

# 파이썬 프로그래밍 강의 노트 #03

---

## 변수와 사용자 입력

# 변수의 필요성

- 계산하면서 값을 기록했다가 다시 사용할 때가 있음
  - 예:  $3 + 4 + 10 =$
  - 사람은 단순히 계산할 수 있음
  - 컴퓨터는 3과 4를 더한 후 결과값을 기억해야 함
- 휴대폰에 입력되어 있는 주소록에서 이름으로 검색하는 것도 마찬가지
  - 사용자가 입력한 이름을 저장하고 주소록에 있는 이름들과 비교해야 함
- 파이썬을 비롯한 프로그래밍 언어에서는 값을 기억시키기 위해 메모리를 사용함

# 변수의 필요성

- 다음 연산에서  $3 + 4 + 10$ 은 중복됨

$3 + 4 + 10$	# (1)
$3 + 4 + 10 + 20$	# (2)

- 파이썬 인터프리터가  $3 + 4 + 10$ 의 결과값을 임시로 저장하지는 않음
  - 프로그래머가 직접 해야 함
  - (1)에서  $3 + 4 + 10$ 의 결과값을 메모리에 저장했다가 (2)에서 재사용하는 것이 바람직함

# 간단한 하드웨어 설명

---

## □ 프로세서(CPU)

- 계산 담당
- 아주 작은 양의 메모리만 있음

## □ 메모리

- 컴퓨터에서 기억 공간으로 사용될 수 있는 하드웨어 요소
- 일반적으로 바이트 단위로 주소가 있음 (정수 주소)
- 주기억(primary memory) 장치라고도 함

## □ 보조 기억 (secondary storage) 장치

- 파일 등에 자료를 저장할 수 있음
- ssd/hdd 등

# 변수

## □ 변수 (variable)

- 값을 담아두는 메모리 공간
- 또는 메모리 공간에 저장된 값

```
tempSum = 3 + 4 + 10 # (1) tempSum은 3 + 4  
                    #      + 10의 결과를 저장  
print(tempSum + 20) # (2) tempSum을 이용해서  
                    #      3 + 4 + 10의 결과를  
                    #      재사용
```

- (1)에서의 tempSum은 공간
- (2)에서의 tempSum은 공간에 있는 값

# 대입 연산자와 대입문

- 변수는 메모리 공간 또는 메모리 공간의 값으로 사용됨
- 언제 어떤 역할을 하는 것일까?

```
tempSum = 3 + 4 + 10      # (1)
```

- = 는 대입 연산자
- 변수가 대입 연산자를 기준으로 왼쪽에 있으면 메모리 공간
- 대입 연산자의 오른쪽에 있으면 메모리 공간의 값을 나타냄

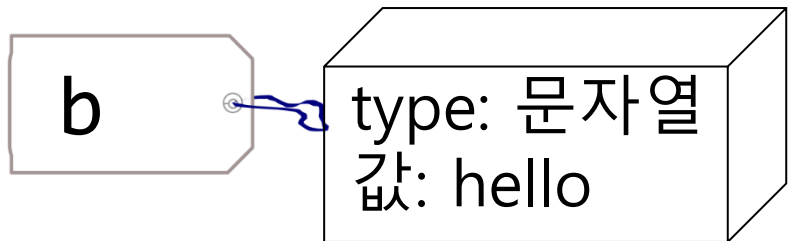
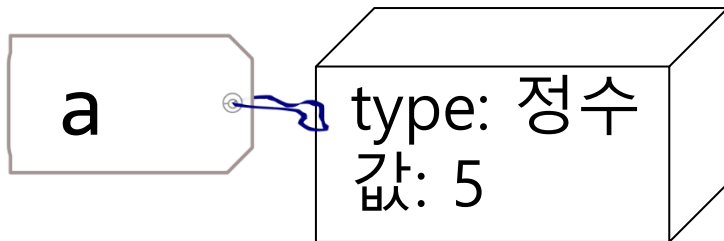
# 대입 연산자와 대입문

- 대입 연산자를 사용하는 (1)번 코드는 대입 명령문
  - 명령문
    - 파이썬에서 코드를 구성하고 실행하는 단위
- 변수 이름
  - 프로그래밍 언어에 따라서 조금씩 다르게 처리됨(설명됨)
    - 메모리 공간에 있는 값 (공통으로 많이 쓰임)
    - 메모리 공간에 붙은 이름
    - 메모리 공간에 붙은 이름표
  - 파이썬에서는 변수 이름은 메모리 공간에 있는 숫자, 문자열, 객체(object)를 가리키는 이름표의 역할을 함
  - 변수 이름표를 메모리 공간에 붙이는 것을 "변수가 메모리 공간 또는 메모리 공간에 있는 값을 참조(reference)한다"라고 함

# 변수 생성 및 대입문

- 파이썬의 변수는 처음 값이 저장(대입)될 때만 들어짐

```
a = 5          # a라는 이름의 변수 생성  
b = "hello"    # b라는 이름의 변수 생성
```



