2018학년도 2학기 언어와 컴퓨터

제7강 모듈과 함수

박수지

서울대학교 인문대학 언어학과

2018년 10월 1일 월요일

오늘의 목표

1. List comprehension으로 리스트를 만들 수 있다.
2. `str.join()` 메서드로 문자열을 결합할 수 있다.
3. `import` 문을 사용하여 모듈이나 객체를 가져올 수 있다.
4. `def` 문을 사용하여 함수를 정의할 수 있다.

List comprehension: 기본 사용법

```
1 [x for x in range(1, 11)]
```

특정한 조건을 만족시키는 항목만 뽑아서 리스트 만들기

```
1 [x for x in range(1, 11) if (10 % x) == 0]
```

열의 모든 항목에 대해 같은 작업을 반복하여 리스트 만들기

```
1 [2 * x + 1 for x in range(1, 11)]
```

https://docs.python.org/ko/3/tutorial/datastructures.html#list-comprehensions

형식

```
<구분자>.join(<문자열의 열>)
```

기본 사용법

```
'**'.join('배고파')
'␣'.join(['집에', '가고', '싶어'])
```

https://docs.python.org/ko/3/library/stdtypes.html#str.join

숫자나 글자만 뽑아서 연결하기

```
1 word = '배고파(...)'
2 ''.join(char for char in word if char.isalnum())
```

구구단 2단 만들기

```
1 print('\t'.join(str(2*x) for x in range(1, 10)))
```

용어집

https://docs.python.org/ko/3/glossary.html

모듈 파이썬 코드의 조직화 단위를 담당하는 객체. 모듈은 임의의 파이썬 객체들을 담는 이름 공간을 갖습니다.

- IDLE의 “Run module”

임포팅 한 모듈의 파이썬 코드가 다른 모듈의 파이썬 코드에서 사용될 수 있도록 하는 절차.

import 문

형식

```
from␣<모듈>␣import␣<이름>
```

```
import␣<모듈>
```

sin, pi 가져오기

```
>>> from math import sin, pi
>>> sin(pi / 6)
0.49999999999999994
```

math 모듈 가져오기

```
>>> import math
>>> math.sin(math.pi / 6)
0.49999999999999994
```

https://docs.python.org/ko/3/reference/simple_stmts.html#import

내장 모듈 사용 예시

`timeit` 모듈의 `timeit` 함수로 코드 실행 시간 측정하기

```
>>> from timeit import timeit
>>> timeit("'ab'.lower()")
0.09514389399991931
>>> timeit("'a'.lower(); 'b'.lower()")
0.19719874600195908
```

길이 2인 문자열 한 개를 소문자로 바꾸는 것이 길이 1인 문자열 두 개를 소문자로 바꾸는 것보다 빠르다.

모듈의 이름 공간

random 모듈에 어떤 객체가 있는지 찾기

```
>>> import random
>>> dir(random)
['BPF', 'LOG4', 'NV_MAGICCONST', 'RECIP_BPF', 'Rando
m', 'SG_MAGICCONST', 'SystemRandom', 'TWOPI', '_Buil
tinMethodType', '_MethodType', '_Sequence', '_Set',
'__all__', '__builtins__', '__cached__', '__doc__',
'__file__', '__loader__', '__name__', '__package__',
 '__spec__', '_acos', '_bisect', '_ceil', '_cos', '_
e', '_exp', '_inst', '_itertools', '_log', '_os', '_
pi', '_random', '_sha512', '_sin', '_sqrt', '_test',
 '_test_generator', '_urandom', '_warn', 'betavariat
e', 'choice', 'choices', 'expovariate', 'gammavariat
e', 'gauss', 'getrandbits', 'getstate', 'lognormvari
ate', 'normalvariate', 'paretovariate', 'randint', '
random', 'randrange', 'sample', 'seed', 'setstate',
'shuffle', 'triangular', 'uniform', 'vonmisesvariate
', 'weibullvariate']
```

용어집

https://docs.python.org/ko/3/glossary.html

함수 호출자에게 어떤 값을 돌려주는 일련의 문장들. 없거나 그 이상의 인자 가 전달될 수 있는데, 바디의 실행에 사용될 수 있습니다. 매개변수와 메서드와 함수 정의 섹션도 보세요.

매개변수 함수 (또는 메서드) 정의에서 함수가 받을 수 있는 인자 (또는 어떤 경우 인자들) 를 지정하는 이름 붙은 엔티티.

인자 함수를 호출할 때 함수 (또는 메서드) 로 전달되는 값.

메서드 클래스 바디 안에서 정의되는 함수. 그 클래스의 인스턴스의 어트리뷰트로서 호출되면, 그 메서드는 첫 번째 인자 (보통 self 라고 불린다) 로 인스턴스 객체를 받습니다. 함수와 중첩된 스코프를 보세요.

함수 호출 예시

```
>>> int('1111', base=2)
15
```

```
    함수 int
함수 호출 int(x='1111', base=2)
매개변수 base
   인자 '1111', 2
 반환값 15
```

주의

매개변수나 반환값이 없는 함수도 있다.

def 문

형식

```
1 def␣<함수명>(<매개변수>):
2 ␣␣␣␣<함수를 호출할 때 실행할 스위트>
3 ␣␣␣␣(return␣<반환할 표현식>)
```

예시

```
>>> def get_square_area(side):
...     return side ** 2
...
>>> get_square_area(5)
25
```

https://docs.python.org/ko/3/reference/compound_stmts.html#function

매개변수가 0개인 함수

```
1 def get_five():
2     return 5
3
4 a = get_five()
5 print(a)
```

반환값이 없는 함수

```
1 def hungry(x):
2     print('배고파!')
3
4 a = hungry(3)
5 print(a)
```

기존 프로그램을 함수로 고쳐 쓰기

요일 계산하기

기존 프로그램 day.py

```
1  # 설정
2  week = '월화수목금토일'
3
4  # 입력
5  day = input('오늘은 무슨 요일입
       니까? ')
6  N = int(input('며칠 후의 요일을
       계산할까요? '))
7
8  # 계산 및 출력
9  new_day = week[(week.index(
       day) + N) % 7]
10 print('일 후는 요일입니
       다.'.format(N, new_day))
```

def 문

```
1  def get_day_after(today, N):
2    week = '월화수목금토일'
3    return week[(week.index(
           today) + N ) % 7]
```

기존 프로그램을 함수로 고쳐 쓰기

단어 빈도 계산하기

새 프로그램

word_freq_functions.py

```
1 ....
```

지키면 좋은 것들

- 한 가지 함수는 한 가지 작업만 처리하도록 한다.
- 함수의 이름은 snake_case로 짓는다.
  비교 camelCase
- def 문 스위트 첫 줄에 독스트링을 작성한다.

숙제

한글 처리 프로그램

`unicodedata` 모듈의 `name` 함수를 사용하여

1. 문자열이 한글인지 판정하는 함수를 작성하기
2. 한글 문자열에 받침이 있는지 판정하는 함수를 작성하기
3. 문자열에 맞는 주격 조사(이/가)를 반환하는 함수를 작성하기

요약

List comprehension

```
[<표현식> for <반복자> in <열> (if <조건>)]
```

str.join()

```
<구분자>.join(<열>)
```

모듈에서 객체 가져오기

```
from <모듈> import <객체>
```

함수 정의하기

```
def 문
```

더 생각해 볼 것

- `f-string`에 대해서 알아보자.
- `zip()` 함수에 대하여 알아보자.