

파이썬

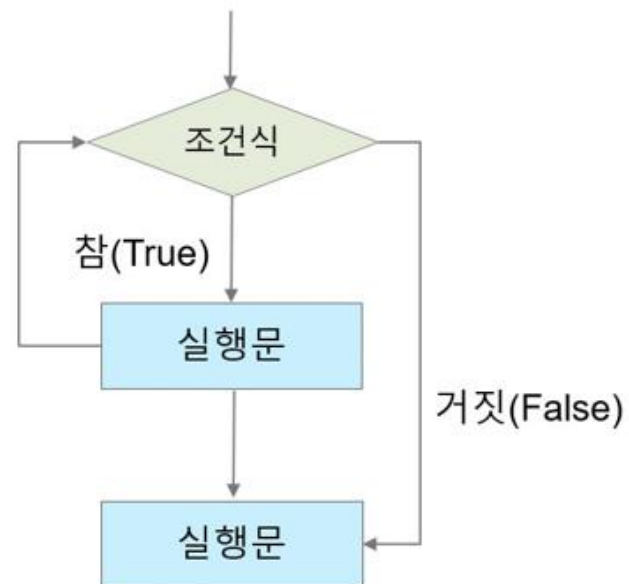
8강. 제어문(조건문)

1. 제어문

- 제어문은 프로그램의 흐름을 변경해 주는 조건문과 특정 명령문들을 반복해서 수행하는 반복문으로 구성된다.
- 특히 제어문은 프로그래밍의 논리적인 순서를 정의할 때 이용되며, 그 사용범위는 무궁무진하다.
- 따라서 어떤 문제를 해결하기 위해서 논리적인 절차를 기술할 때 반드시 필요한 프로그래밍 문법으로 문제 해결 능력에 따라서 고급 개발자의 진입 여부가 결정된다.

2. 조건문

- 조건문은 특정 조건식에 따라서 실행문의 실행이 결정되는 명령문으로 파이썬에서는 if, elif, else 등의 명령어를 제공한다.
- 조건식을 대상으로 비교 판단하는 조건문의 순서도이다.
- 명령문의 형식에 따라서 조금씩 다르지만 제어문은 조건식의 비교 판단에 의해서 수행되는 것을 알 수 있다.



3. 단일 조건문

- 단일 조건문은 조건식 한 개를 기준으로 비교 판단하여 처리하는 형식을 갖는다. if 명령어 다음에 조건식을 지정하고, 콜론(:) 기호를 이용하여 수행할 실행문들을 블록 처리한다.
- C, Java에서는 실행문들의 블록으로 중괄호({}) 기호를 이용하지만 파이썬은 처리할 실행문들을 콜론과 들여쓰기 (Indentation)로 정의한다.
- 단일 비교 판단문의 기본 형식은 들여쓰기는 Tab 키 (뿔)를 이용한다.

```
if 조건식 :  
    실행문  
    실행문  
    } if 블록
```

3. 단일 조건문

chapter03.lecture.step01_if.py ↵

Python Console ↵

```
var = 10 # if 블록에서 사용될 변수 ↵  
if var >= 5 : # 조건식 ↵  
    print('var=', var)  
    print('var는 5보다 크다.')  
    print('조건이 참인 경우  
    실행') ↵  
  
print('항상 실행') ↵
```

```
var= 10 ↵  
var는 5보다 크다.  
조건이 참인 경우  
실행 ↵  
  
항상 실행 ↵
```

3. 단일 조건문

- var변수는 if 블록에서 조건식으로 사용되는 변수이다. 조건식은 산술, 관계, 논리 연산자를 이용하여 정의할 수 있다.
- 현재 조건이 'var >= 5'이므로 var는 5보다 크거나 같기 때문에 조건은 참(True)이다.
- 따라서 콜론과 들여쓰기로 블록이 지정된 3개의 실행문이 순서대로 실행되어 print 함수의 결과가 출력된다.
- 반복문이 아니기 때문에 실행문은 1회만 실행된다.
- if 블록이 모두 실행되면 프로그램 순서에 의해서 if 블록 다음 문장이 실행된다..

