변수, 자료형, 변환

오 동 인
경희대학교 의과대학 의공학교실

변수명 만들기

- 변수 (variable)
  - 컴퓨터 메모리 공간에 이름을 붙인 것
  - 프로그램에서 사용되는 수치나 문자 등의 값을 저장하기 위한 공간
  - 변수에 저장된 값을 사용하기 위해서는 프로그램 상에 변수의 이름을 적어주면 됨
    (변수의 참조)
  - 변수의 값은 프로그램을 시작할 때 초기화된 후, 실행 도중에 변경되거나 계산에 활용될 수 있음

- 변수명
  - 변수명은 식별자(identifier)의 일종
  - 파이썬의 변수명을 포함한 식별자 작성 규칙

    - **영문자, 숫자, 밑줄 문자(_)로 구성**
    - **중간에 공백이 사용 불가**
    - **첫 글자는 반드시 영문자/밑줄 문자(_)이며, 숫자로 시작할 수 없음**
    - **대문자와 소문자 구분**
    - **if, while, for 등의 예약어 사용 불가**

좋은 변수명

- 변수명의 올바른 예

varname
varname1
_varname
var_name
VarName
varfor

- 변수명의 틀린 예

varname$
var name
1varname
for

- 예약어 확인

```
import keyword
keyword.kwlist
```

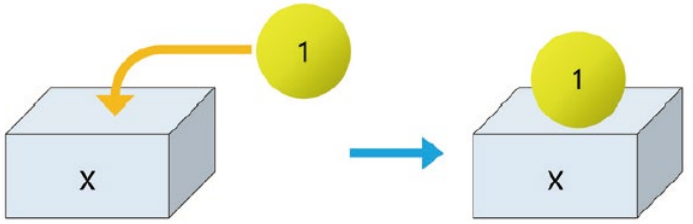
- 변수명의 역할
  - 변수의 역할을 잘 설명하는 이름으로 지어야 함
  - 잘 만들어진 변수 이름은 프로그램을 보다 더 읽기 편하고 이해하기 쉽게 해줌
  - (예) 년, 월, 일을 의미하는 변수를 만들 때 a, b, c 형태의 변수 이름 (X)
    start_year, start_month, start_day 형태의 변수 이름 (O)

https://www.python.org/dev/peps/pep-0008/

대입/할당

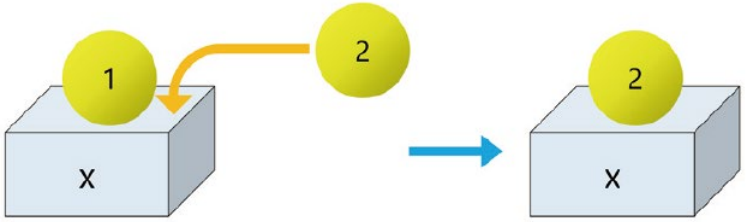
- 대입/할당(assign)
  - 변수에 값을 넣는 것
- 대입 연산자 =
  - 값을 대입하기 위한 연산자
- 대입문/할당문(assignment statement)
  - 대입 연산자를 사용한 문장
  - 변수 초기화(initialization) : 변수에 처음 값을 대입하는 것
  - 변수 x에 1을 대입하는 문장 : x = 1

```
x=1
x
print(x)
```



