



▶ 학습내용

- 01 프로그래밍 언어의 개념을 학습합니다.
- 02 파이썬을 내 컴퓨터에 설치합니다.
- 03 첫 번째 프로그램을 작성해봅니다.
- 04 코드를 파일에 저장하여 실행하는 방법을 알게 됩니다.
- 05 터틀 그래픽으로 여러 가지 그림을 그려봅니다.

▶ LAB

- 01 print() 실습
- 02 원과 다각형 그리기
- 03 터틀 그래픽 더 살펴보기

01. 어디에나 있는 수식



(출처: 영화 '어벤저스'의 한 장면)

01. 어디에나 있는 수식



- 컴퓨터를 사용하면 우리가 복잡한 계산을 직접 해야 할 필요는 없음.
- 우리는 정확한 계산을 하도록 알맞은 수식을 올바르게 작성하여 컴퓨터에 지시해야 함

$$\begin{array}{c} 35 \text{ (식의 값)} \\ \hline (\underbrace{12}_{\text{피연산자}} \underbrace{+}_{\text{연산자}} \underbrace{23}_{\text{피연산자}}) \leftarrow \text{(수식)} \end{array}$$

01. 어디에나 있는 수식



올바른 수식 작성을 위하여

- 연산자 우선순위
- 수학 시간에 배우지 않았던 새로운 연산자
- 연산의 방향

연산자를 잘 사용하면 프로그램 코드의 길이를 줄일 수 있습니다.

02. 산술 연산자



산술은 수에 대한 계산

연산자	기호	사용 예	결괏값
더하기	<u>+</u>	7 + 4	11
빼기	<u>-</u>	7 - 4	3
곱하기	<u>*</u>	7 * 4	28
지수(제곱)	<u>**</u>	<u>7 ** 4</u>	2401
나누기	<u>/</u>	7 / 4	1.75
나누기(몫)	<u>//</u>	<u>7 // 4</u>	1
나머지	<u>%</u>	<u>7 % 4</u>	3
대입(할당)	<u>=</u>	x = 10	

2√4

02. 산술 연산자



더하기(+), 빼기(-), 곱하기(*)

코드 및 실행 결과 1

```
>>> 2+17
```

```
19
```

```
>>> 2-17
```

```
-15
```

코드 및 실행 결과 2

```
>>> 2*17
```

```
34
```

02. 산술 연산자



지수 연산자 (**)

코드 및 실행 결과 1

```
>>> 17 ** 2  
289  
>>> 10*2**7  
1280
```

코드 및 실행 결과 2

```
>>> 2**2**3  
256
```

02. 산술 연산자



나누기 연산자: '/'와 '//'

코드 및 실행 결과 1

```
>>> 17 / 5 # 실수 나눗셈  
3.4
```

코드 및 실행 결과 2

```
>>> 17 // 5 # 정수 몫  
3
```


02. 산술 연산자



나머지 연산자 '%'

코드

```
p = int(input("나누어지는 수를 입력하시오: "))  
q = int(input("나누는 수를 입력하시오: "))  
print("나눗셈의 몫=", p // q)  
print("나눗셈의 나머지=", p % q)
```

실행 결과

```
나누어지는 수를 입력하시오: 17  
나누는 수를 입력하시오: 5  
나눗셈의 몫= 3  
나눗셈의 나머지= 2
```

川

$$\begin{array}{r} 3 \leftarrow \text{몫} \\ 5 \overline{) 17} \\ \underline{15} \\ 2 \leftarrow \text{나머지} \end{array}$$

02. 산술 연산자



그렇다면 이 나머지 연산자를 어디에 이용하면 좋을까요?

코드

```
sec = 1000  
min = sec // 60  
remainder = sec % 60  
print(min, '분', remainder, '초')
```

실행 결과

16 분 40 초

03. 대입 연산자 ' = '



왼쪽 : 변수

오른쪽 : 수식 또는 값

$$X = 100 + 200$$

↑ 변수

↑ 수식 또는 값

'='의 특별함을
2장에 설명했었습니다.

