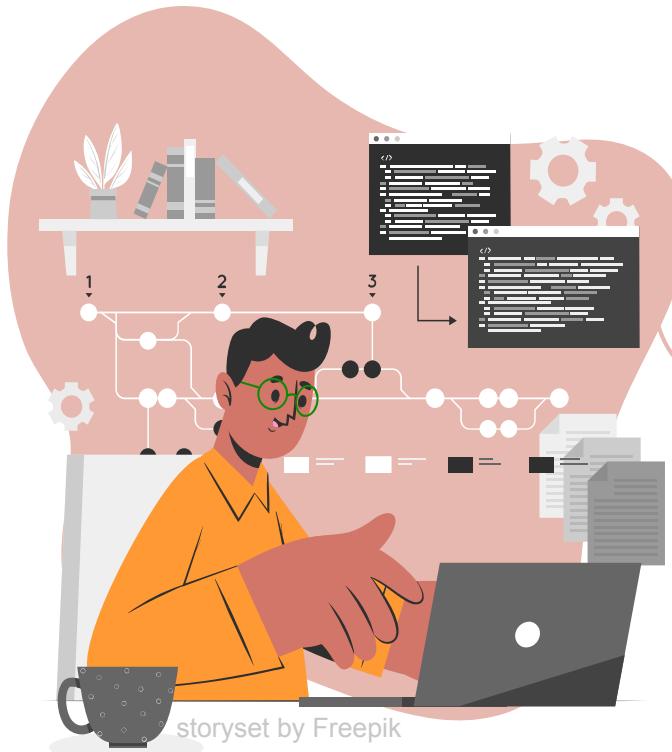
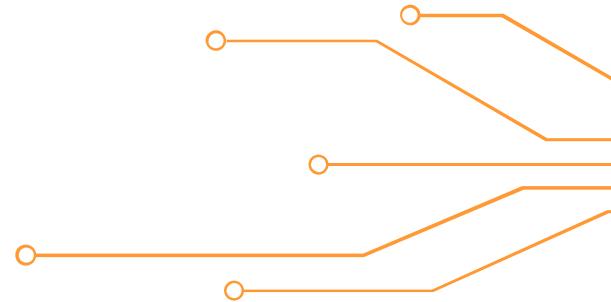
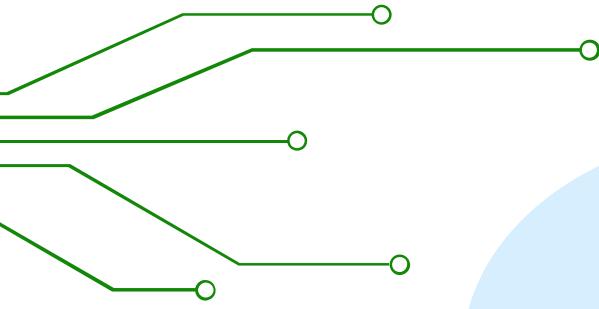


