



▶ 학습내용

01. 변수가 무엇인지 이해합니다.
02. 변수의 이름을 정하고 수나 문자열을 저장할 수 있습니다.
03. 입력과 출력의 방법을 살펴봅니다.
04. 변수 사용의 편리함을 알 수 있습니다.

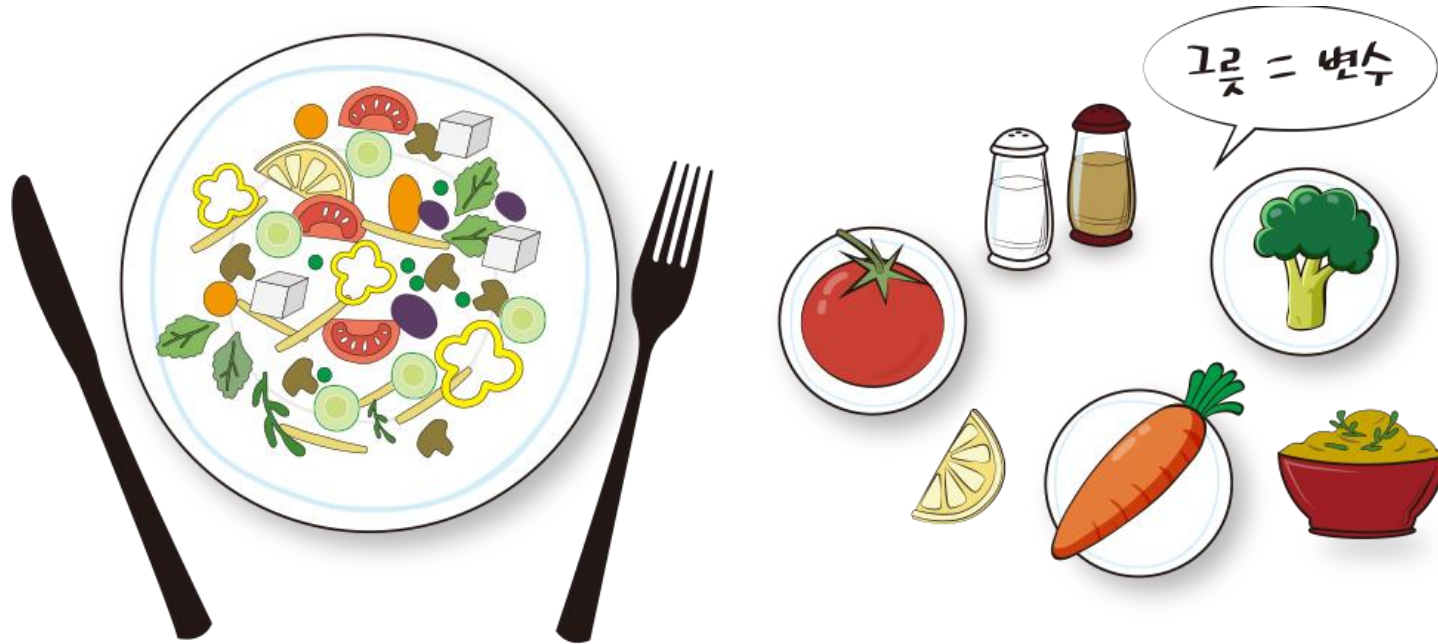
▶ LAB

01. 변수는 어디에 유용할까?
02. 내가 원하는 원 그리기
03. 천둥번개가 발생한 곳은 얼마나 떨어져 있나?

01. 변수가 무엇이죠?



- 변수 : 처리할 자료나 결괏값을 보관하는 공간
- 변수를 이용하여 컴퓨터 메모리에 이름을 붙이고 자료를 저장할 수 있음



01. 변수가 무엇이죠?



변수에 100을 저장하고 이를 출력하는 프로그램을 작성해보겠습니다.
변수에 100을 저장하는 것은 다음과 같습니다.

