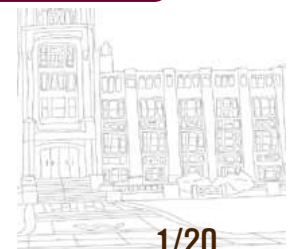




3장. 조건문

파이썬



Contents



❖ 목차

- 1. 불
- 2. if 조건문
- 3. else 구문
- 4. elif 구문
- 5. 연습문제



0. 학습목표

- ❖ 불에 대해 설명할 수 있다.
- ❖ If 조건문의 기본적인 사용 할 수 있다.
- ❖ 현실의 조건과 프로그래밍에서의 조건에 차이가 있다는 것을 이해한다



1. 불(Boolean)

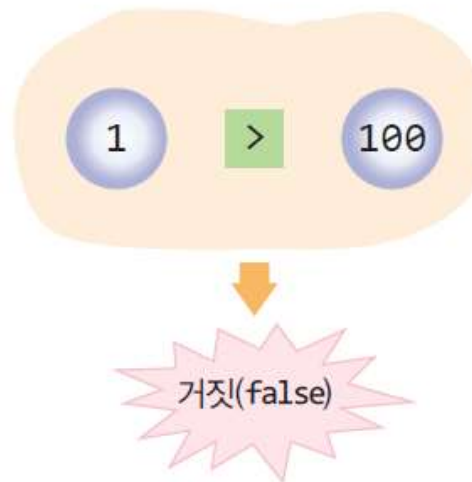
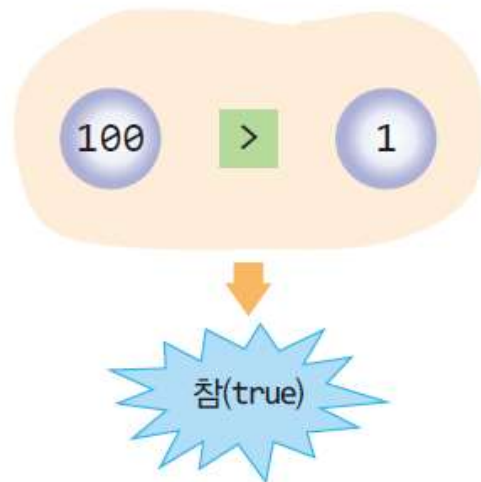
- ❖ 불 (Boolean): 불린, 불리언, 불, 부울
 - 참과 거짓을 나타내는 값으로서의 기본적 자료형
 - 참과 거짓, 오직 두 가지 값만 가짐

```
print(True)  
print(False)
```



True
False

- ❖ 관계 수식은 참(True)이나 거짓(False)을 생성한다.



1. 불(Boolean)

❖ 연산자

- 비교 연산자 통해 불 만듦

연산	의미
$x == y$	x와 y가 같은가?
$x != y$	x와 y가 다른가?
$x > y$	x가 y보다 큰가?
$x < y$	x가 y보다 작은가?
$x \geq y$	x가 y보다 크거나 같은가?
$x \leq y$	x가 y보다 작거나 같은가?



1. 불(Boolean)

- 숫자 & 문자열에 적용

```
print(10 == 100)
print(10 != 100)
print(10 < 100)
print(10 > 100)
print(10 <= 100)
print(10 >= 100)
```



False
True
True
False
True
False

- 문자열 비교시 사전 순서대로 앞에 있는 것이 작은값을 갖는다

```
print("가방" == "가방")
print("가방" != "하마")
print("가방" < "하마")
print("가방" > "하마")
```



True
True
True
False



1. 불(Boolean)

- 범위 구하기

```
x = 25  
print(10 < x < 30)  
print(40 < x < 60)
```



True
False



■ 논리 연산자 통한 불 연산

연산자	설명
not	불을 반대로 전환합니다.
and	피연산자 두 개가 모두 참일 때 True를 출력하며, 나머지의 경우는 모두 False를 출력합니다.
or	피연산자 두 개 중에 하나만 참이라도 True를 출력하며, 두 개가 모두 거짓일 때만 False를 출력합니다.

```
print(not True)
print(not False)
```

→

```
False
True
```

x = 10

`is_under_20 = x < 20`

```
print("is_under_20: ", is_under_20)
```

```
print("not is_under_20: ", not is_under_20)
```

```
is_under_20: True
not is_under_20: False
```



1. 불(Boolean)

■ and 연산자

좌변	우변	결과
True	True	True
True	False	False
False	True	False
False	False	False

■ or 연산자

좌변	우변	결과
True	True	True
True	False	True
False	True	True
False	False	False



1. 불(Boolean)

- Ex)

```
# and 연산자
print(True and True)
print(True and False)
print(False and True)
print(False and False)
print()
```

```
# or 연산자
print(True or True)
print(True or False)
print(False or True)
print(False or False)
```



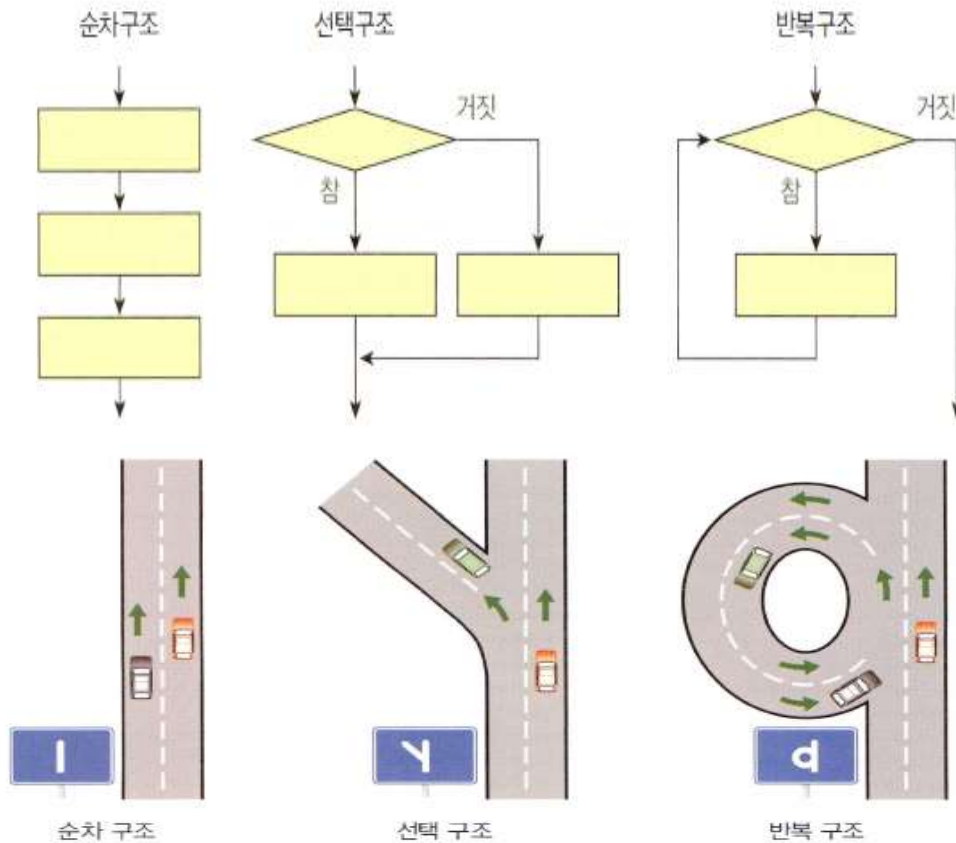
1. 불(Boolean)

