

2018학년도 2학기 언어와 컴퓨터

제2강 시작하기

박수지

서울대학교 인문대학 언어학과

2018년 9월 5일 수요일

오늘의 목표

- 1 '언어와 컴퓨터'의 의미를 설명할 수 있다.
- 2 파이썬 인터프리터를 두 가지 방법으로 사용할 수 있다.
- 3 사칙 연산을 파이썬의 식으로 표현할 수 있다.
- 4 오류 메시지를 보고 침착할 수 있다.

‘언어와 컴퓨터’의 의혹

- 1 ‘언어’가 무엇인가?
- 2 ‘컴퓨터’가 무엇인가?
- 3 ‘와’가 무엇인가?

언어

인간의 의사소통에 사용되는 **기호** 체계의 일종

언어 기호

형식 말소리 → 음성 언어

문자 → 문자 언어

손짓 → 시각 언어

내용 의미

음성 형식의 단위

음소 뜻을 변별하는 최소 단위

형태소 뜻을 가지는 최소 단위

단어 홀로 쓰일 수 있는 최소 단위

문장 완결된 내용을 나타내는 최소 단위

자의성, 사회성, 역사성,
분절성, 추상성

규칙성, 체계성, 창조성

예: “올해 봄에 꽃나물을 심었다.”

컴퓨터

현대적 정의

- 1 **전자회로**를 이용하여
cf. 기계식 컴퓨터: 해석기관
- 2 **입력된 데이터**를
수치→텍스트·그림·소리·영상
- 3 **자동으로 처리**한 후
- 4 **결과**를 즉시 **출력**하거나
- 5 **저장**하여 관리하는 기계

구성

하드웨어: 물리적 기계 장치

- 입력장치
- 출력장치
- 중앙처리장치 (CPU)
- 기억장치

소프트웨어: 프로그램 + 데이터

- 시스템 소프트웨어
- 응용 소프트웨어

와

‘언어와 컴퓨터’의 의도

(텍스트로 이루어진) 언어 데이터를 자동으로 처리하는 방법

언어 데이터

- 텍스트(문자)
- 소리(말소리)
- 영상(손짓)

자동으로 처리하는 수단

프로그램 하드웨어의 동작을
지시하고 제어하는
명령어들의 집합

자연언어처리

자연 언어 한국어, 일본어, 영어, ...

형식 언어 수식, 분자식, 논리식, ...

기술

- 종의성 해소
- 개체명 인식
- 품사 태깅
- 상호참조 해결
- 의미역 부착
- ...

응용 분야

- 기계번역
- 정보검색
- 대화 시스템
- 질의응답
- 감정분석
- ...

자연언어처리의 어려움

중의성 (ambiguity, 애매성) — 자연 언어의 특성

하나의 형식이 둘 이상의 의미를 지니는 것

중의적인 문장

- 말이 많다.
- 성질이 사나운 말의 주인이 맞았다.
- 나는 은수와 재영이를 도왔다.
- 은수는 나보다 재영이를 더 좋아한다.
- 그 사람은 누구든지 동료로 맞고 싶어한다.

자연언어처리에 관해서는 《컴퓨터언어학》에서 계속...

《언어와 컴퓨터》를 배워서 실제로 할 수 있는 일

- 말뭉치에서 가장 긴 단어를 찾아낼 수 있다.
- 텍스트의 단어 목록을 만들고 빈도를 계산할 수 있다.
- 특정 단어와 각운이 맞는 단어를 모두 찾아낼 수 있다.
- 특정 언어에서 가능한 모든 자음군의 목록을 만들 수 있다.
- 말뭉치에서 문장당 평균 단어 수를 계산할 수 있다.
- ...

위에서 나열한 작업을 수행하는 Python 프로그램을 작성할 수 있다.

Python의 역사

창시자 하위도 판 로섬 (Guido van Rossum, 1956-)

연혁

- 1989년 12월 개발 시작
- 1991년 2월 20일 0.9.0 배포
- ...
- 2000년 10월 16일 2.0
- 2008년 12월 3일 3.0

이름의 유래 Monty Python's Flying Circus

Guido van Rossum. (2009). A Brief Timeline of Python.

<http://python-history.blogspot.kr/2009/01/brief-timeline-of-python.html>

“Python”의 두 가지 의미

1 언어

프로그램을 처리하도록 컴퓨터에 내리는 명령을 기술하는 언어

2 인터프리터

고급 언어를 0과 1로 이루어진 기계어로 번역하고 실행하는 장치

- 대화형 (interactive): 문장을 프롬프트에 입력하기
- 스크립트 (script): 저장된 문장을 읽고 실행하기

고급 언어 사람의 언어에 가까운 프로그래밍 언어

■ Python, Java, C, R, Julia, ...

저급 언어 컴퓨터 내부 표현에 가까운 언어

■ 어셈블리어, 기계어

왜 파이썬인가?

“Hello, World!”를 화면에 출력시키는 스크립트를 비교해 보자.

C: 6행

```
1 #include<stdio.h>
2 int main(int argc, char * argv[])
3 {
4     printf("Hello, World!\n");
5     return 0;
6 }
```

Java: 5행

```
1 public class HelloWorld {
2     public static void main(String[] args) {
3         System.out.println("Hello, World!");
4     }
5 }
```

Python: 1행

```
1 print("Hello, World!")
```

그 외의 장점

- 인간의 언어에 가깝다.
- 매우 널리 쓰인다.
- 공개된 모듈이 많다.
- NLTK가 있다.