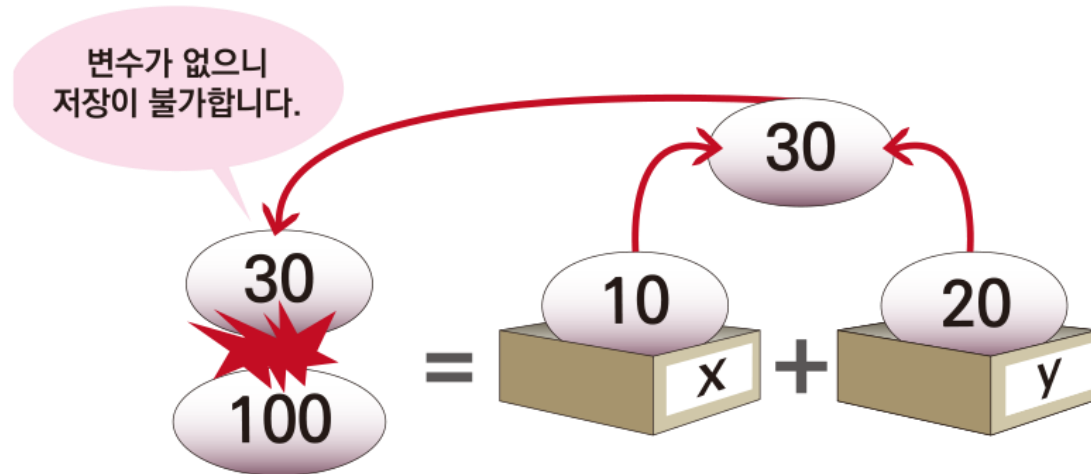
03. 대입 연산자 ' = '

13



잘못된 예

$$100 = x + y$$



03. 대입 연산자 ' = '



여러 개의 변수에 동일한 값을 저장하는 프로그램을 작성해 보겠습니다.

코드

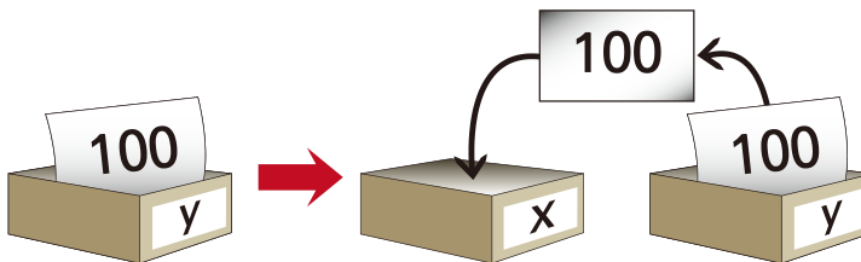
```
x = y = 100  
print(x)  
print(y)
```



실행 결과

100
100

y = 100
x = y



04. 복합 대입 연산자



복합 대입 연산자(compound assignment operator): 대입 연산자와 다른 연산자를 합쳐 놓은 것

서로 의미가 같습니다.

$x \text{ += } 2 \iff x = x + 2$

$x \text{ -= } 2$
 $x \text{ *= }$

$x = x - 2$

04. 복합 대입 연산자



복합 연산자	의미
$x += y$	$x = x + y$
$x -= y$	$x = x - y$
$x *= y$	$x = x * y$
$x /= y$	$x = x / y$
$x //= y$	$x = x // y$
$x \% = y$	$x = x \% y$

04. 복합 대입 연산자



서로 의미가 다릅니다.

$$\underline{x = x * 2 + 3} \longleftrightarrow \underline{x *= 2 + 3}$$

5

~~~~~

$x *= 5$



# 04. 복합 대입 연산자



다음 예제를 통해 복합 대입 연산자 '+=', '-='를 이해해 보겠습니다.

## 코드