- ❖ 논리 연산자 활용
 - and / or 연산자



2. if 조건문

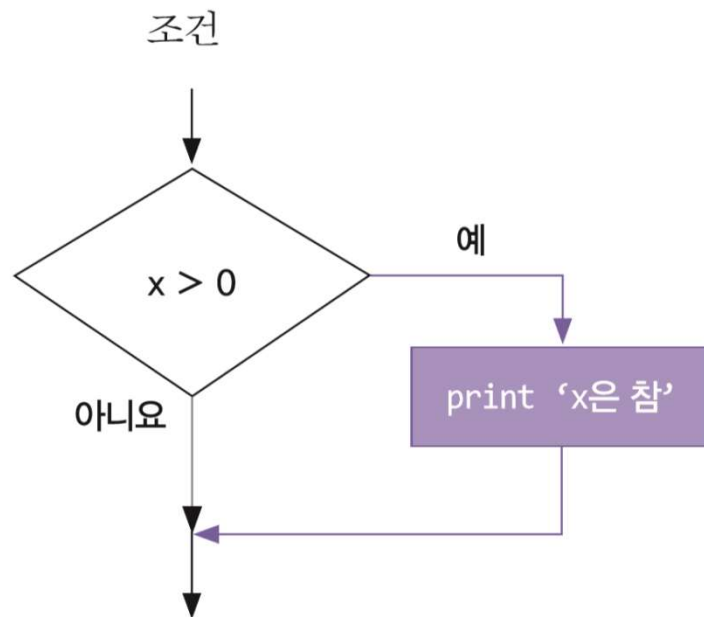
- ❖ 프로그램을 작성할 때, 사용할 수 있는 3가지의 기본 제어 구조가 있다.
 - 순차 구조 (sequence) : 명령들이 순차적으로 실행되는 구조
 - 선택 구조 (selection) : 둘 중의 하나의 명령을 선택하여 실행되는 구조
 - 반복 구조 (iteration) : 동일한 명령이 반복되면서 실행되는 구조



2. if 조건문

❖ 조건이란?

- 제시한 명제가 참이나 거짓인지 판단하고, 판단에 따른 명령을 내리는 것



```
Python 3.6.4 Shell
File Edit Shell Debug Options Window Help
Python 3.6.4 (v3.6.4:d48eceb, Dec 19 2017, 06:04:44)
Type "copyright", "credits" or "license()" for more
>>> a = 3
>>> if a > 2:
>>>     print("a는 2보다 큼니다.")

a는 2보다 큼니다.
>>> |
```

Screenshot of a Python 3.6.4 Shell window showing the execution of an if statement. The code defines a variable 'a' as 3 and uses an if statement to check if 'a' is greater than 2. The output shows the message 'a는 2보다 큼니다.' (a is greater than 2).



2. if 조건문

❖ 조건이란?

- 제시한 명제가 참이나 거짓인지 판단하고, 판단에 따른 명령을 내리는 것

a = 3

a라는 변수에 숫자 3 할당

if a > 2:

만약 a > 2 라는 조건을 만족하면

print("a는 2보다 큽니다.")

"a는 2보다 큽니다."라는 문장을 출력



2. if 조건문

❖ 조건이란?

- 제시한 명제가 참이나 거짓인지 판단하고, 판단에 따른 명령을 내리는 것

Source1	화면 출력
<pre>a = input("과일 이름을 입력하세요 : ") if a == '사과': print("입력한 과일은", a, "입니다.")</pre>	<p>숫자를 하나 입력하세요 : 사과</p> <p>입력한 과일은 사과 입니다.</p>

Source2	화면 출력
<pre>a = int(input("숫자를 하나 입력하세요 : ")) if a == 3: print("입력한 숫자는", a, "입니다.")</pre>	<p>숫자를 하나 입력하세요 : 3</p> <p>입력한 숫자는 3 입니다.</p>



2. if 조건문

❖ 알고리즘과 순서도



개찰구로 간다.

교통카드를 태그한다.

개찰구를 통과한다.

	단말	순서도의 시작과 끝
	흐름선	작업 흐름을 명시
	준비	작업 단계 시작 전 준비 (변수 및 초기치 선언 등)
	처리	처리하여야 할 작업을 명시(변수에 계산 값 입력 등)
	입력	일반적인 데이터의 입력 또는 결과의 출력
	판단	조건에 따라 흐름선을 선택(일반적으로 참, 거짓 구분)
	프린트	프린터를 이용한 출력(서류 등의 지면에 출력)



2. if 조건문

❖ 알고리즘과 순서도

실제행동	의사코드
개찰구로 간다. 만약 교통카드가 있다면 교통카드를 태그한다. 교통카드가 없으면 일회용 교통카드를 발급받는다. 개찰구를 통과한다.	개찰구 이동 if 교통카드: 교통카드를 태그 else: 일회용 교통카드 발급 개찰구 통과



2. if 조건문

❖ 연습문제

- 아래의 결과가 출력되도록 프로그램을 작성하시오

비밀번호는 무엇입니까? 0808

출력 : 비밀번호가 틀렸습니다.

비밀번호는 무엇입니까? 1229

출력 : 문이 열렸습니다.



