

스크립트 프로그래밍

04 선택문

2016 2학기 (02분반)

강승우

학습 목표

- 비교연산자를 사용하여 부울식을 작성할 수 있다(4.2절).
- `random.randint(a, b)`나 `random.random()` 함수를 사용하여 랜덤 숫자를 생성할 수 있다(4.3절).
- 부울식을 가진 프로그램(AdditionQuiz)을 작성할 수 있다(4.3절).
- 단일 if 명령문을 사용하여 선택 제어를 구현할 수 있다(4.4절).
- 단일 if 명령문을 가진 프로그램(GuessBirthday)을 작성할 수 있다(4.5절).
- 이중 if-else 명령문을 사용하여 선택 제어를 구현할 수 있다(4.6절).
- 중첩 if와 다중 if-elif-else 명령문을 사용하여 선택 제어를 구현할 수 있다(4.7절).
- if 명령문에서 흔히 발생하는 오류를 회피할 수 있다(4.8절).
- 선택문을 가진 프로그램을 작성할 수 있다(4.9– 4.10절).
- 논리연산자(and, or, not)를 사용하여 여러 조건들을 조합할 수 있다(4.11절).
- 여러 조건들이 조합된 선택문을 사용할 수 있다(LeapYear, Lottery)(4.12– 4.13절).
- 조건식을 사용한 표현식을 작성할 수 있다(4.14절).
- 연산자 우선순위와 결합법칙을 이해할 수 있다(4.15절).

선택문

- 선택문

- 조건에 따라 명령문의 실행을 선택할 수 있도록 하기 위해 사용
- 파이썬에서도 다른 프로그래밍 언어와 마찬가지로 2개 혹은 그 이상의 항목 중 한 항목을 선택할 수 있는 선택문 제공

원의 넓이 계산

```
if radius < 0:
```

```
    print("wrong input")
```

```
else:
```

```
    area = radius * radius * math.pi
```

```
    print("Area is ", area)
```

- 부울식(Boolean expression)으로 표현된 조건 사용

부울 타입, 부울값과 부울식

- 부울식 (Boolean expression)
 - 부울값인 True 혹은 False로 평가되는 연산식
- 비교 연산과 부울값
 - 반지름이 0보다 큰지, 작은지 혹은 0과 같은지와 같이 두 값을 비교해야 할 경우가 있음
 - 두 값의 비교: 비교 연산자(관계 연산자)를 이용
 - `<`, `<=`, `>`, `>=`, `==`, `!=`
 - 비교 연산의 결과 값은 True 혹은 False의 부울값
- 부울 변수 (Boolean variable)
 - 부울값을 저장하는 변수
 - True 혹은 False 중 하나의 값을 가짐
 - True를 나타내기 위해 1, False를 나타내기 위해 0 사용
 - `print(int(True))`, `print(int(False))`
 - True, False는 예약어
 - bool 함수: 숫자값을 부울값으로 변환
 - `print(bool(0))`, `print(bool(5))`

사례 연구: 랜덤 숫자 생성하기

- 초등학생을 위한 덧셈 퀴즈 프로그램을 작성해 보자.
 - 이 프로그램은 2개의 10진 정수인 `number1`과 `number2`를 랜덤하게 생성하고, '1 + 7은 얼마입니까?'와 같은 질문을 초등학생에게 한다.
 - 입력한 답이 맞는지 혹은 틀린지에 대한 메시지를 화면에 출력한다

