UNIT 6

변수와 입력 사용하기

6 변수와 입력 사용하기

변수와 입력 사용하기

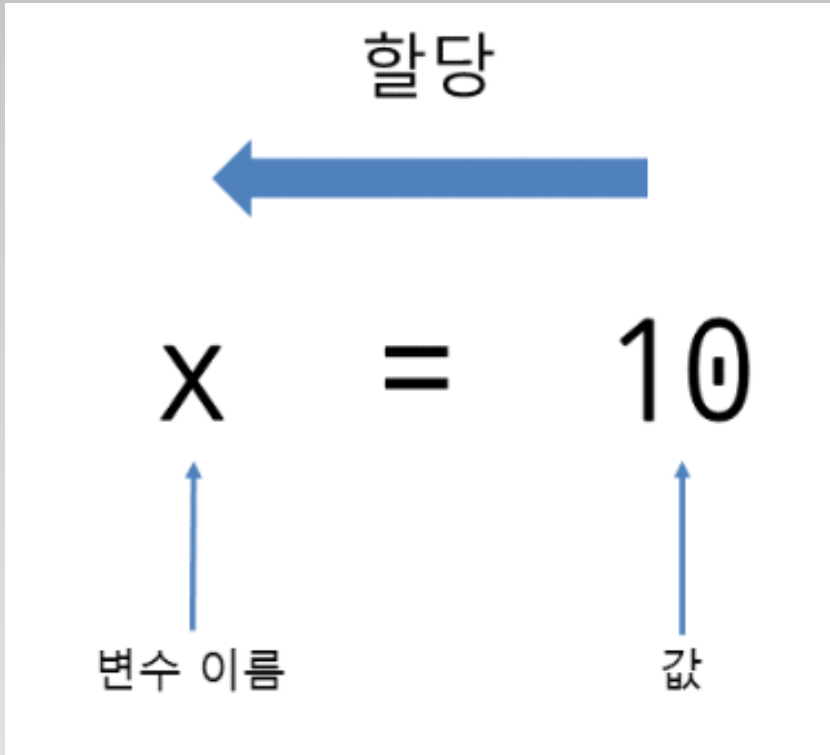
- 변수(variable)를 만들어 결과를 저장하는 방법을 알아보자

6.1 변수 만들기

» 변수 만들기

- 파이썬에서는 다음 그림과 같은 형식으로 코드를 입력하여 변수를 만듦

▼ 그림 변수 만들기



6.1 변수 만들기

» 변수 만들기

- x = 10이라고 입력하면 10이 들어있는 변수 x가 만들어짐
- 즉, 변수이름 = 값 형식임
- 변수가 생성되는 동시에 값이 할당(저장)됨
- 변수 이름은 원하는 대로 지으면 되지만 다음과 같은 규칙을 지켜야 함

- 영문 문자와 숫자를 사용할 수 있습니다.
- 대소문자를 구분합니다.
- 문자부터 시작해야 하며 숫자부터 시작하면 안 됩니다.
- _(밑줄 문자)로 시작할 수 있습니다.
- 특수 문자(+, -, *, /, $, @, &, % 등)는 사용할 수 없습니다.
- 파이썬의 키워드(if, for, while, and, or 등)는 사용할 수 없습니다.

6.1 변수 만들기

변수 만들기

- 파이썬 셸에서 변수를 만들어보자
- 〉〉〉에 다음 코드를 입력

```
>>> x = 10
>>> x
10
```

- 변수 x를 만들면서 10을 할당함
- 파이썬 셸에서는 변수를 입력한 뒤 엔터 키를 누르면 변수에 저장된 값이 출력됨
- 변수에는 숫자뿐만 아니라 문자열도 넣을 수 있음

```
>>> y = 'Hello, world!'
>>> y
'Hello, world!
```

- ' '(작은따옴표)로 묶은 문자열 Hello, world!를 변수 y에 할당함

6.1 변수 만들기

변수의 자료형 알아내기

- 파이썬에서는 변수의 자료형이 중요함
- type에 변수를 넣으면 변수(객체)의 자료형이 나옴

- type(변수)

```
>>> type(x)
<class 'int'>
>>> type(y)
<class 'str'>
```

- x에는 정수 10이 들어있으므로 int, y에는 문자열 Hello, world!가 들어있으므로 str이라고 나옴(int는 정수(integer), str은 문자열(string)에서 따옴)
- 변수의 자료형은 변수에 들어가는 값에 따라 달라짐
- 파이썬에서 변수를 사용하다 보면 자료형이 맞지 않아 발생하는 문제를 자주 접하게 됨
- type으로 자료형 뭔지 알아보면 문제를 쉽게 해결할 수 있음

6.1 변수 만들기

» 참고 | 기호는 같다는 뜻 아닌가요?