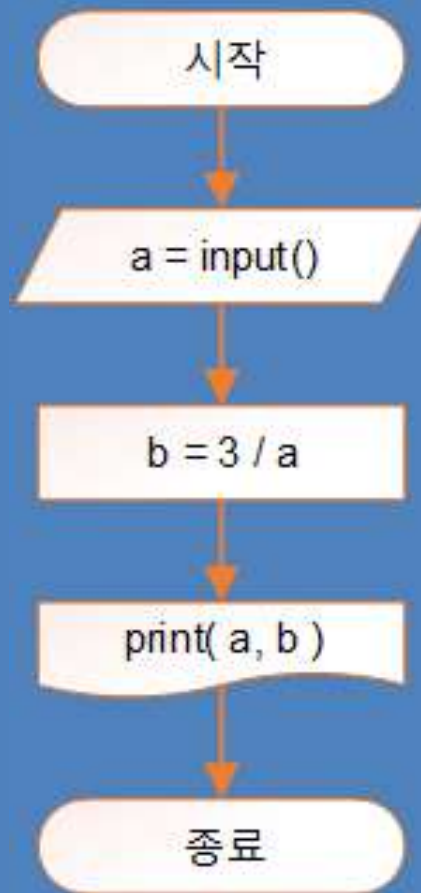
2. if 조건문

- ❖ 소프트웨어는 컴퓨터가 할 일의 목록이라 표현할 수 있다.
 - 프로그래머가 컴퓨터에게 할 일의 목록을 내주면, 컴퓨터는 소프트웨어에 기록되어 있는 목록을 보고 그대로 수행한다.
 - 프로그래밍에서 "흐름(Flow)"은 컴퓨터에서 내려지는 명령의 순서를 가리키는 말.
 - 프로그램의 흐름을 가르는 문장 : 나눌 분(分), 갈림길 기(岐), 분기문

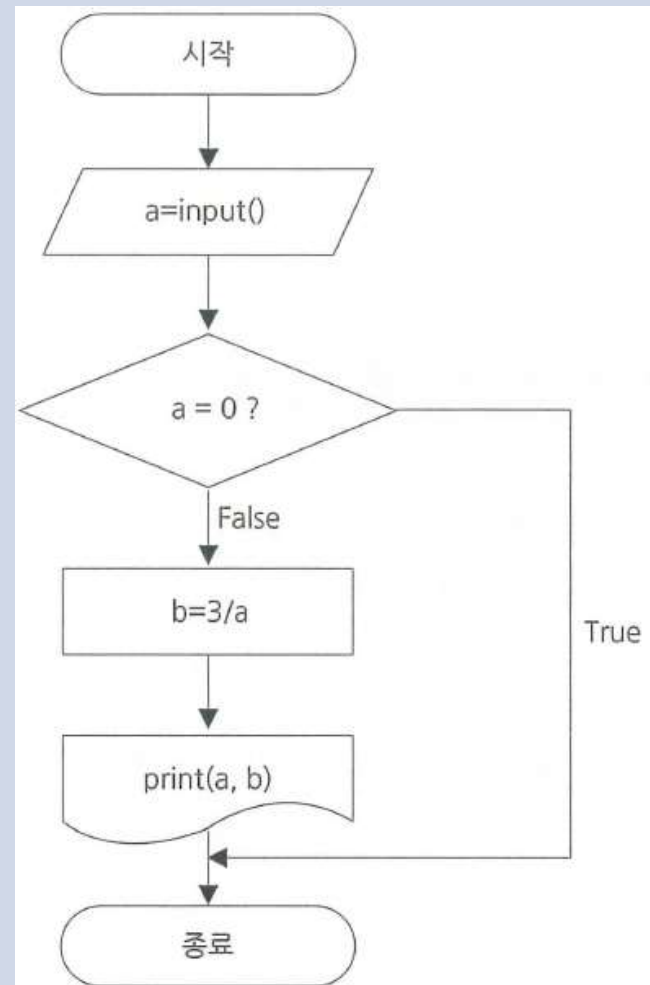
한 방향으로만 흐르는 프로그램	조건에 의해 흐름이 나뉘는 프로그램
<ol style="list-style-type: none">1. a에 사용자로부터 입력 받은 수를 저장하라.2. b에 $3 \div a$의 결과를 저장하라.3. a와 b를 출력하라.	<ol style="list-style-type: none">1. a에 사용자로부터 입력 받은 수를 저장하라.2. a가 0이라면 프로그램을 종료하고, 그렇지 않으면 다음 명령을 수행하라.3. b에 $3 \div a$의 결과를 저장하라.4. a와 b를 출력하라.

2. if 조건문

한 방향으로만 흐르는 코드



조건에 의해 흐름이 나뉘는 코드



2. if 조건문

❖ 조건문

- 조건에 따라 코드 실행하거나 / 실행하지 않도록 만드는 구문
- 코드의 실행 흐름 변경 (조건 분기)

❖ 기본 형태

```
# 입력을 받습니다.  
number = input("정수 입력> ")  
number = int(number)  
  
# 양수 조건  
if number > 0:  
    print("양수입니다")  
  
# 음수 조건  
if number < 0:  
    print("음수입니다")  
  
# 0 조건  
if number == 0:  
    print("0입니다")
```

```
if <조건>:  
    <조건이 참일 때 문장>
```



2. if 조건문

- ❖ 영어에서 if문 "만약 ~ 라면 " 의 뜻
 - 예) "만약 입력 받은 수 a가 0**이라면**"
- ❖ if문의 조건은 참 아니면 거짓으로 평가될 수 있어야 함.
- ❖ 조건 뒤에 있는 **콜론 :** 은 해당 조건이 참일 경우에 실행할 코드 블록을 위치시키기 위함임.
- ❖ **else** 절로 코드가 흐르는 경우는 if 절의 조건이 거짓으로 평가되었을 때

```
if 조건:
    명령1
    명령2
    ....
else:
    명령3
    명령4
    ...
```

if 뒤에 흐름을 가를 조건이 위치하고, 그 뒤에 콜론 : 이 옵니다.

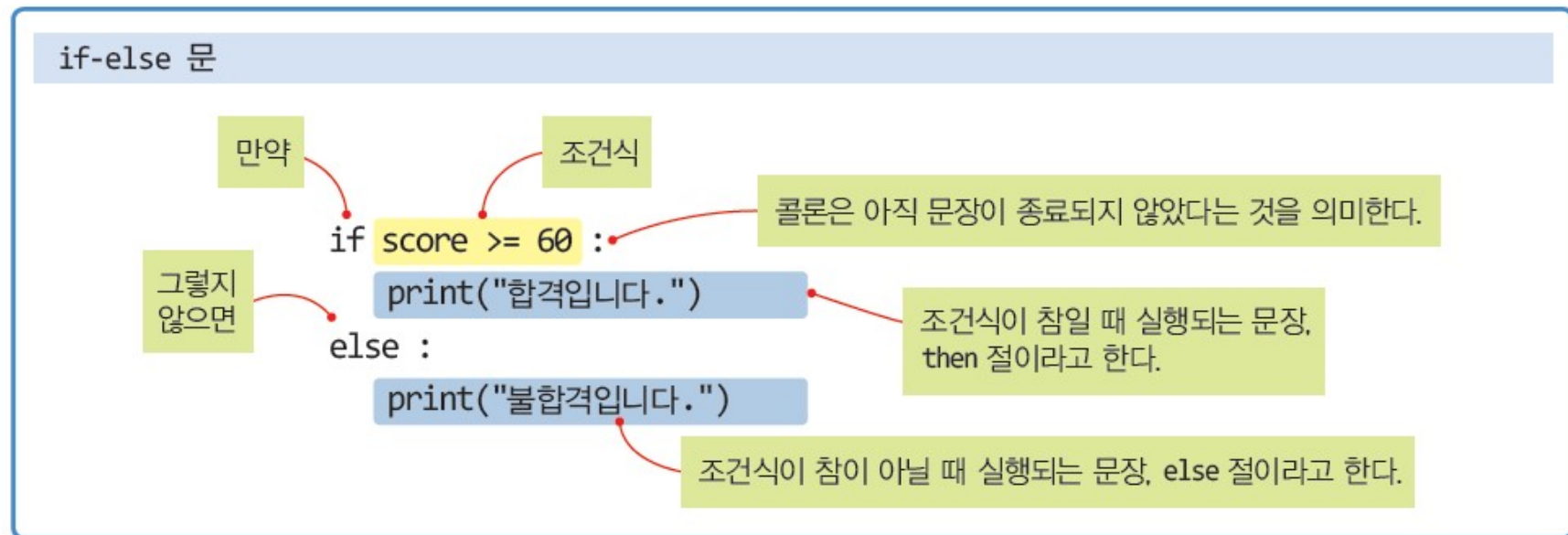
: 뒤에는 들여쓰기로 이루어진 코드블록이 옵니다.
이 코드블록은 if문의 조건이 True일 경우 실행됩니다.

