

# 제어문

if / elif / else

---

# 제어 (Control)

---

- 프로그램의 흐름 (Flow)
  - 프로그램 내의 구문(statement)들은 기본적으로 위부터 아래 방향으로 순서대로 한번씩 실행된다.
  - 그러나 상황에 따라서 이러한 구문들의 흐름을 제어하여 실행 순서를 변경할 수도 있다.
  - 파이썬의 제어 구문에는 3가지 종류가 있다.
    - **if** : 특정 조건의 만족 여부에 따라 다른 내용을 실행
    - **while** : 특정 조건이 만족하는 동안 반복하여 실행
    - **for** : 특정 횟수만큼 반복하여 실행

# 제어문 (Control Statement)

---

- 선택 구문 - if

if 조건문 :

조건이 참(True)인 경우 수행할 구문1

조건이 참(True)인 경우 수행할 구문2

...

- 조건문을 검사하여 그 결과가 참(True)이면 그 아래에 있는 명령어 및 구문들을 순서대로 한번씩 실행한다.
- 조건문을 검사한 결과가 거짓(False)이면 그 아래에 있는 구문들을 무시하고 실행하지 않는다.

# 제어문 (Control Statement)

- 선택 구문 - if

if **조건문**:

조건이 참(True)인 경우 수행할 구문1

조건이 참(True)인 경우 수행할 구문2

...

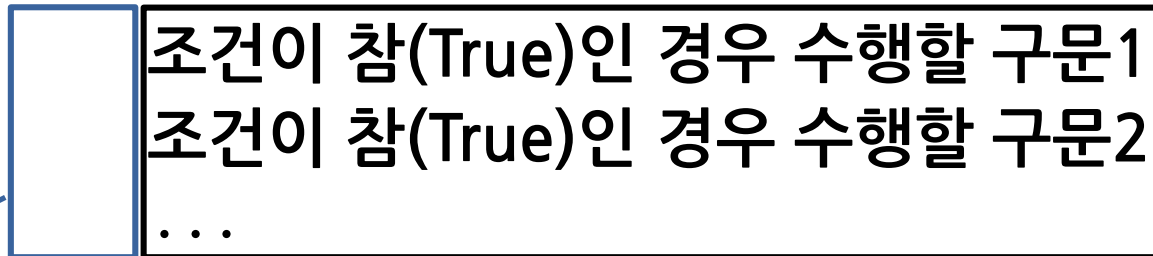
※ 조건문은 반드시 참/거짓을  
판별할 수 있는 논리 연산이어야  
한다.

※ 조건문 뒤에는  
콜론 ':' 기호를  
기재해야 한다.

# 제어문 (Control Statement)

- 선택 구문 - if

if 조건문 :



※ if 구문 아래에 위치해 있는 구문들은 동일한 간격의 들여쓰기(indentation)를 해야 한다.

※ 들여쓰기는 일반적으로 탭(tab) 또는 공백(space) 4회를 사용한다.

※ 동일한 들여쓰기 간격으로 모여 있는 구문들을 블록(block)이라고 한다.

※ 지금 이 블록은 if 구문에 속해 있으므로 if 블록이라고 부른다.

# 제어문 (Control Statement)

- 선택 구문 - if

if 조건문 :

조건이 참(True)인 경우 수행할 구문1

조건이 참(True)인 경우 수행할 구문2

...

- 실수 3.14가 정수 2보다 크면  
“3.14가 큰 수입니다.” 및  
“2가 작은 수입니다.” 라는  
메시지를 순서대로 화면에  
출력하시오.

```
1 if 3.14 > 2:  
2     print("3.14가 큰 수입니다.")  
3     print("2가 작은 수입니다.")
```

3.14가 큰 수입니다.  
2가 작은 수입니다.

# 제어문 (Control Statement)

---

- 선택 구문 - else

if 조건문 :

조건이 참(True)인 경우 수행할 구문1

...

else :

조건이 거짓(False)인 경우 수행할 구문1

...

- 조건문을 검사하여 그 결과가 참(True)이면 if 블록의 구문들을 순서대로 한번씩 실행한다.
- 조건문을 검사한 결과가 거짓(False)이면 else 블록, 즉 else 아래에 있는 구문들을 순서대로 한번씩 실행한다.

# 제어문 (Control Statement)

- 선택 구문 - else

if 조건문 :

조건이 참(True)인 경우 수행할 구문1

else ::

조건이 거짓(False)인 경우 수행할 구문1  
...

※ else 구문 아래에 위치해 있는 구문들도 동일한 간격으로 들여쓰기를 해야 한다.

