

Python

조건문

Spring 2025



AI융합학과

Seongbok Baik

sbbaik@dju.kr

00 Text Book



교재명	으뜸 파이썬
저자	박동규, 강영민
출판사	생능출판사
발행년	2024.06.14



학습목표

- 순차적 실행과 조건문, 반복문과 같은 제어문의 차이를 이해하고 활용할 수 있다.
- 조건문의 개념과 사용법에 대해 이해한다.
- 조건식에 대해 알아보고 복잡한 조건식을 구성할 수 있다.
- 블록과 들여쓰기에 대해 이해하고 중첩된 블록을 구성할 수 있다.
- if 문과 if-else 문 등 다양한 조건문에 대해 이해한다.

3.1 순차문 **sequential statements**

- 순차적 구조
 - 먼저 나타나는 순서에 따라 실행되는 프로그래밍 언어의 문장



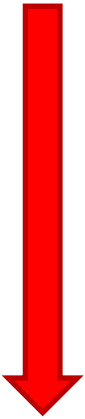
코드 3-1: 순차적 실행 구조를 이용한 변수의 덧셈
seq_test.py

```
num = 100
print('num =', num) # 1000| 출력됨
num = num + 100
print('num =', num) # num에 100이 더해져 2000| 출력됨
num = num + 100
print('num =', num) # num에 다시 100이 더해져 3000| 출력됨
```

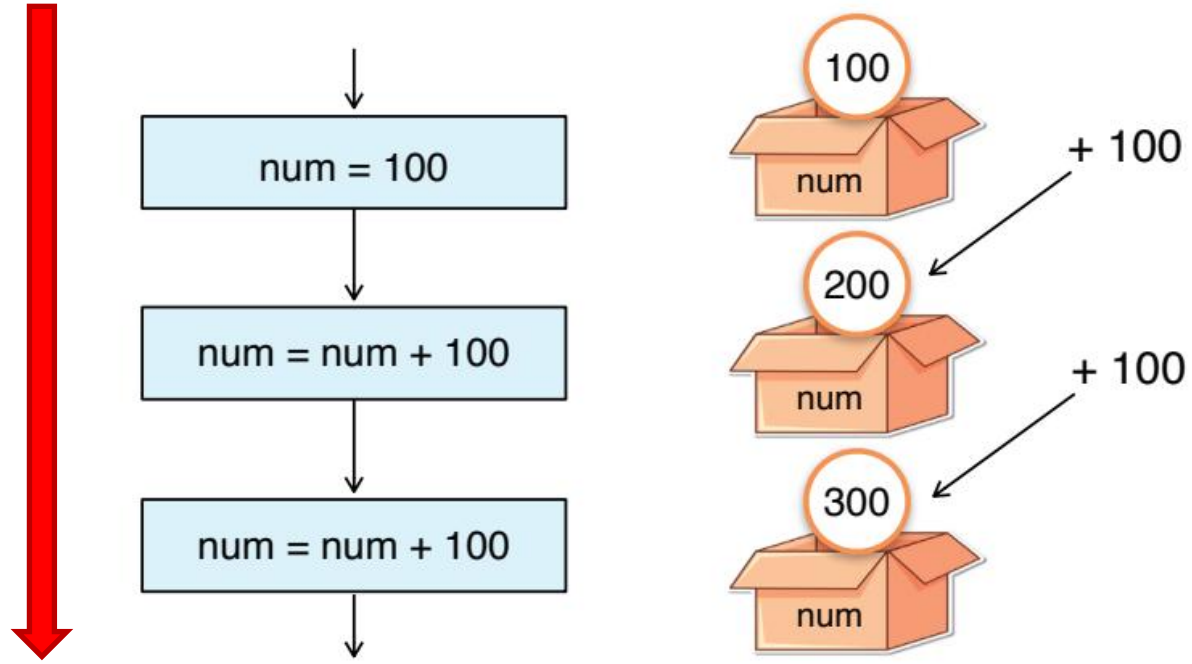
변수의 값은 실행 순서에 따라 순차적으로
100에서 200, 300으로 변한다