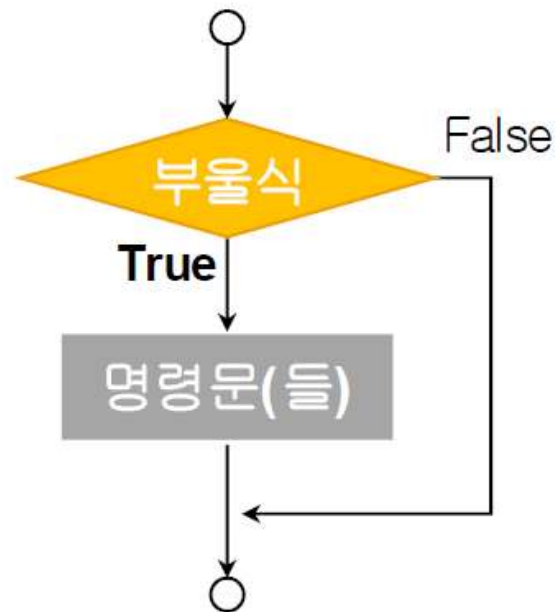
사례 연구: 랜덤 숫자 생성하기

- `randint(a, b)`
 - a와 b를 포함하여 두 숫자 사이의 랜덤 정수를 생성
 - 예
`randint(0, 9)`
 - `random` 모듈을 임포트해야 한다
`import random`
- `randrange(a, b)`
 - a와 b-1 사이의 랜덤 정수를 생성
 - `randint(a, b-1)`과 동일한 기능
- `random()`
 - $0 \leq r < 1.0$ 인 랜덤 실수 r을 생성

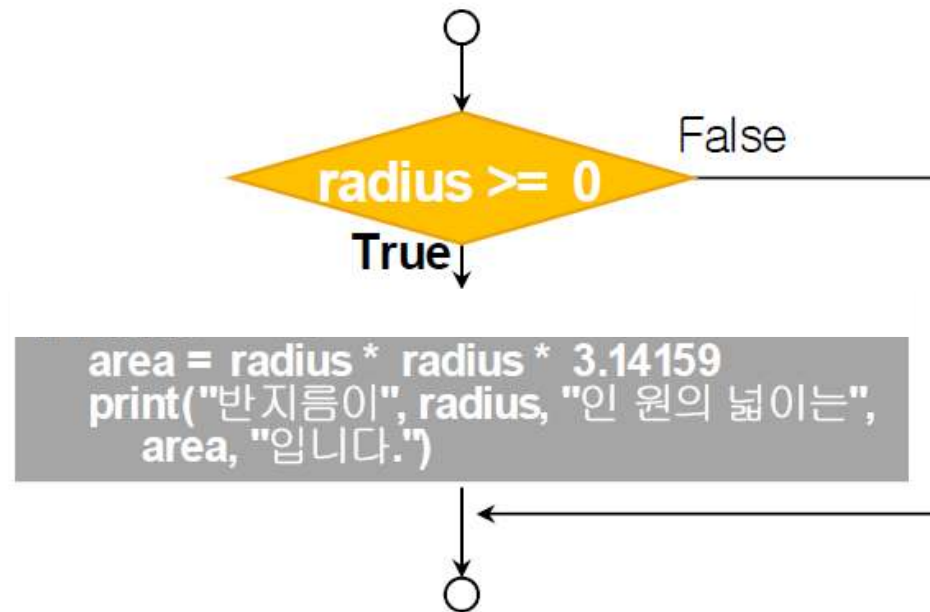
단일 if 문

구문형식

if 부울식:
명령문(들)



```
if radius >= 0:  
    area = radius * radius * 3.14159  
    print("반지름이", radius, "인 원의 넓이는",  
          area, "입니다.")
```



단일 if 문

- if 블록 내부의 명령문
 - if 행의 다음 행부터 들여쓰기 되어야 함
 - 동일한 개수의 공백으로 들여쓰기 되어야 함 (교재에서는 공백 4개 사용)
- if 행
 - 부울식 다음 콜론(:)

```
if i > 0:  
print("i는 양수")
```