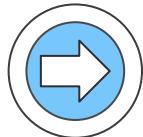
# 컴퓨팅 사고와 프로그래밍



CH 01.  
**HELLO  
PYTHON !**

# 강의안내





학습목표



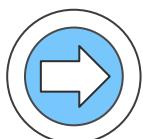
이론



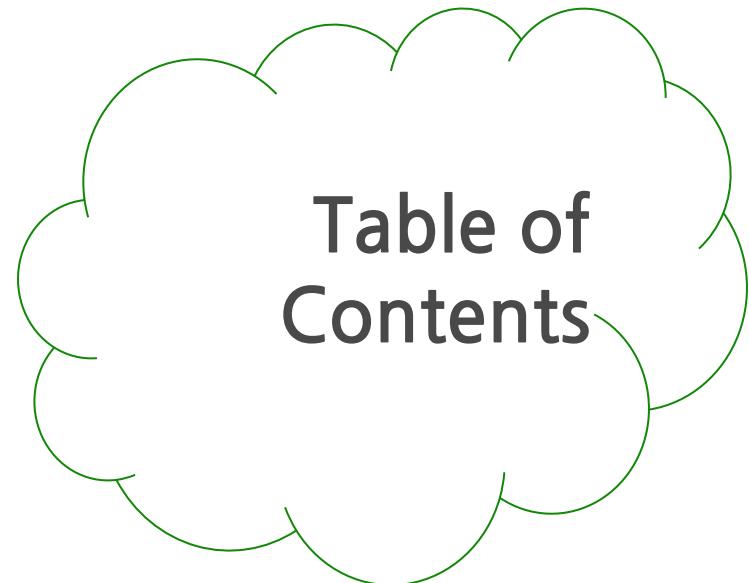
실습



요약



퀴즈



만들면서  
배우는  

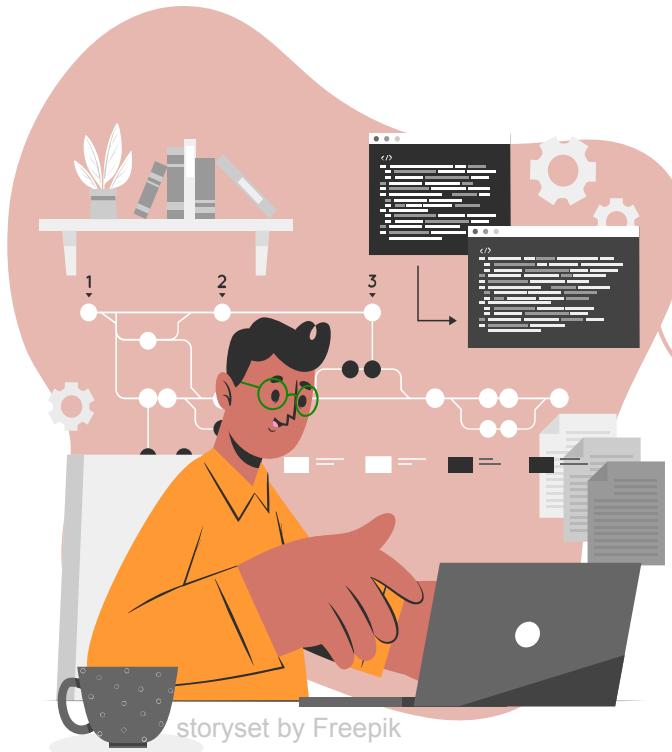



## 컴퓨팅 사고와 프로그래밍

- |            |                          |
|------------|--------------------------|
| 1 - 파이썬 소개 | 7 - 함수                   |
| 2 - 변수     | 8 - <a href="#">프로젝트</a> |
| 3 - 계산 기능  | 9 - 리스트와 딕셔너리 1          |
| 4 - 자료의 종류 | 10 - 리스트와 딕셔너리 2         |
| 5 - 조건     | 11 - GUI 프로그래밍 1         |
| 6 - 반복     | 12 - GUI 프로그래밍 2         |