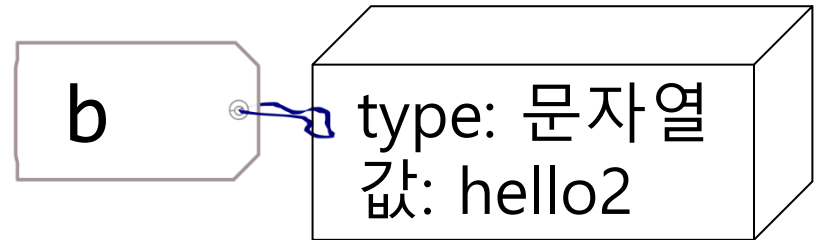
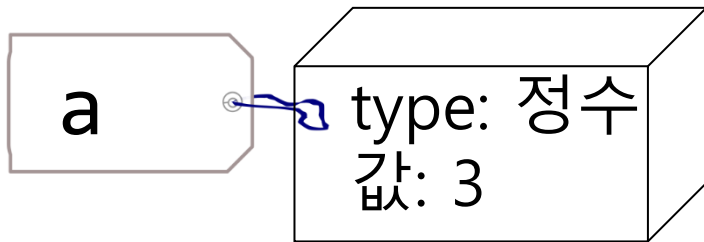
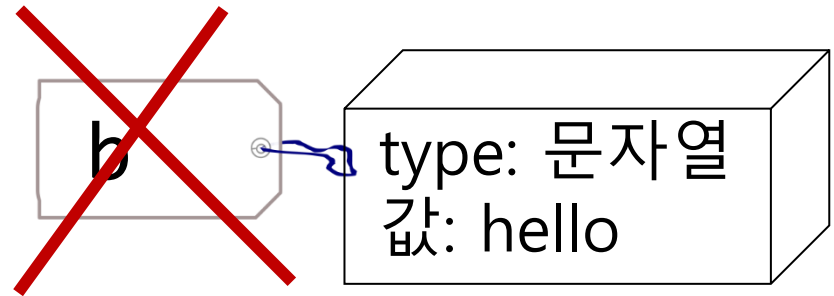
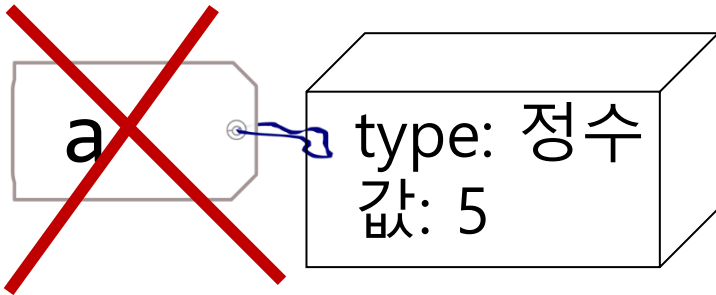
```
a = 5
```

- '='는 대입 연산자
- `a = 5`는 대입문이라고 함
- 변수가 대입문에서 처음 사용되면 새로 생성됨



# 변수 생성 및 대입문

```
a = 3      # 기존 a라는 이름표를 다른 공간에 붙임  
b = "hello2"  # b라는 이름표를 새로운 곳에 붙임
```



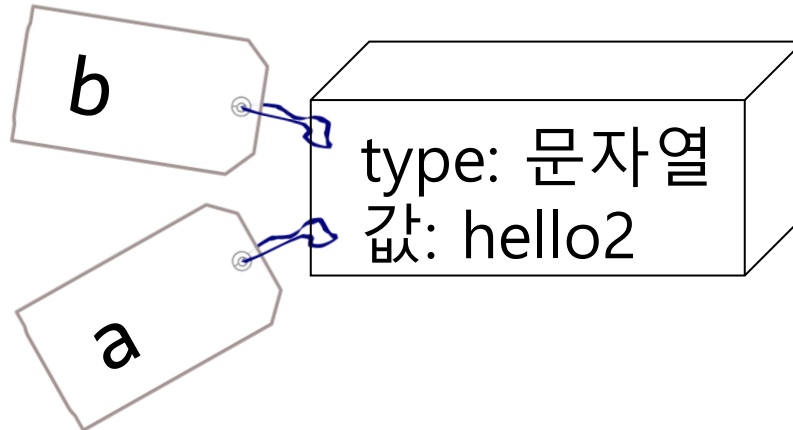
# 변수 생성 및 대입문

```
a = "hello2" # a를 기존 "hello2" 공간에 붙임
```

type: 정수  
값: 5

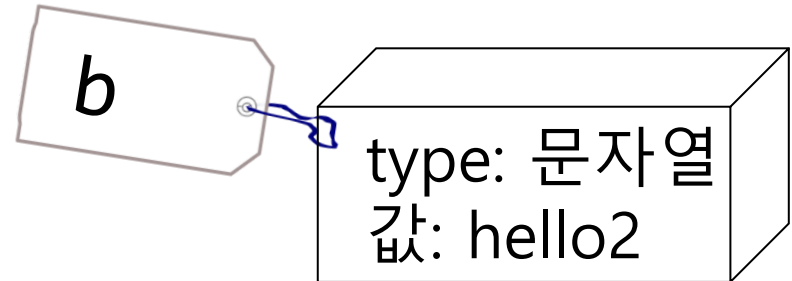
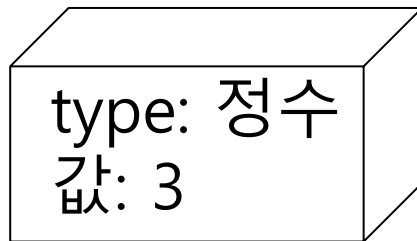
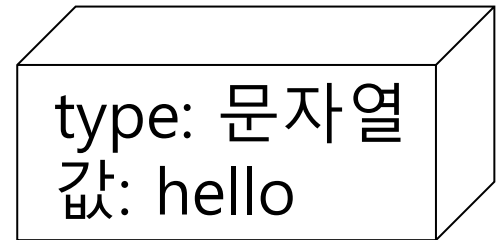
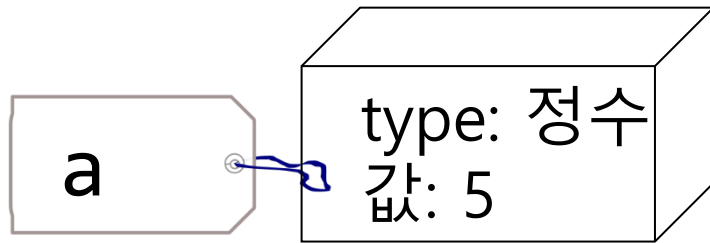
type: 문자열  
값: hello