- 변수 값의 변경
  - 변수에 있는 값은 프로그램이 실행되는 도중에 변경 가능

```
x=1
x
x=2
x
```

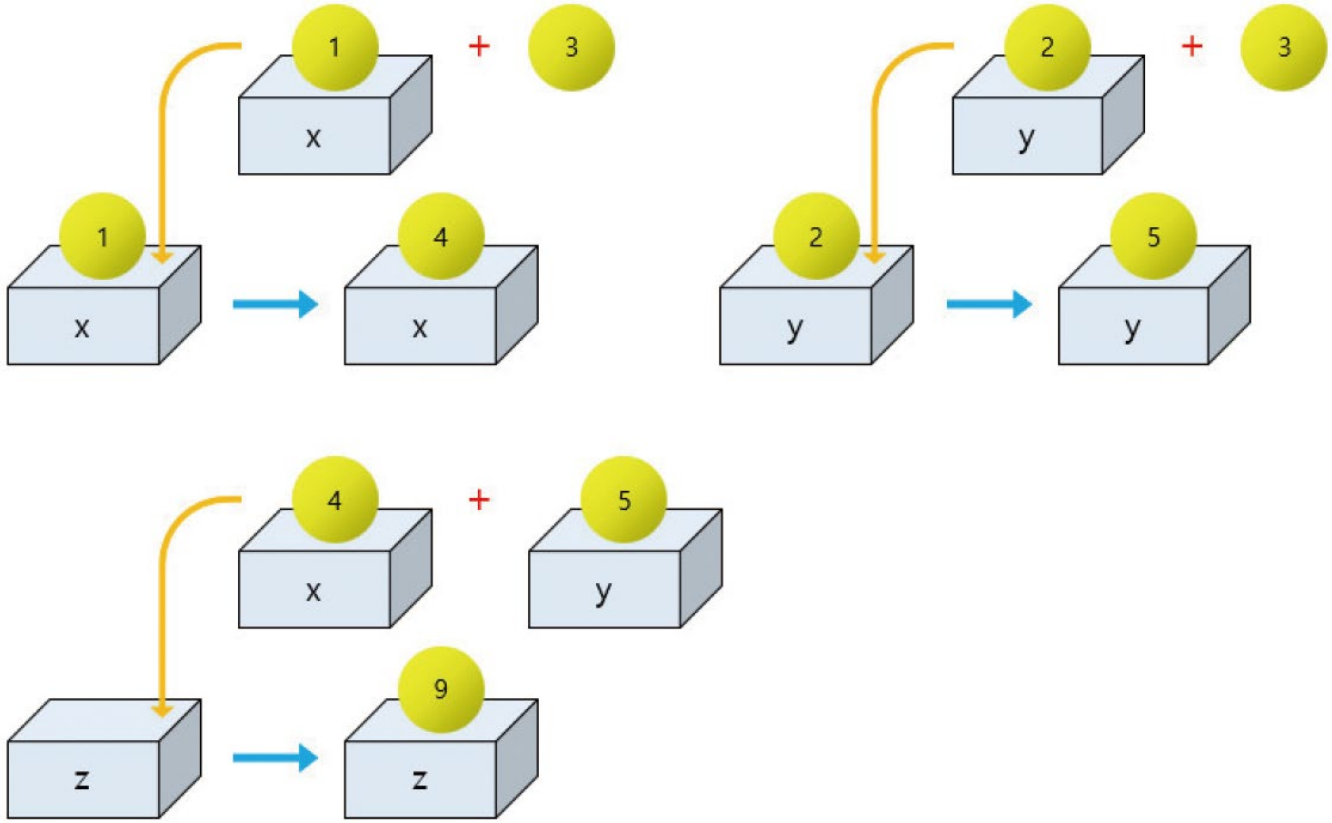


변수에 값 대입하기

- 변수 값의 수식에서의 활용
  - 변수의 값은 수식에서 계산에 활용될 수 있음
  - 계산된 결과 값은 다시 변수에 대입하여 저장할 수 있음

```
x=1
print(x)
y=2
print(y)
x=x+3
print(x)
y=y+3
print(y)
z=x+y
print(z)

del x
```



두 변수의 값 교체

- Ex) 변수 x에 3을 대입하고, 변수 y에 4를 대입한 후, 변수 t를 이용하여 x, y 값을 교체

```
# 두 변수 값 교체

x = 3
y = 4
print(x, y)

t = x
x = y
y = t
print(x, y)
```

```
# 잘못된 변수 교체

x = 3
y = 4
print(x, y)

x = y
print(x, y)

y = x
print(x, y)
```

```
# 다중 대입문 사용

x, y = 3, 4
print(x, y)
x, y = y, x
print(x, y)
```