파이썬 인터프리터: 스크립트 모드

Visual Studio Code(이하 VSC)에서 지난 시간에 한 일

- 1 hello.py 파일 작성 후 저장
- 2 마우스 우클릭 후 Run Python File in Terminal 선택
- 3 터미널 창에서 결과 확인

주의 파이썬 버전이 3.7이 맞는지 확인한다.

파이썬 인터프리터: 스크립트 모드

VSC에서 지금 할 일 — 일반적인 실행 방식

1 hello.py 파일 작성 후 저장

```
1 # -*- coding: utf-8 -*-  
2 name = input('이름을 입력하세요: ')  
3 print(name, '님, 반갑습니다.')
```

2 메뉴에서 View > Terminal 선택

3 터미널 창에서 python hello.py 입력

윈도 python [filename]

맥 python3 [filename]

파이썬 인터프리터: 대화형 모드

VSC에서 더 할 일

- 1 hello.py 파일에 `2 + 3` 추가 입력
- 2 `2 + 3` 위에서 마우스 우클릭 후 Run Selection/Line in Terminal 선택
- 3 터미널 창에서 결과 확인
- 4 몇 가지를 더 해 보다가...
- 5 터미널 창에서 `exit()` 입력

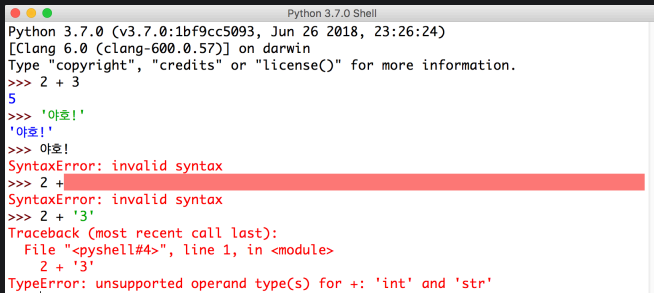
파이썬 인터프리터: 대화형 모드

VSC 밖에서 할 일

앱(응용 프로그램) > Python 3.7 > IDLE 실행

IDLE 통합 개발·교육 환경

Integrated Development and Learning Environment



```
Python 3.7.0 Shell
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 26 2018, 23:26:24)
[Clang 6.0 (clang-600.0.57)] on darwin
Type "copyright", "credits" or "license()" for more information.
>>> 2 + 3
5
>>> '야호!'
'야호!'
>>> 야호!
SyntaxError: invalid syntax
>>> 2 + 
SyntaxError: invalid syntax
>>> 2 + '3'
Traceback (most recent call last):
  File "<pyshell#4>", line 1, in <module>
    2 + '3'
TypeError: unsupported operand type(s) for +: 'int' and 'str'
```


앞으로 사용할 표기법

스크립트 모드

```
filename.py  
1 | print(4 * 6)
```

파일로 작성해서
filename.py로 저장하기

대화형 모드

```
>>> print(4 * 6)  
24
```

>>> 프롬프트(prompt)
프롬프트에 코드를 직접
입력해서 실행하기

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

퀴즈

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 이차방정식 $3x^2 + 2x - 8 = 0$ 의 근

정보

대화형 모드에서 화살표 위쪽 키를 누르면 직전에 입력한 식이 나온다.

스크립트 모드와 대화형 모드의 차이

대화형 모드

```
>>> 7 + 2
9
>>> 7 - 2
5
>>> 7 * 2
14
>>> 7 / 2
3.5
>>> 7 // 2
3
>>> 7 % 2
1
>>> 7 ** 2
49
```

지금 할 일

- 1 오른쪽의 소스코드 스크립트를 실행(run)한 결과를 확인한다.

arith.py

```
1 7+2
2 7-2
3 7*2
4 7/2
5 7//2
6 7%2
7 7**2
```

스크립트 모드와 대화형 모드의 차이

print_arith.py

```
1 print(7_+_2)
2 print(7_-_2)
3 print(7_*_2)
4 print(7_/_2)
5 print(7_//_2)
6 print(7_%_2)
7 print(7_**_2)
```

지금 할 일

- 1 왼쪽의 소스코드 스크립트를 실행해 보고, 앞서 실행한 arith.py와 결과가 어떻게 다른지 살펴본다.

스크립트 모드와 대화형 모드의 차이

print() 함수의 비밀

```
>>> 'hello'
'hello'
>>> print('hello')
hello
>>> 'hello', 'world'
('hello', 'world')
>>> print('hello', 'world')
hello world
```

코딩 관습

반점 다음 한 칸을 띄운다.

print_hello.py

```
1 'hello'
2 print('hello')
3 'hello', 'world'
4 print('hello', 'world')
```

지금 할 일

- 1 위의 소스코드 스크립트를 실행할 때 나올 결과를 추측하거나 확인한다.

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.

flush: whether to forcibly flush the stream.

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
```

《언어와 컴퓨터》에서 배우는 것

파이썬을 사용하여 텍스트 데이터를 자동으로 처리하는 방법

파이썬 인터프리터

파이썬 언어로 작성된 코드를 실행하는 장치

- 대화형 모드
- 스크립트 모드

산술 연산자

사칙연산, 몫, 나머지, 거듭제곱

오류 메시지

SyntaxError, TypeError, ZeroDivisionError 등

다음 시간 예고

- 자료형: 정수, 부동소수점, 문자열, 불(논리)
- 변수