실행결과

```
num = 100
num = 200
num = 300
```



3.1 순차문 sequential statements



[그림 3-1] [코드 3-1]의 순차적인 실행 순서를 나타내는 흐름도와 변수 값의 변화

3.1 순차문 sequential statements

- 순차문 sequential statements 이외의 흐름문 flow statements
- 제어문 control statements : 프로그램의 흐름을 제어하는 역할
 - 조건문 conditional statements
 - » if 문, if-else 문
 - 반복문
 - » for 문, while 문
 - 반복문의 흐름 변경
 - » Break 문, continue 문

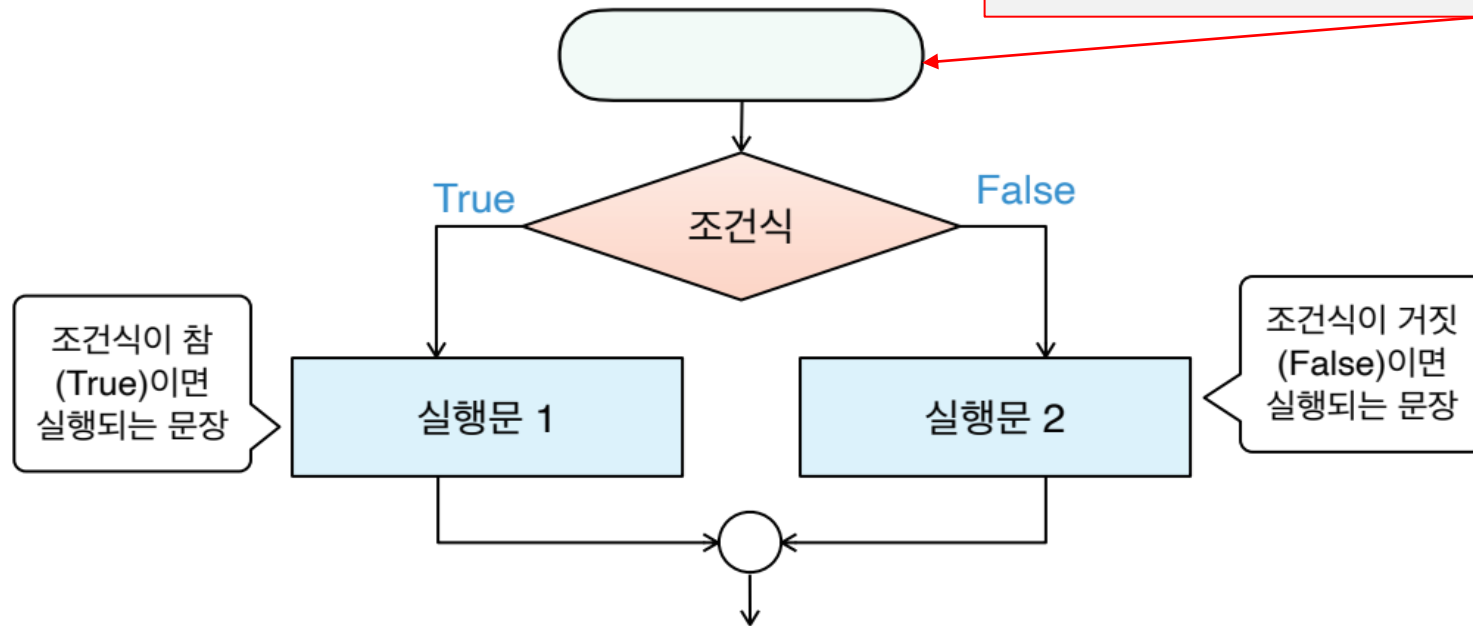
순차적 실행문 말고도 조건문, 제어문 등의 다양한 문장으로 복잡한 일을 수행할 수 있다.