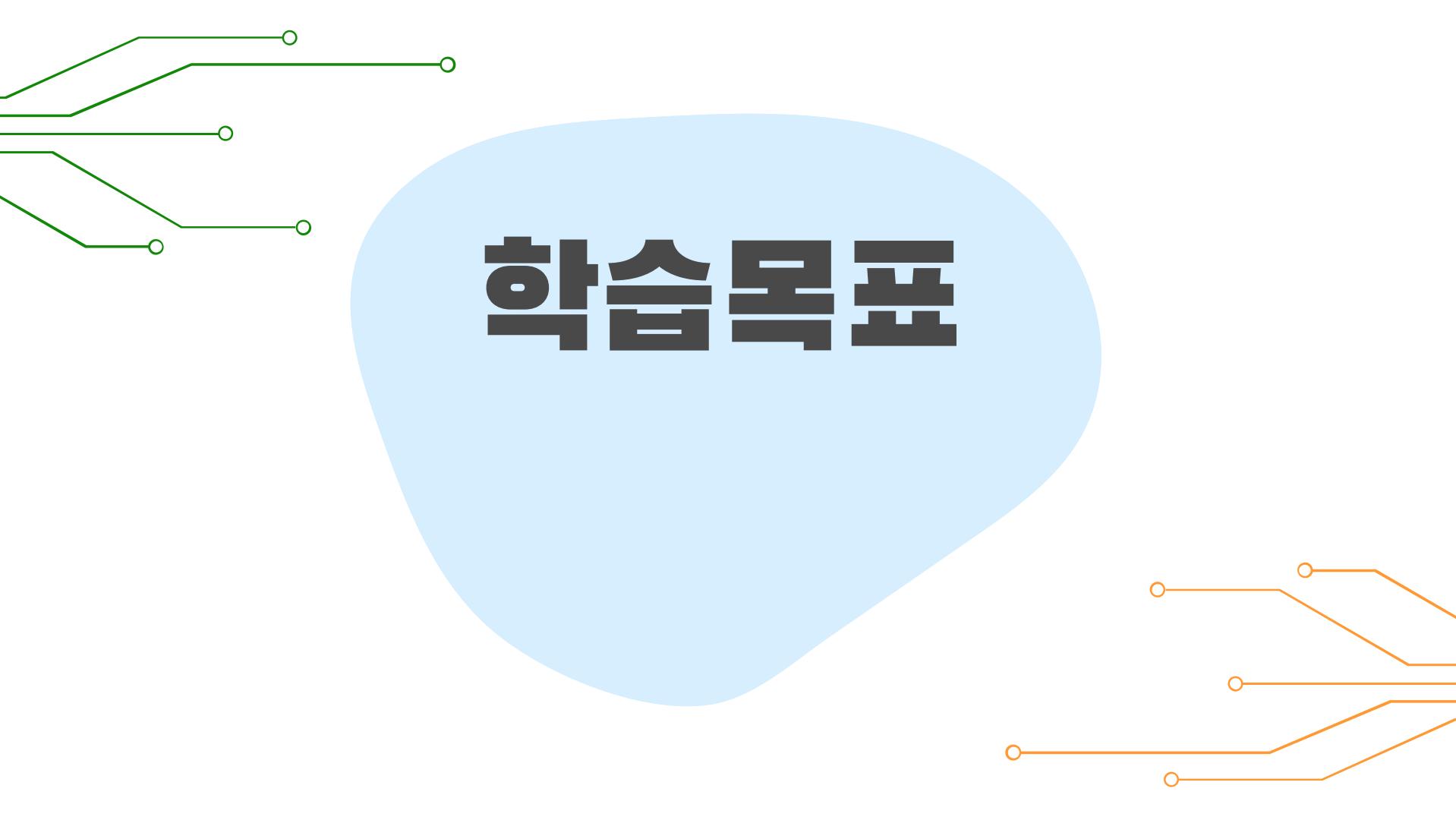


# 컴퓨팅 사고와 프로그래밍



CH 01.  
**HELLO  
PYTHON !**

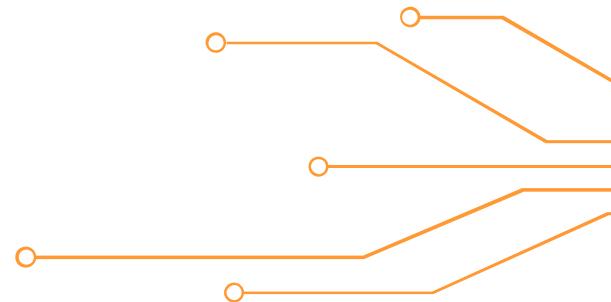
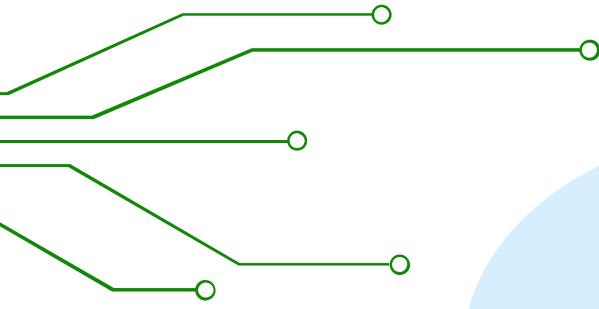
# 학습목표



1. 프로그래밍 언어의 역할을 설명해본다.
2. 파이썬을 컴퓨터에 설치한다.
3. 간단한 파이썬 프로그램을 작성한 후 실행해본다.
4. 터틀 그래픽으로 선과 도형을 그려본다.

