파이썬 프로그래밍

파이썬 모듈

한국기술교육대학교
온라인평생교육원

▣ 모듈의 정의

•모듈: 파이썬 프로그램 파일로서 파이썬 데이터와 함수등을 정의하고 있는 단위
- 서로 연관된 작업을 하는 코드들을 묶어서 독립성을 유지하되 재사용 가능하게 만드는 단위
- 모듈을 사용하는 측에서는 모듈에 정의된 함수나 변수 이름을 사용

•모듈의 종류
- 표준 모듈
  ◦파이썬 언어 패키지 안에 기본적으로 포함된 모듈
  ◦대표적인 표준 모듈 예◦math, string
- 사용자 생성 모듈◦개발자가 직접 정의한 모듈
- 써드 파티 모듈◦다른 업체나 개인이 만들어서 제공하는 모듈

•모듈이 정의되고 저장되는 곳은 파일
- 파일 : 모듈 코드를 저장하는 물리적인 단위
- 모듈 : 논리적으로 하나의 단위로 조직된 코드의 모임

•파이썬 모듈이 정의되는 파일의 확장자: .py

•다른 곳에서 모듈을 사용하게 되면 해당 모듈의 .py는 바이트 코드로 컴파일 되어 .pyc로 존재한다.

▪test.py 자체가 기본적으로 test라고 하는 모듈을 만듦
▪실행코드들만 존재하면 다른 모듈이나 프로그램에서 활용 X
▪함수 f나 class 모듈을 import해서 식별자를 호출해서 이용
▪pyc 파일은 자바의 바이트 코드와 유사
▪pyc가 만들어지면 py가 없더라도 pyc를 활용하여 import가능

▣ 모듈의 정의

2. 사용자 모듈 만들기와 호출하기

```
#File: mymath.py
mypi = 3.14

def add(a, b):
    return a + b

def area(r):
    return mypi * r * r
```

▪mypi 변수, add 함수, area 함수 존재

▪mymath --> 모듈객체

•모듈 이름은 해당 모듈을 정의한 파일 이름에서 .py를 제외한 것
  - 모듈을 불러오는 키워드: import

•모듈에서 정의한 이름 사용하기

```
import mymath

print dir(mymath)     # mymath에 정의된 이름들 확인하기
print mymath.mypi     # mymath 안에 정의된 mypi를 사용한다
print mymath.area(5) # mymath 안에 정의된 area를 사용한다
```

```
['__builtins__', '__doc__', '__file__', '__name__', '__package__', 'add', 'area', 'mypi']
3.14
78.5
```

▪dir(mymath) --> mymath안에 들어있는 모든 이름을 리스트로 출력

▪'__'가 없는 식별자는 평범한 식별자

▪모듈명 + 모듈 안 존재하는 member

▣ 모듈의 정의

3. 모듈을 왜 사용하는가?

•함수와 모듈
- 함수: 파일 내에서 일부 코드를 묶는 것
- 모듈: 파일 단위로 코드들을 묶는 것◦비슷하거나 관련된 일을 하는 함수나 상수값들을 모아서 하나의 파일에 저장하고 추후에 재사용하는 단위

•모듈 사용의 이점
- 코드의 재사용
- 프로그램 개발시에 전체 코드들을 여러 모듈 단위로 분리하여 설계함으로써 작업의 효율을 높일 수 있음
- 별도의 이름 공간(스코프)를 제공함으로써 동일한 이름의 여러 함수나 변수들이 각 모듈마다 독립적으로 정의될 수 있다.

•별도 파일 내에 파이썬 코드를 저장할 때 (즉, 모듈을 코딩할 때) 한글 처리
- 파일의 맨 위에 다음 코드를 넣어 준다.◦# -*- coding:utf-8 -*-

•모듈은 하나의 독립된 이름 공간을 확보하면서 정의된다

▪함수와 모듈의 공통점 : 관련된 코드를 한 곳으로 묶는 것
▪동일한 레벨에서 여러 개의 함수가 한 모듈 내 존재 가능
▪모듈 : 파일 하나가 모듈
▪여러 개의 모듈을 기능, 역할 단위로 나누어 관련 내용을 채워 넣는 것
▪분업도 가능
▪a 모듈에 aaa --> b 모듈에 aaa
▪#으로 시작됨 --> 주석
▪주석 안 내용은 코딩이라는 키 값의 값으로 utf-8 을 적어줌
▪코딩이 utf-8 방식으로 저장됨
▪파이썬과 개발자 간의 약속
▪주석에 마음대로 한글 작성 가능