코드

```
>>> x = 100
```



코드

```
>>> print(x)
```

실행 결과

```
100
```

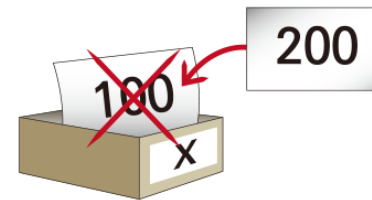
02. 변수는 수를 저장합니다.



변수에 100을 저장하였다가 그 변수에 다시 200을 저장하여 출력하는 프로그램을 작성해 보겠습니다.

코드 및 실행 결과

```
>>> x = 100  
>>> x = 200  
>>> print(x)  
200
```



02. 변수는 수를 저장합니다.



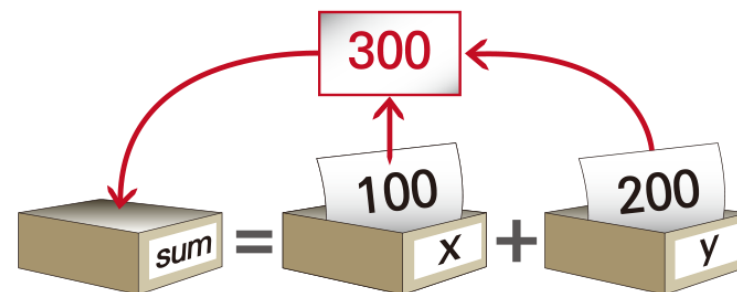
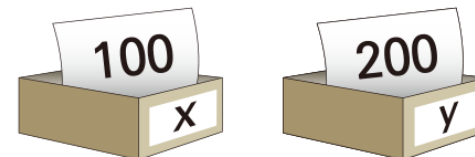
2개의 변수 x와 y를 생성하여 100과 200을 저장해 보겠습니다. 그리고 x와 y의 값을 더하여 sum에 저장하고 출력하는 프로그램을 작성해 보겠습니다.

코드

```
>>> x = 100  
>>> y = 200
```

코드 및 실행 결과

```
>>> sum = x + y  
>>> print(sum)  
300
```



03. '='는 특별해요



- 프로그램에서 '=': 대입을 뜻함

$$\begin{array}{ccc} x & = & y \\ \text{좌변} & = & \text{우변} \end{array}$$

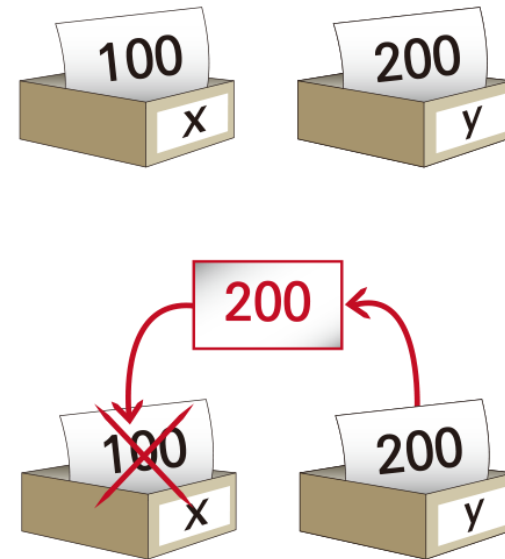
03. '='는 특별해요



$x = y$ 에서 우변의 값이 좌변에 저장됩니다.

코드 및 실행 결과

```
>>> x = 100
>>> y = 200
>>> x = y
>>> print("x = ", x)
x = 200
>>> print("y = ", y)
y = 200
```



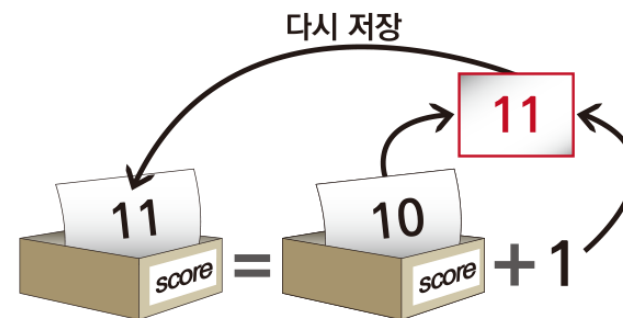
03. '='는 특별해요



'='를 이용하여 변수 자신의 값을 갱신할 수 있습니다.