- 다중 대입문 **(multiple assignment statement)**
  - 형식1 : a = b = c = 1 (여러 개의 변수에 같은 값을 순차적으로 대입)
  - 형식2 : a, b = 1, 2
    - = 양쪽에 여러 개의 변수, 여러 개의 수식을 한번에 기입
    - “,”로 구분하며 양쪽의 변수 및 표현의 개수는 동일해야 함
    - 형식2의 방식을 사용하여 두 변수의 값을 서로 바꿀 수 있음

변수 활용

```
# 변수의 소, 대문자 구분

varname = 1
VarName = 3
print(varname, VarName)
```

```
# 변수포함 계산식의 직접 적용

a = 3
b = 5
c = a * a + b * b
print(a, b, c)
```

```
# 문자열 포함 변수의 계산식 적용

s1 = "파이썬"
s2 = " "
s3 = "프로그래밍"
s4 = s1 + s2 + s3
print(s4)
```

사용자로부터 문자열 입력 받기

- 데이터 입력
  - 실행 도중 사용자로부터 값을 입력 받으며, 입력된 값은 변수에 저장된 후 계산 등에 사용됨
- 문자열 입력 받기
  - 문자열(string) : 문자들이 모인 것
  - Input() 함수
    - 키보드로 입력된 값을 문자열로 반환
    - Input() 함수의 인수는 사용자 입력을 돕기 위한 안내 문구 등을 표시하는 문자열 작성
- 문자열 입력 과정
  - Input() 함수 실행
    - ① "첫 번째 이름 : " 문자열 출력
    - ② 사용자로부터 입력을 기다림
    - ③ 사용자가 키보드로 값을 입력하고 Enter 키를 누름
    - ④ 입력한 값이 문자열로 반환하여 변수 name에 대입됨

```
name = input( " 첫 번째 이름 : ")
첫 번째 이름 : 홍길동
>>> name
'홍길동'
```

사용자로부터 문자열 입력 받기

- Ex) 사용자로부터 학교, 학과, 학번과 성명을 입력 받아 각각 변수 univ_name, dept_name, stud_num과 변수 name에 저장하고 해당 변수의 값 출력

```
# 학번, 이름 입력 받고 출력

univ_name = input("학교 : ")

dept_name = input("학과 : ")

stud_num = input("학번 : ")

name = input("이름 : ")

print("학교 :", univ_name, "학과 :", dept_name, "학번 :", stud_num, "이름 :", name)

type(stud_num)

type(name)
```

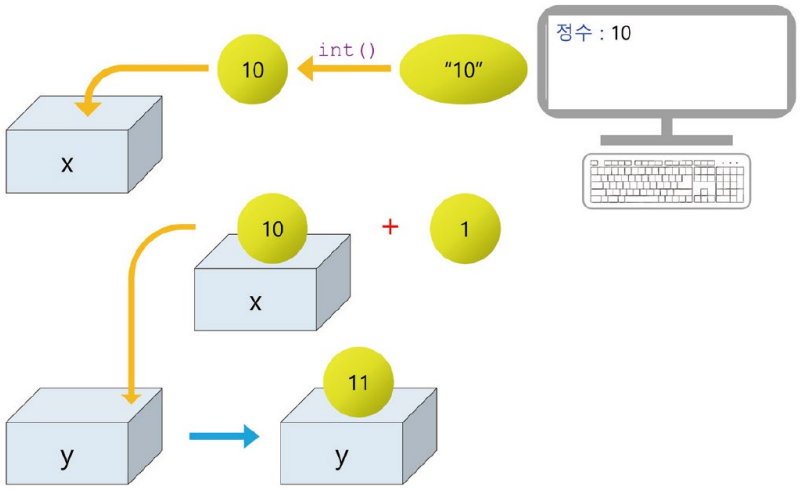
사용자로부터 정수 입력 받기

- 문자열의 산술 연산 오류
  - input() 함수는 사용자의 입력을 문자열로 반환하기 때문에 입력 받은 숫자 형태의 문자열에 산술 연산을 적용하면 오류가 발생

```
>>> x = input("정수 : ")
정수 : 10
>>> y = x + 1
Traceback (most recent call last):
  File "<pyshell#9>", line 1, in <module>
    y = x + 1
TypeError: can only concatenate str (not "int") to str
```

