

# 조건문 다시 보기

구구단 예제로 다시 돌아간다

```
def dan(d):  
    a = 0  
    for i in range(8):  
        a = a + 1  
        b = d * a  
        print(d, 'x', a, '=', b)  
    print()
```

```
print('start')  
d = 1  
for i in range(8):  
    d = d + 1  
    if d < 5:  
        dan(d)  
print('end')
```

## 5단만 제외한다

무식한 방법과 조금 나은 방법

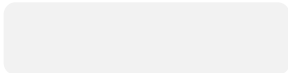
```
print('start')
d = 1
for i in range(8):
    d = d + 1
    if d < 5:
        dan(d)
    if d > 5:
        dan(d)
print('end')
```

```
print('start')
d = 1
for i in range(8):
    d = d + 1
    if d < 5 or d > 5:
        dan(d)
print('end')
```

## 5단만 제외한다

이번에는 != , == 연산자를 사용한다.

```
print('start')
d = 1
for i in range(8):
    d = d + 1
    if d != 5:
        dan(d)
print('end')
```

```
print('start')
d = 1
for i in range(8):
    d = d + 1
    if d == 5:
        
    dan(d)
print('end')
```

이 부분에 들어갈 수 있는 명령이 있다.

# 순서대로 하던 일을 멈추는 명령어 **continue, break**

멈추기는 멈추는데 그 다음 행동이 다르다.

```
print('start')
d = 1
for i in range(8):
    d = d + 1
    if d == 5:
        continue 반복문 계속 진행한다.
    dan(d)
print('end')
```

```
print('start')
d = 1
for i in range(8):
    d = d + 1
    if d == 5:
        break 반복문을 끝낸다.
    dan(d)
print('end')
```

# 이해 했나?

다음 실행 결과를 예측하라.

```
for i in range(4):  
    print('A', end='')  
    if i == 2:  
        break  
    print('B', end='')  
print('C')
```

```
for i in range(4):  
    print('A', end='')  
    if i == 2:  
        continue  
    print('B', end='')  
print('C')
```

## 이해 했나?

다음 실행 결과를 예측하라.

```
for i in range(4):  
    print('A', end='')  
    if i == 2:  
        break  
    print('B', end='')  
print('C')
```

ABABAC

```
for i in range(4):  
    print('A', end='')  
    if i == 2:  
        continue  
    print('B', end='')  
print('C')
```

ABABAABC

## 논리 연산자 and, or

계산 후 남기는 값이 True, False

```
>>> True and True
```

```
True
```

둘다 True일 때  
True

```
>>> True and False
```

```
False
```

```
>>> False and True
```

```
False
```

```
>>> False and False
```

```
False
```

```
>>> True or True
```

```
True
```

하나만 True여도  
True

```
>>> True or False
```

```
True
```

```
>>> False or True
```

```
True
```

```
>>> False or False
```

```
False
```

# not

참이면 거짓을 남기고, 거짓이면 참을 남긴다.

```
>>>not False
```

```
True
```

```
>>>not True
```

```
False
```



# 조건문 동작 방식

조건문의 엄밀한 동작 방식

<pre>if True:     print("True") else:     print("False")</pre>	<pre>if False:     print("True") else:     print("False")</pre>
True	False

# if가 숫자를 대하는 방식

-10이 참인지 거짓인지?

```
a = -1
if a:
    print("True")
else:
    print("False")
```

# -1은 True

음수도 True

```
a = -1
if a:
    print("True")
else:
    print("False")
```

True

# 0만 False, 0이 아닌 것은 True

0이 기준이다.

```
a = 0
if a:
    print("True")
else:
    print("False")
```

False

```
a = 2
if a:
    print("True")
else:
    print("False")
```

True

# None

아무 것도 아닌데, not을 만나면 True가 된다.

```
>>>a=1
```

```
>>>a
```

```
1
```

```
>>>a=None
```

```
>>>a          아무것도 출력되지 않는다.
```

```
>>>not None
```

```
True
```

```
>>>None == True
```

```
False
```

```
>>>None == False
```

```
False
```

## 30 | 2와 5사이 값인가?

```
a = 2
b = 3
c = 5
if (a < b and b < c):
    print(b, "is between", a, "and", c)
else:
    print(b, "is not between", a, "and", c)
```

3 is between 2 and 5

```
a = 2
b = 7
c = 5
if (a < b < c): 다른 언어에서는 지원 안됨
    print(b, "is between", a, "and", c)
else:
    print(b, "is not between", a, "and", c)
```

7 is not between 2 and 5

## 점수를 A, B, C, D, F로 출력하기

알고 있는 조건문으로 코딩

```
s=75
if (s>=90):
    print('A')
if (s>=80 and s<90):
    print('B')
if (s>=70 and s<80):
    print('C')
if (s>=60 and s<70):
    print('D')
if (s<60):
    print('F')
print('end')
```

사용된 조건문 총 5개  
항상 5번의 조건을 따진다.

C

## if의 또 다른 모습

```
s = 95
if s >= 90:
    print("A")
elif s >= 80 and s < 90:
    print("B")
elif s >= 70 and s < 80:
    print("C")
elif s >= 60 and s < 70:
    print("D")
elif s < 60:
    print("F")
else:
    print("wrong score")
```

한 덩어리

True 조건을 만나면,  
그 아래 조건들은 따지지 않는다.

A