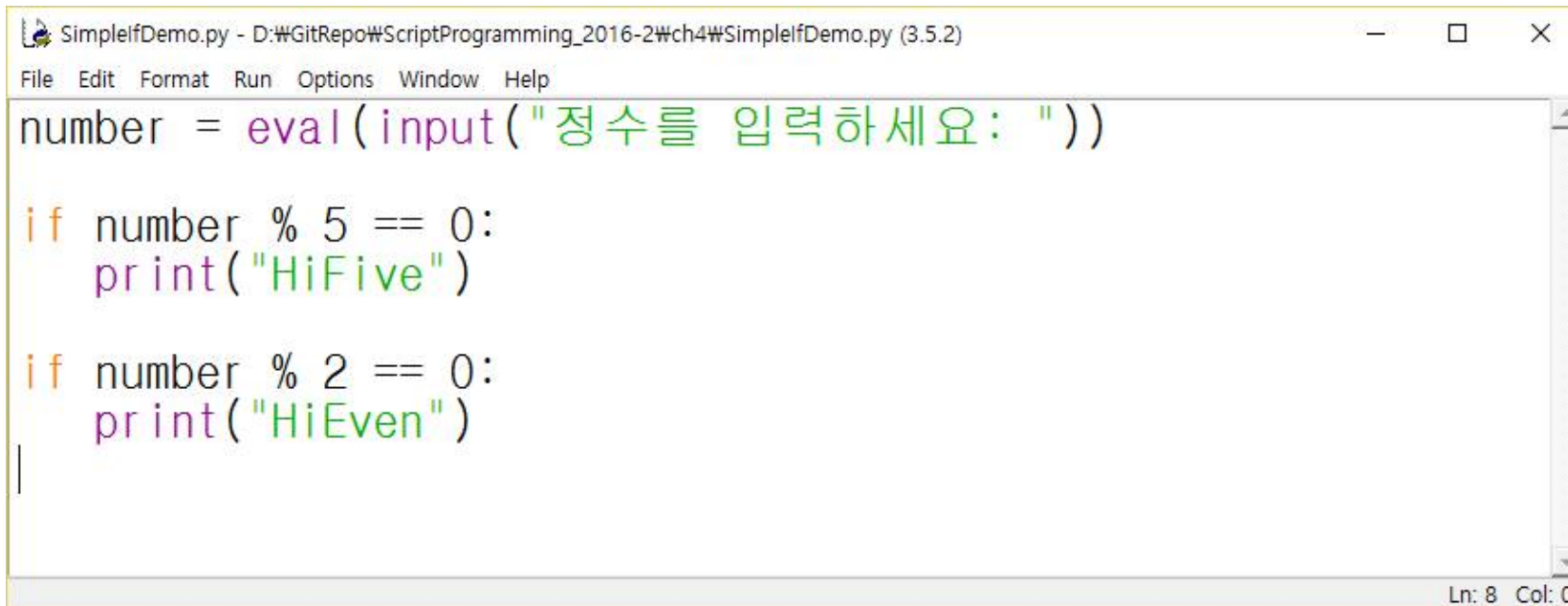
(a) 올바르지 않음

```
if i > 0:  
    print("i는 양수")
```

(b) 올바름

예제

- 사용자로부터 정수를 입력받고(라인 1), 입력된 정수가 5로 나누어지면 HiFive를 출력하며(라인 3-4), 2로 나누어지면 HiEven을 출력하는 프로그램



```
SimpleIfDemo.py - D:\GitRepo\ScriptProgramming_2016-2\ch4\SimpleIfDemo.py (3.5.2)
File Edit Format Run Options Window Help
number = eval(input("정수를 입력하세요: "))

if number % 5 == 0:
    print("HiFive")

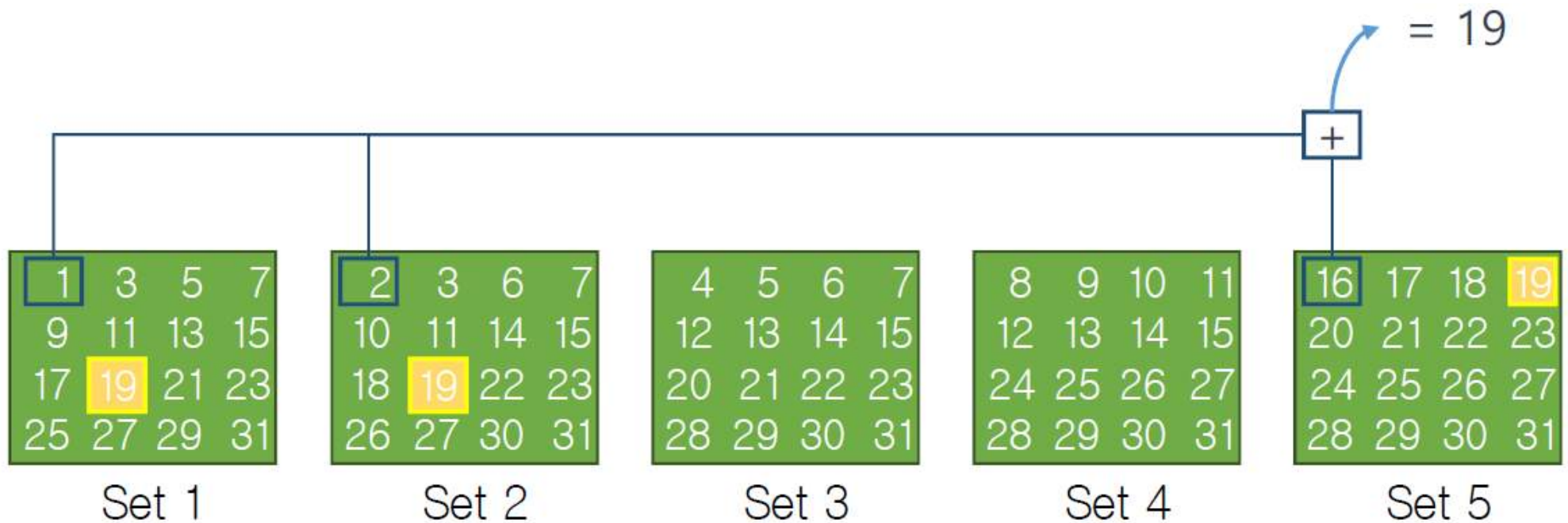
if number % 2 == 0:
    print("HiEven")
|
```