type: 정수  
값: 3



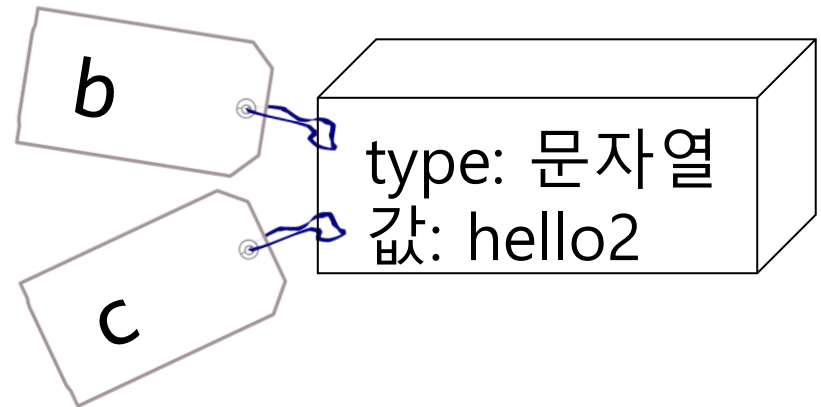
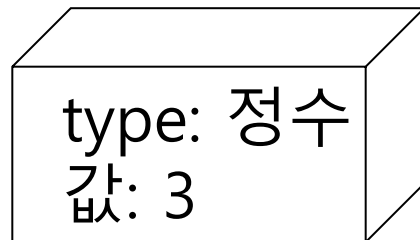
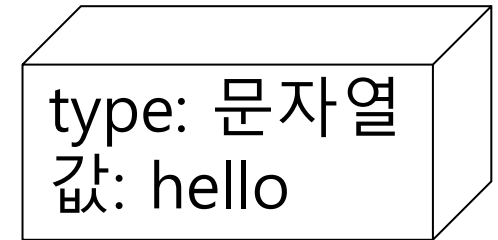
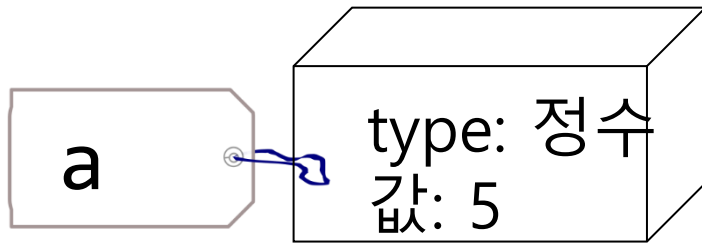
# 변수 생성 및 대입문

`a = 5` # a를 기존 5 공간에 붙임



# 변수 생성 및 대입문

`c = b`      # `c`를 생성하고 `b` 이름표 공간에 붙임



# 변수와 자료형

- 파이썬에서 변수는 대입 연산자의 왼쪽에 처음 나타날 때 생성됨

```
>>> x = 1 # 1을 넣으면 자료형이 숫자
>>> type(x)
<class 'int'> # int 타입. 단 class 'int'로 정수
                # 객체 타입이라는 점 주의하자.
>>> x = "hello" # "hello"을 넣으면 자료형이
                # 문자열로 변경
>>> type(x)
<class 'str'>
```

# 변수 이름 짓기와 변수 생성 방법

## □ 파이썬의 변수 이름 짓기 규칙

- 이름 길이에 제한 없음
- 변수는 반드시 유니코드 문자나 '\_'로 시작
- 문자나 '\_'로 시작한 후에는 숫자도 사용 가능
- 변수 이름에 공백이 없어야 함
- 특수 문자 활용 불가
- 키워드(예약어)가 아니어야 함
- 대문자와 소문자를 다르게 구별함
- 좋은 예: a, a2, \_a, MyVariable, \_variable, myVariable
  - MyVariable과 myVariable은 다른 변수
- 나쁜 예: 2a, a\$, My Variable, for

# 적절한 변수 이름

변수 이름	설명
a	영문자로 시작했고, 한 글자로만 구성
a2	영문자로 시작했고, 두 번째 글자는 숫자를 사용
_a	밑줄 문자로 시작 가능
__variable	밑줄 문자를 두 개 붙여서 시작하는 것도 가능
MyVariable	대문자와 소문자를 섞어서 사용 가능
myVariable	대문자와 소문자를 섞어서 사용 가능

## 변수 이름의 잘못된 예

변수 이름	설명
2a	이름은 숫자로 시작 못함
a\$	이름에 특수 기호 사용 못함
My Variable	이름 중간에 공백 문자 포함 못함
for	파이썬의 키워드는 이름으로 사용 못함



# 실습문제 1

## □ 문제

- 커피 원두 100g의 가격이 10000원
- 커피 원두는 최소 200g부터 구매할 수 있고, 100g 단위로 구매 가능
- 200g, 300g, 400g을 구매할 때의 가격을 화면에 출력