3. 단일 조건문

- 단일 조건문의 두 번째 형식은 다음과 같다.
- If에서 정의한 조건식이 참이면 첫 번째 블록이 수행되고, 조건식이 거짓이면 두 번째 블록이 수행된다.

```
if 조건식 :  
    실행문  
    실행문↵  
else :↵  
    실행문  
    실행문↵
```

} 첫 번째

} 두 번째

3. 단일 조건문

- 단일 조건문의 두 번째 형식은 다음과 같다.
- If에서 정의한 조건식이 참이면 첫 번째 블록이 수행되고, 조건식이 거짓이면 두 번째 블록이 수행된다.

chapter03.lecture.step01_if.py ↵

Python Console ↵

```
# 100~85 : '우수', 84~70: '보통', 69이하 : '저조' ↵  
score = int(input('점수 입력 : ')) ↵  
if score >= 85 and score <=100 :  
    print('우수') ↵  
else : ↵  
    if score >= 70 :  
        print('보통') ↵  
  
else : ↵  
    print('저조') # 69이하 ↵
```

점수 입력 : >? 92 ↵
우수 ↵

점수 입력 : >? 80 ↵
보통 ↵

점수 입력 : >? 55 ↵
저조 ↵

4. 중첩 조건문

- 중첩 조건문은 if 블록 내에 또 다른 if 블록이 포함된 형태를 갖는다.
- 주로 여러 개의 조건식을 비교 판단하여 로직을 구현할 때 이용된다.
- 중첩 비교판단문의 형식은 다음과 같다.
- 첫 번째 조건식1이 참이면 실행문1이 실행되고, 거짓이면 elif 명령어 뒤에 오는 두 번째 조건식2를 지정할 수 있다.
- 두 번째 조건식2가 참이면 실행문2가 실행되고, 거짓이면 마지막 else 블록의 실행문n이 실행된다.

```
if 조건식1 :  
    실행문1  
elif 조건식2 :  
    실행문2  
else :  
    실행문n
```

4. 중첩 조건문

- 키보드로 점수를 입력받아서 100~85점이면 '우수', 84~70점이면 '보통', 69이하 이면 '저조' 라고 평가하는 중첩 비교 판단문의 예문이다.
- '단일 비교 판단문 형식2'의 예문과 결과는 동일하지만 중첩 비교 판단문을 이용하면 프로그램의 가독성이 좋아진다.

chapter03.lecture.step01_if.py ↵

```
score = int(input('점수 입력 : '))
grade = '' # 등급 ↵

if score >= 85 and score <=100 :
    grade = '우수' ↵
elif score >= 70 :
    grade = '보통' ↵
else : ↵
    grade = '저조' ↵

print('당신의 점수는 %d이고, 등급은 %s'%(score, grade)) ↵
```

Python Console ↵

```
↵
점수 입력 : >? 89 ↵
당신의 점수는 89이고, 등급은 우수입니
다. ↵
↵
점수 입력 : >? 75 ↵
당신의 점수는 75이고, 등급은 보통입니
다. ↵
↵
점수 입력 : >? 60 ↵
당신의 점수는 60이고, 등급은 저조 ↵
```

5. 삼항 조건문

- 삼항 조건문은 삼항 연산자라고 불려진다. 조건식이 참인 경우와 거짓인 경우의 처리할 문장을 한 줄로 작성하고, 조건식의 비교 판단 결과에 따라서 선택적으로 각 실행문이 실행된다.
- 삼항 조건문의 형식은 다음과 같다.

```
변수 = 참    if (조건문)  else 거짓 ;
```

5. 삼항 조건문

- 삼항 조건문은 삼항 연산자라고 불려진다. 조건식이 참인 경우와 거짓인 경우의 처리할 문장을 한 줄로 작성하고, 조건식의 비교 판단 결과에 따라서 선택적으로 각 실행문이 실행된다.
- 삼항 조건문의 형식은 다음과 같다.

```
변수 = 참    if (조건문)  else 거짓 ;
```

5. 삼항 조건문

- 일반 조건문에서 num이 5보다 크거나 같으면 2를 곱하고, 그렇지 않으면 2를 더하는 명령문을 3항 연산자로 코딩한 예문이다.
- if 다음에 조건식을 지정하고, if 앞부분에 참일 경우 실행되는 문장, 뒷부분은 거짓일 경우 실행되는 문장을 쓴다.
- num은 5보다 크거나 같기 때문에 num에 2가 곱해져서 result(result2)에는 18이 계산된다.

5. 삼항 조건문

chapter03.lecture.step01_if.py ↵

Python Console ↵

```
# (1) 일반 조건문  
num = 9 # 초기화  
result = 0 ↵
```

```
if num >= 5 : ↵  
    result = num * 2 ↵  
else : ↵  
    result = num + 2 ↵  
print('result =', result) ↵
```

result = 18 ↵

```
# (2) 3항 연산자 ↵  
# 형식) 변수 = 참 if (조건문) else 거짓 ↵  
result2 = num * 2 if num >= 5 else num + 2 ↵  
print('result2 =', result2) # 18 ↵
```

result2 = 18 ↵

THANK YOU