

3차

python에서 딥러닝까지

Python 기초반

주제 | 문자열 입출력 Mission ~ 연산자(2)

Contents

1. 문자열 입출력함수 Mission
2. 자료형과 변수 그리고 주석
 - 1) 변수와 자료형
 - 2) 주석
3. 연산
 - 1) 연산1: 산술, 문자열, 복합할당연산자
 - 2) 연산2: 비교, 논리, 멤버십 연산자

Next

문자열 입출력 함수(교재 66p)

문자열 입력함수: input()

- 문자열 입력함수: input()
 - ()안의 내용을 출력하고 변수에 입력된 내용을 저장하는 함수
 - 문법: <변수> = input("입력을 받기 전 출력할 안내 메시지")

문자열 출력함수: print()

- 문자열 출력함수: print()
 - () 안의 내용을 출력하는 함수
 - 문법: print("출력하고자 하는 문자열")
- 문자열 포매팅
 - way1) print("%s" %<변수, 여러 개 일 때는 ()로 묶기>)
 - way2) print(f"{변수명1} ~~ {변수명2}")
- 여러 개의 문자열을 한 번에 출력하기
 - print("문자열1", "문자열2", "문자열3")

문자열 출력함수: print()

[print() 명령어 Tip]

- `print('문자열1, 문자열2, ... , sep='사이에 넣고 싶은 문자 or 문자열 ')`
 - `sep`의 약자, default값으로 ' '(space bar)가 들어간다.
 - 각 문자열들의 사이의 구분문자를 설정해준다.
- `print('문자열' , end='문장의 끝에 넣고 싶은 문자 or 문자열'`
 - default값으로 '\n'(한 줄 내리기)가 들어간다.
 - `print`로 출력되는 문자열의 마지막

문자열 입출력 함수 연습문제

- 연습문제

: str이라는 변수에 이름을 입력받은 후 “<입력 받은 이름>님 안녕하세요?”

라는 문자열을 “포매팅 way1”을 활용하여 출력하기

```
이름을 입력하세요: 수진쌤  
수진쌤님 안녕하세요?
```

문자열 입출력 함수 Mission

- Mission

: ‘이름’과 ‘지금 하고 싶은 말’을 input으로 받은 후, 이를 아래와 같이 포매팅 way2(f-string)로 출력하기

```
이름을 입력하세요>>> 수진쌤
```

```
지금하고 싶은 말은?>>> 아..! 퇴근하고 싶다!
```

```
수진쌤은 아.. 퇴근하고 싶다!이라고 말했다.
```


Next

자료형과 변수 그리고 주식

프로그래밍... 왜 하는 것일까?

- “프로그래밍”의 의미
: 프로그램을 만드는 일
- 프로그래밍의 목적
: “문제를 해결”하기 위함.
- 프로그래밍 언어란?
: 프로그램을 만들기 위해 필요한 언어!! 언어마다 그 언어의 " 철학 " 이 있다.
(C, C++, C#, JAVA, Python, JS... 많은 프로그래밍 언어들이 존재한다.)

자료형(data type)이란? (교재 68p)

- 정의: 프로그래밍을 할 때 쓰이는 숫자, 문자열 등 자료 형태로 사용하는 모든 자료들의 형태
- 종류
 - int (정수형): -7, 0, 1, 125 등 정수로 표현된 숫자, ex) 안읽은 카카오톡 메시지 개수
 - string (문자열): ' ' 또는 " " 안에 들어가는 문자열, ex) 카카오톡 메시지
 - Boolean (불린형): 참/거짓으로 나타나는, ex) 카카오톡 메시지 안읽음 표시 '1'
 - float (실수형): 0.001, 0.25 등 실수형 표현
 - list : 리스트(중요), [] 안에 정의하는 순서형 자료형 - 이후에 할 내용
 - 이외에... tuple, dictionary, bool, set 등 다양한 자료형이 존재

변수와 자료형

- 변수 (variables) - 교재 37p
 - : 각 데이터를 담는 그릇
 - 정수, 실수, 문자열, 심지어 Screen과 Turtle 같은 객체들도 담을 수 있다.
 - 정의: 프로그램이 실행되는 동안 “변할 수 있는 값”
 - 장점: “유지보수성”
 - ↔ 상수: 프로그램이 실행되는 동안 변하지 않고 고정된 값
- 자료형: 데이터를 담는 그릇들의 종류



