

# AI Programming

Lecture 4

# Assignment 2 Solution

```
name = input("이름이 어떻게 되시나요? ")
print("만나서 반갑습니다 %s 씨.\n" % name)

hobby = input("취미가 무엇인가요? ")
print("저도 %s 좋아합니다.\n" % hobby)

age = int(input("나이가 어떻게 되시나요? "))
print("내년이면 %d살이 되시는군요.\n" % (age + 1))

height = float(input("키가 어떻게 되시나요? "))
print("반올림하면 %.1f cm 이군요.\n" % height)

print("만나서 반가웠습니다 %s 씨.\n다음에는 %s를 같이 합시다." % (name, hobby))
```

# Assignment 2 Solution

```
name = input("이름이 어떻게 되시나요? ")
print("만나서 반갑습니다 {0:s} 씨.\n".format(name))

hobby = input("취미가 무엇인가요? ")
print("저도 {0:s} 좋아합니다.\n".format(hobby))

age = int(input("나이가 어떻게 되시나요? "))
print("내년이면 {0:d} 살이 되시는군요.\n".format(age+1))

height = float(input("키가 어떻게 되시나요? "))
print("반올림하면 {0:.1f} cm 이군요.\n".format(height))

print("만나서 반가웠습니다 {0:s} 씨.\n다음에는 {1:s}를 같이 합시다.".format(name, hobby))
```

# Assignment 2 Solution

```
name = input("이름이 어떻게 되시나요? ")  
print(f"만나서 반갑습니다 {name} 씨.\n")
```

```
hobby = input("취미가 무엇인가요? ")  
print(f"저도 {hobby} 좋아합니다.\n")
```

```
age = int(input("나이가 어떻게 되시나요? "))  
print(f"내년이면 {age + 1}살이 되시는군요.\n")
```

```
height = float(input("키가 어떻게 되시나요? "))  
print(f"반올림하면 {height:.1f} cm 이군요.\n")
```

```
print(f"만나서 반가웠습니다 {name} 씨.\n다음에는 {hobby}를 같이 합시다.")
```

# Preview

- Ch. 3 변수와 데이터형

- 3.4 데이터 표현 단위와 진수 변환

- Ch. 4 연산자

- 4.2 산술 연산자
  - 4.3 관계 연산자
  - 4.4 논리 연산자

## 3.4 데이터 표현 단위와 진수 변환

# Bit and Byte

- **Bit**

- The smallest unit of data
- 1 bit: 0 or 1 → two types of information
- $n$ -bit:  $2^n$  information

- **Byte**

- 1 byte = 8 bits

전기 스위치								
의미	꺼짐 , 꺼짐		꺼짐 , 켜짐		켜짐 , 꺼짐		켜짐 , 켜짐	
2진수	00		01		10		11	
10진수	0		1		2		3	

# Bit and Byte

- **Binary number systems**

- Decimal: 0, 1, 2, ..., 9
- Binary: 0 and 1
- Hexadecimal: 0, 1, ..., 14, 15
  - Numbers larger than 10: A, B,... ,F

표 3-3 10진수, 2진수, 16진수 변환표

10진수(0~9)	2진수(0~1)	16진수(0~F)
00	0000	0
01	0001	1
02	0010	2
03	0011	3
04	0100	4
05	0101	5
06	0110	6
07	0111	7
08	1000	8
09	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F



# Conversion

- **Binary**

- 0b
- `int('number', 2)`

- **Hexadecimal**

- 0x
- `int('number', 16)`

- **Octal**

- 0o
- `int('number', 8)`

```
>>> a = 0b1001
>>> print(a)
9
>>> b = int('1001', 2)
>>> print(b)
9
>>> c = 0x1F
>>> print(c)
31
>>> d = int('1F', 16)
>>> print(d)
31
```

# Conversion

- 진수 변환

- `hex( )`, `oct( )`, `bin( )`

```
bin(11); bin(0o11); bin(0x11)  
oct(11); oct(0b11); oct(0x11)  
hex(11); hex(0b11); hex(0o11)
```