```
x = 1000  
print("초깃값 x=", x)  
x += 2;  
print("x += 2 후의 x=", x)  
x -= 2;  
print("x -= 2 후의 x=", x)
```

## 실행 결과

초깃값 x= 1000  
x += 2 후의 x= 1002  
x -= 2 후의 x= 1000

# 05. 연산자의 우선순위



우선순위(precedence) : 하나의 수식에 있는 여러 연산 중에서 어떤 연산을 먼저 수행할지를 결정하는 규칙



$$\underline{X + y \times z}$$

①

②

# 05. 연산자의 우선순위



| 연산자                            | 설명                       |                                                                                             |
|--------------------------------|--------------------------|---------------------------------------------------------------------------------------------|
| **                             | 지수 연산자                   | <br>높음   |
| ~, +, -                        | 단항 연산자                   |                                                                                             |
| *, /, %, //                    | 곱셈, 나눗셈, 나머지 연산자, 나눗셈(몫) |                                                                                             |
| +, -                           | 덧셈, 뺄셈                   |                                                                                             |
| >>, <<                         | 비트 이동 연산자                |                                                                                             |
| &                              | 비트 AND 연산자               | <br>낮음 |
| ^,                             | 비트 XOR 연산자, 비트 OR 연산자    |                                                                                             |
| <=, <, >, >=                   | 비교 연산자                   |                                                                                             |
| <>, ==, !=                     | 동등 연산자                   |                                                                                             |
| =, %=, /=, //, -=, +=, *=, **= | 대입 연산자                   |                                                                                             |
| is, is not                     | 아이덴티티 연산자                |                                                                                             |
| in, not in                     | 소속 연산자                   |                                                                                             |
| not, or, and                   | 논리 연산자                   |                                                                                             |

# 05. 연산자의 우선순위



왼쪽에서 오른쪽으로 결합

$2 + 3 - 4;$

①

②



오른쪽에서 왼쪽으로 결합

$a = b = c;$

①

②



# 05. 연산자의 우선순위



기본 처리 순서

$i = 2 + 3 \times 4;$

①  $3 \times 4$

②  $2 + 12$

③  $i = 14$

괄호(( ))를 사용한 순서 변경

$i = (2 + 3) \times 4;$

①  $2 + 3$

②  $5 \times 4$

③  $i = 20$

# 05. 연산자의 우선순위



사용자로부터 3개의 수를 입력받아서 평균을 출력하는 프로그램을 작성해 보겠습니다.

## 코드

```
x = int(input("첫 번째 수: "))  
y = int(input("두 번째 수: "))  
z = int(input("세 번째 수: "))  
  