변수 만들기과 할당연산자

- 변수를 만드는 방법

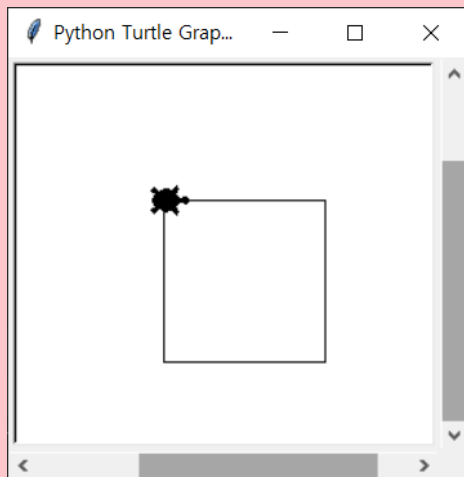
: [문법] 변수이름 = 데이터

- 할당 연산자(=)

- 수학에서는 ‘=’는 ‘같다’의 의미로 사용되지만, 프로그래밍에서의 “=”은 “왼쪽의 값을 오른쪽에 집어 넣는다(할당한다)”의 의미이다.
- 그런 의미에서 “=”을 “할당 연산자”라고 부른다.

변수의 장점

- turtle을 활용하여 사각형 그리기 예제를 통해 보도록 하자
만약, 사각형의 한 변의 길이를 200으로
늘리고 싶다면?



```
# turtle 사각형1
import turtle

# 스크린과 거북이 선언
win = turtle.Screen()
win.setup(500, 500)
t = turtle.Turtle('')

# 그림그리기
t.forward(100)
t.right(90)
t.forward(100)
t.right(90)
t.forward(100)
t.right(90)
t.forward(100)
t.right(90)
```

```
# turtle 사각형2
import turtle

# 변수들 선언
width = 500
height = 500
lineLength = 100
diffAngle = 90

# 스크린과 거북이 선언
win = turtle.Screen()
win.setup(width, height)
t = turtle.Turtle('')

# 그림그리기
t.forward(lineLength)
t.right(diffAngle)
t.forward(lineLength)
t.right(diffAngle)
t.forward(lineLength)
t.right(diffAngle)
t.forward(lineLength)
t.right(diffAngle)
```

변수의 이름을 짓는 규칙

- 데이터를 표현할 수 있는 이름으로 짓는다. (중요!! 가독성 높이기)
- 문자부터 시작해야 한다
- 대소문자는 구분해야 한다
- _로 시작할 수 있다
- 미리 예약된 키워드(예약어)는 사용할 수 없다.

기본 자료형: 숫자형

- 숫자형

- 정수형(integer, int)

- : 소수점이 없는 숫자의 형태. (ex. -1, 0, 2, 3 ...)

- 실수형(float)

- : 소수점이 있는 숫자의 형태. (ex. -2.6, 253.2, 0.0001...)

기본 자료형: 문자열, 불린형

- 문자열(string, str)
 - 문자를 나열한 형태의 자료형
 - “ ”(큰 따옴표) or ‘ ’(작은 따옴표)로 문자열의 시작과 끝을 나타낸다.
 - 123 vs “123”
 - 문자열안에 “ “ or ‘ ’을 넣고 싶은 경우엔 어떻게 할까?
 - TIP. print문에서 sep과 end 속성 활용
- 불린형(Boolean, bool)
 - : 참(True) or 거짓(False)로 나타나는 자료형

주석(comments)

- 작성한 코드를 설명하기 위한 문장
- python 인터프리터가 해석하지 않는 문장이다.
(코드를 읽는 사람에게 설명을 하기 위한 문장)
- 주석을 상세하게 적어 줄 수록 코드를 이해하기 쉬워진다.(“가독성”이 좋아진다.)
- 주석을 쓰는 방법
 - 한 줄 주석: 문장의 맨 앞에 # 입력(맨처음이 #으로 시작하면 주석 처리)
 - 여러 줄 주석: ''' ''' 또는 """ """ 로 코드들을 묶어준다
- Pycharm 단축키: 주석 처리를 원하는 문장들 드래그 & “Ctrl+ /”

주석(comments): 예시

```
# 1차 교습 mission
'''
~~~~~
[수업 내용]
1. python 개발 환경 구축(Anaconda, Pycharm, Git&Github 설치 및 가입)
2. 설치한 프로그램들의 연동(가상환경-Pycharm, Pycharm-Github)
3. 간단한 코드 작성 후 Github에 Commit & Push 수행
~~~~~
'''

# Mission1. 감격스러운 나의 첫 코드: Hello_world


# Mission2. python과 친해지기
## Mission2-1: 정사각형 그리기

## Mission2-2: 정삼각형 그리기

## Mission2-3: 별모양 그리기
```