## □ 요구사항

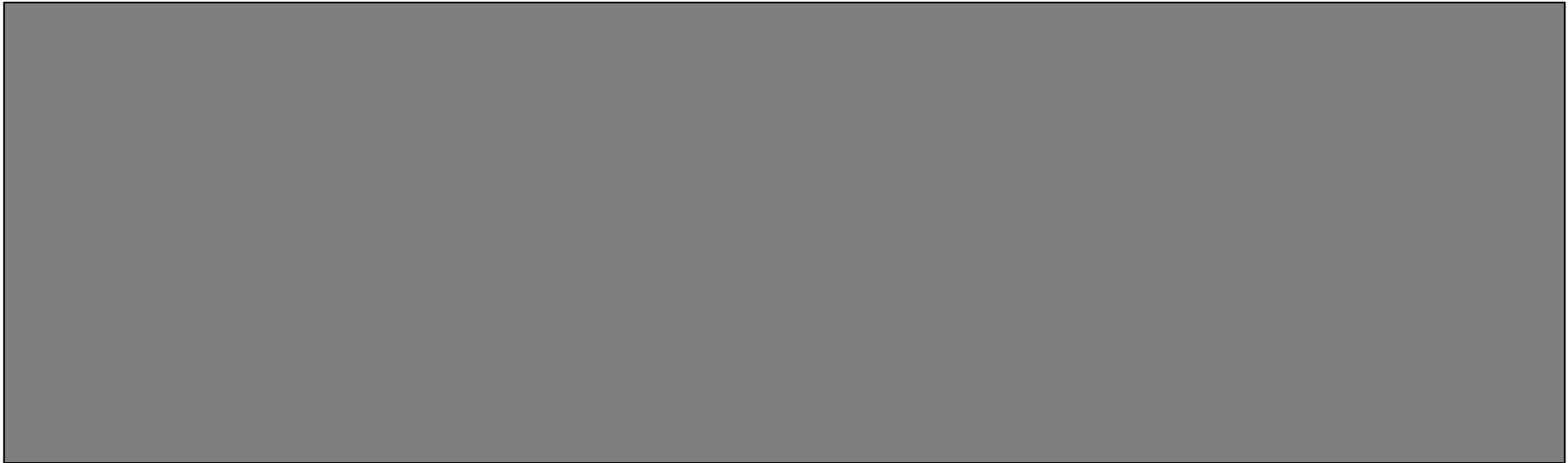
- 변수에 커피 원두 100g의 가격을 저장한 후, 산술 연산을 통해 200, 300, 400g의 가격을 산출해서 출력
- 화면 출력 내용은 다음 형태로 처리

커피 원두	200g	가격:	20000원
커피 원두	300g	가격:	30000원
커피 원두	400g	가격:	40000원

# 실습문제 1

---

## ▣ 최종 코드



# 복합 연산자

- 복합 연산자는 대입 연산과 산술 연산을 함께 처리하는 방법

- 사용법

변수 복합연산자 표현식

- 연산자의 왼쪽에 있는 변수 값과 오른쪽 표현식의 결과 값으로 산술 연산을 한 후, 다시 왼쪽 변수에 저장

- 변수는 이미 생성된 변수여야 함

- 예

a += 3

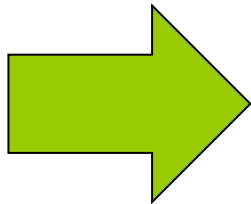
# 복합 연산자

복합 연산자	산술 연산	사용 예	a=2로 연산한 결과값
변수 += 값	변수 = 변수 + 값	a += 2	4
변수 -= 값	변수 = 변수 - 값	a -= 2	0
변수 *= 값	변수 = 변수 * 값	a *= 2	4
변수 /= 값	변수 = 변수 / 값	a /= 2	1.0
변수 //= 값	변수 = 변수 // 값	a //= 2	1
변수 %= 값	변수 = 변수 % 값	a %= 2	0
변수 **= 값	변수 = 변수 ** 값	a **= 2	4

# 복합 연산자

▣ 다음 코드를 실행시키고 결과 화면에 출력하기

```
a = 5  
a += 2  
a -= 2.1  
a *= 2.1  
a /= 2.1  
a = 5  
a //= 2  
a %= 2  
a **= 2
```



## 실습문제 2

### □ 문제

- 커피 원두 100g의 가격이 10000원
- 커피 원두는 최소 200g부터 구매할 수 있고, 100g 단위로 구매 가능
- 200g, 300g, 400g을 구매할 때의 가격을 화면에 출력

### □ 요구사항

- 변수에 커피 원두 100g의 가격을 저장한 후, 복합 연산을 통해 100g의 단가를 차례대로 더해서 200, 300, 400g의 가격을 출력
- 화면 출력 내용은 다음 형태로 처리

커피 원두	200g	가격:	20000원
커피 원두	300g	가격:	30000원
커피 원두	400g	가격:	40000원

## 실습문제 2

---

### ▣ 최종 코드



# 서식에 맞춰 문자열을 생성하고 화면에 출력

## □ 서식 출력의 필요성

- 화면에 거의 동일한 문자열을 출력하면서 약간의 내용만 바꿔 반복적으로 출력하는 경우를 고려
- 예를 들어, 메일이나 문자를 받았는데 단체 메일 또는 단체 문제 내용인 것 같은데 시작 부분은 "김가온님께" 같은 형태로 개인에게 따로 쓴 것처럼 되어 있는 경우
- 신입생 환영회나 새로운 모임에 가서 순서대로 자신의 이름을 소개하는 경우

"안녕하세요. 제 이름은 파이썬입니다"

"안녕하세요. 제 이름은 이서우입니다"

"안녕하세요. 제 이름은 김가온입니다"



# 서식에 맞춰 문자열을 생성하고 화면에 출력

## □ 형식 지정자와 값

- 서식 문자열을 지정하려면 문자열에서 값을 변경할 부분에 '%'와 자료형을 나타내는 문자를 붙여 넣음

## □ 형식 지정자

- '%'와 자료형을 나타내는 문자열을 묶어서 지칭

## □ 자신을 소개하는 문자열을 서식으로 만들어보기

"안녕하세요. 제 이름은 %s입니다"

## □ 문자열에 %s같은 형식 지정자가 있다면, 파이썬은 %s 부분을 다른 값으로 치환

## □ 형식 지정자를 치환할 값은 "% 값" 또는 "%값"으로 작성 → 형식 지정자가 주어진 값으로 치환된 새로운 문자열이 생성됨

# 문자열과 다른 자료형 섞기

- 서식을 만들어놓고 일부 내용을 바꿔서 문자열을 생성할 수 있음
- 서식 문자열을 지정하려면 값을 변경해야 하는 부분에 '%' 글자와 자료형을 나타내는 문자를 붙여서 지정 → 형식 지정자