코드 및 실행 결과

```
>>> score = 10  
>>> score = score + 1  
>>> print(score)  
11
```



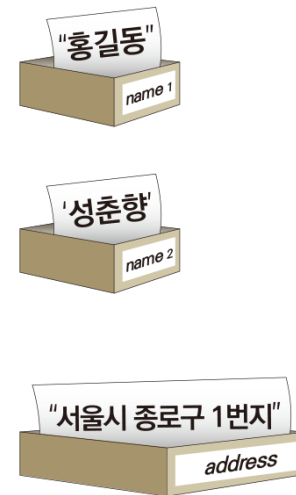
04. 변수는 문자열을 저장합니다



name과 address라는 변수에 우리의 이름과 주소를 저장하고 출력해 보겠습니다.

코드

```
>>> name1 = "홍길동"
>>> name2 = '성춘향'
>>> address = "서울시 종로구 1번지"
```



코드

```
>>> print(name1)
>>> print(name2)
```

실행 결과

홍길동
성춘향

코드

```
>>> print(address)
```

실행 결과

서울시 종로구 1번지

04. 변수는 문자열을 저장합니다



파이썬에서 따옴표가 있으면 문자열이고 따옴표가 없으면 숫자입니다. 그래서 아래의 코드를 통해 그 차이점을 살펴보겠습니다.

코드

```
>>> print("23" + "56")
```

실행 결과

2356

코드

```
>>> print(23 + 56)
```

실행 결과

79

05. 변수의 이름 짓기



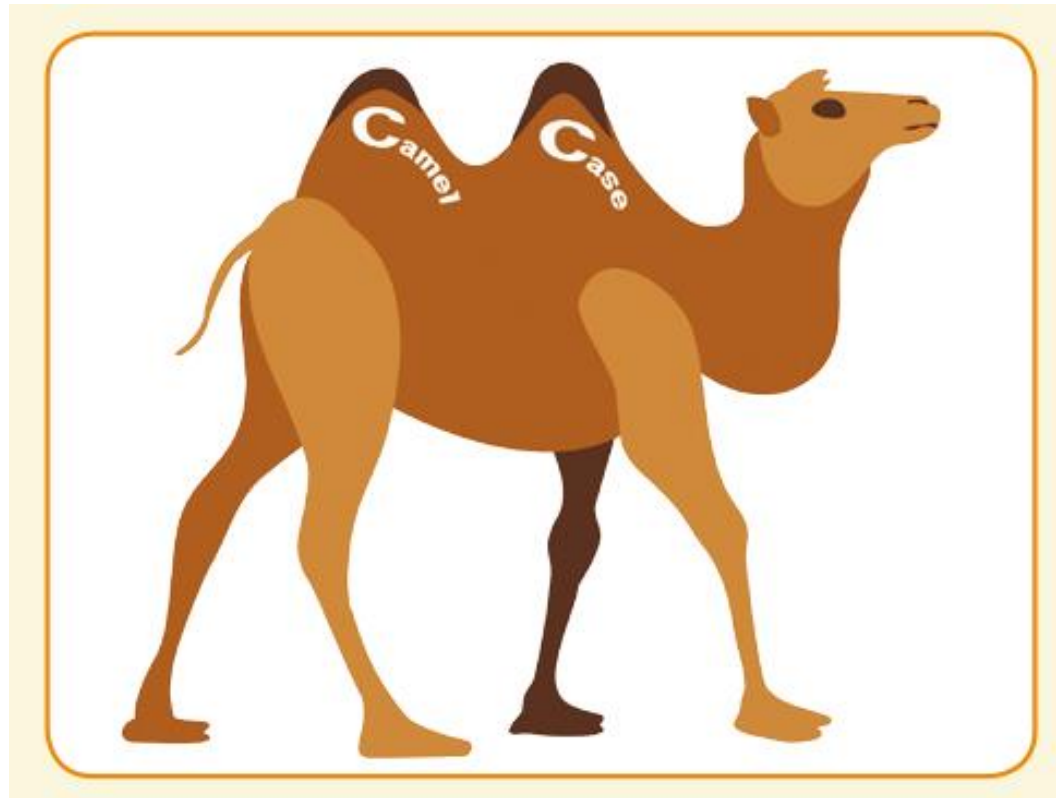
- 변수도 변수와 변수들을 구별하기 위해 이름이 필요
- 작명의 규칙
 1. 변수의 이름은 영문자와 숫자, 밑줄 문자(_)로 이루어집니다.
 2. 변수의 이름은 중간에 공백이 들어가면 안 됩니다.
 3. 변수의 이름은 숫자로 시작할 수 없습니다.
 4. 변수의 이름에 쓰이는 영문자는 대문자와 소문자를 구별합니다.

