# 데이터 처리를 위한 Python 프로그래밍 입문

2-2강. 변수

**ERICA 2018-2** 

#### 강의 내용

- ▶ 변수
- ▶ 변수 할당
- ▶ 변수 삭제
- ▶ 변수명
- ▶ 변수 재지정
- ▶ 복합 대입 연산자

#### 변수

#### ▶ 변수

- ▶ 변할 수 있는 수
- 한 숫자를 저장하거나 여러 가지 연산 결과를 보관했다가 다시 찾기 위한 정보 저 장 공간 : 정보 저장소 역할
- ▶ 내용을 몇 번이고 썼다 지웠다 할 수 있는 메모지
- ▶ 한 개 프로그램에 여러 개의 변수를 사용함 변수명 지정 유의
- ▶ 변수에 저장 내용
  - ▶ 숫자만이 아니다.
  - ▶ 변할 수 있는 정보를 저장하는 곳 : 숫자, 문자열, 함수명 등

## 변수 할당(1/2)

- ▶ 할당문 연산자
  - ▶ 기호: =

```
>>>
>>> number_1=26
>>> number_2=3*4*9
>>>
>>>
>>>
>>>
1
26
>>> number_1
26
>>> number_2
108
>>>
```

#### 변수 할당(2/2)

- ▶ 할당되지 않는 변수
  - ▶ 할당되지 않은 변수는 사용할 수 없음
  - ▶ 에러문 인지

```
>>>
>>>
number
Traceback (most recent call last):
   File "<pyshell#11>", line 1, in <module>
     number
NameError: name 'number' is not defined
>>>
>>> number = 6
>>> number
6
>>>
```

#### 변수 삭제

- ▶ 예약어 del()
  - ▶ 한 개 변수 삭제

▶ 여러 변수 동시 삭제

```
>>>
>>> num_1 = 6
>>> num_1
6
>>> del(num_1)
>>> num_1
Traceback (most recent call last):
   File "<pyshell#135>", line 1, in <module>
        num_1
NameError: name 'num_1' is not defined
>>>
```

```
>>>
>>> num_3, num_4
(4, 6)
>>>
>>> del(num_3, num_4)
>>> num_3, num_4
Traceback (most recent call last):
   File "<pyshell#100>", line 1, in <module>
        num_3, num_4
NameError: name 'num_3' is not defined
>>>
>>>
>>>
```

#### 변수명(1/4)

- ▶ 변수명 지정 유의
  - ▶ 변수의 이름은 문자, 숫자, 그리고 \_(Underscore)만 포함해야 함
    - ▶ \$같은 특수문자 사용 불가
  - ▶ 변수명은 문자 또는 \_(Underscore)로만 시작할 수 있음
    - ▶ 숫자로 시작하는 변수명 사용 불가
  - ▶ Python은 대문자와 소문자를 구분함
    - ▶ Score와 score는 다른 변수명이다

#### 변수명(2/4)

- ▶ 변수명 지정 유의(계속)
  - ▶ Python에서 다른 목적을 위해 미리 지정한 단어들은 사용할 수 없음
    - ▶ 미리 지정된 예약어는 변수명으로 사용 불가
    - ▶ 예약어 확인

```
>>> import keyword
>>> keyword.kwlist
['False', 'None', 'True', 'and', 'as', 'assert', 'break', 'class', 'continue', 'def', 'del', 'elif', 'else', 'except', 'finally', 'for', 'from', 'global', 'if', 'import', 'in', 'is', 'lambda', 'nonlocal', 'not', 'or', 'pass', 'raise', 'return', 'try', 'while', 'with', 'yield']
>>> |
```

## 변수명(3/4)

▶ 변수명 지정 유의(계속)

```
>>>
>>> Coffee$ = 3
SyntaxError: invalid syntax
>>>
>>> 79frend = 9
SyntaxError: invalid syntax
>>>
>>> False = 9
SyntaxError: can't assign to keyword
>>>
>>>
>>>
```

#### 변수명(4/4)

▶ 의미있는 변수명 지정

```
>>> a=80
>>> b=96
>>> c=88
>>> d=a+b+c
>>> e=d/3
>>> e
88.0
>>>
```

```
>>> math=80
>>> english=96
>>> history=88
>>> sum=math+english+history
>>> average=sum/3
>>> average
88.0
>>>
```

#### 여러 변수 할당

▶ 한 문장에서 여러개의 변수를 할당 가능

```
>>>
>>> num_1, num_2 = 69
Traceback (most recent call last):
 File "<pyshell#70>", line 1, in <module>
    num_1, num_2 = 69
TypeError: 'int' object is not iterable
>>>
>>> num_1, num_2 = 6.9
>>> num_1
>>> num_2
>>>
>>> num_3, num_4 = 6.9.4
Traceback (most recent call last):
  File "<pyshell#76>", line 1, in <module>
   num_3, num_4 = 6,9,4
ValueError: too many values to unpack (expected 2)
l>>>
```

#### 변수 재지정(1/2)

- ▶ 변수 재지정(재할당)
  - ▶ a라는 변수에 숫자 3을 할당한 후
  - ▶ a라는 변수에 "Python 완전정복" 문자열 재할당
  - ▶ 재 할당시 어떠한 알림메시지도 없음을 유의

```
>>> a = 3
>>> a
>>> a
3
>>>
>>> a = "Python 완정정복"
>>> a
'Python 완정정복'
>>>
```

#### 변수 재지정(2/2)

▶ 변수 할당시 수식 활용

```
>>> num_1 = 3
>>> num_2 = num_1 + 3
>>>
>>> num_2
>>> num_1
3
>>> num_2
6
```

▶ = 의 우측 값이 좌측 변수에 할당됨

```
>>> num_1
3
>>> num_2
6
>>> num_1 = num_2
>>> num_2
6
>>> num_1
6
>>> num_1
6
>>>
```

## 변수 연산

▶ 숫자, 변수 혼용 연산가능

```
| >>> num_1
6
| >>> num_2
| 3
| >>> 4*num_1+num_2
| 27
| >>>
```

## 복합 대입 연산자(1/2)

▶ 기호별 기능

복합 대입 연산자	내 용
a += 1	a = a + 1
a -= 1	a= a - 1
a *= 1	a= a * 1
a /= 1	a= a / 1
a //= 1	a= a // 1
a %= 1	a= a % 1

## 복합 대입 연산자(2/2)

▶ 기호별 기능 활용

```
>>> cou = 0
>>> cou = 0
>>> cou +=1
>>> cou +=1
>>> cou *=3
>>> cou 3
>>>
```

```
>>> n_3 = 3

>>> n_3 *= 6

>>> n_3 = 6

>>> n_3 = 4

>>> n_3 = 4

>>> n_3 //=4

>>> n_3 //=4

>>> n_3 %=2

>>> n_3 1

>>> n_3 %=2
```

#### Thank you