Ln: 8 Col: 0

사례 연구: 생일 맞추기

- 며칠에 태어났는지 다섯 번의 질문만으로 맞추기

사례 연구: 생일 맞추기



사례 연구: 생일 맞추기

- 19는 2진수로 10011

- 10000 \rightarrow 16
- 00010 \rightarrow 2
- 00001 \rightarrow 1

- 31은 2진수로 11111

- 10000 \rightarrow 16
- 01000 \rightarrow 8
- 00100 \rightarrow 4
- 00010 \rightarrow 2
- 00001 \rightarrow 1

- 생일이 될 수 있는 1과 31 사이의 10진수에 대한 2진수는 최대 다섯 자리로 표현됨

- $b_5b_4b_3b_2b_1$

- b_k 자리가 1이면 그 생일은 반드시 Set_k 집합에 있음

이중 if-else 문

구문형식

if 부울식:

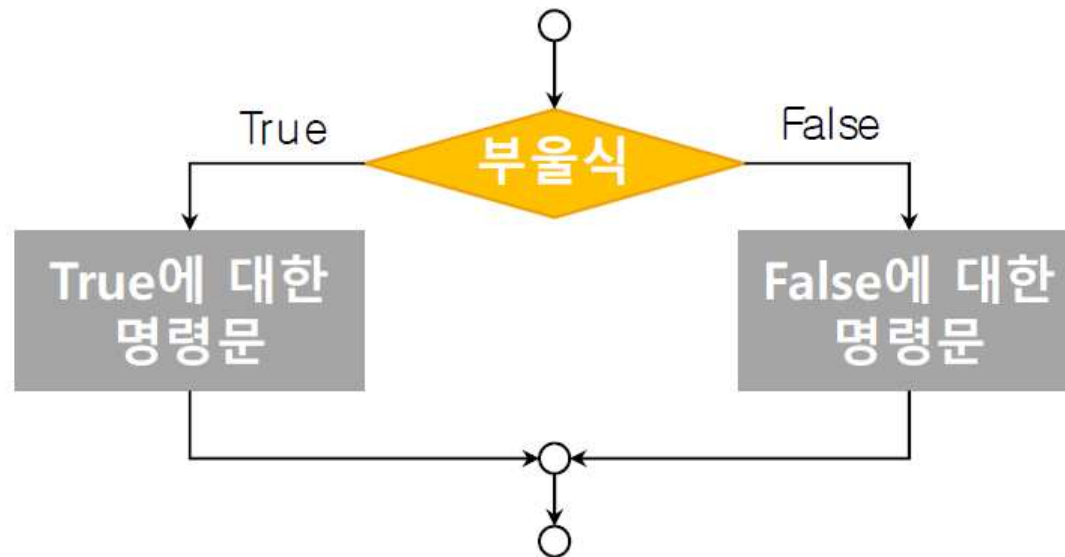
명령문(들)

#참인 경우

else:

명령문(들)

#거짓인 경우



if-else 문 예제

```
if radius >= 0:
    area = radius * radius * math.pi
    print("반지름이", radius, "인 원의 넓이는", area, "입니다.")
else:
    print("음수가 입력됨")
```

사례 연구: 뺄셈 퀴즈

- 초등학생을 위한 뺄셈 프로그램을 개발해 보자.
 - 이 프로그램은 $\text{number1} \geq \text{number2}$ 조건을 만족하는 2개의 10진수 정수인 number1 과 number2 를 랜덤하게 생성하고 학생들에게 “9 – 2는 얼마입니까?”와 같은 질문을 한다.
 - 학생이 답을 입력한 후에, 그 답의 정확성 여부를 나타내는 메시지를 출력한다.