3.2 if 조건문

- 조건문 conditional statements

- 특정한 조건에 따라 실행 내용이 달라지는 문장 형태
- 조건식은 True 혹은 False를 반환
- True면 실행문 1, False면 실행문 2를 실행

흐름도 :
이와 같은 작업의 흐름을 나타내는 그림을 흐름도
라고 한다.

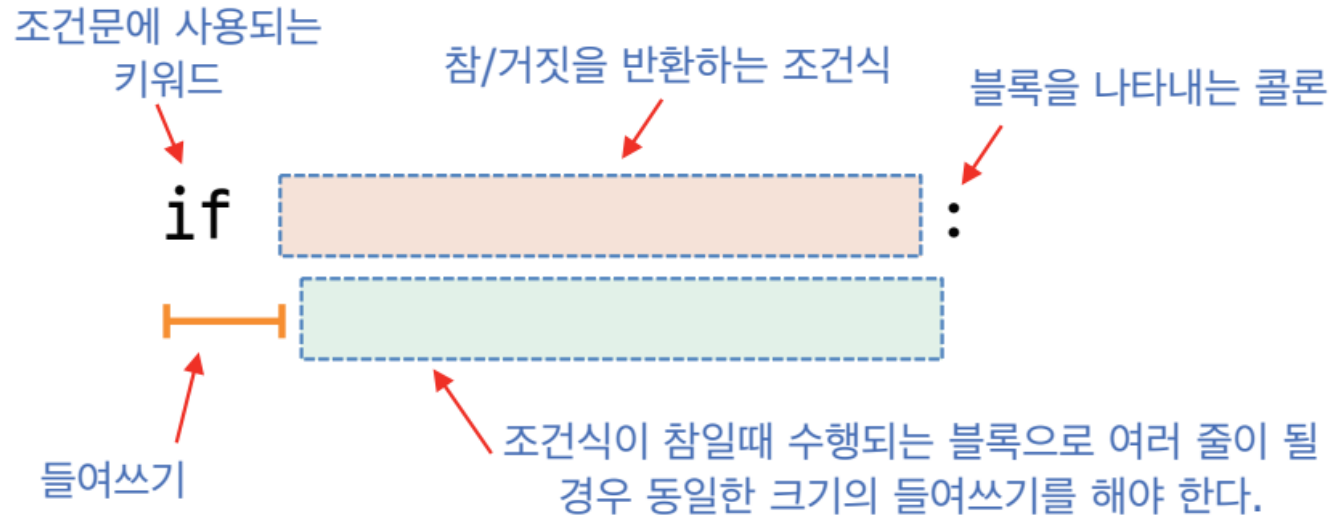


3.2 if 조건문

- 조건문의 구체적 상황
- 상황 1 : 나이가 20세 미만이면 '청소년 할인'을 출력하는 기능
- 상황 2 : 1,000걸음 이상을 걸으면 '목표 달성'을 출력하는 기능

3.2 if 조건문

- if 조건문의 사용법



[그림 3-3] if 조건문의 사용법

파이썬은 들여쓰기가 아주 중요하다

3.2 if 조건문

- if 조건문의 사용법

상황 1 :

나이(age)가 20세 미만이면 '청소년 할인'을 출력



```
if age < 20 :  
    print('청소년 할인')
```

상황 2 :

걸음(walk_count)이 1000 이상이면 '목표 달성'을 출력



```
if walk_count >= 1000 :  
    print('목표 달성')
```

- (상황 1) - 콜론(:)앞에 나타나는 조건문 절에서 < 연산자를 이용해 나이(age)가 20세 미만인 경우에만 print('청소년 할인')이라는 코드를 실행
- (상황 2) - 조건문 절에서 >= 연산자를 이용해 걸음(walk_count)이 1,000 이상이 되면 print('목표 달성')이라는 코드를 실행