변수와 주석 연습문제

- Mission

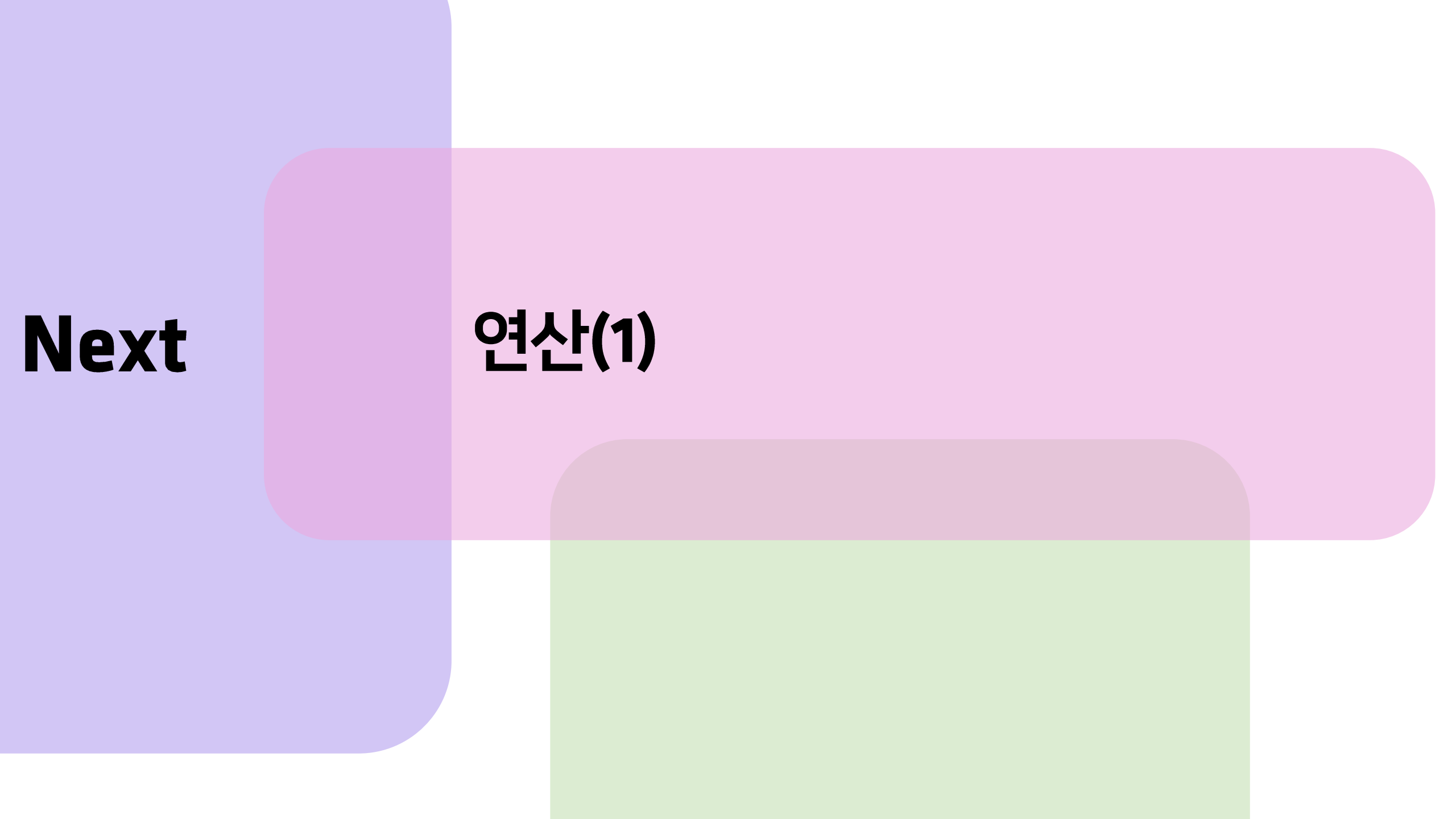
: “2nd_week_mission.py”의 ‘자료형과 변수’ 의 연습문제를 참고하여 문제를 풀어보자

- 변수의 자료형을 확인하는 명령어: `type(변수명)`

변수 Mission

- 속으로 20초를 세어 맞히는 프로그램(교재 71p)
: 하지만 우리는 깔끔하게 10초를 세는 프로그램을 만들어 보겠습니다.
- 활용할 모듈: `import time`
- 현재 시간을 반환하는 함수: `time.time()`
- `start`와 `end`에 Enter를 누른 시간을 저장 후, 그 차이를 출력해보자.
- 절댓값을 구하는 함수: `abs()`

```
Enter를 누르고 속으로 10초를 세는 게임입니다. Enter를 누르면 시작합니다!  
10초를 다 세셨다면 Enter를 눌러 stop!!  
실제시간: 7.922857999801636초  
차이: -2.0771420001983643 초
```



Next

연산(1)

산술연산자 - 교재 35p

- 여러가지 연산자

- 더하기: +
- 빼기: -
- 곱하기: *
- 나누기
 - 실수 몫: /
 - 정수 몫: //
- 나머지: %
- 거듭제곱: **

- 연산자에는 우선순위가 있다.