- 정수 입력
  - int(input()) 형식으로 정수 입력
  - int() 함수 : 문자열을 정수로 변환, 형 변환(type conversion)

```
>>> x = int(input("정수 : "))
정수 : 10
>>> y = x + 1
>>> y
11
```



사용자로부터 문자열/정수 입력 받기

- Ex) 사용자로부터 2개의 정수를 입력 받아 각각 변수 x와 y에 대입하고, print()함수를 이용하여 덧셈한 결과를 출력

```
>>> x = int(input("정수 1 : "))
정수 1 : 4
>>> y = int(input("정수 2 : "))
정수 2 : 2
>>> print(x + y)
6
```

```
>>> x = input("정수 : ")
정수 : 10a
>>> x = int(input("정수 : "))
정수 : 10a
```

```
# 이름을 문자열로 name에 대입하고, 출생년도를 정수로 입력 받아 year에 대입하여 출력

name = input("이름 : ")
year = int(input("출생년도 : "))
print("이름 :", name, "출생년도 :", year)
```

변수1, 변수2 = Input('문자열').split('기준문자열')
Ex) a, b = map(int, input('숫자 두 개를 입력하세요: ').split(','))

자료형의 종류

- 자료형(data type)
  - 프로그램에서 사용할 수 있는 자료의 종류
  - 변수에는 정수, 실수, 문자열 등의 다양한 값들을 대입하여 저장할 수 있음

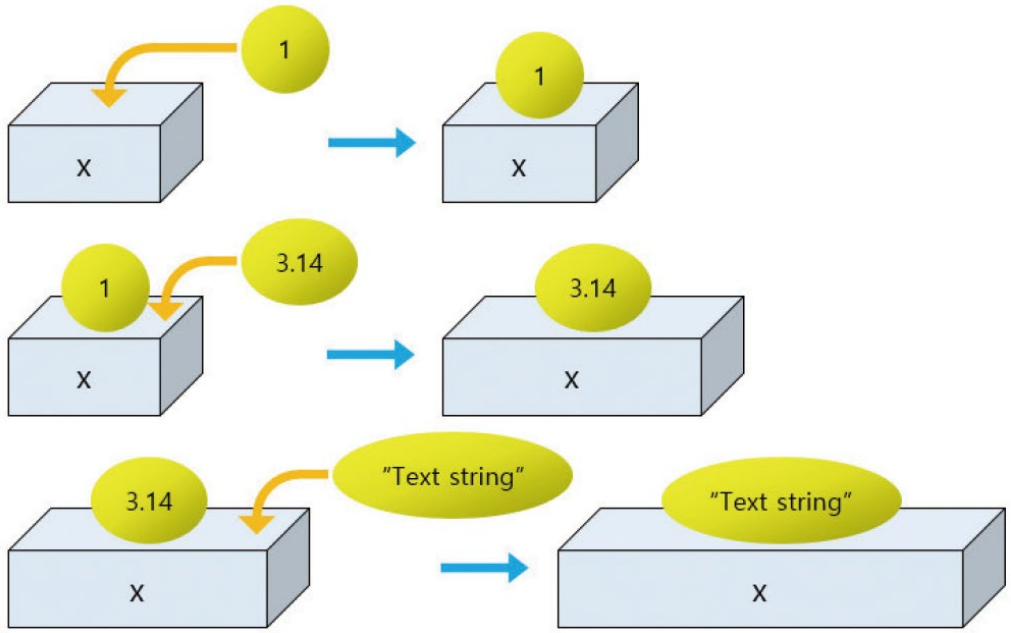
자료 유형	자료형	예
정수(integer)	int	-1, 0, 1, 2, 3
실수 (floating-point)	float	3.14, 0.12
문자열(string)	str	“Text”, ‘String’, “123”