if문의 조건을 충족하지 않을 때, 즉 조건 평가의 결과가 False일
때의 흐름은 else로 향합니다.
else 뒤에도 코드블록이 오므로 콜론 : 이 위치해야 합니다.



2. if 조건문

- ❖ 영어에서 if문 "만약 ~ 라면 " 의 뜻
 - 예) "만약 입력 받은 수 a가 0이라면"
- ❖ if문의 조건은 참 아니면 거짓으로 평가될 수 있어야 함.
- ❖ 조건 뒤에 있는 **콜론 :** 은 해당 조건이 참일 경우에 실행할 코드 블록을 위치시키기 위함임.
- ❖ **else** 절로 코드가 흐르는 경우는 if 절의 조건이 거짓으로 평가되었을 때

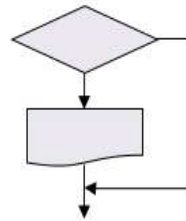


2. if 조건문

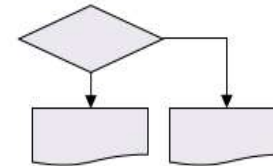
❖ 만약 ~ 라면 ~ 한다.

- 제시한 명제가 참이나 거짓인지 판단하고, 판단에 따른 명령을 내리는 것

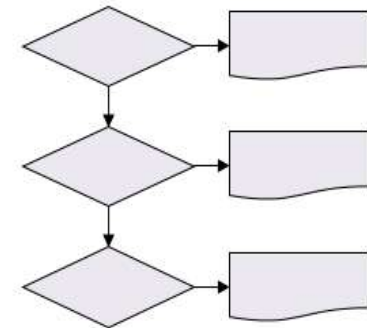
만약 일요일에 날씨가 좋으면 소풍을 가자.
조건 실행



만약 일요일에 날씨가 좋으면 소풍을 가자. 좋지 않으면 카페에 가자
조건 실행 조건2 실행



만약 일요일에 날씨가 좋으면 소풍을 가자. 비가 오면 카페에 가자. 눈이 오면 운동장에 가자.
조건 실행 조건2 실행 조건3 실행



2. if 조건문

- ❖ 여러 개의 조건을 다룰 때는?
 - if와 함께 elif 절 사용

```
if 조건1 :  
    코드블록
```

첫 번째 조건은 항상 if로 시작합니다.

```
...  
elif 조건2 :  
    코드블록
```

두 번째 조건부터는 elif를 이용합니다.

```
...  
elif 조건3 :  
    코드블록
```

```
...  
elif 조건4 :  
    코드블록
```

```
...  
else :  
    코드블록
```

마지막의 else는 생략할 수 있습니다.

```
...
```

2. if 조건문

- ❖ if 조건문 다음 줄에서 들여쓰기를 하지 않으면 에러가 발생한다.
- ❖ if 조건문을 사용할 때 주의할 점은 비교할 때는 ==로 사용해야 한다.

```
>>> x = 10
>>> if x == 10:
...     print('10입니다.')
...
10입니다.
```

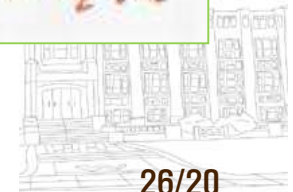
• 들여쓰기 4칸

• 빈 줄에서 엔터 키를 누름

```
>>> x = 10
>>> if x == 10:
... print('10입니다.')
File "<stdin>", line 2
    print('10입니다.')
    ^
IndentationError: expected an indented block
```

• 들여쓰기를 하지 않으면

• 들여쓰기를 하지 않으면 에러가 발생함



2. if 조건문

- ❖ if 다음 줄에서 들여쓰기를 하지 않으면 에러가 발생한다.
- ❖ if 조건문을 사용할 때 주의할 점은 비교할 때는 ==로 사용해야 한다.

```
>>> if x = 10:  .....if에 =를 사용하면
```

```
File "<stdin>", line 1
```

```
    if x = 10:
```

```
        ^
```

```
SyntaxError: invalid syntax  .....문법 에러가 발생함
```

```
>>> if x == 10:  .....조건식 뒤에 :를 빼먹음
```

```
File "<stdin>", line 1
```

```
    if x == 10
```

```
        ^
```

```
SyntaxError: invalid syntax  .....문법 에러가 발생함
```



2. if 조건문

- ❖ if 다음 줄에서 들여쓰기를 하지 않으면 에러가 발생한다.
- ❖ if 조건문을 사용할 때 주의할 점은 비교할 때는 ==로 사용해야 한다.

```
x = 10
if x == 10:
    print('x에 들어있는 숫자는')
    print('10입니다.') # unexpected indent 에러 발생
```

• 들여쓰기 4칸

• 들여쓰기 8칸



2. if 조건문

- ❖ if 다음 줄에서 들여쓰기를 하지 않으면 에러가 발생한다.
- ❖ if 조건문을 사용할 때 주의할 점은 비교할 때는 ==로 사용해야 한다.

```
x = 10
if x == 10:
    print('x에 들어있는 숫자는')
    print('10입니다.')
```

• 들여쓰기 4칸

• 들여쓰기 없음

```
x = 5          # x에 5를 할당
if x == 10:    # x가 5라서 조건식을 만족하지 않음
    print('x에 들어있는 숫자는')
    print('10입니다.')
```

• 들여쓰기 4칸

• 들여쓰기가 없어서 if와는 상관없는 코드가 됨



2. 중첩 if 조건문

❖ if 를 여러번 사용하는 중첩 if 조건문

```
x = 15

if x >= 10:
    print('10 이상입니다.')

    if x == 15:
        print('15입니다.')

        if x == 20:
            print('20입니다.')
```

• 들여쓰기 4칸

• 들여쓰기 4칸

• 들여쓰기 8칸

• 들여쓰기 4칸

• 들여쓰기 8칸



2. if 조건문

❖ if 예제

- if 조건문에서 else 절은 필수 사항이 아니다.

```
print('수를 입력하세요 : ')
a = int(input())

if a == 0 :
    print('0은 나눗셈에 이용할 수 없습니다.')
else :
    print('3 /', a, '=', 3/a)
```



```
print('수를 입력하세요 : ')
a = int(input())

if a == 0 :
    print('0은 나눗셈에 이용할 수 없습니다.')
    sys.exit(0)

print('3 /', a, '=', 3/a)
```

2. if 조건문

❖ 연습문제

- 아래의 결과가 출력되도록 프로그램을 작성하시오.

