

» 학습목표

- 파이썬 프로그램으로 수학과 관련된 문제를 해결 할 수 있는
수학문제 풀기의 프로젝트 프로그램을 제작할 수 있다.

Unit 1. 파이썬으로 수학 문제 풀기 1

- 01** 수학과 컴퓨터의 밀접한 관계
- 02** 집합
- 03** 소인수 분해를 하는 프로그램
- 04** 평균 분산 표준편차

슬라이드1. 수학과 컴퓨터의 밀접한 관계

» 컴퓨터(Computer)의 어원 : '계산하다'를 뜻하는 Compute에서 나옴

원래 컴퓨터라는 단어는 '돈을 받고 원하는 계산을 하는 사람' 혹은 '계산을 전문으로 하는 직업'을 일컫는 말.

현대식 컴퓨터가 없던 시절, 수학이나 과학 연구 혹은 군사 목적으로 복잡한 계산을 하려면 수학 지식을 갖춘 전문가를 고용해서 월급을 주고 복잡한 계산을 시켰는데, 이들이 바로 '최초의 컴퓨터'였음

뛰어난 성능을 자랑하는 최신 컴퓨터부터 스마트폰, 태블릿 PC와 같이 최첨단 전자제품이 넘쳐나는 지금, '컴퓨터'라는 직업은 이미 사라진 지 오래지만 여전히 컴퓨터와 수학은 떼려야 뗄 수 없는 관계임.

애초에 수학 문제를 풀려고 만든 것이 컴퓨터였을 뿐만 아니라, 컴퓨터 역시 수학 이론을 바탕으로 만들어졌기 때문.

1. 수학과 컴퓨터의 밀접한 관계

» 아직 중학교 수학을 배우지 않았어요!

Day 19와 20에서는 파이썬을 사용해서 중학교 과정의 수학 문제를 풀어보는 예제 프로그램을 만드는데, 구체적인 주제는 다음 다섯 가지입니다.

- ① 집합
- ② 소인수분해
- ③ 통계(평균, 분산, 표준편차)
- ④ 이차방정식
- ⑤ 함수의 그래프

다섯 가지 주제를 아직 배우지 않은 초등학생이나 중학생이라면 186쪽으로 넘어가서 이 책을 마무리해도 됩니다. 배우긴 했지만 오래되어 기억나지 않는다면 '이렇게 하는구나' 정도로 실습을 따라할 것을 권합니다. 프로그래밍에 익숙해지는데 분명 도움이 될 것입니다.

슬라이드 2. 집합

» 수학의 집합에 해당하는 기능이 기본으로 있기 때문에 모듈을 사용하지 않고도 집합과 관련된 연산을 쉽게 할 수 있음

» 집합 연산을 하는 프로그램

파이썬의 집합 기능 1

```
A = {1, 2, 3, 4}
B = {3, 4, 5, 6}
```

```
print(A)
print(B)
```

```
print(1 in A)
print(6 in A)
print(len(A))
```

```
print(A | B)
print(A & B)
print(A - B)
```

파이썬의 집합 기능 2

```
C = {x for x in range(1, 11)}
D = {x for x in range(1, 11) if x % 3 == 0 }
```

```
print(C)
print(D)
```

```
print(C < D)
print(C > D)
```

2. 집합

» 실행결과

```
{1, 2, 3, 4}
```

```
{3, 4, 5, 6}
```

```
True
```

```
False
```

```
4
```

```
{1, 2, 3, 4, 5, 6}
```

```
{3, 4}
```

```
{1, 2}
```

```
{1, 2, 3, 4, 5, 6, 7, 8, 9, 10}
```

```
{9, 3, 6}
```

```
False
```

```
True
```

```
>>>
```