"안녕하세요. 제 이름은 %s입니다"

- 값을 지정하는 방법 "% 값" 또는 "%값"을 서식 문자열 뒤에 붙임

## 문자열과 다른 자료형 섞기

- 서식에 있는 형식 지정자를 뒤에 있는 값으로 치환해서 문자열 생성

```
>>> "안녕하세요. 제 이름은 %s입니다" % "파이썬"  
'안녕하세요. 제 이름은 파이썬입니다'
```



"안녕하세요. 제 이름은 %s입니다" % "파이썬"

# 문자열과 다른 자료형 섞기

## ▣ 형식 지정자

형식 지정자	설명
%s	문자열
%d	정수
%f	실수

```
>>> s = "원의 반지름: %d" % 20 # 문자열 생성 후 출력
>>> print(s)
원의 반지름: 20
>>> print("원의 반지름: %f" % 20) # print( ) 명령에
생성되는 문자열을 바로 전달
20.000000
```

# 서식 문자열과 변수 사용

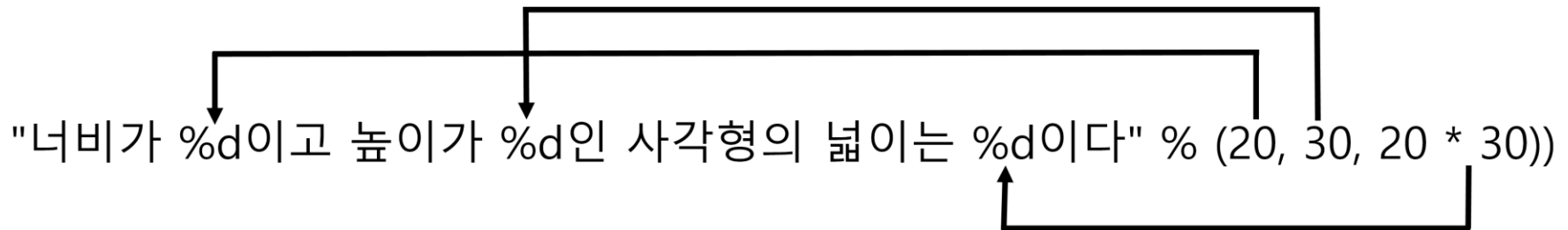
- ▣ 서식 문자열에 치환되는 값을 문자열 또는 숫자값 대신 변수로 사용 가능

```
address = "종로구 홍지문 2길" # 사용자가 입력한  
                                # 정보라고 가정  
temperature = 24 # 인터넷에서 취득한 온도라고 가정  
  
print("지역: %s" % address)  
print("온도: %d도" % temperature)
```

## 여러 개 값을 치환

- 두 개 이상의 값을 형식 지정자를 이용해서 치환하는 경우, "% (값1, 값2, ...)" 형태로 지정

```
>>> print("너비가 %d이고 높이가 %d인 사각형의 넓이  
        는 %d이다" % (20, 30, 20 * 30))  
너비가 20이고 높이가 30인 사각형의 넓이는 600이다
```



# 형식 지정자의 다양한 활용법

## □ 소수점 이하 자릿수 지정

- 실수값을 '%'와 'f'사이에 마침표와 자릿수를 적으면 소수점 자릿수 지정 가능

## □ 원주율을 소수점 둘 째 자리까지 출력하는 코드

```
>>> print("원주율: %.2f" % 3.1415)  
원주율: 3.14
```

## □ 소수점 이하 자릿수를 지정할 때 기존 값이 잘리면 반올림함

```
>>> print("원주율: %.3f" % 3.1415)  
원주율: 3.142  
>>> print("원주율: %.6f" % 3.1415)  
원주율: 3.141500
```

# 전체 자릿수 지정

- "%10s"처럼 형식 지정자의 중간에 숫자를 넣으면 전체 자릿수 지정 가능
  - 값의 길이가 전체 자릿수보다 작으면 빈칸으로 채워지고 값은 오른쪽으로 정렬

```
>>> print("%10s" % "한글")  
      한글
```

								한	글
--	--	--	--	--	--	--	--	---	---

- 값이 전체 자릿수보다 길다면 그대로 출력

```
>>> print("%5s" % "한글을 출력합니다")  
한글을 출력합니다
```



# 전체 자릿수 지정

## □ 숫자도 마찬가지로

```
>>> print("%8d" % 10)
      10
```

## □ 실수값을 출력할 때는 자릿수에 맞춰 0으로 채움

```
>>> print("%8f" % 3.1415)
3.141500
```

## □ 실수값을 출력하면서 오른쪽으로 정렬하려면 소수점까지 표시

```
>>> print("%8.2f" % 3.1415)
      3.14
```

# 서식 문자열에서 % 출력

- ▣ 서식 문자열을 이용해서 '%' 문자를 출력하려면 "%%"처럼 두 번 붙여 사용

```
>>> print("오늘의 습도는 %d%%입니다" % 70)  
오늘의 습도는 70%입니다
```