- 수학에서는 =(등호) 기호는 양 변이 같다는 뜻?
- 프로그래밍 언어에서 =는 변수에 값을 할당(assignment)한다는 의미
- 수학의 등호와 같은 역할을 하는 연산자는 ==임

6.1 변수 만들기

변수 여러 개를 한 번에 만들기

- 파이썬에서는 변수 여러 개를 한 번에 만들 수도 있음

```
>>> x, y, z = 10, 20, 30
>>> x
10
>>> y
20
>>> z
30
```

- 변수이름1, 변수이름2, 변수이름3 = 값1, 값2, 값3 형식으로 변수를 ,(콤마)로 구분한 뒤 각 변수에 할당될 값을 지정
- 변수와 값의 개수는 동일하게 맞춰주어야 하며 나열된 순서대로 값이 할당
- 변수와 값의 개수가 맞지 않으면 이렇게 에러가 발생

```
>>> x, y, z = 10, 20
Traceback (most recent call last):
  File "<pyshell#3>", line 1, in <module>
    x, y, z = 10, 20
ValueError: not enough values to unpack (expected 3, got 2)
```

6.1 변수 만들기

변수 여러 개를 한 번에 만들기

- 변수 여러 개를 만들 때 값이 모두 같아도 된다면 다음과 같은 방식도 사용할 수 있음

```
>>> x = y = z = 10
>>> x
10
>>> y
10
>>> z
10
>>>
```

- 변수 3개를 만들면서 모두 같은 값을 할당
- 변수1 = 변수2 = 변수3 = 값 형식으로 변수 여러 개를 =로 연결하고 마지막에 값을 할당해주면 같은 값을 가진 변수 3개가 만들어짐

6.1 변수 만들기

변수 여러 개를 한 번에 만들기

- 두 변수의 값을 바꾸려면 어떻게 해야 할까?
- 변수를 할당할 때 서로 자리를 바꿔주면 됨

```
>>> x, y = 10, 20
>>> x, y = y, x
>>> x
20
>>> y
10
```

- x는 20, y는 10이 나왔음
- 변수1, 변수2 = 변수2, 변수1 형식으로 두 변수의 값을 바꿀 수 있음

6.1 변수 만들기

» 참고 | 변수 삭제하기

- 변수 삭제는 del을 사용

del 변수

```
>>> x = 10
>>> del x
>>> x
Traceback (most recent call last):
  File "<pyshell#2>", line 1, in <module>
    x
NameError: name 'x' is not defined
>>>
```

- 변수 x를 삭제하여 변수가 없어졌으므로 x가 정의되지 않았다는 메시지와 함께 NameError가 발생
- 지금은 변수 삭제가 큰 의미가 없지만 나중에 리스트를 사용할 때 del이 유용하게 쓰임

6.1 변수 만들기

» 참고 | 빈 변수 만들기

- 변수를 만들 때 x = 10과 같이 할당할 값을 지정해줌
- 값이 들어있지 않는 변수는 만들 수 없을까?
- 값이 들어있지 않은 빈 변수를 만들때는 None을 할당해주면 됨

```
>>> x = None
>>> print(x)
None
>>> x
>>> (아무것도 출력되지 않음)
```

- print로 변수 x의 값을 출력해보면 None이 나옴
- 파이썬에서 None은 아무것도 없는 상태를 나타내는 자료형
- 보통 다른 언어에서는 널(null)이라고 표현함

6.2 변수로 계산하기

변수로 계산하기

- 변수를 활용하여 계산을 해보자

```
>>> a = 10
>>> b = 20
>>> c = a + b
>>> c
30
```

- 변수 a, b에 숫자를 할당한 뒤에 a와 b의 값을 더해서 변수 c에 할당함
- 변수는 변수끼리 계산할 수 있고, 계산 결과를 다른 변수에 할당할 수 있음

6.2 변수로 계산하기

산술 연산 후 할당 연산자 사용하기

- 변수 a의 값을 20 증가시키려면 어떻게 해야 할까?
- a + 20처럼 20을 더하면 30이 나오지만 a의 값을 다시 출력해보면 10이 나옴

```
>>> a = 10
>>> a + 20
30
>>> a
10
```