- 애매하면 먼저 계산할 식을 ()로 묶어주자

우선순위	연산자	설명
1	(값...), [값...], {키: 값...}, {값...}	튜플, 리스트, 딕셔너리, 세트 생성
2	x[인덱스], x[인덱스:인덱스], x(인수...), x.속성	리스트(튜플) 첨자, 슬라이싱, 함수 호출, 속성 참조
3	await x	await 표현식
4	**	거듭제곱
5	+x, -x, ~x	단항 덧셈(양의 부호), 단항 뺄셈(음의 부호), 비트 NOT
6	*, @, /, //, %	곱셈, 행렬 곱셈, 나눗셈, 버림 나눗셈, 나머지
7	+, -	덧셈, 뺄셈
8	<<, >>	비트 시프트

9	&	비트 AND
10	^	비트 XOR
11		비트 OR
12	in, not in, is, is not, <, <=, >, >=, !=, ==	포함 연산자, 객체 비교 연산자, 비교 연산자
13	not x	논리 NOT
14	and	논리 AND
15	or	논리 OR
16	if else	조건부 표현식
17	lambda	람다 표현식

문자열 연산

- 문자열 더하기

- python에서는 문자열끼리 더할 수 있다.
- [문법] <문자열1> + <문자열2> (ex. “안녕하세요.” + “반갑습니다“)
결과: “문자열1문자열2“(ex. “안녕하세요.반갑습니다”)

- 문자열 곱하기

- 나아가, python에서는 문자열을 곱할 수도 있다.
- [문법] <문자열> * 정수 (ex. “정신나갈꺼같아” * 3)
결과: <문자열을 정수만큼 반복>
(ex. “정신나갈꺼 같아정신나갈꺼같아정신나갈꺼같아”)

복합할당 연산자

- 정의

: 연산자와 할당연산자를 합쳐 놓은 것.

- 연산자계의 “단축키”라고 생각하면 됨.

연산자명	설명식	복합 대입 연산자	결과
<code>+=</code>	$a = a + 1$	<code>a += 1</code>	4
<code>-=</code>	$a = a - 1$	<code>a -= 1</code>	2
<code>*=</code>	$a = a * 2$	<code>a *= 2</code>	6
<code>/=</code>	$a = a / 2$	<code>a /= 2</code>	1.5
<code>%=</code>	$a = a \% 2$	<code>a %= 2</code>	1
<code>//=</code>	$a = a // 2$	<code>a //= 2</code>	1
<code>**=</code>	$a = a ** 2$	<code>a **= 2</code>	9

연산(1) 연습문제 & Mission

- 2nd_week_mission.py의 연산(1) 연습문제 및 Mission을 풀어보자
 - 산술 연산
 - 문자열 연산
 - 복합할당 연산

Next

연산(2)

비교연산(관계 연산)

- 두 개의 피연산자(변수나 상수)를 “비교”하는 데 사용하는 연산자
- 결과: 참(True) 또는 거짓(False)

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

논리 연산

- 논리 값을 판단해주는 연산자
- True(참,1)//False(거짓,0)

Operator	Description	Example
and	논리 AND 연산. 둘다 참일때만 참	(a and b) = False
or	논리 OR 연산. 둘 중 하나만 참이여도 참	(a or b) = True
not	논리 NOT 연산. 논리 상태를 반전	not(a and b) = True

a	b	AND	OR	XOR	NAND	NOR
0	0	0	0	0	1	1
0	1	0	1	1	1	0
1	0	0	1	1	1	0
1	1	1	1	0	0	0

멤버십 연산

- 포함관계를 나타내는 연산(결과 : True / False)
- in : 포함되어 있다.
- not in: 포함되어 있지 않다.

in 연산자 : 포함하는지 검사합니다.

not in 연산자 : 포함되어 있지 않은지 검사 합니다.

연산(2) Mission

- 2nd_week_mission.py의 연산(2) 연습문제 및 Mission을 풀어보자
 - 비교 연산
 - 논리 연산
 - 멤버십 연산