# 실습문제 3

## □ 문제

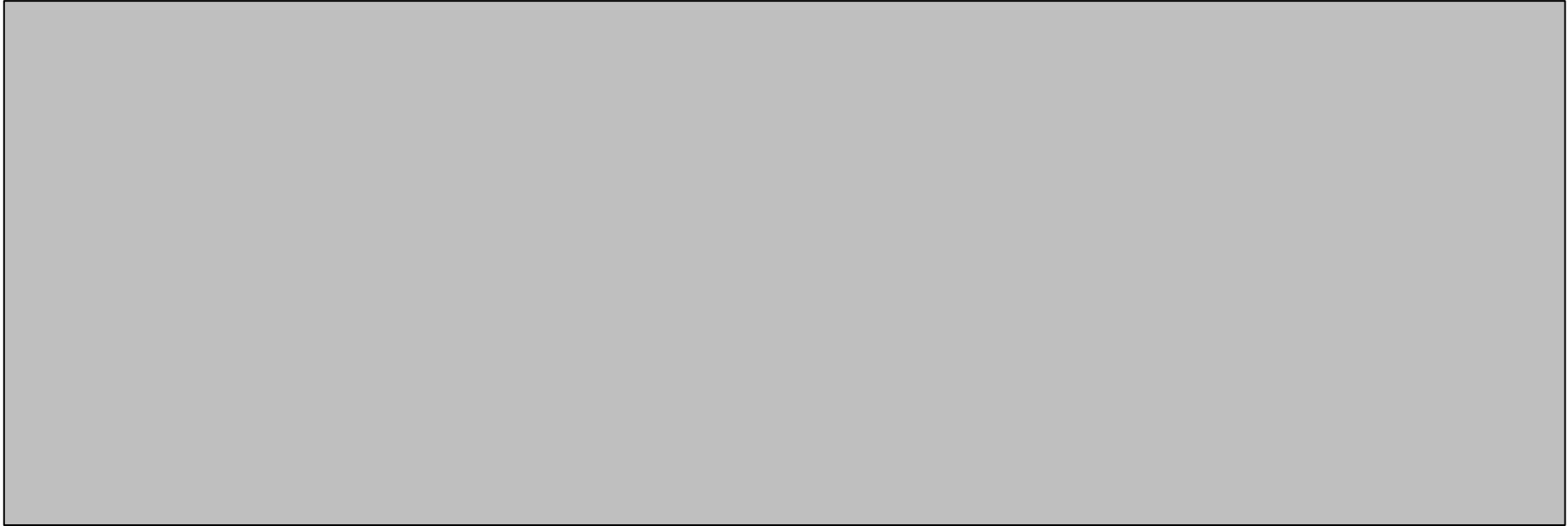
- '2022년 아카데미 영화제 작품상은 "코다"가 받았다'라는 문장에서 연도와 작품 이름을 바꿀 수 있게 서식을 활용해서 출력하는 코드를 작성하고, 다음 표에 있는 값으로 바꿔 출력하는 코드를 작성

연도	영화 제목
2020	기생충
2021	노매드랜드
2022	코다

## 실습문제 3

---

### ▣ 최종 코드



## 실습문제 4

---

### □ 문제

- 실습문제 1에서 사용한 공식을 이용해서 원두 200g과 300g의 가격을 화면에 출력

### □ 요구사항

- 커피 무게를 `weight`라는 변수에 저장 후 산술 연산식과 `print()` 명령문에서 사용
- 커피 무게와 가격을 형식 지정자를 이용해서 치환

## 실습문제 4

---

### ▣ 최종 코드



# f-string

- 파이썬 3.6부터 사용 가능

- 서식 지정

```
f"서식 문자열"
```

- 서식 문자열에서 값을 넣어야 하는 부분에 {변수} 형태로 지정 가능

```
f"너비가 {width}이고 높이가 {height}인 사각형의 넓이는 {area}이다"
```

- 만약 '{' 또는 '}'를 서식 문자열에 넣고 싶으면 "{{" 또는 "}}" 형태로 쓰면 됨

```
f"{{{({width} + {height}) * 2 }}}"
```

# 실습문제 5

---

## □ 문제

- 실습문제 3을 f-문자열을 이용해서 출력

## □ 최종 코드





# 키보드로 입력 받기

---

## □ 출력

- print() 명령을 사용함

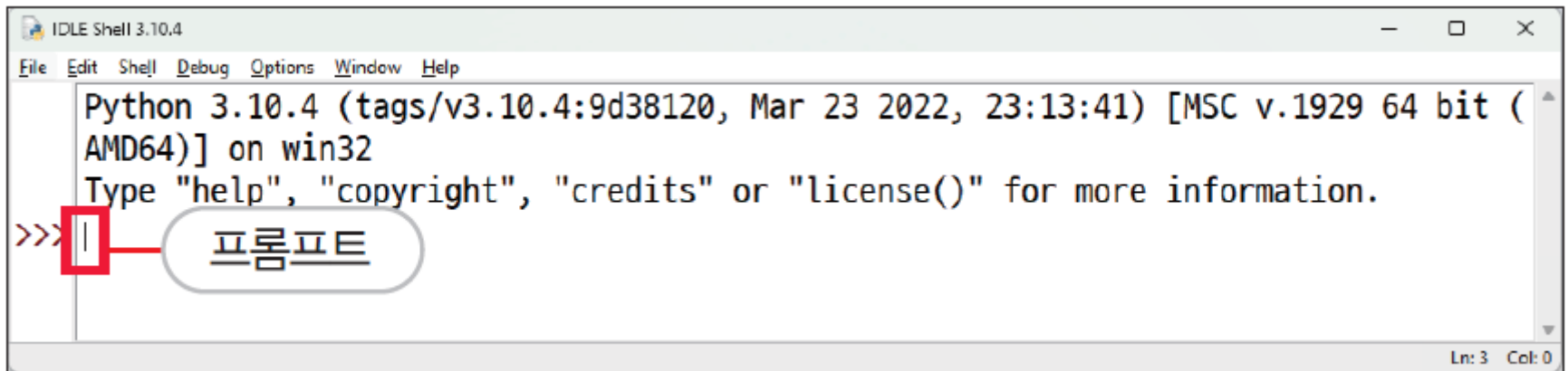
## □ 입력

- 사용자로부터 문자열/숫자 등을 입력 받아서 프로그램에서 사용
- input() 명령을 사용함
- 아래 코드를 파이썬 쉘에서 실행시켜 볼 것

```
input()
```