- a + 20은 a에 20을 더하기만 할 뿐 계산 결과를 유지하지 않음
- 변수 한 개에서 값의 변화를 계속 유지하려면 계산 결과를 다시 변수에 저장해야 함

```
>>> a = 10
>>> a = a + 20    # a와 20을 더한 후 결과를 다시 a에 저장
>>> a
30
```

6.2 변수로 계산하기

산술 연산 후 할당 연산자 사용하기

- a = a + 20과 같이 a에 20을 더한 값을 다시 a에 할당해주면 계산 결과가 계속 유지됨
- 파이썬에서는 변수를 두 번 입력하지 않도록 산술 연산 후 할당 연산자를 제공

```
>>> a = 10
>>> a += 20    # a와 20을 더한 후 결과를 다시 a에 저장
>>> a
30
```

- a에는 10이 들어있고 a += 20을 수행하면 a에는 10과 20을 더한 결과인 30이 들어감
- +=처럼 산술 연산자 앞에 =(할당 연산자)를 붙이면 연산 결과를 변수에 저장
  (+ =처럼 두 연산자를 공백으로 띄우면 안 됨)
- 덧셈(+=)만 해보았는데 뺄셈(-=), 곱셈(*=), 나눗셈(/=, //=), 나머지(%=)도 같은 방식
- 똑같이 연산(-, *, /, //) 후 할당(=) 한다는 뜻

6.2 변수로 계산하기

산술 연산 후 할당 연산자 사용하기

- 산술 연산 후 할당 연산자를 사용할 때는 주의할 점이 있음
- 만들지 않은 변수 d에 10을 더한 후 다시 할당하면 에러가 발생

```
>>> d += 10    # d는 만들지 않은 변수
Traceback (most recent call last):
  File "<pyshell#2>", line 1, in <module>
    d += 10
NameError: name 'd' is not defined
```

- 계산 결과를 d에 할당하기 전에 d와 10을 더해야 하는데 d라는 변수가 없어서 덧셈이 안 되기 때문임
- 연산 후 할당을 하려면 값이 들어있는 변수를 사용해야 함

6.2 변수로 계산하기

참고 | 부호 붙이기

- 계산을 하다 보면 부호를 붙여야 하는 경우도 생김
- 값이나 변수 앞에 양수, 음수 부호를 붙이면 됨

```
>>> x = -10
>>> +x
-10
>>> -x
10
```

- 양수, 음수 부호 붙이기는 수학 시간에 배운 내용과 같음
- -10에 + 부호를 붙이면 부호의 변화가 없고, -10에 - 부호를 붙이면 양수 10이 됨

6.3 입력 값을 변수에 저장하기

입력 값을 변수에 저장하기

- 지금까지 변수를 만들 때 10, 'Hello, world!' 등의 값을 직접 할당해주었음
- 이렇게 하면 고정된 값만 사용할 수 있음
- 매번 다른 값을 변수에 할당하려면 어떻게 해야 할까?

6.3 입력 값을 변수에 저장하기

input 함수 사용하기

- 이때는 input 함수를 사용하면 됨
  - `input()`
- >>>에 input()을 입력한 뒤 엔터 키를 누르면 다음 줄로 넘어감
- 이 상태에서 Hello, world!를 입력한 뒤 엔터 키를 누름

```
>>> input()
Hello, world! (입력)
'Hello, world!'
```

- 입력한 문자열이 그대로 출력됨
- input 함수는 사용자가 입력한 값을 가져오는 함수

6.3 입력 값을 변수에 저장하기

input 함수의 결과를 변수에 할당하기

- input 함수의 결과를 변수에 할당해보자
  - `변수 = input()`
- input 함수의 결과를 변수 x에 할당함
- Hello, world!를 입력한 뒤 엔터 키를 누름

```
>>> x = input()
Hello, world! (입력)
>>>
```

- 변수 x에 입력한 문자열이 저장됨
- x의 값을 출력해보면 방금 입력한 'Hello, world!'가 나옴

```
>>> x
'Hello, world!'
```

- 여기서 한 가지 불편한 점은 input 함수가 실행된 다음에는 아무 내용이 없어서 입력을 받는 상태인지 출력이 없는 상태인지 알 수가 없다는 점