중첩 if 문과 다중 if-elif-else 문

- 중첩 if 문
 - if 또는 if-else 문 내부에 어떤 파이썬 명령문도 올 수 있음. 여기에는 if, if-else 문도 포함
 - 외부 if 문 안에 내부 if 문이 중첩
- 다중 if 문
 - 들여쓰기가 너무 깊어지는 것을 방지
 - 프로그램을 읽기 쉽게
 - if-elif-else
 - elif: else if의 축약

중첩 if 문과 다중 if-elif-else 문

```
if score >= 90.0:
    grade = 'A'
else:
    if score >= 80.0:
        grade = 'B'
    else:
        if score >= 70.0:
            grade = 'C'
        else:
            if score >= 60.0:
                grade = 'D'
            else:
                grade = 'F'
```

(a)

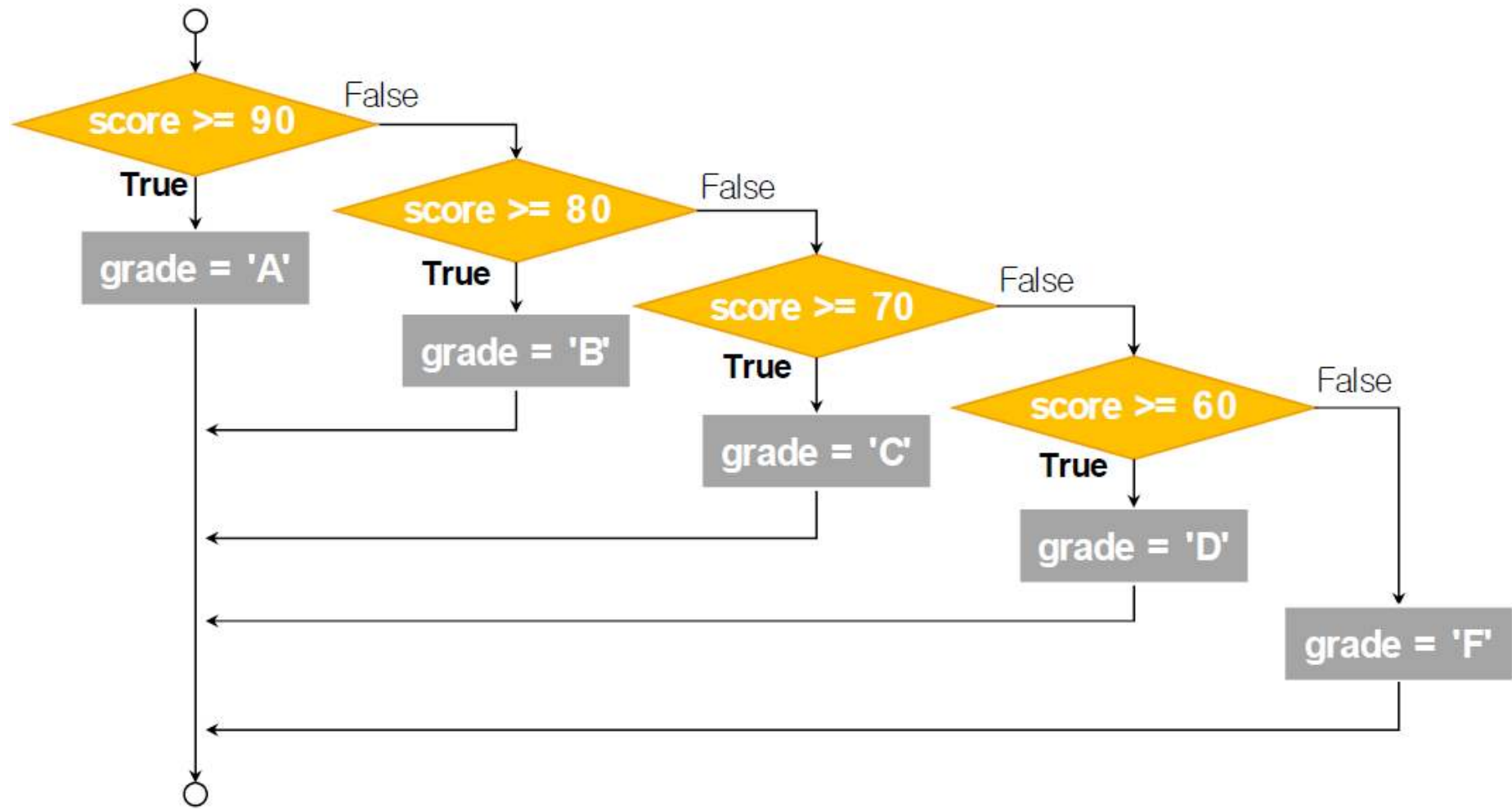
동일

```
if score >= 90.0:
    grade = 'A'
elif score >= 80.0:
    grade = 'B'
elif score >= 70.0:
    grade = 'C'
elif score >= 60.0:
    grade = 'D'
else:
    grade = 'F'
```