# 키보드로 입력 받기

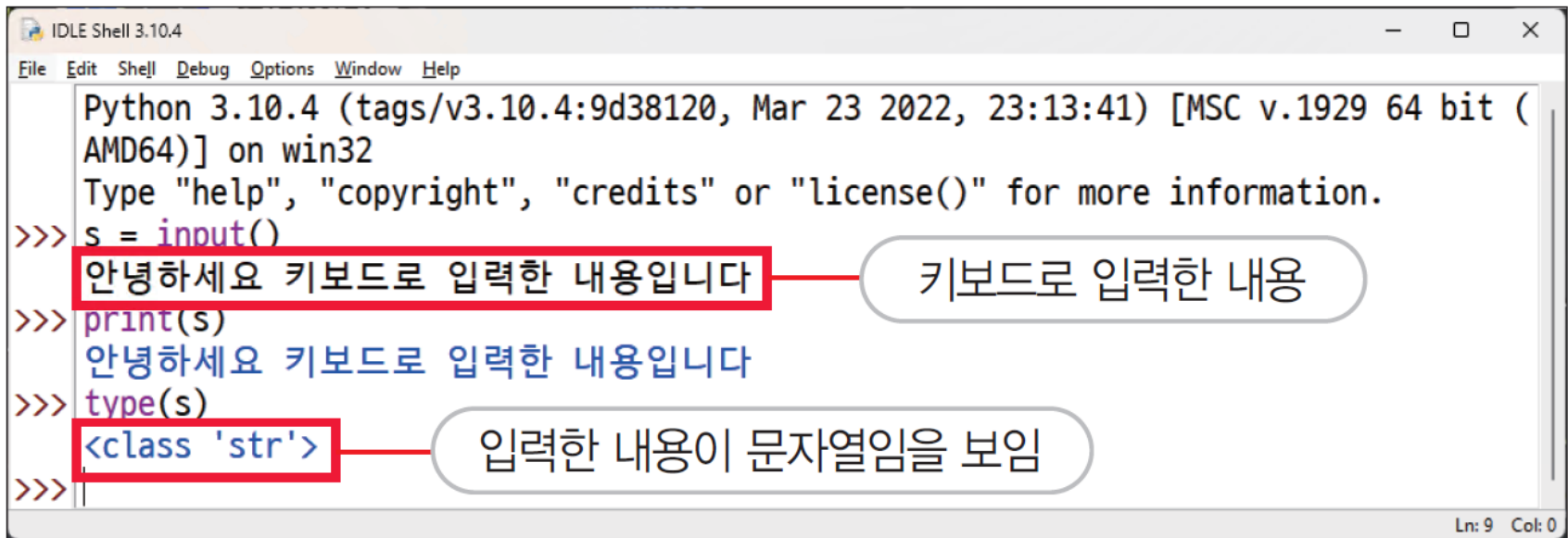
- 프롬프트 (prompt)는 화면에서 입력을 기다릴 때 깜빡 거리는 것을 의미함
- input() 명령은 실행되면 사용자에게 프롬프트를 보이고, 입력 받을 때까지 대기함



The screenshot shows the IDLE Shell 3.10.4 window. The title bar reads "IDLE Shell 3.10.4". The menu bar includes "File", "Edit", "Shell", "Debug", "Options", "Window", and "Help". The main text area displays the following text: "Python 3.10.4 (tags/v3.10.4:9d38120, Mar 23 2022, 23:13:41) [MSC v.1929 64 bit (AMD64)] on win32" followed by "Type 'help', 'copyright', 'credits' or 'license()' for more information." Below this, the prompt ">>>" is shown with a vertical cursor bar. A red square highlights the cursor bar, and a red line points from this square to a rounded rectangle containing the Korean text "프롬프트". The status bar at the bottom right indicates "Ln: 3 Col: 0".

# 키보드로 입력 받기

```
s = input()  
print(s)
```



The screenshot shows the IDLE Shell 3.10.4 interface. The Python version is 3.10.4 (tags/v3.10.4:9d38120, Mar 23 2022, 23:13:41) [MSC v.1929 64 bit (AMD64)] on win32. The user has entered the following commands and received the following output:

```
>>> s = input()  
안녕하세요 키보드로 입력한 내용입니다  
>>> print(s)  
안녕하세요 키보드로 입력한 내용입니다  
>>> type(s)  
<class 'str'>
```

Annotations in the image:

- A red box highlights the input string "안녕하세요 키보드로 입력한 내용입니다" in the first command, with a callout bubble pointing to it containing the text "키보드로 입력한 내용".
- A red box highlights the output "<class 'str'>" in the third command, with a callout bubble pointing to it containing the text "입력한 내용이 문자열임을 보임".

The status bar at the bottom right indicates "Ln: 9 Col: 0".

# 키보드로 입력 받기

- 앞에서 사용한 input() 명령은 프롬프트를 통해서 입력을 기다림을 사용자에게 알려줌
- 하지만 어떤 내용을 입력해야 하는지는 나타내지는 않음
- input() 명령의 또 다른 사용 방법: 무엇을 어떻게 입력해야 하는지 사용자에게 전달 가능

```
input([사용자에게 전달할 내용(문자열)])
```

- 사용 예

```
input("이름을 입력하세요: ")  
input("글자 한 개를 입력하세요: ")  
input("1~100 사이의 정수 한 개를 입력하세요: ")  
input("단어 한 개를 입력하세요: ")
```

