

Chapter 3

Logic

Contents

- Boolean Operators
- Relational Operators
- Arithmetic Operators
- Membership Operator
- Computational Problem

Boolean Operators

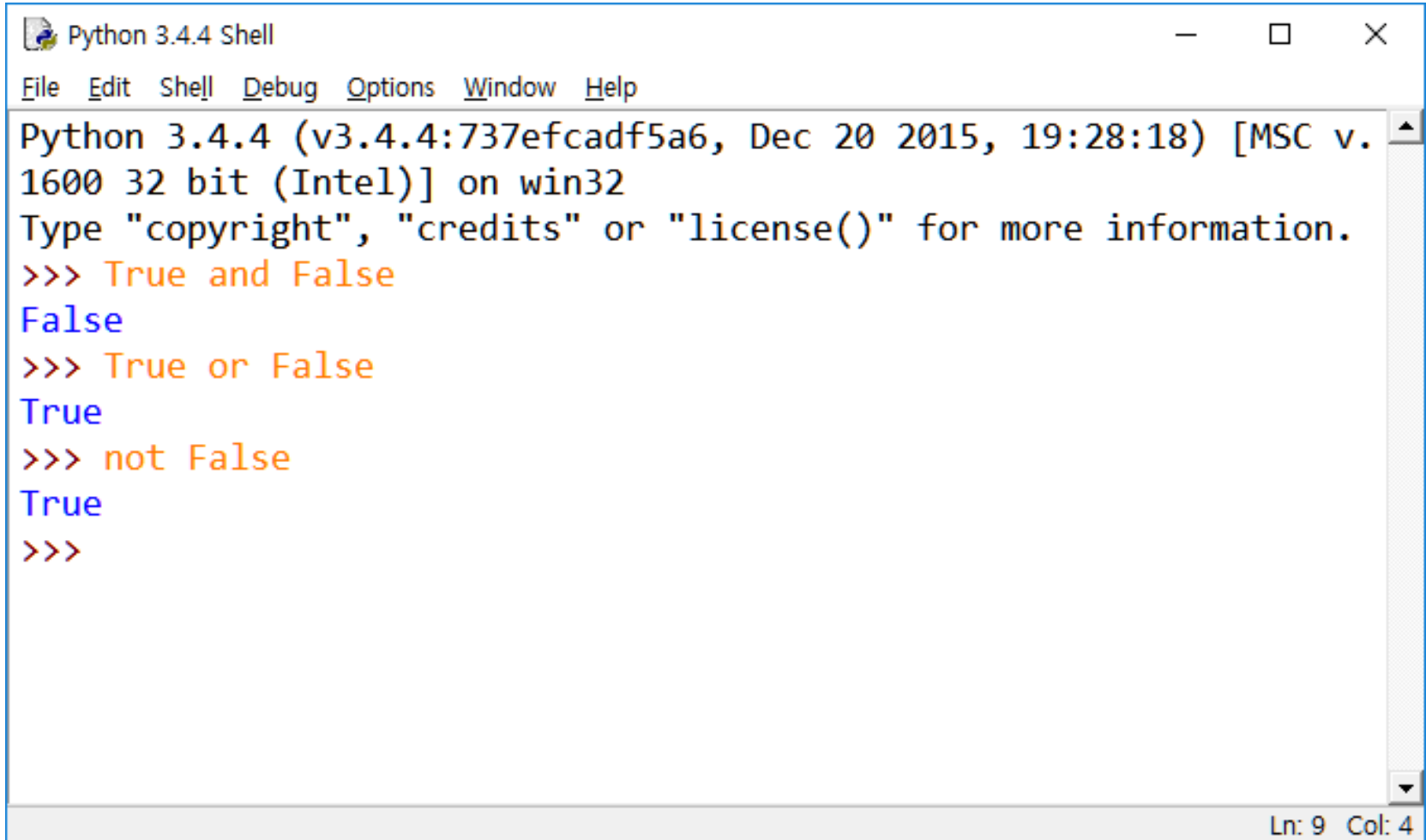
- Python에서는 3개의 Boolean 연산자를 사용할 수 있음

and, or, not

Expression	Meaning
x or y	if x is False, then y, else x
x and y	if x is False, then x, else y
not x	if x is False, then True, else False