올바른 예	잘못된 예
sum _count number_of_pictures King3	2nd_base # 숫자로 시작할 수 없습니다. money# # #과 같은 기호는 사용할 수 없다.

05. 변수의 이름 짓기



- 변수의 이름을 지을 때는 변수의 역할을 가장 잘 설명하는 이름 사용



06. 한 번에 여러 값을 출력하기



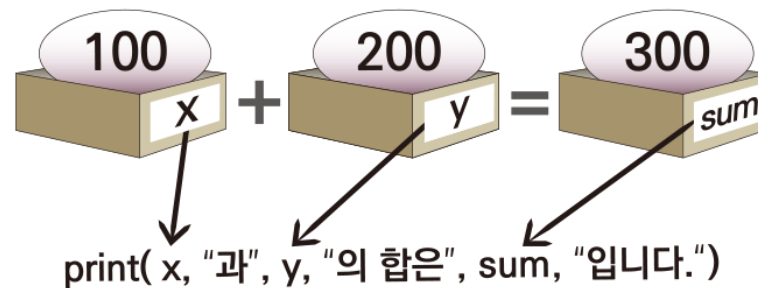
print()에서 쉼표(,)를 이용하여 화면에 여러 개의 값을 한 번에 출력하는 프로그램을 살펴보겠습니다.

코드

```
x = 100  
y = 200  
sum = x + y  
print(x, "과", y, "의 합은", sum, "입니다.")
```

실행 결과

100 과 200 의 합은 300 입니다.



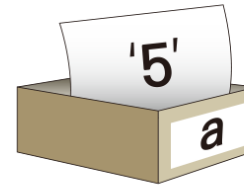
07. 문자열 입력받기



사용자로부터 숫자나 문자를 입력받아 출력해 보겠습니다.

코드 및 실행 결과

```
>>> a = input()  
5  
>>> a  
'5'
```



코드 및 실행 결과

```
>>> a = input("입력")  
입력 Hello  
>>> a  
'Hello'
```



07. 문자열 입력받기



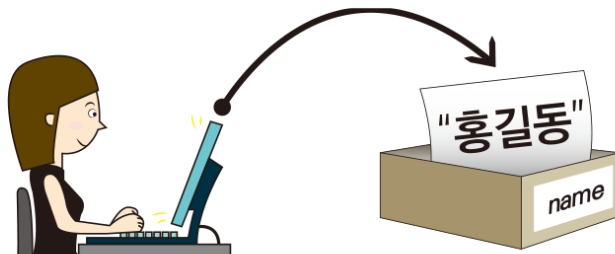
사용자로부터 이름을 입력받아 환영 메시지를 출력하는 프로그램을 작성해 보겠습니다.

코드

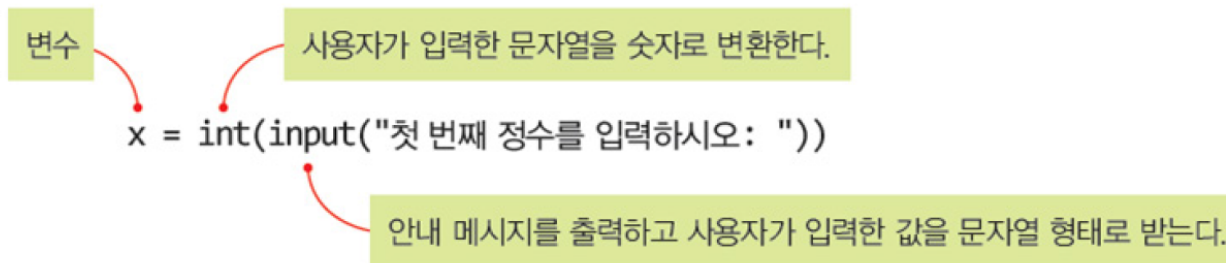
```
name = input("이름을 입력하시오: ")  
print(name, "씨, 안녕하세요?")  
print("파이썬에 오신 것을 환영합니다.")
```

