현재 위치를 기준으로 한 해당 파일의 경로

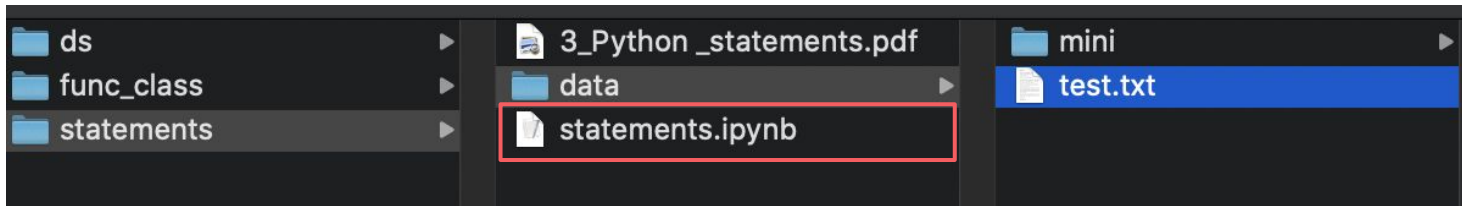


statements.ipynb 에서 작업중인 경우 test.txt를 읽으려면?
./data/test.txt

파일 읽고 쓰기

open(**파일경로**, 모드)

- **상대경로** 현재 위치를 기준으로 한 해당 파일의 경로



statements.ipynb 에서 작업중인 경우
“ds”폴더에 있는 sample.txt를 읽으려면?
../ds/sample.txt

파일 읽고 쓰기

open(파일경로, 모드)

- 윈도우 환경

`r 'User/park/python/test.txt'`

`'User//park//python//test.txt'`

파일 읽고 쓰기

open(파일경로, **모드**)

모드	설명
r	파일 읽기 (default)
w	파일 쓰기 (파일이 존재하지 않는 경우 파일 생성, 존재하는 경우 덮어씀)
a	파일 쓰기 (파일이 존재하면 뒤에 이어서 씀)
x	파일 추가하기 (파일이 존재하지 않을 경우 생성. 존재하면 에러)

텍스트 타입 :
default
바이너리 타입:
모드의 두번째
글자에 b만
추가해서 파일
타입 명시해줌.
ex) rb(바이너리
파일 읽기)

파일 읽고 쓰기

close()

파일을 열었으면 닫아야 함.

```
f = open('test.txt', 'r')
```

```
f.close()
```

파일 읽고 쓰기

with

with 문을 벗어나면 파일을 자동으로 닫아줌

```
with open(파일경로, 모드) as 파일객체명:  
    실행문장
```

파일 읽고 쓰기

write

파일 쓰기

```
with open('test2.txt', 'w') as f:  
    f.write('안녕하세요\n')  
    f.write('지금은 파일 읽고 쓰기를 하고 있습니다.')
```

test2.txt

안녕하세요
지금은 파일 읽고 쓰기를 하고 있습니다.

파일 읽고 쓰기

파일 읽기 `read`, `readline`, `readlines`

read 파일 전체를 한번에 읽어서 하나의 string 타입으로 반환

readline 파일을 한줄 읽어서 하나의 string 타입으로 반환

readlines 파일 전체를 한번에 읽어서 한 줄이 원소인 list 타입으로 반환

파일 읽고 쓰기

파일 읽기 `read`, `readline`, `readlines`

`read` 파일 전체를 한번에 읽어서

하나의 string 타입으로 반환

```
with open('test.txt', 'r') as f:  
    text = f.read()  
    print(text)
```

어떤 주문

이해인

행복하다고 말만 하지 마시고
행복한 모습
환한 웃음으로 보여주셔요
사랑한다고 말만 하지 마시고
사랑하는 모습
한결같은 참을성으로 보여주셔요
행복과 사랑에도 겸손이 필요해요
너무 가볍게 말하지 마세요

파일 읽고 쓰기

파일 읽기 `readline`, `readline`, `readlines`

`readline` 파일을 한줄 읽어서 하나의 string 타입

으로 반환

```
with open('test.txt', 'r') as f:
    text = f.readline()
    print(text)
```

어떤 주문

```
with open('test.txt', 'r') as f:
    while True:
        text = f.readline()
        if not text: break
        print(text)
```

어떤 주문

이해인

행복하다고 말만 하지 마시고

행복한 모습

환한 웃음으로 보여주셔요

사랑한다고 말만 하지 마시고

사랑하는 모습

한결같은 참을성으로 보여주셔요

행복과 사랑에도 겸손이 필요해요

너무 가볍게 말하지 마세요

파일 읽고 쓰기

파일 읽기 `read`, `readline`, `readlines`

`readlines` 파일 전체를 한번에 읽어서 한 줄이 원소인 list 타입으로 반환

```
with open('test.txt', 'r') as f:
    text = f.readlines()
    print(text)
```

```
['어떤 주문\n', '\n', '이해인\n', '\n', '행복하다고 말만  
하지 마시고\n', '행복한 모습\n', '환한 웃음으로 보여주셔요\n', '사랑한다고 말만 하지 마시고\n', '사랑하는 모습\n', '한  
결같은 참을성으로 보여주셔요\n', '행복과 사랑에도 겸손이 필요해  
요\n', '너무 가볍게 말하지 마세요']
```

Reference

파이썬 공식 문서

러닝 파이썬

혼자 공부하는 파이썬

생활코딩 파이썬

처음 시작하는 파이썬

점프 투 파이썬