Boolean Operators

True, False, and, or, not의 대소문자에 유의

A screenshot of a Python 3.4.4 Shell window. The window has a title bar with the text 'Python 3.4.4 Shell' and standard window controls (minimize, maximize, close). Below the title bar is a menu bar with 'File', 'Edit', 'Shell', 'Debug', 'Options', 'Window', and 'Help'. The main text area shows the following content:

```
Python 3.4.4 (v3.4.4:737efcadf5a6, Dec 20 2015, 19:28:18) [MSC v.
1600 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> True and False
False
>>> True or False
True
>>> not False
True
>>>
```

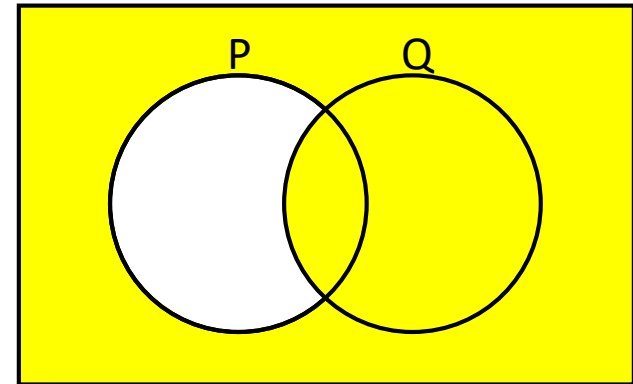
The status bar at the bottom right indicates 'Ln: 9 Col: 4'.

Boolean Operators

■ Implication

- Python 에는 Implies 연산자가 없음
- 조합 명제(compound proposition)로 표현 가능

Implication		
P	Q	P implies Q
False	False	True
False	True	True
True	False	False
True	True	True



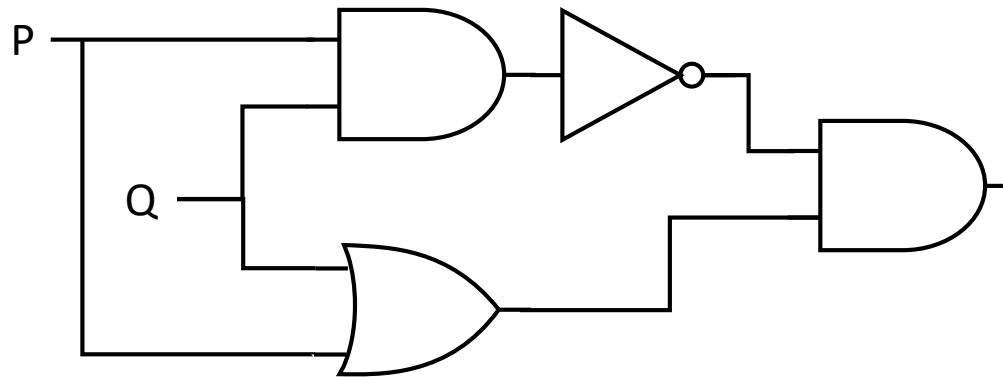
$$P \text{ implies } Q \Rightarrow (\text{not } P) \text{ or } Q$$

Boolean Operators

Exercise!

다음 Logic gate에 대한 Python 연산 표현은?