※ 단어 else 뒤에 콜론 ':' 기호를 기재해야 한다.

- ※ 반드시 if 구문이 먼저 나온 뒤 else 구문이 뒤따라 나오며, if 구문 없이 else 구문만 독립적으로 사용할 수는 없다.



# 제어문 (Control Statement)

- 선택 구문 - else

**if 조건문 :**

조건이 참(True)인 경우 수행할 구문1

...

**else :**

조건이 거짓(False)인 경우 수행할 구문1

...

- 실수 3.14가 정수 2보다 크면  
“3.14가 큰 수입니다.”를,  
그렇지 않으면 “2가 큰 수  
입니다.”를 화면에 출력하시오.

```
1 if 3.14 > 2:  
2     print("3.14가 큰 수 입니다.")  
3 else:  
4     print("2가 큰 수 입니다.")
```

3.14가 큰 수 입니다.

# 제어문 (Control Statement)

---

- 선택 구문 - elif

if 조건문1 :

조건1이 참인 경우 수행할 구문1

...

elif 조건문2 :

조건1이 거짓이고 조건2가 참인 경우 수행할 구문1

...

elif 조건문3 :

조건1, 2가 거짓이고 조건3이 참인 경우 수행할 구문1

...

...

else :

조건이 거짓(False)인 경우 수행할 구문1

...

# 제어문 (Control Statement)

---

- 선택 구문 - elif

if 조건문1 :

조건1이 참인 경우 수행할 구문1

...

elif 조건문2 :

조건1이 거짓이고 조건2가 참인 경우 수행할 구문1

...

elif 조건문3 :

조건1, 2가 거짓이고 조건3이 참인 경우 수행할 구문1

...

- 여러 조건문을 순서대로 검사하여 그 결과가 참(True)일 때 해당 블록의 구문들을 순서대로 한번씩 실행한다.

# 제어문 (Control Statement)

---

- 선택 구문 - elif

if 조건문1 :

조건1이 참인 경우 수행할 구문1

...

elif 조건문2:

조건1이 거짓이고 조건2가 참인 경우 수행할 구문1

...

elif 조건문3:

조건1, 2가 거짓이고 조건3이 참인 경우 수행할 구문1

...

※ if 구문 없이 elif 구문만 독립적으로 사용할 수는 없다.

※ elif는 개수의 제한 없이 필요한 만큼 사용할 수 있다.

# 제어문 (Control Statement)

- 선택 구문 - elif
  - 사용자에게 정수를 1개 입력 받아서, 그 값이 양수이면 “양수입니다.”를, 음수이면 “음수입니다.”를, 둘 다 아니면 “0입니다.”를 화면에 출력하시오.

```
1 v = int(input("정수를 입력하세요: "))
2
3 if v > 0:
4     print("양수입니다.")
5 elif v < 0:
6     print("음수입니다.")
7 else:
8     print("0입니다.")
```

정수를 입력하세요: 3  
양수입니다.

정수를 입력하세요: -7  
음수입니다.

정수를 입력하세요: 0  
0입니다.

# 논리 연산자 다시보기

- 참 또는 거짓의 논리 값을 결과로 도출하는 연산자

연산자	기능	참고사항
<	작다	동일한 자료형 또는 동일한 자료 구조끼리 비교 가능하다.
>	크다	
<=	작거나 같다	
>=	크거나 같다	
==	같다	서로 다른 자료형 또는 자료 구조 간의 비교가 가능하다.
!=	같지 않다	

# 논리 연산자 다시보기

- 참 또는 거짓의 논리 값을 결과로 도출하는 연산자

연산자	기능	설명
not	부정	논리 값을 반대로 만든다.
and	그리고	양 쪽의 논리 값이 모두 참이면 전체가 참이 된다. 둘 중 하나라도 거짓이면 전체가 거짓이 된다.
or	또는	양 쪽의 논리 값 중 하나라도 참이면 전체가 참이 된다. 둘 모두 거짓이면 전체가 거짓이 된다.

# 제어문 (Control Statement)

- 선택 구문의 여러 가지 예

```
1 a = 1
2 b = 2
3 c = 3.14
4 d = "Hello"
5 e = True
6
7 if a > 0 and b < 0:
8     print("a는 양수이고 b는 음수인 경우 출력됩니다.")
9 elif c <= b:
10    print("일단 a가 양수가 아니거나 b가 음수가 아닙니다.")
11    print("그 상황에서 c가 b보다 작거나 같은 경우 출력됩니다.")
12 elif d[0] == "h" or e == True:
13    print("일단 a가 양수가 아니거나 b가 음수가 아닙니다.")
14    print("또한 c가 b보다 큼니다.")
15    print("그 상황에서 d의 첫 글자가 'h'이거나 e가 참인 경우 출력됩니다.")
16 else:
17    print("위의 모든 조건들이 전부 거짓인 경우 출력됩니다.")
```

일단 a가 양수가 아니거나 b가 음수가 아닙니다.