15세 이상이면

출력 : 이 영화를 보실 수 있습니다.

영화의 가격은 10,000원입니다.

15세 미만이면

출력 : 이 영화를 보실 수 없습니다.

다른 영화를 보시겠습니까?



2. if 조건문

❖ 연습문제

- 아래의 결과가 출력되도록 프로그램을 작성하시오.

(Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday)

사용자로부터 요일(월 ~ 일)에 해당하는 값을 입력 받아
영문 요일로 출력하는 프로그램을 작성하시오.

입력 : 월

출력 : Monday



2. if 조건문

❖ 연습문제

- 아래의 결과가 출력되도록 프로그램을 작성하시오

(January, February, March, April, May, June, July, August, September, October, November, December)

사용자로부터 월(1월 ~ 12월)에 해당하는 값을 입력 받아
영문 월로 출력하는 프로그램을 작성하시오.

입력 : 1월

출력 : January



2. if 조건문

❖ 연습문제

- 아래의 결과가 출력되도록 프로그램을 작성하시오

아이가 태어난 지 몇 개월입니까? 5
파상풍 예방접종 대상자입니다.
폐렴구균 예방접종 대상자입니다.

대상 감염병	출생-1개월	1개월	2개월	6개월	15개월
결핵					
B형간염					
파상풍					
폐렴구균					



2. if 조건문

❖ 날짜/시간 활용

- 현재 날짜/시간 출력

```
# 날짜/시간과 관련된 기능을 가져옵니다.
```

```
import datetime
```

```
# 현재 날짜/시간을 구합니다.
```

```
now = datetime.datetime.now()
```

```
# 출력합니다.
```

```
print(now.year, "년")
```

```
print(now.month, "월")
```

```
print(now.day, "일")
```

```
print(now.hour, "시")
```

```
print(now.minute, "분")
```

```
print(now.second, "초")
```



2. if 조건문

❖ 연습문제

- 아래의 결과가 출력되도록 프로그램을 작성하시오

현재 날짜와 시간을 format() 함수를 사용하여
출력하는 프로그램을 작성하시오.



2. if 조건문

❖ 연습문제

- 아래의 결과가 출력되도록 **조건문** 프로그램으로 작성하시오

현재 날짜와 시간을 기준으로 오전과 오후를
구분하여 출력하는 프로그램을 작성하시오.



2. if 조건문

❖ 연습문제

- 아래의 결과가 출력되도록 프로그램을 작성하시오

현재 날짜와 시간을 기준으로 계절(봄 ~ 겨울)
구분하여 출력하는 프로그램을 작성하시오.

봄 : 3월 ~ 5월

여름 : 6월 ~ 8월

가을 : 9월 ~ 11월

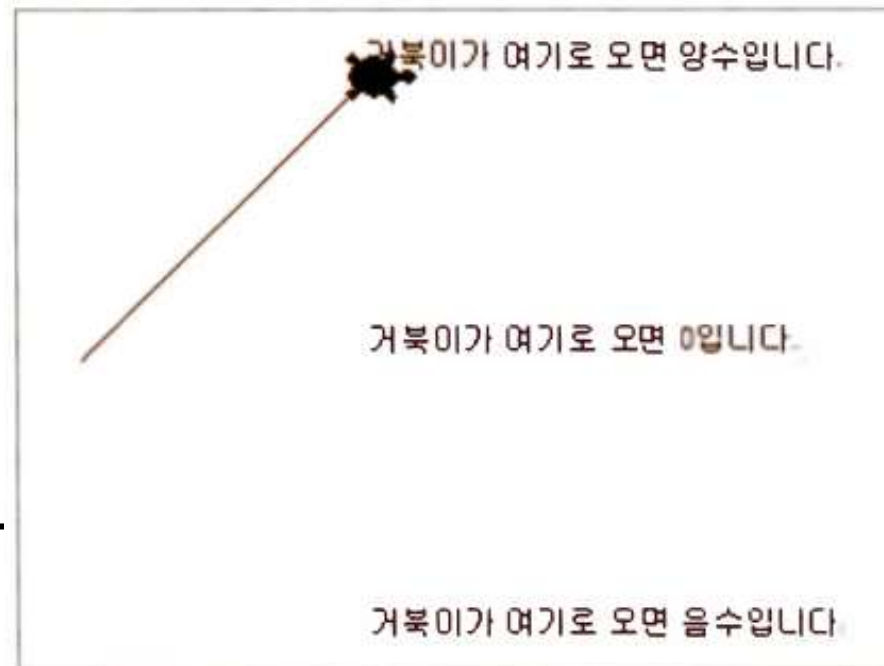
겨울 : 12월 ~ 2월



2. if 조건문

❖ 연습문제

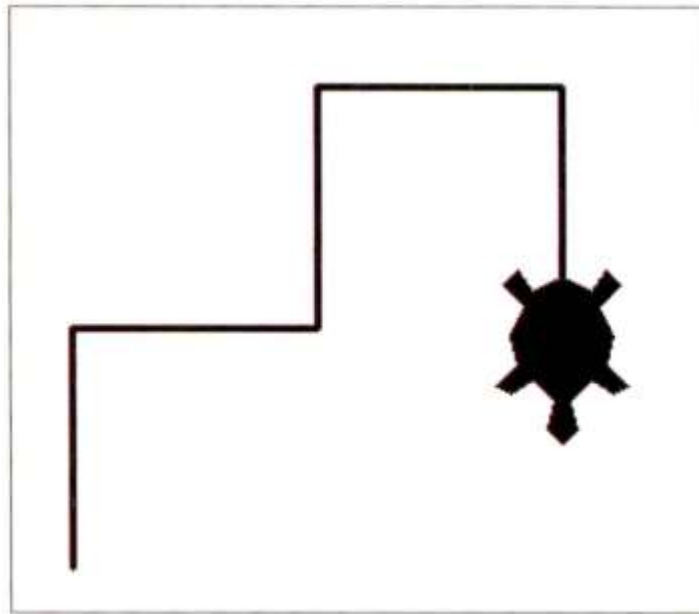
사용자로부터 정수를 받아서 정수의 부호에 따라
거북이를 (100, 100), (100, 0), (100, -100)으로
움직이는 프로그램을 작성하시오.