더 낫다

(b)

실행 흐름도



예제

- 연도에 해당하는 12지신을 찾는 프로그램
 - 12지신은 12년 사이클에 기반
 - 원숭이, 닭, 개, 돼지, 쥐, 소, 호랑이, 토끼, 용, 뱀, 말, 양 중 하나의 동물
- 연도를 12로 나눈 나머지를 기반으로 12지신을 구분

예제 코드

```
year = eval(input("연도를 입력하세요: "))
zodiacYear = year % 12
if zodiacYear == 0:
    print("원숭이")
elif zodiacYear == 1:
    print("닭")
elif zodiacYear == 2:
    print("개")
elif zodiacYear == 3:
    print("돼지")
elif zodiacYear == 4:
    print("쥐")
elif zodiacYear == 5:
    print("소")
elif zodiacYear == 6:
    print("범")
elif zodiacYear == 7:
    print("토끼")
elif zodiacYear == 8:
    print("용")
elif zodiacYear == 9:
    print("뱀")
elif zodiacYear == 10:
    print("말")
else:
    print("양")
```

선택문에서 흔히 발생하는 오류들

```
radius = -20
```

```
if radius >= 0:
```

```
    area = radius * radius * 3.14
```

```
print("The area is", area)
```

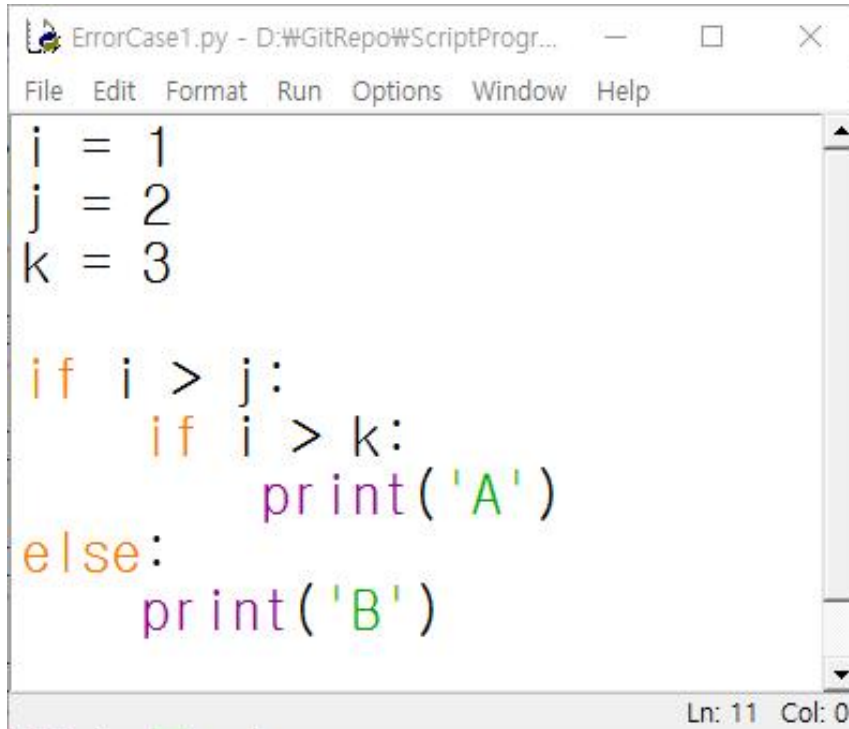
```
radius = -20
```

```
if radius >= 0:
```

```
    area = radius * radius * 3.14
```

```
print("The area is", area)
```

