데이터 처리를 위한 Python 프로그래밍 입문

3-1강. 데이터 타입

ERICA 2018-2

강의 내용

- 정수와 실수
- 문자열
  - 문자열 기호
  - 문자열 길이
  - 문자열 일부분 추출
- 불리언

정수와 실수(1/3)

- 연산에 사용되는 숫자
  - 자연수, 정수, 실수 등
- Python 제공 숫자 형태
  - 정수형, 실수형
- 정수형 : 소수점이 없는 수로써 양수와 음수를 포함
  - 1
  - int형
- 실수형 : 소수점을 포함하는 수
  - 2.0
  - float형

정수와 실수(2/3)

- 데이터 타입 확인 방법
  - type()

```
>>>
>>> type(9)
<class 'int'>
>>> type(9.3)
<class 'float'>
>>>
>>>
>>>
>>> n=3
>>> type(n)
<class 'int'>
>>>
```

Python은 대소문자를 구분함으로
함수명 입력시 대소문자 유의해야 합니다.

```
>>>
>>> Type(9)
Traceback (most recent call last):
  File "<pyshell#200>", line 1, in <module>
    Type(9)
NameError: name 'Type' is not defined
>>>
```

정수와 실수(3/3)

- 연산 결과 데이터 타입
  - int형을 이용한 연산 결과는 int형
  - float을 이용한 연산 결과는 float 형
  - 그러나 나눗셈 연산은 예외

```
>>>
>>> type(4*2)
<class 'int'>
>>> type(4.0*2.0)
<class 'float'>
>>>
>>> type(4/2)
<class 'float'>
>>> 4/2
2.0
>>>
```

문자열 기호(1/2)

- Python에서 문자 표현
  - 따옴표 Quote(‘ ‘)를 이용
  - String형

```
>>>
>>> a = "Python"
>>> type(a)
<class 'str'>
>>>
```

문자열 기호(2/2)

- 문자열 기호
  - 문자열 시작과 끝에 기호 “a” , ‘b’, “””c”””, ‘’’d‘’’를 쌍으로 구분(**1, 3개**)
  - **그러나 “”e””, ‘’f’’는 사용 못함(2개)**

```
>>> print("a")
a
>>> print('b')
b
>>> print("""c""")
c
>>> print('''d''')
d
>>> print(""e"")
SyntaxError: invalid syntax
>>> print(''f'')
SyntaxError: invalid syntax
>>>
>>>
>>>
```

문자열 시작과 끝은 쌍으로 이뤄져야 함.
'시작하고 "으로 끝내지 못함, Error발생

```
'''
>>> print('Python")

SyntaxError: EOL while s
>>>
```

문자열 길이(1/3)

- 문자열 길이
  - len()
  - 띄어쓰기 부분도 하나의 문자열로 판단

```
>>>
>>>
>>> length = len("I like Python")
>>> length
13
>>>
```

문자열 길이(1/3)

- 문자열 길이
  - 한글, 영어 문자열 길이 확인

```
>>>
>>> str_1 = "안녕"
>>> str_2 = "Python"
>>> len(str_1)+len(str_2)
8
>>>
```

문자열 길이(2/3)

- 문자열의 문자 순번 부여(indexing)

```
>>>
>>> A = "Python"
>>> num_1 = A[0]
>>> num_2 = A[1]
>>> num_3 = A[2]
>>> num_1, num_2, num_3
('P', 'y', 't')
>>>
```

문자열 길이(2/3)

- 문자열의 문자 순번 부여(indexing)

Python은 대소문자를 구분함으로
변수명 입력시 대소문자 유의하여야 합니다.

```
>>>
>>> A[0]
'P'
>>> a[0]
Traceback (most recent call last):
  File "<pyshell#253>", line 1, in <module>
    a[0]
NameError: name 'a' is not defined
>>>
```

```
>>> a = "Life is too short, You need Python"
>>> a[0]
 'L'
>>> a[12]
's‘
 >>> a[-1]
 'n'
>>> a[-0]
>>> a[-2]
```

```
>>> a[-0]
 'L'
>>> a[-2]
 'o'
```

문자열 길이(3/3)

- 문자열의 일부 내용 변경 불가

```
>>>
>>> b = "Hello, Python!"
>>> b[0]
'H'
>>> b[0] = J
Traceback (most recent call last):
  File "<pyshell#259>", line 1, in <module>
    b[0] = J
NameError: name 'J' is not defined
>>>
>>>
>>> b[6]
' '
>>> b[13]
'!'
>>> b[14]
Traceback (most recent call last):
  File "<pyshell#264>", line 1, in <module>
    b[14]
```

문자열은 띄어쓰기, 특수문자를 인식하며,
없는 문자열 index불가 Error메세지 확인바랍니다.