2. if 조건문

❖ 연습문제

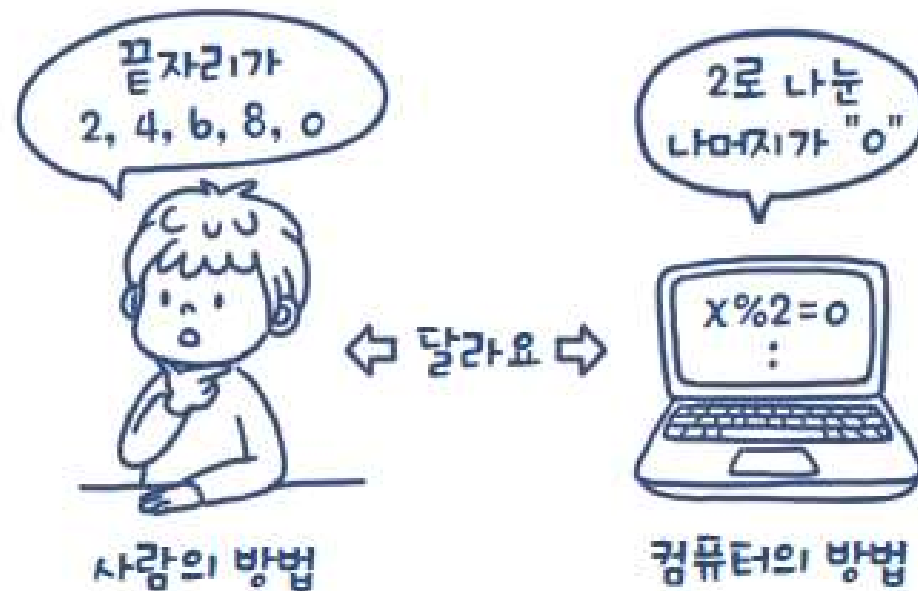
사용자가 'l'을 하면 거북이가 왼쪽으로 100픽셀 이동하고 'r'을 입력하면 오른쪽으로 100픽셀 이동하는 프로그램을 작성하시오.



3. if 조건문을 효율적으로 사용하기

❖ 컴퓨터의 조건

- 프로그래밍에서 우리의 기존 생각과 다른 것이 더 효율적일 수 있다
- 다양한 코드를 많이 공부하자
 - 알고리즘 공부 등



3. if 조건문을 효율적으로 사용하기

❖ 연습문제

- 아래의 결과가 출력되도록 프로그램을 작성하시오(예: 36,535)

사용자로부터 임의의 정수를 입력 받아
홀수와 짝수를 구분하여 출력하는 프로그램을
작성하시오. (1 ~ 65,535)



3. if 조건문을 효율적으로 사용하기

❖ 연습문제

- 아래의 결과가 출력되도록 프로그램을 작성하시오

사용자로부터 임의의 정수를 입력 받아
홀수와 짝수를 구분하여 출력하는 프로그램을
작성하시오. (1 ~ 65,535) - 두번째 방법



3. if 조건문을 효율적으로 사용하기

❖ 연습문제

사용자로부터 연도를 입력 받아 윤년을 판단하는 프로그램을 작성하시오.

<조건>

1. 연도가 4로 나누어지지만 100으로는 나누어 지지 않는 연도
2. 연도가 400으로 나누어지는 연도



3. if 조건문을 효율적으로 사용하기

❖ 연습문제 : random() 사용

동전 던지기 게임을 시작합니다.

뒷면입니다.

게임이 종료되었습니다.



또는



3. if 조건문을 효율적으로 사용하기

❖ 연습문제 : turtle + image 사용

동전 던지기 그래픽 게임을 시작합니다.
뒷면 이미지 출력



또는



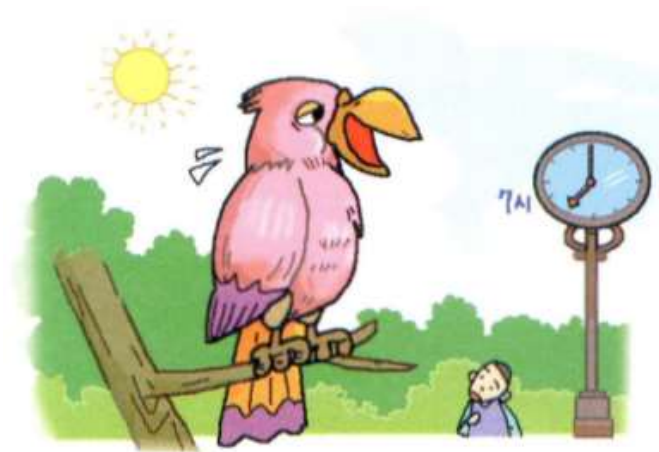
3. if 조건문을 효율적으로 사용하기

❖ 연습문제 : random() 사용

동물원에 있는 종달새는 다음의 조건이 만족되면 노래를 한다.

<조건>

1. 오전 6시부터 오전 9시 까지
2. 날씨가 화창하다.



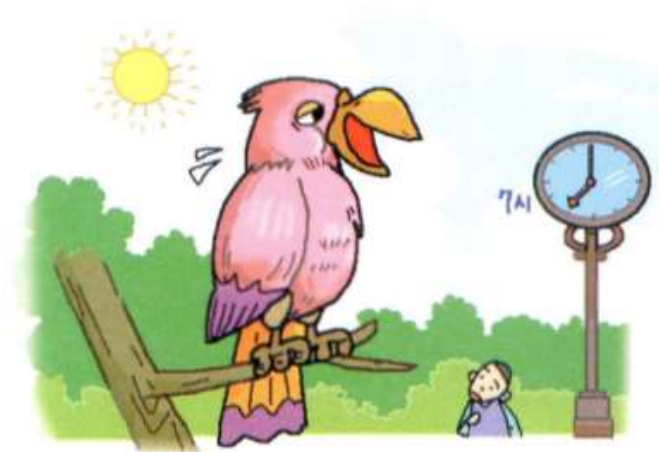
3. if 조건문을 효율적으로 사용하기

❖ 연습문제 : random() 사용

동물원에 있는 종달새는 다음의 조건이
만족되면 노래를 한다.