2. 집합

» 파이썬 문장과 수학 개념 비교

파이썬 문장	수학 개념	설명	실행 결과
<code>A = {1, 2, 3, 4}</code>	$A = \{1, 2, 3, 4\}$	1, 2, 3, 4를 원소로 하는 집합 A	<code>{1, 2, 3, 4}</code>
<code>B = {3, 4, 5, 6}</code>	$B = \{3, 4, 5, 6\}$	3, 4, 5, 6을 원소로 하는 집합 B	<code>{3, 4, 5, 6}</code>
<code>1 in A</code>	$1 \in A$	1이 집합 A의 원소인가?	<code>True</code>
<code>6 in A</code>	$6 \in A$	6이 집합 A의 원소인가?	<code>False</code>
<code>len(A)</code>	$n(A)$	집합 A의 원소 개수	<code>4</code>
<code>A B</code>	$A \cup B$	A와 B의 합집합	<code>{1, 2, 3, 4, 5, 6}</code>
<code>A & B</code>	$A \cap B$	A와 B의 교집합	<code>{3, 4}</code>
<code>A - B</code>	$A - B$	A와 B의 차집합	<code>{1, 2}</code>
<code>C = {x for x in range(1,11) }</code>	$C = \{x x \text{는 } 1 \text{ 이상 } 11 \text{ 미만의 정수}\}$	1 이상 11 미만의 정수를 원소로 하는 집합 C	<code>{1, 2, 3, 4, 5, 6, 7, 8, 9, 10}</code> ※ 11은 제외
<code>D = {x for x in range(1,11) if x % 3 == 0 }</code>	$D = \{x x \text{는 } 1 \text{ 이상 } 11 \text{ 미만의 } 3 \text{의 배수}\}$	1 이상 11 미만의 3의 배수를 원소로 하는 집합 D	<code>{9, 3, 6}</code> ※ 원소의 순서는 상관없으므로 {3, 6, 9}와 같음
<code>C < D</code>	$C \subset D$	C는 D의 (진)부분집합인가?	<code>False</code>
<code>C > D</code>	$C \supset D$	D는 C의 (진)부분집합인가?	<code>True</code>

슬라이드 3. 소인수분해를 하는 프로그램

» # 소인수분해 프로그램

```
x = int(input("?"))
d = 2
```

소인수분해할 숫자를 입력받아 정수로 바꿈
가장 작은 소수인 2부터 나눔

```
while d <= x:
```

```
    if x % d == 0:
```

```
        print(d)
```

```
        x = x / d
```

```
    else:
```

```
        d = d + 1
```

x가 d로 나누어지면(나머지가 0이면)

d는 x의 약수이므로 출력

x를 d로 나눠서 다시 x에 저장

나누어지지 않으면 1을 더해서 반복

» 실행결과

```
? 12
```

```
2
```

```
2
```

```
3
```

```
>>>
```


슬라이드 4. 평균 분산 표준편차

» 평균·분산·표준편차를 계산하는 방법

- 1 | 평균 : 여러 개의 자료값을 모두 더한 다음 자료의 개수로 나눈 값
- 2 | 분산 : 편차 제곱의 평균값. 즉, 각 자료값에서 평균값을 뺀 값(편차)의 제곱을 모두 더해서 자료의 개수로 나눈 값
- 3 | 표준편차 : 분산의 제곱근(\sqrt{x}) 값

4. 평균 분산 표준편차

» 평균 · 분산 · 표준편차 구하기

```
# 평균, 분산, 표준편차를 구하는 프로그램  
import math
```

```
# 자료값 리스트  
d = [1, 2, 3, 4, 5]  
print(d)
```

```
# 평균 구하기  
mean = sum(d) / len(d)  
print(mean)
```

```
# 분산 구하기  
vsum = 0  
for x in d:  
    vsum = vsum + (x - mean)**2  
var = vsum / len(d)  
print(var)
```

```
# 표준편차 구하기  
std = math.sqrt(var)  
print(std)
```