avg = (x + y + z) / 3  
print("평균 =", avg)
```

## 실행 결과

첫 번째 수: 100  
두 번째 수: 90  
세 번째 수: 80  
평균 = 90.0

$$\underbrace{x + y}_{②} + \underbrace{z / 3}_{①}$$

③

$$\underbrace{(x + y + z)}_{①} / 3$$

②  
③

연산의 순서

# Lab. 다항식의 계산



- $x=-1$ ,  $y=3$ 일 때,  $(-y)^3+2x^2y$ 의 값을 계산하여 출력하는 프로그램

## 실행 결과

다항식의 계산 결과 : -21



생각 1 : 프로그램의 순서를 생각해봅니다.

변수  $x$ ,  $y$ 에 값 저장



다항식을 파이썬의  
연산자를 이용하여 표현



계산결과 출력하기

# Lab. 화씨온도를 섭씨온도로 변환하기

25



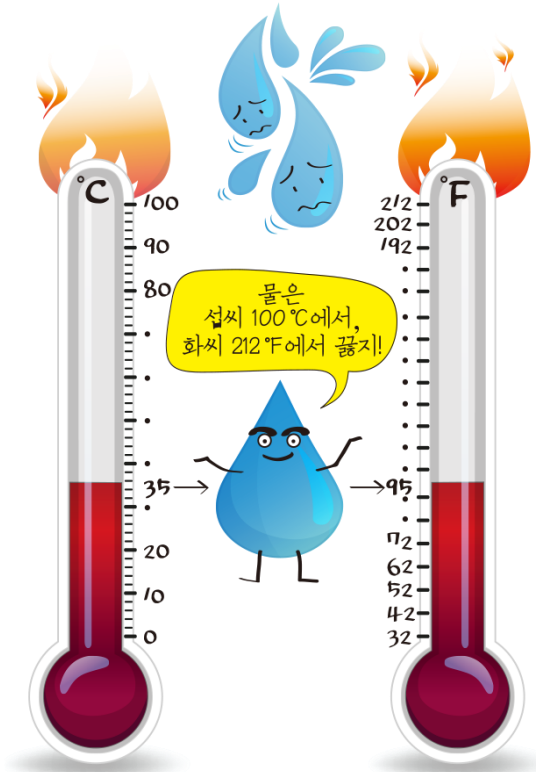
- 화씨온도를 입력 받아서 섭씨온도로 바꾸는 프로그램

$$C = (F - 32) \times \frac{5}{9}$$

## 실행 결과

화씨온도: 100

섭씨온도: 37.77777777777778





# Lab. 두 점 사이의 거리 구하기

26



- 사용자로부터 두 점의 좌표 (x1, y1)과 (x2, y2)를 입력받아 두 점 사이의 거리를 계산하는 프로그램

$$\sqrt{(x2-x1)^2 + (y2-y1)^2}$$

\*Q 5

## 실행 결과

x1: 0

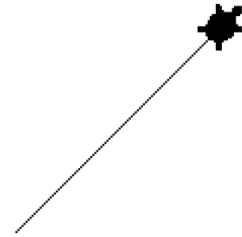
y1: 0

x2: 100

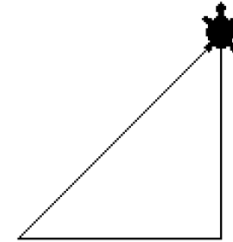
y2: 100

두 점 사이의 거리= 141.4213562373095

# Lab. 두 점 사이의 거리 확인하기



- ① 터틀을 좌표 (0, 0)에서  $45^\circ$  회전하여 141.4만큼의 거리를 이동시킴