▣ 모듈의 정의

4. 모듈이 지닌 이름들 알아보기

•dir(모듈): 모듈이 지니고 있는 모든 이름들을 리스트로 반환

```
import string
print dir(string)
```

```
['Formatter', 'Template', '_TemplateMetaclass', '__builtins__', '__doc__', '__file__',
'__name__', '__package__', '_float', '_idmap', '_idmapL', '_int', '_long', '_multimap',
'_re', 'ascii_letters', 'ascii_lowercase', 'ascii_uppercase', 'atof', 'atof_error', 'atoi',
'atoi_error', 'atol', 'atol_error', 'capitalize', 'capwords', 'center', 'count', 'digits',
'expandtabs', 'find', 'hexdigits', 'index', 'index_error', 'join', 'joinfields', 'letters',
'ljust', 'lower', 'lowercase', 'lstrip', 'maketrans', 'octdigits', 'printable', 'punctuation',
'replace', 'rfind', 'rindex', 'rjust', 'rsplit', 'rstrip', 'split', 'splitfields', 'strip', 'swapcase',
'translate', 'upper', 'uppercase', 'whitespace', 'zfill']
```

▪string --> 파이썬에서 같이 설치되는 표준 모듈

▣ 모듈의 정의

5. 이름 공간을 제공하는 다른 예들

•독립된 이름 공간(스코프)을 제공하는 것들
- 모듈
- 클래스
- 객체
- 함수

•string 모듈 이름 공간에 변수 a를 생성한다.
- 표준 모듈에 사용자가 정의하는 이름을 생성하는 것은 비추천
- 단치 모듈 자체가 독립적인 이름 공간을 제공한다는 것을 알려줌

```
import string
string.a = 1
print string.a
```

```
1
```

▪string 모듈 바깥에서 새로운 변수를 정의하여 삽입 가능
▪표준모듈에 새로운 변수를 정의하여 삽입하는 것은 추천 X

▣ 모듈의 정의

5. 이름 공간을 제공하는 다른 예들

•클래스도 독립적인 이름 공간

```
class C:            # 클래스도 독립적인 이름 공간
    a = 2           # 클래스 이름 공간 내에 변수 선언
    pass            # 클래스 정의

c = C()             # 클래스 인스턴스 객체 생성
c.a = 1             # 클래스에서 생성된 인스턴스 객체도 별도의 이름 공간
print c.a
print c.__class__.a
print C.a
```

```
1
2
2
```

▪class 정의 방식 : 'class' 키워드+ class 이름+ 콜론(:)+ 아래 내용 기입
▪ a = 2의 a가 class C의 변수
▪c = C ( ) --> 생성자 호출
▪생성자가 호출되면 해당 클래스의 인스턴스가 반환됨
▪c.a = 1 --> 인스턴스 c 안에 a가 존재하고 a 값은 1
▪클래스와 인스턴스는 이름 공간이 독립적
▪함수도 독립적인 이름공간을 제공

▣ 모듈의 정의

5. 이름 공간을 제공하는 다른 예들

•함수도 독립적인 이름 공간
 - 다만 함수 내에서 선언된 로컬 변수는 함수 외부에서 접근할 수 없다.

```
x = 10    # 현재 모듈 내부에 정의되는 이름
def f():
    a = 1
    b = 2 # 현재 모듈에 정의되는 함수 f 내에 이름 a,b를 정의하고있다.
            함수도 독립적인 이름 공간
f.c = 1
print f.c
print
print f.a
```

```
1


---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
<ipython-input-6-1110995cdbd5> in <module>()
      6 print f.c
      7 print
----> 8 print f.a

AttributeError: 'function' object has no attribute 'a'
```

▪x = 10의 x는 global 변수 해당
▪a, b는 지역 변수
▪정의가 끝난 F 함수에 F 함수 바깥에서 새 변수 삽입 가능
▪일반적인 함수가 가지고 있는 지역변수 a, b는 바깥에서 코딩 X