

2021 학년도 2학기 언어와 컴퓨터

제2강 Python 기초 문법

박수지

서울대학교 인문대학 언어학과

2020년 9월 6일 월요일

오늘의 목표

- 1 사칙 연산을 파이썬의 식으로 표현할 수 있다.
- 2 오류 메시지를 보고 침착할 수 있다.
- 3 표현식과 문장이 무엇인지 설명할 수 있다.
- 4 값의 이름을 바르게 지을 수 있다.
- 5 값의 자료형이 무엇인지 알고 다른 자료형으로 변환할 수 있다.
- 6 문자열의 메소드를 사용할 수 있다.

대화형 모드를 계산기처럼 사용하기

사칙 연산

```
>>> 7 + 2
9
>>> 7 - 2
5
>>> 7 * 2
14
>>> 7 / 2
3.5
```

뭔가 낯선 것

```
>>> 7 // 2
[ ]
>>> 7 % 2
[ ]
>>> 7 ** 2
[ ]
```

지금 할 일

- 1 왼쪽의 계산 결과를 보고 *, /, //, %, **의 의미를 각각 추측한다.

코딩 관습

연산자 앞뒤로 한 칸 띄운다.

산술 연산

연산자 우선순위

- 1 괄호
- 2 거듭제곱
- 3 곱셈과 나눗셈
- 4 덧셈과 뺄셈
- 5 왼쪽에서 오른쪽으로

괄호

소괄호만 사용한다.

예시

```
>>> 3 / (4 / (5 / 6))
0.625
>>> (3 / 4) / (5 / 6)
0.8999999999999999
>>> 4 * 3 ** 2
36
>>> (4 * 3) ** 2
144
>>> 4 * (3 ** 2)
36
```

산술 연산

연습

다음을 파이썬 대화형 모드 인터프리터에서 계산하라.

- 1 가로가 2.6, 세로가 4.9인 직사각형의 넓이
- 2 가로가 1.8, 세로가 5.3인 직사각형의 둘레의 길이
- 3 한 변의 길이가 3.1인 정육면체의 부피
- 4 윗변의 길이가 2.6, 아랫변의 길이가 4.9, 높이가 8인 사다리꼴의 넓이
- 5 이차방정식 $3x^2 + 2x - 8 = 0$ 의 근

질문

대화형 모드에서 할 수 있는 일

help() 함수 사용하기

도움말

```
>>> help(print)
Help on built-in function print in module builtins:

print(...)
print(value, ..., sep=' ', end='\n', file=sys.stdout, flush=False)

Prints the values to a stream, or to sys.stdout by default.
Optional keyword arguments:
file: a file-like object (stream); defaults to the current sys.stdout.
sep:   string inserted between values, default a space.
end:   string appended after the last value, default a newline.
```

무슨 말인지 모르겠어도 괜찮아요.

대화형 모드에서 할 수 있는 일

type() 함수 사용하기

값의 자료형

```
>>> type(2)
<class 'int'>
>>> type(2.0)
<class 'float'>
>>> type(2j)
<class 'complex'>
>>> type('2')
<class 'str'>
```

자료형의 이름

정수 integer
실수* floating-point number
복소수 complex number
문자열 string

*사실은 실수를 표현하려고 애쓴 것임

대화형 모드에서 할 수 있는 일

`dir()` 함수 사용하기

각 자료형의 객체로 할 수 있는 것

```
>>> dir(2)
[무엇이 길게 나오지만]
>>> dir(2.0)
[무슨 뜻인지 모르겠지만]
>>> dir(2j)
[일단 지금은 이 네 가지가]
>>> dir('2')
[서로 다른 것만 확인합시다]
```

무슨 말인지 모르겠어도 괜찮아요.

대화형 모드에서 할 수 있는 일

일부러 실수해 보기

괄호와 따옴표를 안 닫기

```
>>> print('hello')
hello
>>> print('hello'
...
... )
hello
>>> print('hello
File "<stdin>", line 1
print('hello
      ^
SyntaxError: EOL while scanning string literal
```

대화형 모드에서 할 수 있는 일

일부러 실수해 보기

산술 표현식에서 무엇인가를

```
>>> 2 + 3
5
>>> 2 +
File "<stdin>", line 1
2 +
    ^
SyntaxError: invalid syntax
```

빠뜨리기

```
>>> 2 * (3 + 4)
14
>>> 2 * (3 + 4
...
...
... )
14
```

대화형 모드에서 할 수 있는 일

일부러 실수해 보기

연산자에 이상한 값 넣기

```
>>> 3 / 0
```

```
Traceback (most recent call last):  
File "<stdin>", line 1, in <module>  
ZeroDivisionError: division by zero
```

```
>>> 4 + '5'
```

```
Traceback (most recent call last):  
File "<stdin>", line 1, in <module>  
TypeError: unsupported operand type(s) for +: 'int' and 'str'
```