- ② 터틀을 좌표 (0, 0)에서 x축으로 100, y축으로 100 이동하여 점 (100, 100)으로 이동

터틀이 그린 자취가 ①과 ②에서 만나면 계산 값이 올바르게 되는 것!!



생각 1 : 프로그램의 순서를 생각해 봅니다.

터틀 그래픽을 사용할  
준비 작업



① 터틀을 좌표 (0, 0)에서  $45^\circ$  회전하여 141.4의 거리만큼 이동시킴



② 터틀을 좌표 (0,0)에서 x축으로 100, y축으로 100 이동하여 점 (100, 100)으로 이동

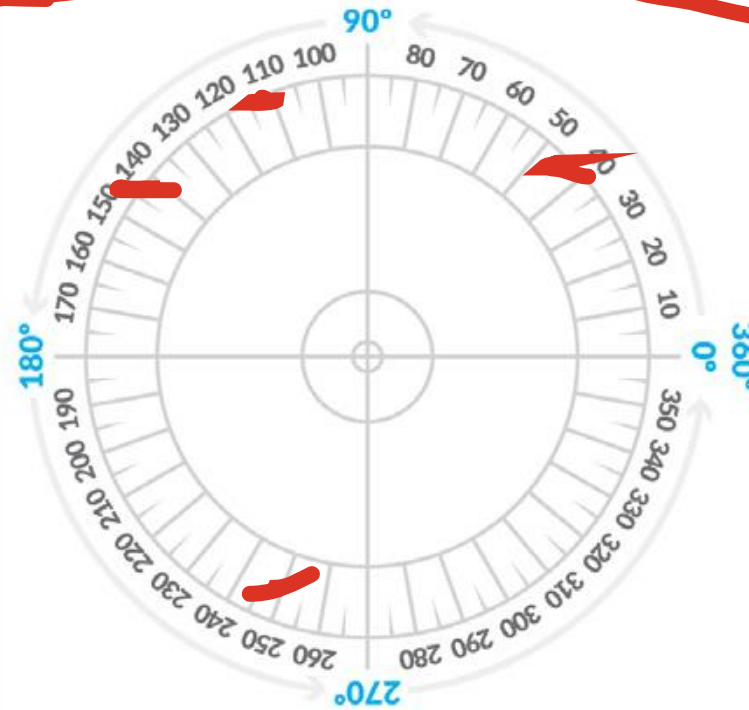


①과 ②가 만나는가,  
안 만나는가를 확인

# Lab. 두 점 사이의 거리 확인하기



`t.setheading()` : 터틀의 머리 방향을 특정한 각도로 설정합니다.



# Lab. 그리니치 표준시-세계시간의 기준점



☺ 표시된 도시는 서머타임 시행 중이며, 세계시간은 PC환경에 따라 오차가 있을 수 있습니다.

(출처 : 네이버 해외정보)

# Lab. 그리니치 표준시-세계시간의 기준점

## 실행 결과

현재 한국 시간: 12시23분



생각 1 : 프로그램의 순서를 생각해 봅니다.

time( )을 이용하여 그리니치 시간 계산에 필요한 자료 구하기



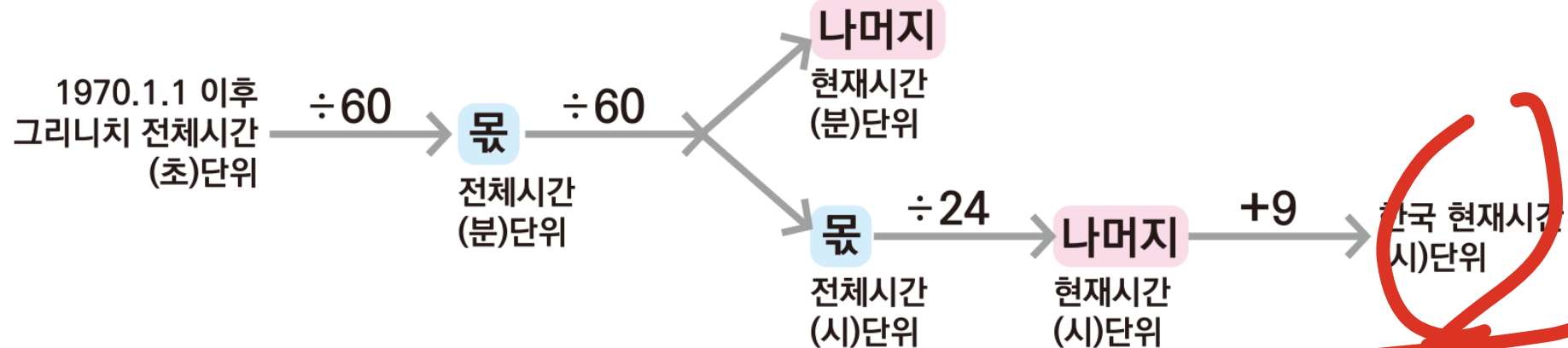
그리니치 시간의 '시'계산



그리니치 시간의 '분'계산



현재시간 출력



# Lab. 계산대 프로그램



물건의 가격과 지불한 금액을 입력하면 거스름돈을 알려주는 계산대 프로그램을 작성해 보겠습니다. 대신 동전의 개수는 최대한 적게 하도록 합니다.(단, 물건가격과 투입한 돈은 100원 단위로 입력되고, 거스름돈은 500원, 100원짜리로만 거슬러 줍니다.)



## 실행 결과

투입한 돈: 5000

물건가격: 2600

거스름돈: 2400

500원 동전의 개수: 4

100원 동전의 개수: 4

1 1 0 0 0  
2 0 0

—

1000  
2600