XOR(eXclusive OR) 연산



Relational Operators

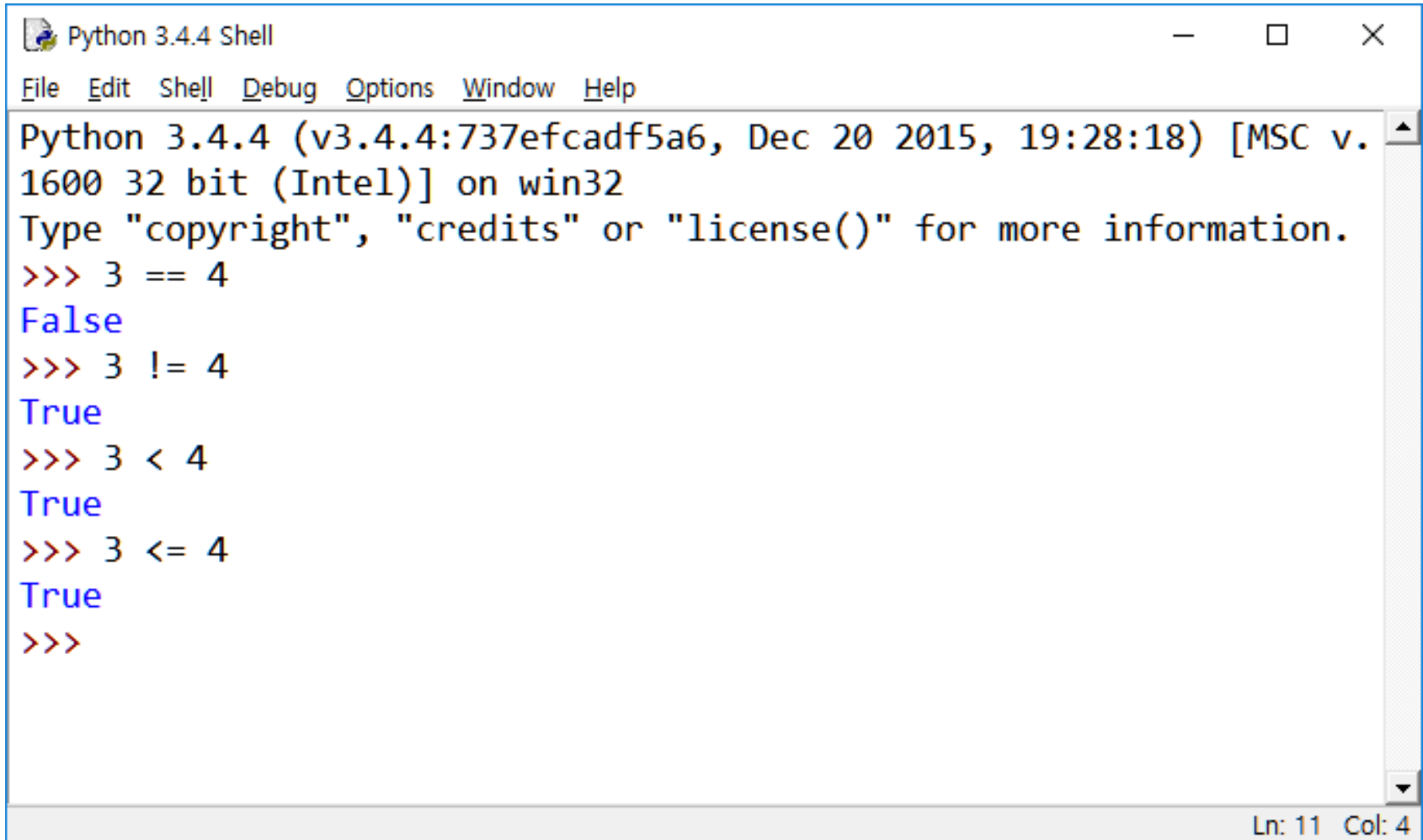
■ Relational Operators

- 크기 및 동일 여부를 비교하여 True/False를 반환

Mathematical Symbol	Python Operators
\equiv	<code>==</code>
\neq	<code>!=</code>
$<$	<code><</code>
\leq	<code><=</code>
$>$	<code>></code>
\geq	<code>>=</code>

Relational Operators

■ 숫자의 비교

A screenshot of a Python 3.4.4 Shell window. The window has a title bar with the text 'Python 3.4.4 Shell' and standard window controls (minimize, maximize, close). Below the title bar is a menu bar with 'File', 'Edit', 'Shell', 'Debug', 'Options', 'Window', and 'Help'. The main text area shows the Python interpreter's startup message: 'Python 3.4.4 (v3.4.4:737efcadf5a6, Dec 20 2015, 19:28:18) [MSC v. 1600 32 bit (Intel)] on win32'. It then prompts the user to type 'copyright', 'credits', or 'license()' for more information. Following this, several comparison operations are entered at the prompt: '3 == 4' returns 'False', '3 != 4' returns 'True', '3 < 4' returns 'True', and '3 <= 4' returns 'True'. The prompt '3 >>>' is shown at the bottom. The status bar at the bottom right indicates 'Ln: 11 Col: 4'.

```
Python 3.4.4 Shell
File Edit Shell Debug Options Window Help
Python 3.4.4 (v3.4.4:737efcadf5a6, Dec 20 2015, 19:28:18) [MSC v.
1600 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> 3 == 4
False
>>> 3 != 4
True
>>> 3 < 4
True
>>> 3 <= 4
True
>>>
```

Ln: 11 Col: 4

Relational Operators

Exercise!