문자열 일부분 추출(1/2)-슬라이싱

- 문자열 일부분 추출
  - [n:m]
  - m은 포함하지 않음

```
>>>
>>> str = "Hello"
>>>
>>> str[1:3]
'el'
>>>
```

a = "Life is too short, You need Python"

```
>>> a[0:2]
>>> a[5:7]
```

```
>>> a[19:]
>>> a[:17]
>>> a[:]
```

```
>>> a[19:-7]
```

'You need'

```
>>>
>>> str_1= 'red apple'
>>> str_2= " yellow banana"
>>> color_1 = str_1[:4]
>>> fruit_1 = str_1[4:]
>>> color_2 = str_2[:7]
>>> fruit_2 = str_2[8:]
>>> color_1 + fruit_1
'red apple'
>>> color_2 + fruit_2
' yellowbanana'
>>>
```

문제) 세부분으로 분리해보세요

```
>>> a = "20010331Rainy"
>>> year =
>>> day =
>>> weather =
>>> year
 '2001'
>>> day
'0331'
>>> weather
'Rainy'
```

문자열나누기

```
>>> a = "20010331Rainy"
>>> date = a[:8]
>>> weather = a[8:]
>>> date
'20010331'
>>> weather
'Rainy'
```

문자열 포매팅

- 문자열 내의 특정한 값을 바꿔야 할 경우가 있을 때 이것을 가능하게 해주는 것이 바로 문자열 포매팅 기법이다

"현재 온도는 20도 입니다."

"현재 온도는 25도입니다"

문자열 포맷코드

코드	설명
%s	문자열 (String)
%c	문자 1개(character)
%d	정수 (Integer)
%f	부동소수 (floating-point)
%o	8진수
%x	16진수

1) 숫자 바로 대입

```
>>> "I eat %d apples."  % 3
'I eat 3 apples
```

2) 문자열 바로 대입

```
>>> "I eat  %s  apples."  %  'five'
'I eat five apples.'
```

3) 숫자 값을 나타내는 변수로 대입

```
>>> number = 3
>>> "I eat %d apples." % number
'I eat 3 apples.'
```

4) 2개 이상의 값 넣기

```
>>> number = 10
>>> day = "three"
>>> "I ate %d apples. so I was sick for %s days." % (number, day)
'I ate 10 apples. so I was sick for three days.'
```

```
>>> "Error is %d%%." % 98
'Error is 98%.'
```

포맷 코드와 숫자 함께 사용하기

1) 정렬과 공백

```
>>> "%10s" % "hi"
'        hi'

>>> "%-10sjane." % 'hi'
'hi        jane.'
```

2) 소수점 표현하기

```
>>> "%0.4f" % 3.42134234
'3.4213'

>>> "%10.4f" % 3.42134234
'    3.4213'
```

불리언

- boolean형
  - True : 참(맞음, 같음)
  - False : 거짓(틀림, 다름)

```
>>>
>>> a = True
>>> type(a)
<class 'bool'>
>>>
```

대소문자 유의하여야 합니다.

```
>>>
>>> b = TRUE
Traceback (most recent call last):
  File "<pyshell#6>", line 1, in <module>
    b = TRUE
NameError: name 'TRUE' is not defined
>>> c = true
Traceback (most recent call last):
  File "<pyshell#7>", line 1, in <module>
    c = true
NameError: name 'true' is not defined
>>>
```

문자관련함수들

문자 개수 세기(count)

```
>>> a = "hobby"
>>> a.count('b')
2
```

문자열 중 문자 b의 개수를 반환한다

위치 알려주기1(find)

```
>>> a = "Python is best choice"
>>> a.find('b') 10
>>> a.find('k') -1
```

b가 처음으로 나온 위치를 반환한다

문자나 문자열이 존재하지 않는다면 -1을 반환한다

문자관련함수들

위치 알려주기**2(index)**

```
>>> a = "Life is too short"
>>> a.index('t')
 8
>>> a.index('k')
Traceback (most recent call last): File "<stdin>", line 1,
in <module> ValueError: substring not found
```

문자 t가 맨 처음으로 나온 위치를 반환한다

find 함수와 다른 점은 문자열 안에 존재하지 않는 문자를 찾으면 오류가 발생한다는 점이다

문자관련함수들

문자열 삽입(join)

```
>>> a= ","
>>> a.join('abcd')
'a,b,c,d'
```

문자관련함수들

소문자를 대문자로 바꾸기**(upper)**

```
>>> a = "hi"
>>> a.upper()
 'HI'
```

대문자를 소문자로 바꾸기**(lower)**

```
>>> a = "HI“
 >>> a.lower()
 'hi'
```

문자관련함수들

왼쪽 공백 지우기**(lstrip)**

```
>>> a = "   hi    "
>>> a.lstrip()
'hi    '
```

오른쪽 공백 지우기**(rstrip)**

```
>>> a= "    hi    "
>>> a.rstrip()
'    hi'
```

양쪽 공백 지우기**(strip)**

```
>>> a = " hi "
>>> a.strip()
'hi'
```

문자관련함수들

문자열 바꾸기(replace)

replace(바뀌게 될 문자열, 바꿀 문자열)

```
>>> a = "Life is too short"
>>> a.replace("Life", "Your leg")
'Your leg is too short'
```

문자관련함수들

문자열 나누기**(split)**

```
>>> a = "Life is too short"
>>> a.split()
['Life', 'is', 'too', 'short']
>>> a = "a:b:c:d"
>>> a.split(':')
['a', 'b', 'c', 'd']
```

a.split()처럼 괄호 안에 아무런 값도 넣어 주지 않으면 공백을 기준으로 문자열을 나누어 준다.

a.split(':')처럼 괄호 안에 특정한 값이 있을 경우에는 괄호 안의 값을 구분자로 해서 문자열을 나누어 준다

Thank you