대화형 모드에서 할 수 있는 일

일부러 실수해 보기

연산자 개수를 0개나 2개로

```
>>> 7 2
File "<stdin>", line 1
7 2
    ^
SyntaxError: invalid syntax
>>> 7 + + 2
9
```

바꾸어 보기

```
>>> 7 + - 2
5
>>> 7 + * 2
File "<stdin>", line 1
7 + * 2
    ^
SyntaxError: invalid syntax
```

질문

표현식

표현식 (expression)

- 값 (value), 변수 (variable), 연산자 (operator) 의 조합
- 하나의 값으로 평가 (evaluate) 된다.
예 표현식 $2 + 2$ 는 4라는 값으로 평가된다.

자료형

주의

값의 자료형에 따라 사용 가능한 연산자가 달라진다.

수치 산술 (사칙연산, 거듭제곱)

문자열 연결, 반복

주요 자료형 (data types)

- 정수, 부동소수점, 문자열
- 복소수, 불

수

수치 자료형

정수 (int), 부동소수점 (float), 복소수 (complex), 불 (bool)

산술 연산자 사용 가능

```
>>> True + 1 + 1.0 + (1+0j)  
(4+0j)
```

부동소수점수

만드는 방법

소수점 입력

```
>>> 1.23
1.23
>>> 4.
4.0
>>> .5
0.5
>>> .
File "<stdin>", line 1
  .
  ^
SyntaxError: invalid syntax
```

정수의 나눗셈

```
>>> 6 / 2
3.0
>>> 6 // 2
3
```

정수 ↔ 부동소수점수

```
>>> int(1.0)
1
>>> float(1)
1.0
```

부동소수점수

실수 \neq 부동소수점수

수의 체계

자연수 \subset 정수 \subset 유리수 \subset 실수 \subset 복소수

컴퓨터의 주요 수 유형

정수(int) $\not\subset$ 부동소수점(float) $\not\subset$ 복소수(complex)

- 부동소수점(floating point): 실수를 표현하는 개념

특징

이진수 & 유한한 메모리 공간

⇒ 모든 부동소수점은 이진수 유한소수로 표현된다.

부동소수점수

오차와 한계

정수가 부동소수점보다 더 정확한 예시

```
>>> 0.1 + 0.2
0.30000000000000004
>>> (1 + 2) / 10
0.3
>>> 2.0 ** 1024
[....]
>>> 2 ** 1024
[....]
```

복소수

허수단위 j 사용

예시

```
>>> (-1) ** (1 / 2)
(6.123233995736766e-17+1j)
>>> type((-1) ** (1 / 2))
<class 'complex'>
>>> 1j ** 2
(-1+0j)
```

주의

SyntaxError가 발생한 경우 괄호의 개수를 다시 살펴보자.

불

비교 연산자

불 자료형의 값으로 평가되는 표현식

등식, 부등식 명제

```
>>> 2 == 3
```

```
False
```

```
>>> 2 != 3
```

```
True
```

```
>>> 2 < 3
```

```
True
```

```
>>> 2 >= 3
```

```
False
```

불

논리 연산자

불 자료형의 값으로 할 수 있는 연산

```
>>> True and True
True
>>> True or False
True
>>> not True
False
```

관찰

and, or, not으로 가능한 모든 연산을 만들어 보자.

불

불과 다른 자료형의 관계

사실

불은 정수의 특수한 경우다. True는 1, False는 0이다.

확인

```
>>> isinstance(True, bool)
True
>>> isinstance(True, int)
True
>>> (True == 1) and (False == 0)
True
```

`isinstance(값, 자료형)` 값이 자료형에 속하는지 확인하는 함수

불

불과 다른 자료형의 관계

사실

0은 거짓(False), 이외의 값은 참(True)으로 해석된다.

확인

```
>>> bool(0)
False
>>> bool(5)
True
>>> bool(3.14)
True
```

관찰

자료형의 이름을 함수처럼 사용하여
자료형을 변환할 수 있다.

질문

문장

문장 (statement)

영향을 주는 코드의 단위

예시

- 1 할당문: 변수 name에 값을 지정해 준다.
예 `name = input('이름을 입력하세요: ')`
- 2 print 문: 변수 name의 값을 표시한다.
예 `print(name, '님, 반갑습니다.')`

할당문

문법

할당문 (assignment statement)의 구성

이름, 할당 연산자(=), 표현식

예시

```
>>> height = 170
>>> height / 2.5
68.0
>>> height = height + 1
>>> height
171
```

할당문

이름의 규약

필수

- 1 키워드를 쓸 수 없다.
- 2 특수 문자는 _(밑줄)만 허용된다.
- 3 숫자로 시작하면 안 된다.
- 4 공백을 포함할 수 없다.