선택문에서 흔히 발생하는 오류들

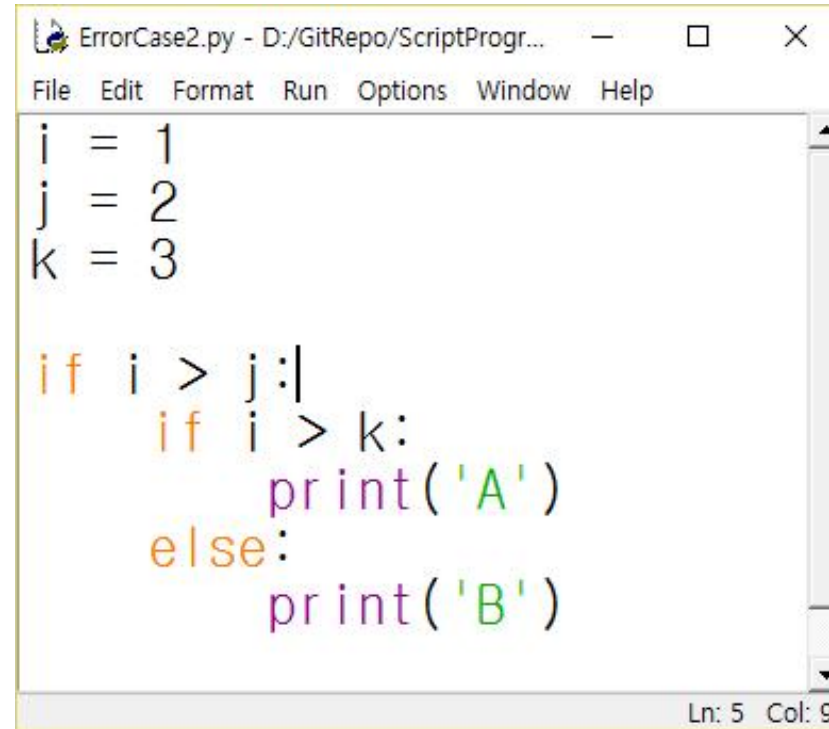


```
File Edit Format Run Options Window Help
i = 1
j = 2
k = 3

if i > j:
    if i > k:
        print('A')
else:
    print('B')
```

Ln: 11 Col: 0

The screenshot shows a Python IDE window titled 'ErrorCase1.py'. The code contains a nested if-else statement. The first if statement 'if i > j:' is followed by an indented 'if i > k:' block with a 'print('A')' statement. This is followed by an 'else:' block with a 'print('B')' statement. The status bar at the bottom indicates 'Ln: 11 Col: 0'.



```
File Edit Format Run Options Window Help
i = 1
j = 2
k = 3

if i > j:|
    if i > k:
        print('A')
else:
    print('B')
```

Ln: 5 Col: 9

The screenshot shows a Python IDE window titled 'ErrorCase2.py'. The code is identical to the first one, but the first if statement 'if i > j:' has a trailing pipe character '|' at the end of the line. The status bar at the bottom indicates 'Ln: 5 Col: 9'.

사례 연구: 체질량 지수 계산하기

- 체질량 지수(Body Mass Index, BMI)는 몸무게에 기반한 건강 측정 지수로서 몸무게(킬로그램)를 키(미터)의 제곱으로 나누어서 계산된다.
- 16세 이상의 성인을 기준으로 한 BMI는 다음과 같이 해석

BMI 수치	상태
18.5 이하	저체중
18.5~24.9	정상
25.0~29.9	과체중
30.0 이상	비만

논리 연산자

- 복합 조건을 만들기 위해 사용
 - 몇 개의 조건식을 조합하여 명령문의 수행 여부를 결정해야 할 경우
 - 논리 연산자를 조건식의 조합에 사용하여 복합 표현식을 구성
- not
 - 논리 부정
- and
 - 논리곱
- or
 - 논리합

- C/C++에서 논리 연산자
 - !a
 - a && b
 - a || b

not 연산자

p	not p	예제 (age=24, gender='F')
True	False	not (age > 18)은 False이다.
False	True	not (gender == 'M')은 True이다.