푸른색 부분을 표현하는 Boolean Expression을 쓰세요

...	-4	-3	-2	-1	0	1	2	3	4	...
-----	----	----	----	----	---	---	---	---	---	-----

Arithmetic Operators

Operator	Python Operators	Example
+	Addition	>>> 4 + 1.2 5.2
-	Subtraction	>>> 3 - 4 -1
*	Multiplication	>>> 3 * 4 12
/	Division	>>> 29 / 10 2.9
//	Floor division	>>> 29 // 10 2
**	Power	>>> 2 ** 10 1024
%	Modulo	>>> 10 % 3 1

Arithmetic Operators

Exercise!

```
>>> 10 + 6 / 2 * 3
```

```
>>> 'Hello' + 'World'
```

```
>>> 12 // 4.0
```

```
>>> '3.14' + 2
```

```
>>> 3 ** 2 ** 2
```

```
>>> 10 + int(5.5)
```

```
>>> 81 % 2 == 0
```

```
>>> 10 + int('5.5')
```

Membership Operators

■ Membership Operators

- 여러 개의 값으로 이루어진 리스트에서 특정 값이 존재 하는지 판별
- 문자열에서 특정 문자 혹은 문자열이 존재하는 지 판별

Operator	Example	Output
in	2 in (1, 2, 3)	True
	'Mon' in ('Sun', 'Mon', 'Tue', 'Wed')	True
	'a' in ('apple', 'banana')	False
	'a' in 'apple'	True
not in	-1 not in (0, 1, 2, 3)	True

Membership Operators

Exercise!

사용자로부터 학점 기호를 입력 받아 올바른 학점 기호인지 True/False로 판단하는 구문

```
grade = input('Enter a letter grade:')  
grade in ('A', 'B', 'C', 'D', 'F')
```

Computational Problem

The Problem

Leap Year Checker

사용자로부터 4자리의 연도를 입력 받아 윤년 여부를
True/False로 출력하는 프로그램을 작성하세요

Leap Year Checker

Problem Analysis

윤년의 정의

- A. 서력 기원 연수가 4로 나누어 떨어지는 해는 윤년으로 한다
(2004년, 2008년, 2012년, 2016년, 2020년, 2024년 ...)
- B. 이 중에서 100으로 나누어 떨어지는 해는 평년으로 한다
(1900년, 2100년, 2200년, 2300년, 2500년 ...)
- C. 그 중에 400으로 나누어 떨어지는 해는 윤년으로 둔다
(1600년, 2000년, 2400년 ...)