또한 c가 b보다 큼니다.

그 상황에서 d의 첫 글자가 'h'이거나 e가 참인 경우 출력됩니다.



# 제어문 (Control Statement)

- 선택 구문의 여러 가지 예

```
1 a = [1, 2, 3]
2
3 if 5 in a:
4     print("정수 5가 리스트 a의 항목인 경우 출력됩니다.")
5 elif 1 not in a:
6     print("일단 정수 5가 리스트 a에 없습니다.")
7     print("그 상황에서 정수 1이 리스트 a의 항목이 아닌 경우 출력됩니다.")
8 elif a > [3, 2, 1]:
9     print("일단 정수 5가 리스트 a에 없습니다.")
10    print("또한 정수 1이 리스트 a의 항목입니다.")
11    print("그 상황에서 리스트 a가 리스트 [3, 2, 1]보다 큰 경우 출력됩니다.")
12 else:
13    print("위의 모든 조건들이 전부 거짓인 경우 출력됩니다.")
14    a.append(4)
15
16 print("if/elif/else 구문이 끝났고, 이어서 그 다음 줄부터 실행됩니다.")
17 print(a)
```

위의 모든 조건들이 전부 거짓인 경우 출력됩니다.

if/elif/else 구문이 끝났고, 이어서 그 다음 줄부터 실행됩니다.

[1, 2, 3, 4]

# 제어문 (Control Statement)

- 선택 구문의 여러 가지 예

```
1 a = 3
2
3 if a:
4     print("값이", a, "입니다.")
```

값이 3 입니다.

```
1 if False:
2     print("if 구문의 조건이 참인 경우 출력됩니다.")
3 else:
4     print("if 구문의 조건이 거짓인 경우 출력됩니다.")
```

if 구문의 조건이 거짓인 경우 출력됩니다.

```
1 if "False":
2     print("if 구문의 조건이 참인 경우 출력됩니다.")
3 else:
4     print("if 구문의 조건이 거짓인 경우 출력됩니다.")
```

if 구문의 조건이 참인 경우 출력됩니다.

# 제어문 (Control Statement)

---

- 선택 구문의 여러 가지 예

```
1 a = []
2
3 if len(a) == 0:
4     print("리스트가 비어 있습니다.")
5 else:
6     print("리스트의 내용은", a, "입니다.")
```

리스트가 비어 있습니다.

```
1 a = []
2
3 if len(a) != 0:
4     print("리스트의 내용은", a, "입니다.")
5 else:
6     print("리스트가 비어 있습니다.")
```

리스트가 비어 있습니다.

# 제어문 (Control Statement)

---

- 선택 구문의 여러 가지 예

```
1 a = []
2
3 if a == []:
4     print("리스트가 비어 있습니다.")
5 else:
6     print("리스트의 내용은", a, "입니다.")
```

리스트가 비어 있습니다.

```
1 a = []
2
3 if a:
4     print("리스트의 내용은", a, "입니다.")
5 else:
6     print("리스트가 비어 있습니다.")
```

리스트가 비어 있습니다.

# 제어문 (Control Statement)

- 선택 구문의 중첩 (nested statement)
  - 구문은 필요에 따라서 얼마든지 중첩되어(nested) 사용할 수 있다.
  - 중첩된 하위 블록은 상위 블록에 종속(dependent)된다.

```
1 a = 6
2
3 if a % 2 == 0 and a % 3 == 0:
4     print("a는 6의 배수입니다.")
```

a는 6의 배수입니다.

```
1 a = 6
2
3 if a % 2 == 0:
4     if a % 3 == 0:
5         print("a는 6의 배수입니다.")
```

a는 6의 배수입니다.

# 제어문 (Control Statement)

- 선택 구문의 중첩 (nested statement)

```
1 a = "1"
2 b = "2"
3 c = "3"
4
5 if a == "1":
6     print(a)
7     if b == "second":
8         print("second")
9     else:
10        if c == "third":
11            print("third")
12        elif c == "3":
13            print(c)
14        else:
15            print("세번째")
16    print(b)
17 else:
18    print("first")
```

1  
3  
2