<조건>

1. 오전 6시부터 오전 9시 까지
2. 오후 2시부터 4시 까지
3. 날씨가 화창하다



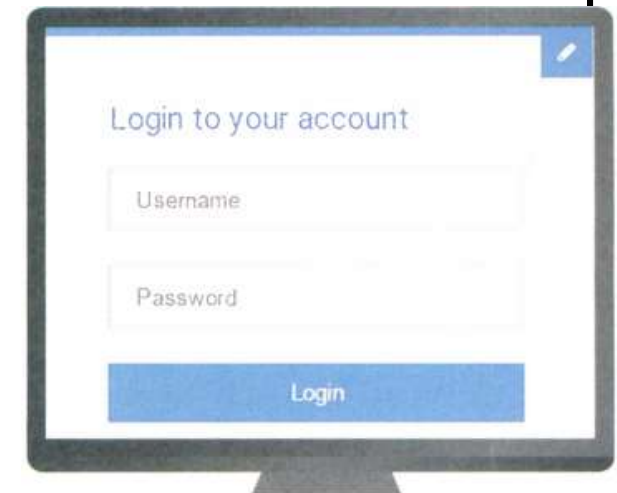
3. if 조건문을 효율적으로 사용하기

❖ 연습문제 : random() 사용

사용자로부터 아이디를 입력 받아 프로그램에 저장된
아이디와 일치하는지 여부를 출력하는 프로그램을
작성하시오.

아이디를 입력하시오 : administrator
환영합니다.

아이디를 입력하시오 : root
아이디를 찾을 수 없습니다.



3. if 조건문을 효율적으로 사용하기

- ❖ 학점을 입력 받아 학생평가를 하는 조건문을 구현하시오.
(float() 함수 사용)

조건	설명(학생 평가)	조건	설명(학생 평가)
4.5	신	1.75~2.3	오락문화의 선구자
4.2~4.5	교수님의 사랑	1.0~1.75	불가촉천민
3.5~4.2	현 체제의 수호자	0.5~1.0	자벌레
2.8~3.5	일반인	0~0.5	플랑크톤
2.3~2.8	일탈을 꿈꾸는 소시민	0	시대를 앞서가는 혁명의 씨앗



3. if 조건문을 효율적으로 사용하기

❖ 연습문제 : random() 사용

사용자는 다음의 3가지 영역중에서 하나를 선택하여 페널티 킥을 한다. 컴퓨터는 난수를 생성하여 3개의 영역중에서 하나를 수비한다. (왼쪽, 중앙, 오른쪽)

어디를 수비하겠습니까?(왼쪽, 중앙, 오른쪽) **왼쪽**
페널티 킥이 성공하였습니다.

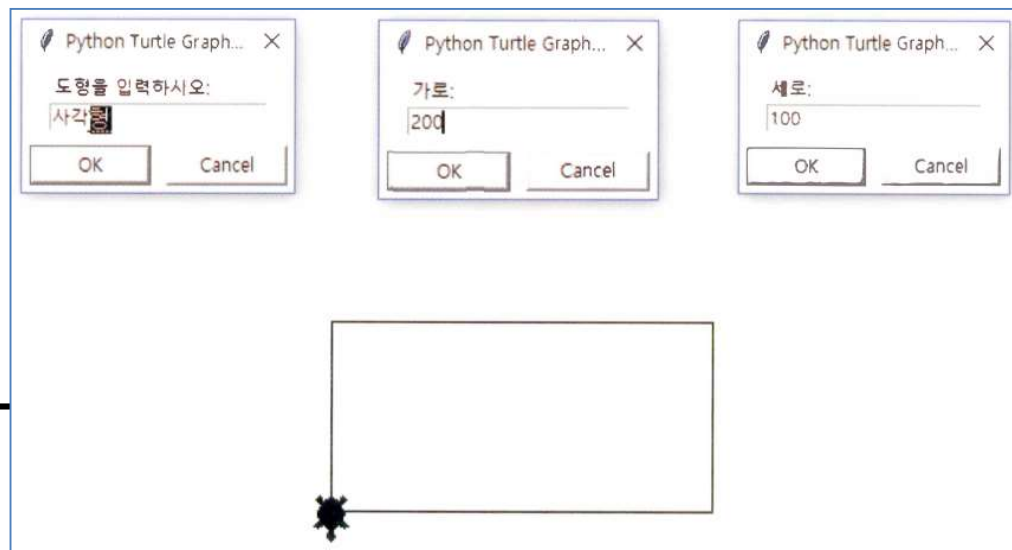


3. if 조건문을 효율적으로 사용하기

❖ 연습문제 : turtle() 사용

사용자는 다음의 3가지 영역중에서 하나를 선택하여 페널티 킥을 한다. 컴퓨터는 난수를 생성하여 3개의 영역중에서 하나를 수비한다.
(왼쪽, 중앙, 오른쪽)

어디를 수비하겠습니까?(왼쪽, 중앙, 오른쪽) **왼쪽**
페널티 킥이 성공하였습니다.



3. if 조건문을 효율적으로 사용하기

❖ 연습문제

- 아래의 결과가 출력되도록 프로그램을 작성하시오

이름을 입력하세요 : 파이썬
키(cm)를 입력하세요 : 176
몸무게(kg)를 입력하세요 : 73

파이썬님의 키는 176 cm이고 몸무게는 73 kg 입니다.
BMI 지수는 23.57 입니다. 과체중 입니다.

BMI기준	18.5~22.9	23~24.9	25~29.9	30 이상
비만분류	정상	과체중	비만	고도비만



3. if 조건문을 효율적으로 사용하기

❖ 연습문제

- 아래의 결과가 출력되도록 프로그램을 작성하시오

화면 출력	
수능 평균 등급 : 2	수능 평균 등급 : 2.1
수능 최저 기준을 만족합니다.	수능 최저 기준을 만족하지 않습니다.
합격입니다.	불합격입니다.



3. if 조건문을 효율적으로 사용하기

❖ 연습문제

- 아래의 결과가 출력되도록 프로그램을 작성하시오(모두 만족시 합격)

100m 기록(초) : 13.8

1000m 기록(초) : 239

윗몸일으키기 기록(회) : 52

좌우 악력 기록(kg) : 56

팔굽혀펴기 기록(회) : 56

모든 조건 만족시 -> 합격

조건에 만족하지 않으면 불합격.

	100m	1000m	윗몸일으키기	좌우악력	팔굽혀펴기
8점 커트라인	13.6	237	51	56	46

3. if 조건문을 효율적으로 사용하기

❖ 연습문제

- 아래의 결과가 출력되도록 프로그램을 작성하시오

공무원 시험은 보통 필수 3과목(국어, 영어, 한국사)와 선택 2과목으로 나뉜다. 이중 어느 한 과목이라도 40점 미만이면 과락이다. 한 해 응시자 중 평균 50% 정도 과락이 발생한다.