## 출력 결과

```
'0b1011'  '0b1001'  '0b10001'  
'0o13'    '0o3'     '0o21'  
'0xb'     '0x3'     '0x9'
```

# Conversion

## • 진수 변환기

- Step 1) 16/10/8/2 진수 종류를 입력 받음
- Step 2) 숫자 값을 입력 받음
- Step 3) 각 진수 변환값 출력

```
if 진수 == 16:  
    int(값, 16)
```

```
Python 3.6.0 Shell  
File Edit Shell Debug Options Window Help  
===== RESTART: C:/CookPython/Code03-04.py =====  
입력 진수 결정(16/10/8/2) : 16  
값 입력 : FF  
16진수 ==> 0xff  
10진수 ==> 255  
8진수 ==> 0o377  
2진수 ==> 0b11111111  
>>>  
Ln: 579 Col: 77
```

# Conversion

```
1 sel = int(input("입력 진수 결정(16/10/8/2) : "))
2 num = input("값 입력 : ")
3
4 if sel == 16 :
5     num10 = int(num, 16)
6 if sel == 10 :
7     num10 = int(num, 10)
8 if sel == 8 :
9     num10 = int(num, 8)
10 if sel == 2 :
11     num10 = int(num, 2)
12
13 print("16진수 ==> ", hex(num10))
14 print("10진수 ==> ", num10)
15 print(" 8진수 ==> ", oct(num10))
16 print(" 2진수 ==> ", bin(num10))
```

## 4.2 산술 연산자

# Arithmetic Operators

연산자	의미	사용 예	설명
=	대입 연산자	$a = 3$	정수 3을 a에 대입
+	더하기	$a = 5 + 3$	5와 3을 더한 값을 a에 대입
-	빼기	$a = 5 - 3$	5에서 3을 뺀 값을 a에 대입
*	곱하기	$a = 5 * 3$	5와 3을 곱한 값을 a에 대입
/	나누기	$a = 5 / 3$	5를 3으로 나눈 값을 a에 대입
//	나누기(몫)	$a = 5 // 3$	5를 3으로 나눈 후 소수점을 버리고 값을 a에 대입
%	나머지값	$a = 5 \% 3$	5를 3으로 나눈 후 나머지값을 a에 대입
**	제곱	$a = 5 ** 3$	5의 3제곱을 a에 대입

# Arithmetic Operators

- Examples

```
a = 5; b = 3  
print(a + b, a - b, a * b, a / b, a // b, a % b, a ** b)
```

출력 결과

```
8 2 15 1.6666666666666667 1 2 125
```

# Arithmetic Operators

- **Priority**

- $( ) \rightarrow \times, \div \rightarrow +, -$

❶  $(a + b) - c$

❷  $a + (b - c)$

❶  $(a + b) * c \rightarrow (2 + 3) * 4 \rightarrow 5 * 4 \rightarrow 20$

❷  $a + (b * c) \rightarrow 2 + (3 * 4) \rightarrow 2 + 12 \rightarrow 14$

>>> 2/2\*4

4.0

>>> 2/(2\*4)

0.25



# Arithmetic Operators

- `int()` & `float()`
  - Type casting (형변환)

[illegible]

## 출력 결과

101 101.123 10000000000000000000000000000000

# Arithmetic Operators

- `int()` & `float()`

- `int(100)`

```
>>> int(100)
100
```

- `int('100.5')`

```
>>> int('100.5')
```

```
Traceback (most recent call last):
```

```
  File "<pyshell#18>", line 1, in <module>
```

```
    int('100.5')
```

```
ValueError: invalid literal for int() with base 10: '100.5'
```