실행 결과

이름을 입력하시오: 홍길동
홍길동 씨, 안녕하세요?
파이썬에 오신 것을 환영합니다.



08. 수 입력받기



앞의 프로그램에서는 항상 100과 200의 합만 출력하였는데, 사용자로부터 그때그때 원하는 정수를 2개 입력받아 그 합을 출력하는 프로그램을 작성해 보겠습니다. 더욱 유용한 덧셈 프로그램이 될 것입니다.

코드

```
x = int(input("첫 번째 정수를 입력하시오: "))  
y = int(input("두 번째 정수를 입력하시오: "))  
sum = x + y  
print(x, "과", y, "의 합은", sum, "입니다.")
```

실행 결과

첫 번째 정수를 입력하시오: 300
두 번째 정수를 입력하시오: 400
300 과 400 의 합은 700 입니다.

08. 수 입력받기



사용자로부터 2개의 정수를 받아서 사칙연산(+, -, *, /)의 결과를 출력하는 프로그램을 작성해 보겠습니다.

코드

```
x = int(input("첫 번째 정수를 입력하시오: "))  
y = int(input("두 번째 정수를 입력하시오: "))  
  
print(x, "+", y, "=", x + y)  
print(x, "-", y, "=", x - y)  
print(x, "*", y, "=", x * y)  
print(x, "/", y, "=", x / y)
```

실행 결과

```
첫 번째 정수를 입력하시오: 300  
두 번째 정수를 입력하시오: 400  
300 + 400 = 700  
300 - 400 = -100  
300 * 400 = 120000  
300 / 400 = 0.75
```

Lab. 변수는 어디에 유용할까?



- 변수를 이용하여 프로그램을 효과적으로 개선해 보세요.

코드

```
print("반지름이 10인 원의 넓이:", 10 * 10 * 3.14)  
print("반지름이 20인 원의 넓이:", 20 * 20 * 3.14)  
print("반지름이 30인 원의 넓이:", 30 * 30 * 3.14)  
print("반지름이 40인 원의 넓이:", 40 * 40 * 3.14)  
print("반지름이 50인 원의 넓이:", 50 * 50 * 3.14)
```

π
3.14159 ...
265358979323

Lab. 내가 원하는 원 그리기



- 사용자로부터 원의 반지름과 색깔을 입력받아 radius와 color 변수에 저장하고 해당 크기와 색깔의 원을 그려주는 프로그램을 작성해 보겠습니다(단, 반지름은 자연수만 입력받는다고 가정하겠습니다)

실행 결과: 셸 창

원의 반지름을 입력하시오: 100

원의 색깔을 입력하시오: blue

실행 결과: 캔버스 창



생각 1 : 프로그램의 순서를 생각해 봅시다.

캔버스에 터틀 불러오기



원의 크기 입력받기



원의 색깔 입력받기



원 그리기

Lab. 천둥번개가 발생한 곳은 얼마나 떨어져 있나?

21

- 번개가 치고 천둥소리가 들리기까지의 시간(초)을 입력하여 자신의 위치에서 천둥번개가 발생한 장소까지의 거리를 계산하는 프로그램을 작성해 봅시다

빛의 속도 = 약 30만 km/s
소리의 속도 = 약 340 m/s



실행 결과

측정 시간(초) 입력: 3

자신의 위치에서 번개가 친 장소까지의 거리= 1020 m

Lab. 천둥번개가 발생한 곳은 얼마나 떨어져 있나?

22



생각 1 : 프로그램의 순서를 생각해 봅니다.

측정한 시간 입력



거리 계산



계산 결과 출력