- 대입에 의한 자료형 지정
  - 변수를 미리 선언하지 않아도, 변수에 대입된 값의 자료형에 따라 변수의 자료형이 자동적으로 정해짐

```
# 정수
x = 1

# 실수
x = 3.14

# 문자열
x = "Text string"
```



파이썬 자료형

- 정수, 실수, 문자열의 기본 자료형 이외 다음의 자료형 지원

- **부울(boolean)** : True 또는 False 의 값을 갖는 자료형
  - bool_data = True
- **리스트(list)** : 대괄호([]) 안에 임의의 객체를 순서 있게 나열한 자료형
  - list_data = [1, 2, 3]
- **튜플(tuple)** : 리스트와 비슷하지만 원소의 값을 변경할 수 없는 자료형
  - tuple_data = (1, 2, 3)
- **집합(set)** : 원소의 값들이 순서에 상관없이 모인 자료형
  - set_data = {2, 3, 1}
- **딕셔너리(dictionary)** : 중괄호({}) 안에 '키: 값'으로 된 쌍이 원소로 구성된 순서가 없는 자료형
  - dict_data = {0:False, 1:True}

자료형 지정과 확인

- Ex) 변수 x에 정수 값 1, 실수 값 3.14, 문자열 값 "Text string"을 순서대로 대입 저장 후, 변수 x의 값과 자료형을 출력하시오.

```
# 자료형 지정과 확인
>>> x = 1
>>> print(x, type(x))
1 <class 'int'>
>>> x = 3.14
>>> print(x, type(x))
3.14 <class 'float'>
>>> x = "Text string"
>>> print(x, type(x))
Text string <class 'str'>
```

```
# 다른 자료형
>>> bool_data = True
>>> print(bool_data, type(bool_data))
True <class 'bool'>
>>> list_data = [1, 2, 3]
>>> print(list_data, type(list_data))
[1, 2, 3] <class 'list'>
>>> tuple_data = (1, 2, 3)
>>> print(tuple_data, type(tuple_data))
(1, 2, 3) <class 'tuple'>
>>> set_data = {2, 3, 1}
>>> print(set_data, type(set_data))
{1, 2, 3} <class 'set'>
>>> dict_data = {0: False, 1: True}
>>> print(dict_data, type(dict_data))
{0: False, 1: True} <class 'dict'>
```