3.2 if 조건문

- 상황 1



코드 3-2: if 조건문을 이용한 출력 기능(조건을 만족하는 경우)

`if_youth_discount.py`

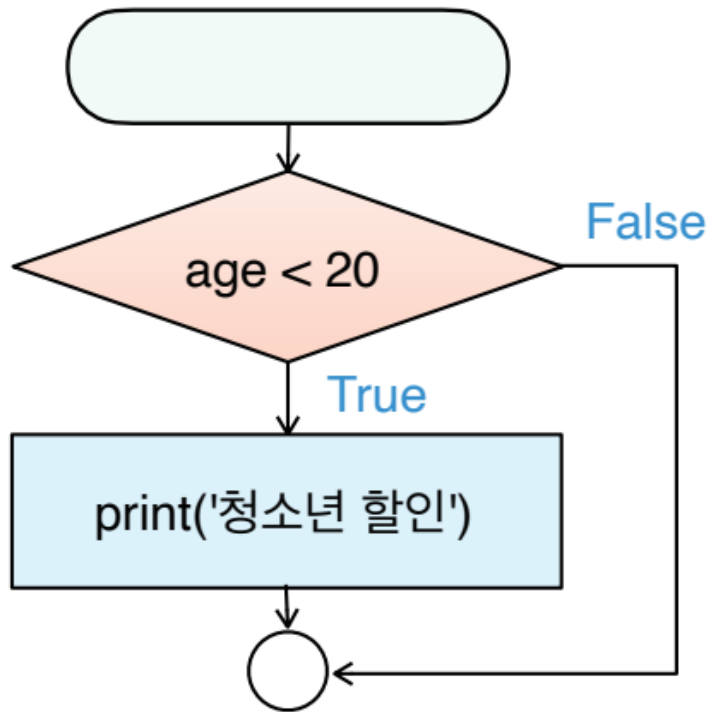
```
age = 18      # age가 20 미만의 값
if age < 20:  # age < 20 조건식의 결과는 True임
    print('청소년 할인')
```