6.3 입력 값을 변수에 저장하기

input 함수의 결과를 변수에 할당하기

- input의 괄호 안에 문자열을 지정해주면 됨
  - 변수 = input('문자열')

```
>>> x = input('문자열을 입력하세요: ')
문자열을 입력하세요: Hello, world! (입력)
>>> x
'Hello, world!'
```

- 실행을 해보면 '문자열을 입력하세요: '처럼 안내 문구가 먼저 나옴
- 문자열을 입력한 뒤 엔터 키를 누르면 입력한 그대로 출력
- 이 문자열은 스크립트 파일 사용자에게 입력받는 값의 용도를 미리 알려줄 때 사용
- 다른 말로는 프롬프트(prompt)라고도 부릅니다(파이썬 프롬프트 〉〉〉와 같은 의미

6.3 입력 값을 변수에 저장하기

두 숫자의 합 구하기

- 조금 응용해서 숫자 두 개를 입력받은 뒤에 두 숫자의 합을 구해보자
- IDLE의 소스 코드 편집 창에 입력

input_integer.py

```
a = input('첫 번째 숫자를 입력하세요: ')
b = input('두 번째 숫자를 입력하세요: ')

print(a + b)
```

- 소스 코드를 실행하면 '첫 번째 숫자를 입력하세요: '가 출력
- 10을 입력하고 엔터 키를 누름
- '두 번째 숫자를 입력하세요: '가 출력되면 20를 입력하고 엔터 키를 누름

실행 결과

```
첫 번째 숫자를 입력하세요: 10 (입력)
두 번째 숫자를 입력하세요: 20 (입력)
1020
```

6.3 입력 값을 변수에 저장하기

두 숫자의 합 구하기

- 10 + 20의 결과는 30이 나와야 하는데 1020이 나옴
- 이런 결과는 input에서 입력받은 값은 항상 문자열 형태이기 때문임
- 10과 20은 겉보기에는 숫자이지만 실제로는 문자열이므로 10과 20를 +로 연결하여 1020이 나오게 됨
- input의 결과를 변수에 저장한 뒤 type을 사용해보면 input의 결과가 문자열(str)이라는 것을 알 수 있음

```
>>> a = input()
10 (입력)
>>> type(a)
<class 'str'>
```

6.3 입력 값을 변수에 저장하기

» 입력 값을 정수로 변환하기

- 10 + 20의 결과가 30이 나오게 하려면 input에서 입력받은 문자열을 숫자(정수)로 만들어주어야 함
  - 변수 = int(input())
  - 변수 = int(input('문자열'))

input_integer.py

```
a = int(input('첫 번째 숫자를 입력하세요: '))    # int를 사용하여 입력 값을 정수로 변환
b = int(input('두 번째 숫자를 입력하세요: '))    # int를 사용하여 입력 값을 정수로 변환

print(a + b)
```

실행 결과

```
첫 번째 숫자를 입력하세요: 10 (입력)
두 번째 숫자를 입력하세요: 20 (입력)
30
```

- 입력 받은 값을 숫자(정수)로 만들려면 int에 input()을 넣어줌
- int는 정수로 된 문자열도 정수로 만들 수 있으므로 문자열 '10'은 정수 10으로 바뀜
- 3.5와 2.1처럼 실수를 더하려면 int 대신 float에 input()을 넣음
- 각자 소스 코드를 수정하여 실수의 합도 구해보자

6.4 입력 값을 변수 두 개에 저장하기

» 입력 값을 변수 두 개에 저장하기

- input 한 번에 값을 여러 개 입력 받으려면 어떻게 해야 할까?
- input에서 split을 사용한 변수 여러 개에 저장해주면 됨(각 변수는 콤마로 구분해줌)

```
• 변수1, 변수2 = input().split()
• 변수1, 변수2 = input().split('기준문자열')
• 변수1, 변수2 = input('문자열').split()
• 변수1, 변수2 = input('문자열').split('기준문자열')
```

- 문자열 두 개를 입력 받아 보자
- IDLE의 소스 코드 편집 창에 입력

input_split_string.py

```
a, b = input('문자열 두 개를 입력하세요: ').split()    # 입력받은 값을 공백을 기준으로 분리

print(a)
print(b)
```