» 실행결과

```
[1, 2, 3, 4, 5]  
3.0  
2.0  
1.4142135623730951  
>>>
```

Unit 2. 파이썬으로 수학 문제 풀기 2

01 이차방정식

02 함수의 그래프

슬라이드 1. 이차방정식

» 이차방정식 근의 공식

$$ax^2 + bx + c = 0 (a \neq 0)$$

$$\text{판별식} : D = b^2 - 4ac$$

- $D > 0$: 근이 두 개일 때

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

- $D = 0$: 근이 한 개일 때

$$x = -\frac{b}{2a}$$

- $D < 0$: 근이 없을 때

1. 이차방정식

» 파이썬의 math 모듈

math 모듈은 이름에서 알 수 있듯이 수학 계산에 필요한 여러 가지 함수를 제공하는 모듈입니다. 앞에서 살펴본 이차방정식을 푸는 프로그램은 `import math`로 math 모듈을 사용합니다. 근의 공식을 풀려면 제곱근($\sqrt{\quad}$)을 구하는 함수인 `math.sqrt(x)`가 필요하기 때문입니다. math 모듈에는 원주율을 의미하는 `math.pi(=3.1415926...)`와 같은 숫자값도 정의되어 있습니다. 뿐만 아니라 `sin(x)`, `cos(x)`, `tan(x)`와 같은 삼각함수, Day 12에서 만든 `factorial(x)` 함수 등 수학 계산에 필요한 여러 가지 기능이 있습니다.

수학 문제에 관심이 많은 사람은 math 모듈의 내용을 한 번 찾아보는 것도 재미있을 것입니다.

- 참고 사이트 : <https://docs.python.org/3/library/math.html>

1. 이차방정식

```
>> import math  
import sys
```

```
print("ax2 + bx + c = 0")
```

```
# 계수 a, b, c를 입력받고, 입력받은 문자열을 소수로 바꿈
```

```
a = float(input("a? "))
```

```
b = float (input("b? "))
```

```
c = float (input("c? "))
```

```
if a == 0:
```

```
    print("a = 0 :이차방정식이 아닙니다.")
```

```
    sys.exit()
```

```
# 이차방정식이 아니면 프로그램 실행을 멈춤
```

1. 이차방정식

$D = b^2 - 4ac$

판별식

```
if D > 0:
    x1 = (-b+math.sqrt(D))/(2*a)
    x2 = (-b-math.sqrt(D))/(2*a)
    print("2개의 해 :", x1, x2)
if D == 0:
    x = -b/(2*a)
    print("1개의 해 :", x)
if D < 0:
    print("해가 없습니다.")
```

» 실행결과

```
ax2 + bx + c = 0
a? 1
b? -3
c? 2
2개의 해 : 2.0 1.0
>>>
```

슬라이드2. 함수의 그래프

```
» import turtle as t
```

```
# 그래프를 그릴 x 좌표 범위
```

```
x_min = -5
```

```
x_max = +5
```

```
# 그래프를 그릴 y 좌표 범위
```

```
y_min = -5
```

```
y_max = +5
```

```
# 그래프를 그릴 간격
```

```
space = 0.1
```

```
# 그릴 함수의 리스트
```

```
func_list = ["y = x*x", "y = abs(x)", "y = 0.5*x + 1"]
```

```
# 좌표 설정, 거북이 속도, 선 굵기
```

```
t.setworldcoordinates(x_min, y_min, x_max, y_max)
```

```
t.speed(0)
```

```
t.pensize(2)
```

```
# x 축 그리기
```

```
t.up()
```

```
t.goto(x_min, 0)
```

```
t.down()
```

```
t.goto(x_max, 0)
```

```
# y 축 그리기
```

```
t.up()
```

```
t.goto(0, y_min)
```

```
t.down()
```

```
t.goto(0, y_max)
```


2. 함수의 그래프

그래프 그리기

t.color("green")

for func in func_list:

 x = x_min

 exec(func)

 t.up()

 t.goto(x, y)

 t.down()

while x <= x_max:

 x = x + space

 exec(func)

 t.goto(x, y)

func_list에 있는 함수를 하나씩 그림

x_min부터 계산을 시작

수식을 계산

계산된 좌표로 이동

x_max까지 그래프를 그림

space만큼 x를 증가시킨 후

수식을 계산.

계산된 좌표로 이동

2. 함수의 그래프

» 실행결과