# 키보드로 입력 받기

- input() 명령에 전달되는 문자열(사용자에게 보여질)은 파이썬에서 허용하는 문자열 상수 또는 문자열 변수는 사용 가능

```
input("글자 한 개를\n입력하세요: ")
input("1~100 사이의\n정수 한 개를\n입력하세요: ")
input('''단어
한 개를
입력하세요: ''')
msg = "단어 한 개를 입력하세요: "
input(msg)
```

# 키보드로 입력 받기

- 입력 받은 내용 저장 후 다시 사용하기
  - input() 명령은 사용자가 입력한 내용을 문자열 형태로 반환함
  - input() 명령이 반환하는 값을 변수에 대입 후 재사용

```
>>> name = input("이름을 입력하세요: ")
이름을 입력하세요: 파이썬
>>> print(name)
파이썬
```

# 키보드로 입력 받기

```
>>> input("글자 한 개를\n입력하세요: ") # 줄바꿈 문자 출력
글자 한 개를
입력하세요:
>>> input("1~100 사이의\n정수 한 개를\n입력하세요: ")
1~100 사이의
정수 한 개를
입력하세요:
>>> input('''단어 # 여러 줄에 표현되는 문자열 출력
한 개를
입력하세요: ''')
단어
한 개를
입력하세요:
>>> msg = "단어 한 개를 입력하세요: "
>>> input(msg)
단어 한 개를 입력하세요:
```

# 키보드로 입력 받기

- 사용자로부터 정수 두 개를 입력 받아서 합계를 구한 후에 출력하는 프로그램을 작성하라

- input() 명령은 문자열만 반환

```
num = input("정수 한 개를 입력하세요: ")  
print(num)
```

- 다른 방법은?

- 정수를 입력 받을 수 있는 명령을 찾아보기
- 숫자로 구성된 문자열을 정수로 변환할 수 있는 명령을 찾아보기



# 숫자 입력 받기

- input() 명령을 이용해서 정수나 실수를 입력 받고 싶다면, 입력 받은 내용을 int()나 float()로 변환해야 함

```
>>> numStr = input("1~100 사이의 정수 한 개를  
입력하세요: ") # (1)  
1~100 사이의 정수 한 개를 입력하세요: 100  
>>> type(numStr) # (2)  
<class 'str'>  
>>> num = int(numStr) # (3)  
>>> type(num) # (4)  
<class 'int'>
```

# 숫자 입력 받기

## ▣ 실수값 입력

```
>>> num = float(input("실수 한 개를 입력하세요.  
입력한 값에 1을 더한 후 출력합니다: ")) # (1)  
실수 한 개를 입력하세요. 입력한 값에 1을 더한 후  
출력합니다: 12.4  
>>> print(num + 1.0) # (2)  
13.4
```

## 실습문제 6

---

### □ 문제

- 사과 1kg의 단가와 사과의 무게를 사용자로부터 입력 받고, 총 사과 가격을 계산하여 화면에 출력하는 코드를 작성하세요.

### 요구사항

- 사과 1kg 의 단가는 원 단위의 정수로 입력 받기
- 무게 1kg는 실수로 입력 받기
- 문자열 출력은 f-문자열 사용

# 실습문제 7

---

## □ 문제

- 직사각형의 가로 길이와 세로 길이를 입력 받아 넓이를 계산하고 출력하는 코드를 작성하세요.

### 요구사항

- 변수 width , height 실수 값으로 입력 받기
- 계산값은 area 로 받고 출력
- 문자열 출력은 f-문자열 사용

# 실습문제 8

---

## □ 문제

- 섭씨 온도를 입력받아 화씨 온도로 변환하고 출력하는 코드를 작성하세요. (화씨 = 섭씨 \* 9/5 + 32)

### 요구사항

- 변수 입력 받기
- 계산값은 변수로 담아 출력
- 문자열 출력은 f-문자열 사용

# 실습문제 9

---

## □ 문제

- 직육면체의 가로, 세로, 높이를 입력 받아 부피를 계산하고 출력하는 코드를 작성하세요.
  - (가로\*세로\*높이)

## 요구사항

- 실수 3개 변수 입력 받기
- 계산값은 변수로 담아 출력
- 문자열 출력은 f-문자열 사용

# 실습문제 10

---

## □ 문제

- 세 개의 숫자를 입력받아 평균을 계산하고 출력하는 코드를 작성하세요.

### 요구사항

- 변수 3개 입력 받기
- 계산값은 변수로 담아 출력
- 문자열 출력은 f-문자열 사용

# 실습문제 11

---

## □ 문제

- 삼각형의 밑변과 높이를 입력받아 넓이를 계산하고 출력하는 코드를 작성하세요.

□ (삼각형의 넓이 = 밑변 \* 높이 / 2)

## 요구사항

- 변수 입력 받기
- 계산값은 변수로 담아 출력
- 문자열 출력은 f-문자열 사용



# 실습문제 12

---

## □ 문제

- 원금을 입력받고 연이율과 기간(년)을 입력받아, 단리 이자를 계산하여 총액을 출력하는 코드를 작성하세요.
  - ( 단리이자 = 원금 \* 연이율 \* 기간)

## 요구사항

- 변수 입력 받기
- 계산값은 변수로 담아 출력
- 문자열 출력은 f-문자열 사용

# 실습문제 13

---

## □ 문제

- 초 단위의 시간을 입력받아 시간, 분, 초로 변환하여 출력하는 코드를 작성하세요.
  - $1\text{hour} = 3600\text{sec}$

### 요구사항

- 변수 입력 받기
- 계산값은 변수로 담아 출력
- 문자열 출력은 f-문자열 사용

# 실습문제 14

---

## □ 문제

- 원화를 입력받아 달러,유로, 엔화로 변환하고 출력하는 코드를 작성하세요.
  - 환율은 사용자 입력 받으세요.

## 요구사항

- 변수 입력 받기
- 계산값은 변수로 담아 출력
- 문자열 출력은 f-문자열 사용

# Geek & geek