실행결과

청소년 할인

3.2 if 조건문

• 상황 1

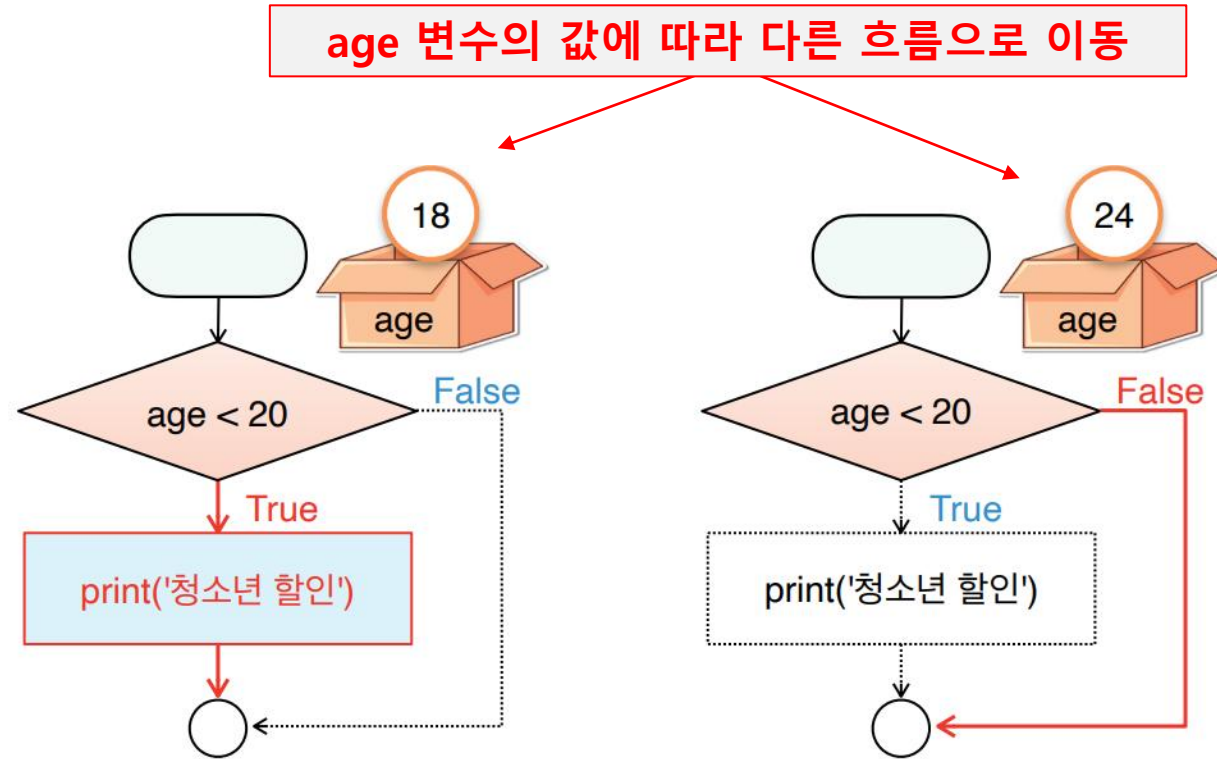


[그림 3-5] 흐름도로 표현한 [코드 3-2]의 실행 구조

- age값에 18을 할당하면 $age < 20$ 이라는 조건문이 참(True)이 되므로 '청소년 할인'이 화면에 출력
- age 값이 20 이상이 되면 아무런 출력을 내보내지 않음

3.2 if 조건문

• 상황 1



[그림 3-6] 변수 age의 변화에 따라 제어되는 [코드 3-2]의 실행 흐름

3.2 if 조건문

- 상황 1



코드 3-3: if 조건문을 이용한 출력 기능(조건을 만족하지 않는 경우)
`if_youth_discount2.py`

```
age = 24      # age가 20 이상의 값  
if age < 20:  # age < 20 조건식의 결과는 False임  
    print('청소년 할인')
```

실행결과 (아무런 출력이 없음)

age 변수의 값이 24이면 아무런 출력도 없음

3.2 if 조건문

- 상황 2
- 걸음걸이 수(walk_count)가 1,000 이상인 경우에 '목표달성'을 출력하는 프로그램



코드 3-4: if 조건문을 이용한 출력 기능(조건을 만족하는 경우)

if_walk_count.py

```
walk_count = 1200
if walk_count >= 1000: # walk_count >= 1000 조건식의 결과는 True임
    print('목표 달성')
```

실행결과

목표 달성

3.2 if 조건문

- 상황 2
- 첫 줄을 `walk_count = 800`과 같이 수정한다면 아무런 출력결과를 볼 수 없음

```
walk_count = 800
```

if 문의 조건을 만족하지 않을 경우 아무것도 출력되지 않음

실행결과(walk_count = 800으로 수정한 후 실행 결과)

3.2 if 조건문

• 상황 2



LAB 3-1 : if 문의 사용법

1. 게임 사용자의 게임 점수(game_score)가 1,000점 이상이면 '당신은 고수입니다'를 출력하는 프로그램을 if 문을 이용하여 작성하시오. 이때 다음과 같이 game_score 값을 화면에 출력하시오. game_score에 800점, 1,300점을 각각 입력하여 출력문을 확인하시오.

```
game_score = 800
```

혹은

```
game_score = 1300  
당신은 고수입니다
```

2. num_a와 num_b에 할당된 값이 같으면 '두 값이 일치합니다.'를 출력하는 프로그램을 if 문을 이용하여 작성하시오. num_a와 num_b에 각각 100과 200이 할당되어 있는 경우와 num_a와 num_b에 300과 300이 할당되어 있는 경우에 대하여 각각 코드를 작성하고 출력문을 확인하시오.

```
num_a = 100, num_b = 200
```

혹은

```
num_a = 300, num_b = 300  
두 값이 일치합니다.
```

3.2 if 조건문

• 3.2.1 조건문과 블록

- 블록 **block**

- 어떤 조건을 만족하는 경우에 특정한 코드를 선택적으로 실행하는 구조
- 이때 실행될 코드 덩어리
- 블록은 반드시 들여쓰기를 해야 한다.