6.4 입력 값을 변수 두 개에 저장하기

» 입력 값을 변수 두 개에 저장하기

- 소스 코드를 실행하면 '문자열 두 개를 입력하세요: '가 출력됨
- Hello Python을 입력하고 엔터 키를 누름

실행 결과

```
문자열 두 개를 입력하세요: Hello Python (입력)
Hello
Python
```

- input에 split을 사용하면 입력받은 값을 공백을 기준으로 분리하여 변수에 차례대로 저장 (split은 분리하다, 나누다라는 뜻)
- 문자열 'Hello Python'을 **공백**을 기준으로 분리하여 'Hello'는 첫 번째 변수 a에 'Python'은 두 번째 변수 b에 저장

```
a , b  = input('문자열 두 개를 입력하세요: ').split()
```

6.4 입력 값을 변수 두 개에 저장하기

» 두 숫자의 합 구하기

- 숫자 두 개를 입력 받아서 두 숫자의 합을 구해보자

input_split_int.py

```
a, b = input('숫자 두 개를 입력하세요: ').split()    # 입력받은 값을 공백을 기준으로 분리

print(a + b)
```

실행 결과

```
숫자 두 개를 입력하세요: 10 20 (입력)
1020
```

- 30이 나와야 하는데 1020이 나옴
- input에서 입력받은 값은 문자열이고, 이 문자열은 split으로 분리해도 문자열이기 때문임
- 문자열 '10 20'을 공백을 기준으로 분리하여 a에는 '10', b에는 '20'이 저장되므로 +로 연결하면 '1020'이 나옴

```
a , b  = input('숫자 두 개를 입력하세요: ').split()
```

6.4 입력 값을 변수 두 개에 저장하기

» 입력 값을 정수로 변환하기

- 10 + 20의 결과가 30이 나오게 하려면 변수 a와 b를 정수로 변환해주어야 함

input_split_int.py

```
a, b = input('숫자 두 개를 입력하세요: ').split()    # 입력받은 값을 공백을 기준으로 분리
a = int(a)    # 변수를 정수로 변환한 뒤 다시 저장
b = int(b)    # 변수를 정수로 변환한 뒤 다시 저장

print(a + b)
```

실행 결과

```
숫자 두 개를 입력하세요: 10 20 (입력)
30
```

- a = int(a)와 같이 int에 변수를 넣은 뒤 다시 변수에 저장해주면 변수가 정수 자료형으로 변환됨
- int(a)처럼 int만 사용하고 결과를 변수에 저장하지 않으면 정수로 변환되지 않음
- print 안에서 int로 변수를 변환하고 바로 더해도 상관없음

```
print(int(a) + int(b))
```

6.4 입력 값을 변수 두 개에 저장하기

map을 사용하여 정수로 변환하기

- map에 int와 input().split()을 넣으면 split의 결과를 모두 int로 변환해줌
  (실수로 변환할 때는 int 대신 float를 넣음)
  - 변수1, 변수2 = map(int, input().split())
  - 변수1, 변수2 = map(int, input().split('기준문자열'))
  - 변수1, 변수2 = map(int, input('문자열').split())
  - 변수1, 변수2 = map(int, input('문자열').split('기준문자열'))
- IDLE의 소스 코드 편집 창에 입력

map_input_split.py

```
a, b = map(int, input('숫자 두 개를 입력하세요: ').split())

print(a + b)
```

실행 결과

```
숫자 두 개를 입력하세요: 10 20 (입력)
30
```

- input().split()을 사용할 때 map을 사용하면 코드를 짧게 줄일 수 있음

6.4 입력 값을 변수 두 개에 저장하기

» 입력받은 값을 콤마를 기준으로 분리하기

- split에 기준 문자열을 지정하여 공백이 아닌 다른 문자로 분리해보자

map_input_split_comma.py

```
a, b = map(int, input('숫자 두 개를 입력하세요: ').split(',')) # 입력받은 값을 콤마를 기준으로 분리

print(a + b)
```

- 소스 코드를 실행한 뒤 '숫자 두 개를 입력하세요: '가 출력되면 10,20을 입력하고 엔터 키

실행 결과

```
숫자 두 개를 입력하세요: 10,20 (입력)
30
```

- split(',')과 같이 분리할 기준 문자열을 콤마로 지정하였으므로 '10,20'에서 10은 a, 20은 b에 저장
- 변수는 값이나 계산 결과를 저장할 때 사용한다는 점, 변수를 만드는 방법, 변수 이름을 짓는 방법만 기억하면 됨
- input과 split의 결과가 문자열라는 점이 중요
- 숫자 계산을 한다면 int, float를 사용해서 결과를 숫자로 변환해주어야 한다는 점
- split의 결과를 모두 int, float로 변환할 때는 map을 사용하면 편리함