위 규칙을 True/False로 판단 할 수 있는
Boolean Expression이 요구 됨

Leap Year Checker

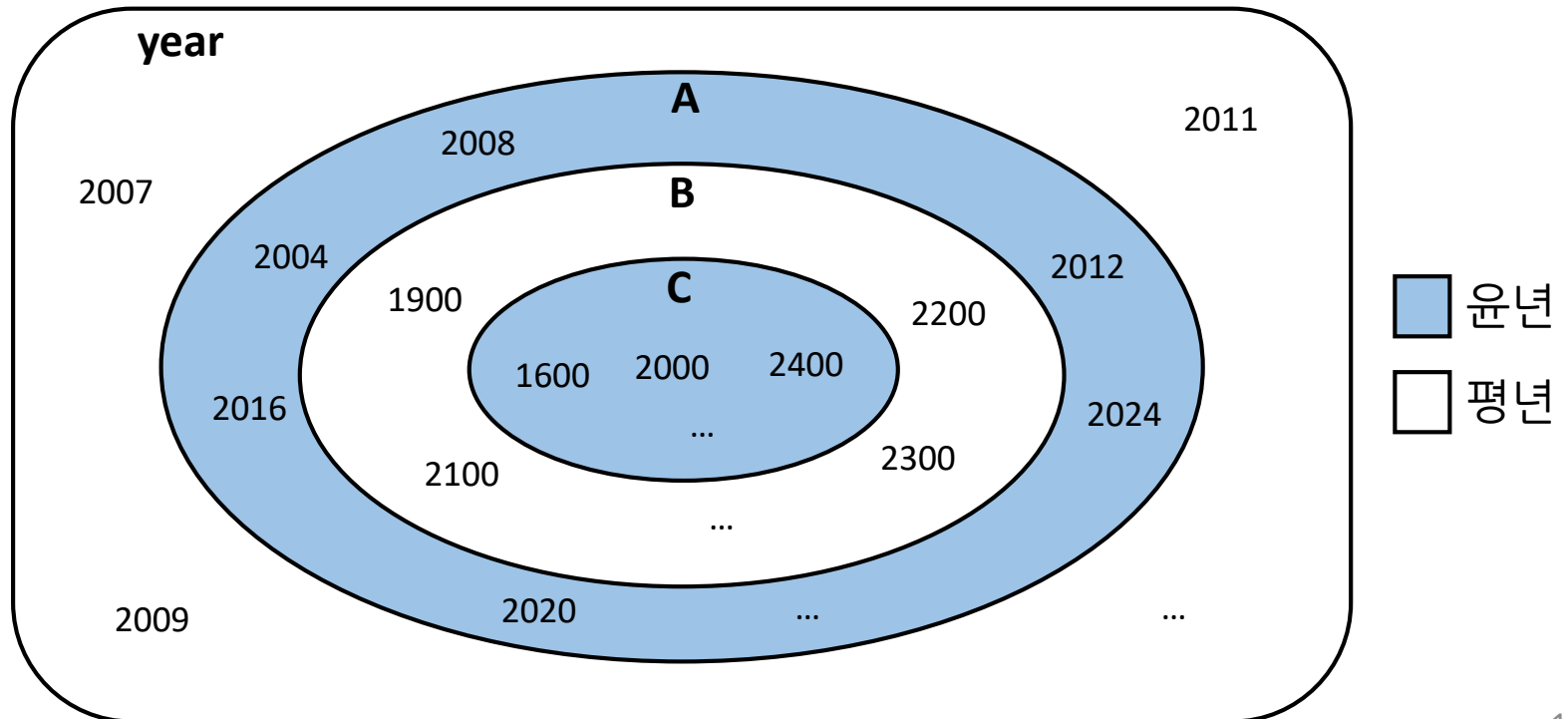
Problem Analysis

윤년의 집합 다이어그램 표현

A. $\text{year} \% 4 == 0$

B. $\text{year} \% 100 == 0$

C. $\text{year} \% 400 == 0$



Leap Year Checker

Data Representation

연도를 저장하기 위한 변수

`year`

`year`의 윤년 여부를 나타내는 변수

`is_leap_year`

Leap Year Checker

Algorithmic Thinking

윤년을 판별하는 Boolean Expression 구현

다이아그램으로부터:

A and not B or C



Python Boolean Expression:

```
(year % 4 == 0)
and
(not (year % 100 == 0))
or
(year % 400 == 0)
```

Leap Year Checker

Program Design

Program Requirements

사용자로부터 연도를 4자리 정수로 입력 받음

결과문을 출력

Leap Year Checker

Program Design

Program Introduction

소개문을 출력

User Input

4자리 정수로 year를 입력 받음

Calculate results

Boolean Expression으로 윤년 여부를 판단하여 is_leap_year에 저장

Display results

결과를 출력

Computational Problem

Program Implementation

```
1 print('Welcome to Leap Year Checker')
2 print('Enter a 4-digits year to check if it is a leap year')
3 year = int(input('Year(yyyy): '))
4 is_leap_year = (year % 4 == 0) and \
5                 (not (year % 100 == 0)) \
6                 or (year % 400 == 0)
7 print(year, 'is a leap year:', is_leap_year)
```

참고: Python 문장이 길 경우
\
기호로 줄 바꿈을 할 수 있음

Computational Problem

Program Execution

```
Welcome to Leap Year Checker
Enter a 4-digits year to check if it is a leap year
Year(yyyy): 2000
2000 is a leap year: True
>>>
```

```
Welcome to Leap Year Checker
Enter a 4-digits year to check if it is a leap year
Year(yyyy): 2017
2017 is a leap year: False
>>>
```