코드 3-5: 들여쓰기 없는 print() 문

if_youth_error.py

```
age = 18
if age < 20:
print('청소년 할인')  # 주의: 들여쓰기 없는 print() 문
```

들여쓰기(indentation) 블록이 필요하다는 의미임

실행결과

IndentationError: expected an indented block

3.2 if 조건문

- 파이썬의 들여쓰기
- 파이썬은 들여쓰기가 매우 중요한 의미를 가지는 프로그래밍 언어
 - C/C++이나 Java, Pascal 등 전통적인 프로그래밍 언어와 다른 특징

[들여쓰기 코드 1] 조건을 만족하는 경우	[들여쓰기 코드 2] 조건을 만족하지 않는 경우
<pre>age = 18 if age < 20: print('청소년 할인') print('입장을 환영합니다')</pre>	<pre>age = 24 if age < 20: print('청소년 할인') print('입장을 환영합니다')</pre>
수행결과 - if 문 내부와 외부 print 문이 수행됨	수행결과 - if 문 외부의 print 문만 수행됨
청소년 할인 입장을 환영합니다	입장을 환영합니다

3.2 if 조건문

- 파이썬의 들여쓰기

[들여쓰기 코드 3] 조건을 만족하는 경우	[들여쓰기 코드 4] 조건을 만족하지 않는 경우
<pre>age = 18 if age < 20: print('나이', age) print('청소년 환영') print('청소년 할인')</pre>	<pre>age = 24 if age < 20: print('나이', age) print('청소년 환영') print('청소년 할인')</pre>
수행결과 - 들여쓰기 블록 전체가 수행됨	수행결과 - 들여쓰기 블록 전체가 수행되지 않음
나이 18 청소년 할인 입장을 환영합니다	

3.2 if 조건문

- 파이썬의 들여쓰기
- **블록**은 흔히 **코드 블록**이라고도 함
- 소스 코드에서 함께 묶을 수 있는 코드의 덩어리를 말한다
- 파이썬은 if문 다음에 :(콜론)이 나오면 다음에 들여쓰기 코드 블록이 나와야 하며 for, while, def, class 등에서도 코드 블록이 사용됨

3.2 if 조건문

- 파이썬의 들여쓰기
- [들여쓰기 코드 5]와 같이 동일한 블록에 대해 들여쓰기의 정도(칸 수)가 일정하지 않으면 "IndentationError: unexpected indent"라는 들여쓰기 오류가 발생
- 동일한 코드 블록에서는 들여쓰기의 칸 수를 반드시 일치시켜야 함
- 스페이스_{space} 4칸을 권장

[들여쓰기 코드 5]

들여쓰기가 잘못된 경우(들여쓰기의 크기가 일정하지 않음)

```
age = 18
if age < 20:
    print('나이', age) # 네 칸 들여쓰기
    print('청소년 환영') # 한 칸 들여쓰기
```

수행결과

IndentationError: unexpected indent 오류 발생

3.2 if 조건문

- 블록의 규칙



NOTE: 들여쓰기 블록의 규칙

파이썬에서 들여쓰기 칸수는 몇 칸이든 상관없지만, 4칸을 권장한다. 대부분의 파이썬 **통합 개발환경IDE: integrated development environments**에서는 자동 들여쓰기 기능을 제공한다. 또한, 들여쓰기를 할 때 탭보다는 스페이스 키를 권장한다. 블록은 여러 줄로 작성할 수 있다. 단, 여러 줄 들여쓰기를 하는 경우 들여쓰기 칸 수가 모두 같아야 한다.