점수를 입력 받은 후 어느 한 과목 이하라도 40점 미만일 경우 “과락”을 출력하고, 반대의 경우 “과락 아님”을 출력하시오.



3. if 조건문을 효율적으로 사용하기

❖ 연습문제

- 아래의 결과가 출력되도록 프로그램을 작성하시오

사이다-700원 콜라-800원 물-1200원

돈을 넣어주세요: 2000
선택) 1-사이다 2-콜라 3-물 : 3
물이 나왔습니다. 덜컹
잔돈 800 원 반환합니다.

사이다-700원 콜라-800원 물-1200원

돈을 넣어주세요 : 700
선택) 1-사이다 2-콜라 3-물 : 2
음료수를 뽑을 수 없습니다.
잔돈 700 원 반환합니다.



3. if 조건문을 효율적으로 사용하기

❖ 연습문제

- 아래의 결과가 출력되도록 프로그램을 작성하시오

입력한 자연수가 한 자리의 수면 "한 자릿수",
두 자리의 수면 "두 자릿수", 세 자리의 수면
"세 자릿수", 그 이상은 "세 자릿수 이상"을
출력한다.



3. if 조건문을 효율적으로 사용하기

❖ 아래 제시한 조건에 알맞은 점수 구하기(float() 함수 사용)

[조건]

국제 리듬 체조대회에서 한 선수가 한 종목의 경기를 마치는 즉시 다섯명의 국제심판이 0 ~ 10 사이의 숫자로 채점한 점수를 각각 입력한다.
이 가운데 최고점과 최저점을 제외한 3 개의 점수만을 평균을 구하여 해당 선수의 최종 점수로 발표한다.

[Hint]

1. 다섯 심판의 점수를 입력받아 다섯개의 변수에 각각 저장한다.
2. 다섯 점수중에서 최고점과 최저점을 구한다.(max, min)
3. 다섯 점수의 총합에서 최고점과 최저점을 뺀 다음
3으로 나누어 평균 점수를 구한다.(sum, avg)
4. 적당한 메시지와 함께 계산된 점수를 출력한다.



3. if 조건문을 효율적으로 사용하기

❖ 아래 제시한 조건에 알맞은 프로그램을 작성하시오.

[조건]

사용자에게 임의의 네자리 숫자로 된 암호를 입력받아 네 자리 가운데 둘 이상의 겹치는 수가 있거나, 네 자리 수가 하나씩 증가하거나, 하나씩 감소하는 경우 적당한 메시지와 함께 입력된 암호를 거절하고 그렇지 않으면 받아들인다.

[Hint]

1. 네 자리 숫자의 암호를 입력받는다.
2. 첫번째부터 마지막 자리까지의 네 개의 숫자를 각각 따로 구분한다.
3. 네 개의 숫자 가운데 중복이 있는지 검사한다.
4. 네 개의 숫자가 하나씩 증가하는 순서로 배치되어 있는지 검사한다.
5. 네 개의 숫자가 하나씩 감소하는 순서로 배치되어 있는지 검사한다.
6. 암호에 오류가 발견되면 "사용할 수 없는 암호입니다" 라는 메시지를 출력한다.
7. 암호가 정상일 경우 "사용할 수 있는 암호입니다"라는 메시지 출력

3. if 조건문을 효율적으로 사용하기

❖ 아래 제시한 조건에 알맞은 프로그램을 작성하시오.

[조건]

주민등록번호를 숫자로만 입력하면 해당 주민등록번호에서 지역번호 부분을 이용하여 출생지를 출력하도록 하여라.

출생지역번호는 생년월일을 표시하는 여섯자리와 성별을 나타내는 자리 다음의 두 자리이다. 예를 들면 770101-1234567일 경우 23이 지역번호이다.

서울	00-08	부산	09-12	인천	13-15
경기	16-25	강원	26-34	충청	35-47
전라	48-66	경상	67-91	제주	92-95

4. False로 변환되는 값

- ❖ If 조건문의 매개변수에 불이 아닌 다른값이 올때는 자동으로 불로 변환해서 처리
- ❖ False로 변환되는 값 : None, 숫자 0과 0.0, 빈 문자열 등등...
- ❖ 다음과 같이 숫자 0 또는 빈 문자열을 if else 조건문의 매개변수에 넣으면 else 구문이 실행된다.

```
print("# if 조건문에 0 넣기")
if 0:
    print("0은 True로 변환됩니다")
else:
    print("0은 False로 변환됩니다")
print()

print("# if 조건문에 빈 문자열 넣기")
if 0:
    print("빈 문자열은 True로 변환됩니다")
else:
    print("빈 문자열은 False로 변환됩니다")
```



if 조건문에 0 넣기
0은 False로 변환됩니다

if 조건문에 빈 문자열 넣기
빈 문자열은 False로 변환됩니다



5. Pass 키워드

- ❖ 프로그램의 골격은 조건문, 반복문, 함수, 클래스 등의 기본 구문이다.

```
# 입력을 받습니다.  
number = input("정수 입력> ")  
number = int(number)  
  
# 조건문 사용  
if number > 0:  
    # 양수일 때: 아직 미구현 상태입니다.  
else:  
    # 음수일 때: 아직 미구현 상태입니다.
```

조건에 맞는 실행문이
작성되지 않았다.



```
> python test.py  
File "test.py", line 8  
    else :  
      ^  
IndentationError: expected an indented block
```



5. Pass 키워드

- ❖ 프로그램의 골격은 조건문, 반복문, 함수, 클래스 등의 기본 구문이다.
- ❖ Pass 키워드를 사용하면 '아무 의미 없는 코드'를 작성할 수 있다.
- ❖ Pass를 한 줄에 한 번 입력하면, 해당 줄은 의미없이 지나가게 된다.

```
# 입력을 받습니다.  
number = input("정수 입력> ")  
number = int(number)  
  
# 조건문 사용  
if number > 0:  
    # 양수일 때: 아직 미구현 상태입니다.  
    pass  
else:  
    # 음수일 때: 아직 미구현 상태입니다.  
    pass
```

Indentation Error
들여쓰기가 잘못되어 있다



6. raise 키워드

- ❖ Raise 키워드는 '이름을 붙여서 오류를 강제로 발생시키는 기능' 을 가지고 있다.
- ❖ 미구현 상태를 표현 : raise NotImplementedError

```
# 입력을 받습니다.  
number = input("정수 입력> ")  
number = int(number)  
# 조건문 사용  
if number > 0:  
    # 양수일 때: 아직 미구현 상태입니다.  
    raise NotImplementedError  
else:  
    # 음수일 때: 아직 미구현 상태입니다.  
    raise NotImplementedError
```

raise
에러 강제 발생

> python test.py

정수 입력> 10

Traceback (most recent call last):

File "test.py", line 7, in <module>

raise NotImplementedError
NotImplementedError





Thank You !

파이썬