- `int(100.5)`

```
>>> int(100.5)
```

```
100
```

- `float(100)`

```
>>> float(100)
```

```
100.0
```

- `float('100.5')`

```
>>> float('100.5')
```

```
100.5
```

# Arithmetic Operators

- `str()`

```
a = 100; b = 100.123  
str(a) + '1'; str(b) + '1'
```

**출력 결과**

```
'1001'
```

```
'100.1231'
```

# Arithmetic Operators

- Compound operator (복합 연산자)

- 산술 연산자 + 대입 연산자

연산자	사용 예	설명
<code>+=</code>	<code>a += 3</code>	<code>a = a + 3</code> 과 동일
<code>-=</code>	<code>a -= 3</code>	<code>a = a - 3</code> 과 동일
<code>*=</code>	<code>a *= 3</code>	<code>a = a * 3</code> 과 동일
<code>/=</code>	<code>a /= 3</code>	<code>a = a / 3</code> 과 동일
<code>//=</code>	<code>a //= 3</code>	<code>a = a // 3</code> 과 동일
<code>%=</code>	<code>a %= 3</code>	<code>a = a % 3</code> 과 동일
<code>**=</code>	<code>a **= 3</code>	<code>a = a ** 3</code> 과 동일

`a++`

`a--`



# Arithmetic Operators

- Exercise

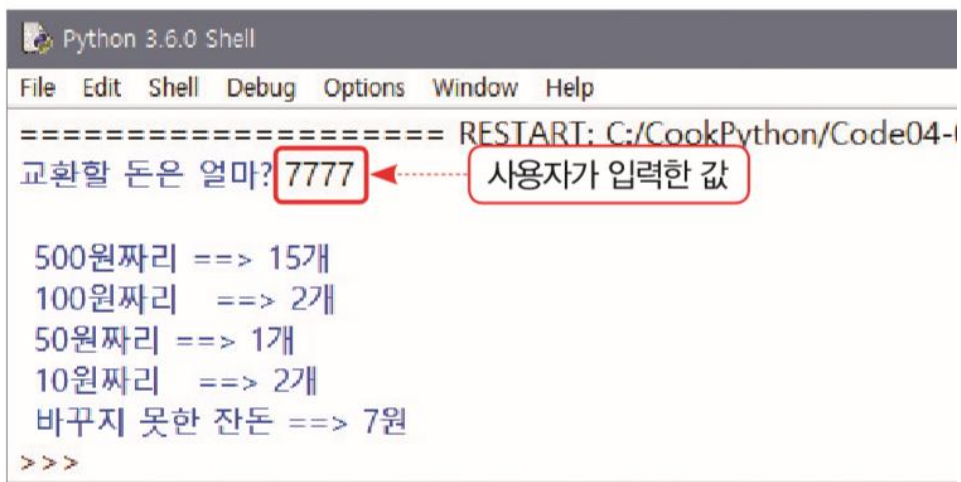
```
a = 10  
a += 5; print(a)  
a -= 5; print(a)  
a *= 5; print(a)  
a /= 5; print(a)  
a //= 5; print(a)  
a %= 5; print(a)  
a **= 5; print(a)
```

출력 결과

15 10 50 10.0 2.0 2.0 32.0

# Arithmetic Operators

## • 동전 교환기



```
Python 3.6.0 Shell
File Edit Shell Debug Options Window Help
===== RESTART: C:/CookPython/Code04-1
교환할 돈은 얼마? 7777
500원짜리 ==> 15개
100원짜리 ==> 2개
50원짜리 ==> 1개
10원짜리 ==> 2개
바꾸지 못한 잔돈 ==> 7원
>>>
```

```
1  ## 변수 선언 부분 ##
2  money, c500, c100, c50, c10 = 0, 0, 0, 0, 0
3
4  ## 메인 코드 부분 ##
5  money = int(input("교환할 돈은 얼마?"))
6
7  c500 = money // 500
8  money %= 500
9
10 c100 = money // 100
11 money %= 100
12
13 c50 = money // 50
14 money %= 50
15
16 c10 = money // 10
17 money %= 10
18
19 print("\n 500원짜리 ==> %d개" % c500)
20 print(" 100원짜리 ==> %d개" % c100)
21 print(" 50원짜리 ==> %d개" % c50)
22 print(" 10원짜리 ==> %d개" % c10)
23 print(" 바꾸지 못한 잔돈 ==> %d원 \n" % money)
```

