

방정식 풀기

배경

목표

이차 방정식의 해를 계산하는 것을 핵심으로 하는 파이썬 모듈을 개발하여 보자. 우선 아래 과제 지침을 보기 전에 어떻게 구현할지 생각해 보는 것도 좋다. 구현 결과는 지침에 정확히 부합해야 한다.

이차 방정식의 해

이차 방정식 $ax^2 + bx + c = 0$ 의 해는 다음과 같다.

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

- 이차 방정식이므로 $a \neq 0$ 인 경우의 해이다. $b^2 - 4ac$ 가 양수일 때는 두 개의 실근, 0일 때는 하나의 중근, 음수일 때는 두 개의 허근을 가진다.
- 한편 $a = 0$ 일 때에는 일차 방정식 $bx + c = 0$ 을 풀면 된다. $b \neq 0$ 일 때의 해는 $x = -c/b$ 이다. 만약 $b = 0$ 이라면 $c = 0$ 일 때는 부정, $c \neq 0$ 일 때는 불능이라고 한다.

구현 지침

패키지 구성

다음과 같은 함수를 작성하여라. 디렉토리 이름은 `smath`에 파일이름은 `poly.py`로 한다. 지정된 `math.sqrt` 이외의 모듈은 사용하지 않는다. `smath` 폴더를 zip으로 압축하여 제출하여라.

```
#
# poly.py - polynomials
#

from math import sqrt

# Polynomial Equation Solver
#
# coefficients : list of coefficients. for example [2, 3, 4] for 2x^2 + 3x + 4
# return list of solutions. for example [-1, 1] for x^2 - 1 = 0
def solve(coefficients) :
    # implement it
    pass
```

작동 방식

다음과 같이 작동하도록 구현하여라.

```
>>> import smath.poly
>>> smath.poly.solve([1, -5, 6])
[3.0, 2.0]
>>> smath.poly.solve([1, 2, 1])
[-1.0]
>>> smath.poly.solve([1, 1, 2.5])
[(-0.5+1.5j), (-0.5-1.5j)]
>>> smath.poly.solve([0, 1, 1])
[-1.0]
>>> smath.poly.solve([1, 1])
[-1.0]
>>> smath.poly.solve([0, 1])
warning: undefined (there is no possible solution)
nan
>>> ans = smath.poly.solve([0, 1])
warning: undefined (there is no possible solution)
>>> ans
nan
>>> ans = smath.poly.solve([0, 0])
warning: indeterminate (0/0 form)
>>> ans
nan
>>> smath.poly.solve([1, 2, 3, 4])
can't solve! [1, 2, 3, 4]
>>> ans = smath.poly.solve([1, 2, 3, 4])
can't solve! [1, 2, 3, 4]
>>> print(ans)
None
```

참고

len()

리스트의 길이는 len() 함수로 알 수 있다.

```
>>> arr = [1, 2, 3]
>>> len(arr)
3
```

nan

숫자가 아니라는 뜻의 `nan` (Not-A-Number)는 다음과 같이 만들 수 있다.

```
>>> float('nan')
nan
```

print vs return

함수는 그 결과를 항상 `return` 값으로 돌려주도록 작성한다. 결과를 `print` 하지 않도록 한다. 그러나 함수에서 결과값이 아니라 예러나 경고를 주고 싶을 때가 있다. 이번 과제에서 메시지는 `print()`로 출력하고 결과값은 `return`으로 돌려주도록 한다. 예를 들어 다음과 같이 함수를 작성하였다고 하자.

```
def add(x, y):
    ans = x + y
    print(x, "+", y, "=", ans)
    return ans
```

다음과 같이 작동한다.

```
>>> add(3, 4)
3 + 4 = 7
7
>>> a = add(3, 4)
3 + 4 = 7
>>> a
7
```