and 연산자

p1	p2	p1 and p2	예제(가정: age=24, gender='F')
False	False	False	(age > 18) and (gender == 'F')는 True
False	True	False	
True	False	False	(age > 18) and (gender != 'F')는 False
True	True	True	

or 연산자

p1	p2	p1 or p2	예제(가정: age=24, gender='F')
False	False	False	(age > 34) or (gender == 'F')는 True
False	True	True	
True	False	True	(age > 34) or (gender == 'M')는 False
True	True	True	

사례 연구: 윤년 판별하기

- 사용자로부터 연도를 입력 받은 후에 윤년 여부를 판별하는 프로그램
 - 어떤 연도가 4로는 나누어지지만 100으로 나누어지지 않거나 혹은 400으로 나누어 진다면 그 연도는 윤년이다.

사례 연구: 복권

- 복권 게임 프로그램

- 이 프로그램은 2자리 숫자를 랜덤하게 생성하여 복권 숫자를 만들고, 사용자로부터 2자리 숫자를 입력 받은 후에 다음 규칙에 따라 사용자의 승리 여부를 결정

1. 사용자의 입력 숫자가 정확한 순서로 복권 숫자와 모두 일치한다면, 1,000만원의 상금을 받는다.
2. 사용자의 입력 숫자가 순서는 맞지 않지만 복권 숫자와 모두 일치한다면, 300만 원의 상금을 받는다.
3. 사용자의 입력 숫자 중 1개 숫자가 복권 숫자와 일치한다면, 100만 원의 상금을 받는다.

사례 연구: 복권

- 당첨 번호 생성 : random 모듈 이용
- 사용자로부터 추첨 번호 입력 받음
- 복권 당첨 번호의 각 자릿수 값을 얻어냄
- 사용자 추첨 번호의 각 자릿수 값을 얻어냄
- 추첨 번호 검사
 - 당첨 번호와 동일
 - 순서는 일치하지 않지만 당첨 번호 숫자와 동일
 - 한 자릿수만 동일
 - 일치하는 숫자 없음

조건식

```
if x > 0:  
    y = 1  
else:  
    y = -1
```

동일

```
y = 1 if x > 0 else -1
```

구문형식

표현식1 if 부울식 else 표현식2

조건식

- number1과 number2 중 큰 값을 max에 할당하는 명령문

```
if number1 > number2:
```

```
    max = number1
```

```
else:
```

```
    max = number2
```


- 이를 조건식을 이용하여 표현하면?

조건식

- 아래의 if-else 문을 조건식을 이용하여 표현하면?

```
if num % 2 == 0:  
    print(str(num) + "는 짝수입니다.")  
else:  
    print(str(num) + "는 홀수입니다.")
```

연산자 우선순위



$+$, $-$ (단항 $+$ 와 $-$)

$**$ (지수)

not

$*$, $/$, $\\//$, $\%$ (곱셈, 나눗셈, 정수 나눗셈, 나머지)

$+$, $-$ (이항 덧셈과 뺄셈)

$<$, $<=$, $>$, $>$ (비교)

$==$, $!=$ (동등)

and

or

$=$, $+=$, $-=$, $*=$, $/=$, $\\//=$, $\%=$ (할당 연산자)

연산자 우선순위와 결합법칙

- 산술적으로 괄호 안에 있는 표현식이 먼저 평가된다
 - 괄호는 중첩될 수 있고, 이 경우에 괄호 내부에 있는 표현식이 먼저 수행된다
- 괄호가 없는 표현식을 평가할 때에는 연산자 우선순위와 결합법칙에 따라 연산자가 수행된다

연산자 결합

- 동일한 우선순위를 가진 연산자는 결합(associativity)에 의해서 연산자의 평가 순서가 결정된다.
- 모든 2진 연산자는 왼쪽-결합(left-associative)이기 때문에 다음 표현식의 결과는 동일

$$a - b + c - d \quad \xlongequal{\text{동일}} \quad ((a - b) + c) - d$$

- 할당 연산자는 오른쪽 결합(right-associative)이기 때문에 다음 표현식의 결과는 동일

$$a = b += c = 5 \quad \xlongequal{\text{동일}} \quad a = (b += (c = 5))$$