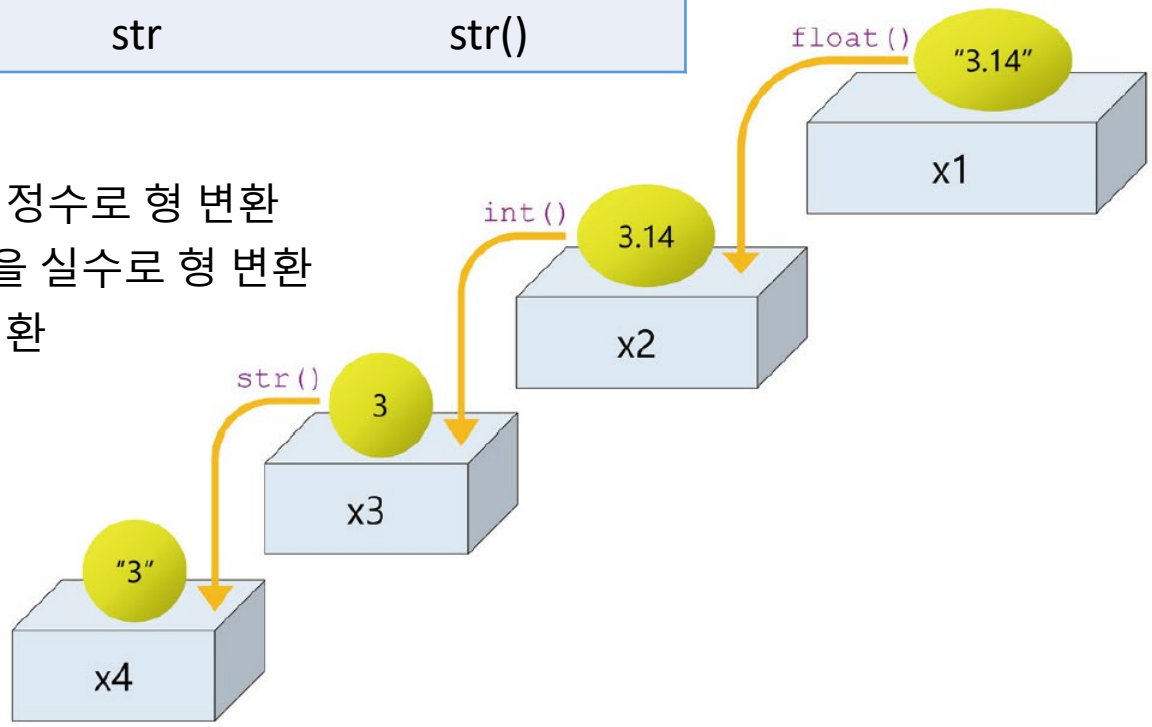
다른 자료형으로 변환하기

- 자료형의 변환
  - 변수에 저장된 정수, 실수, 문자열 등 다양한 자료형들의 값들은 사용자에 의해 강제로 다른 자료형으로 변환될 수 있음

자료 유형	자료형	형 변환 함수
정수(integer)	int	int()
실수 (floating-point)	float	float()
문자열(string)	str	str()

- 형 변환 함수의 변환 과정
  - int() 함수 : 정수 형태의 문자열/실수 값을 정수로 형 변환
  - float() 함수 : 실수 형태의 문자열/정수 값을 실수로 형 변환
  - str() 함수 : 실수/정수 값을 문자열로 형 변환

```
# 형 변환
x1 = "3.14"     # 문자열
x2 = float(x1)  # 실수
x3 = int(x2)    # 정수
x4 = str(x3)    # 문자열
```



문자열 입력 받아 실수형, 정수형 변환

- Ex) 사용자로부터 실수 형태의 문자열을 입력 받아 실수형으로 변환, 출력하고, 정수형, 문자열로 연속 변환하여 각각 변수의 값과 자료형을 함께 출력하시오.

```
 # 문자열 입력 후 실수, 정수, 문자열 변환
>>> text = input("실수 형태의 문자열 : ")
실수 형태의 문자열 : 3.14
>>> fnum = float(text)
>>> inum = int(fnum)
>>> tstr = str(inum)
>>> print(text, type(text), fnum, type(fnum), inum, type(inum), tstr, type(tstr))
3.14 <class 'str'> 3.14 <class 'float'> 3 <class 'int'> 3 <class 'str'>
```

```
x = 3.14
y = int(x)
z = float(y)
print (x, y, z)
```

```
inum = 3
fnum = float(inum)
print(inum, fnum)
```

```
fnum = 3.14
inum = int(fnum)
print(fnum, inum)
```

HW 1

- "이름, 학과, 장래 희망, 출생년도, 생일(월), 생일(일)" 의 여러 개 항목을 동시에 입력 받도록 프로그램 하시오.
- 만 30세 생일이 지난 오늘의 날짜에 당신의 장래 희망이 이루어졌다는 축하메시지를 출력하시오.
- "이름_제출날짜_HW1.py" 파일을 e-campus 에 제출하세요.

  (예시: 홍길동_20210914_HW1.py)

예시)

>>> 당신의 이름, 학과, 장래 희망, 출생년도, 생일(월), 생일(일) 을 ',' 로 구분하여 입력하시오: 홍길동, 의예과, 의사, 2001, 12, 1

의예과를 졸업하신 홍길동님은 2032년 09월 14일 당신의 희망이셨던 의사가 되셨습니다. 축하 드립니다.

EOD