## 4.3 관계 연산자

# Relational Operators

- **Relational operators (관계 연산자)**

- 두 변수 간의 관계를 확인할 때 사용
- Result: Bool-type variable (True and False)
- Applications: `if` and `while` loops

$a < b = \begin{cases} \text{참} & : \text{True} \\ \text{거짓} & : \text{False} \end{cases}$



# Relational Operators

- Relational operators

연산자	의미	설명
==	같다.	두 값이 동일하면 참
!=	같지 않다.	두 값이 다르면 참
>	크다.	왼쪽이 크면 참
<	작다.	왼쪽이 작으면 참
>=	크거나 같다.	왼쪽이 크거나 같으면 참
<=	작거나 같다.	왼쪽이 작거나 같으면 참

# Relational Operators

- Examples

```
a, b = 100, 200
```

```
print(a == b , a != b, a > b , a < b , a >= b , a <= b)
```

출력 결과

False True False True False True

## 4.4 논리 연산자

# Logical Operators

- Logical operators

- and, or, not

연산자	의미	설명	사용 예
and(논리곱)	~이고, 그리고	둘 다 참이어야 참	(a > 100) and (a < 200)
or(논리합)	~이거나, 또는	둘 중 하나만 참이어도 참	(a == 100) or (a == 200)
not(논리부정)	~아니다, 부정	참이면 거짓, 거짓이면 참	not(a < 100)

# Logical Operators

- Bool variables and logical operators

```
>>> True and True
True
>>> True and False
False
>>> True or False
True
>>> not True
False
```

# Logical Operators

- Examples

```
a = 99  
(a > 100) and (a < 200)  
(a > 100) or (a < 200)  
not(a == 100)
```

출력 결과

False True True

# Logical Operators

- **True and False of integers**

- 0: False
- Others: True

```
if(1234) : print("참이면 보여요")  
if(0) : print("거짓이면 안 보여요")
```

# Logical Operators

## • 윤년 계산기

- 규칙 1) 4로 나누어 떨어지고 동시에 100으로 나누어 떨어지지 않으면 윤년이다.
- 규칙 2) 400으로 나누어 떨어지는 해도 윤년에 포함된다.

연도를 입력하세요 : 2020  
2020년은 윤년입니다.

```
year = int(input("연도를 입력하세요 : "))
```

```
if ((year % 4 == 0) and (year % 100 != 0)) or (year % 400 == 0):  
    print(f"{year}년은 윤년입니다.")
```



# Summary

- **Arithmetic operators**

- `+`, `-`, `x`, `/`, ...

- **Relational operators**

- `==`, `!=`, `>`, `<`, ...

- **Logical operators**

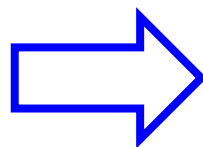
- `and`, `or`, `not`

# Assignment 3

- 고급 진수 변환기

- 진수에 16, 10, 8, 2 이외의 숫자를 입력 받으면 사용자에게 재입력 유도
  - if, !=, and

```
입력 진수 결정(16/10/8/2) : 7  
값 입력 : FF  
Traceback (most recent call last):  
  File "C:/Users/user/Dropbox/수업/2019년  
Code03-04.py", line 13, in <module>  
    print("16진수 ==>", hex(num10))  
NameError: name 'num10' is not defined
```



```
입력 진수 결정(16/10/8/2) : 7  
16, 10, 8, 2 숫자 중 하나만 입력하세요 : 16  
값 입력 : FF  
16진수 ==> 0xff  
10진수 ==> 255  
8진수 ==> 0o377  
2진수 ==> 0b11111111
```