추가

- 1 영문자와 숫자와 밑줄만 사용한다.
- 2 영어 소문자와 밑줄로 단어 경계를 표시한다.
 - my_number (o)
 - myNumber (x)

키워드 ['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']

질문

문자열

문자란 무엇인가?

問 문자열 (string) 이란 무엇인가?

답 문자들의 열 (sequence of characters) 이다.

問 문자란 무엇인가?

컴퓨터의 기준

단어 (word) 공백 **문자**를 경계로 하는 문자들의 연쇄

- 공백 문자: 빈 칸, 탭, 줄바꿈 문자, ...

행 (line) 줄바꿈 **문자**로 끝나는 문자들의 연쇄

- 줄바꿈 문자 = 개행 (開行) 문자 = line break = EOL(end-of-line)

문자열

문자열 연산

연결 (concatenation), 반복 (repetition), 비교

```
>>> '안녕하세요' + '...!'  
'안녕하세요...!'
```

```
>>> 3 * '안녕하세요'  
'안녕하세요안녕하세요안녕하세요'
```

```
>>> '안녕하세요' < '안녕하십니까'  
True
```


문자열

문자열 연산

TypeError: 자료형과 연산자가 맞지 않는 경우

```
>>> '안녕하세요' * '...!'
```

```
Traceback (most recent call last):
```

```
  File "<stdin>", line 1, in <module>
```

```
TypeError: can't multiply sequence by non-int of type 'str'
```

```
>>> 3 + '안녕하세요'
```

```
Traceback (most recent call last):
```

```
  File "<stdin>", line 1, in <module>
```

```
TypeError: unsupported operand type(s) for +: 'int' and 'str'
```

```
>>> '안녕하세요' < 333
```

```
Traceback (most recent call last):
```

```
  File "<stdin>", line 1, in <module>
```

```
TypeError: '<' not supported between instances of 'str' and 'int'
```

문자열

문자열과 수 사이의 형변환

문자열로

```
>>> str(1)
'1'
>>> str(1.0)
'1.0'
>>> str(1j)
'1j'
```

정수로

```
>>> int('11')
11
>>> int('11', base=2)
3
>>> int('11', base=3)
4
>>> int('1.0') # ValueError
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
ValueError: invalid literal for int() with base 10: '1.0'
```

문자열

문자의 열 — 지표(index)

앞에서부터 찾기

```
>>> "안녕하세요" [0]
'안'
>>> "안녕하세요" [1]
'녕'
>>> "안녕하세요" [2]
'하'
>>> "안녕하세요" [3]
'세'
>>> "안녕하세요" [4]
'요'
```

뒤에서부터 찾기

```
>>> "안녕하세요" [-1]
'요'
>>> "안녕하세요" [-2]
'세'
>>> "안녕하세요" [-3]
'하'
>>> "안녕하세요" [-4]
'녕'
>>> "안녕하세요" [-5]
'안'
```

문자열

문자의 열 — 썰기(slicing)

m번째부터 n번째 직전까지

```
>>> "안녕하세요"[0:2]
'안녕'
>>> "안녕하세요"[1:3]
'녕하'
>>> "안녕하세요"[2:4]
'하세'
>>> "안녕하세요"[3:]
'세요'
>>> "안녕하세요"[:3]
'안녕하'
```

여러 칸씩 썰기

```
>>> "안녕하세요"[0:4:2]
'안하'
>>> "안녕하세요"[1::3]
'녕요'
>>> "안녕하세요"[::-1]
'요세하녕안'
```

문자열

메소드

대문자로 바꾸기

```
>>> 'Python'.upper()  
'PYTHON'  
>>> str.upper('Python')  
'PYTHON'
```

빈칸 채우기

```
>>> student = '이름: {}, 나이: {}'  
>>> student.format('강은수', 21)  
'이름: 강은수, 나이: 21'  
>>> student.format('조재영', 20)  
'이름: 조재영, 나이: 20'
```

문자열

메소드

문자의 유형 확인하기

```
>>> 'Python'.isalpha()
True
>>> 'abc'.islower()
True
>>> '123'.isdigit()
True
>>> 'ABC'.isupper()
True
>>> '\n'.isspace()
True
```

문자열

메소드

메소드 찾기

```
>>> dir(str)
[....]
>>> dir('아무말')
[....]
>>> dir(student)
[....]
```

메소드 도움말 보기

```
>>> help(str.upper)
[....]
>>> help('아무말'.upper)
[....]
>>> help(student.upper)
[....]
```

오늘의 내용

개념

표현식, 자료형, 연산자, 문장

산술 연산자

사칙연산, 몫, 나머지, 거듭제곱

오류 메시지

SyntaxError, TypeError, ZeroDivisionError 등