3.2 if 조건문



NOTE: PEP 문서

PEP 8 – Style Guide for Python Code

Author: Guido van Rossum <guido at python.org>, Barry Warsaw <barry at python.org>, Nick Coghlan <ncoghlan at gmail.com>

Status: Active

Type: Process

Created: 05-Jul-2001

Post-History: 05-Jul-2001, 01-Aug-2013

▼ Table of Contents

- [Introduction](#)

파이썬 소프트웨어 재단 홈페이지에는 파이썬의 여러 가지 기능을 규정하는 **파이썬 확장 제안서** PEP: **python enhanced proposals** 문서가 있다. 이 PEP 문서는 파이썬 사용자와 개발자 공동체의 합의된 내용을 담고 있는데 이 중에서 PEP 8 문서는 파이썬의 코딩 스타일을 규정하고 있다. 이 문서의 내용을 살펴보면 다음과 같이 파이썬에서 탭을 사용할지 스페이스를 사용할지에 대한 합의된 내용이 나타나 있다. 그 중에서 탭과 스페이스에 관한 규정이 있는데, 요약하면 들여쓰기에서 스페이스 키를 사용하는 것을 더 권장하며, 탭은 이미 탭으로 들여쓰기가 된 코드와 일관성을 유지하기 위해서만 제한적으로 사용하는 것이 낫다고 규정되어 있다. 코딩 시에 일관성 있는 방식으로 코딩하는 습관을 가지는 것은 개발자에게 매우 중요한 덕목이다.

3.2 if 조건문

- 3.2.2 대화형 모드의 블록



대화창 실습: 대화형 모드를 통한 블록과 들여쓰기 실습

```
>>> num = 300
>>> if num > 200:
...     print('num은 200보다 큼니다.')
...
    num은 200보다 큼니다.
>>>
```

3.2 if 조건문

- 3의 배수 판단



코드 3-6: 3의 배수를 판단하기 위한 모듈로 연산과 조건문

`if_modulo1.py`

```
number = int(input('정수를 입력하세요 : ')) # 입력값을 정수형으로 변환
if number % 3 == 0: # 모듈로 3의 결과가 0이면 3의 배수임
    print(number, '은(는) 3의 배수입니다.')
```

실행결과

정수를 입력하세요 : 15
15 은(는) 3의 배수입니다.

실행결과

정수를 입력하세요 : 16

3.2 if 조건문

- 3과 5의 배수 판단



코드 3-7: 3과 5의 배수를 판단하기 위한 모듈로 연산과 and 조건문
`if_modulo2.py`

```
number = int(input('정수를 입력하세요 : '))  
if number % 3 == 0 and number % 5 == 0:  
    print(number, '은(는) 3의 배수이면서 5의 배수입니다.')
```

실행결과

정수를 입력하세요 : 15
15 은(는) 3의 배수이면서 5의 배수입니다.

3.2 if 조건문



LAB 3-2: 변수와 if 조건식 사용하기

- 1에서 100 사이의 임의의 정수 n 을 입력 받아서 ① n 을 화면에 출력한 후, ② n 이 짝수이면 "...은(는) 짝수입니다."를 다음과 같이 출력하는 프로그램을 작성하여라.

정수를 입력하세요 : 50

$n = 50$

50 은(는) 짝수입니다.

또는

정수를 입력하세요 : 50

$n = 50$

3.2 if 조건문

2. -100에서 100 사이의 임의의 정수 n 를 입력 받아서 ① n 을 화면에 출력한 후, ② n 이 0보다 큰 정수이면 "...은(는) 자연수입니다."를 다음과 같이 출력하는 프로그램을 작성하여라. 그렇지 않을 경우 $x = -10$ 과 같이 x 를 단순 출력하여라.

정수를 입력하세요: 50

$n = 50$

50 은(는) 자연수입니다.

또는

정수를 입력하세요 : -10

$x = -10$

Leistung ist nicht alles / Keinen Studierenden zurücklassen