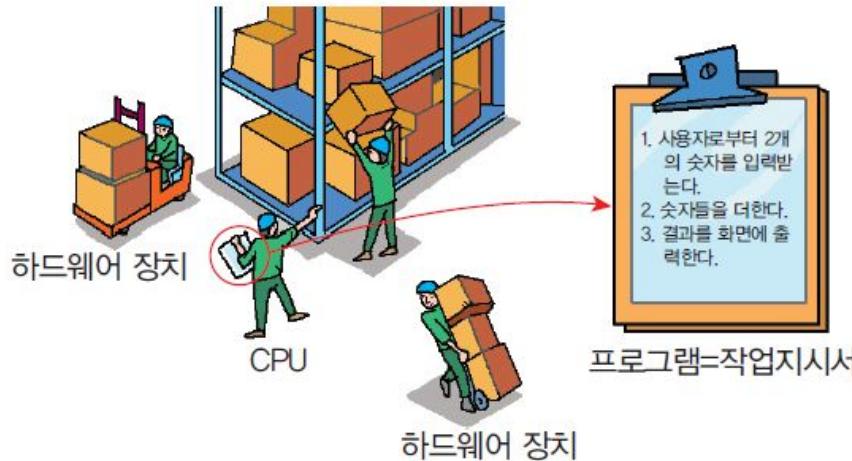


# 이론수업



컴퓨터에 일을 시키려면 인간이 컴퓨터에게 자세한 명령어(instruction)들의 리스트를 주어야 한다.

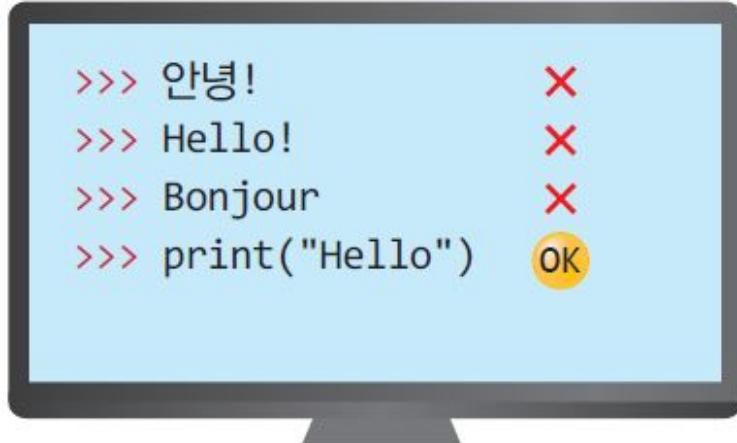
프로그램 (program) : 컴퓨터가 수행할 명령어를 적어놓은 문서





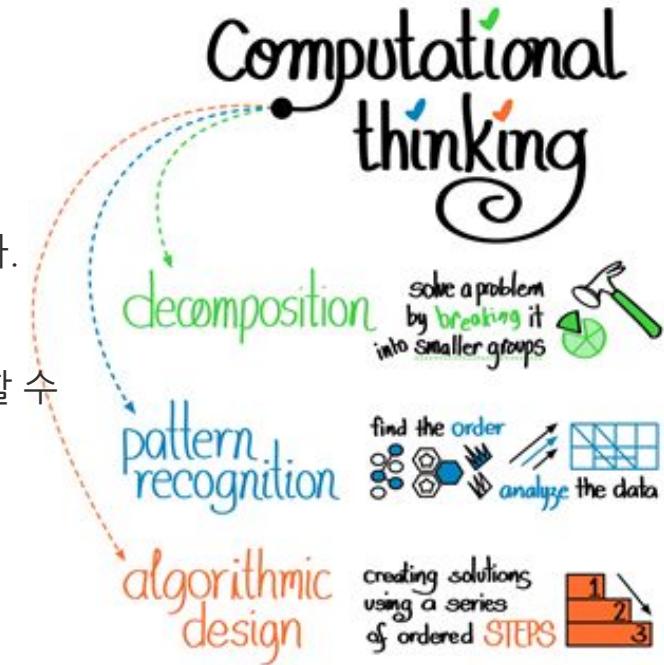
프로그래밍 언어

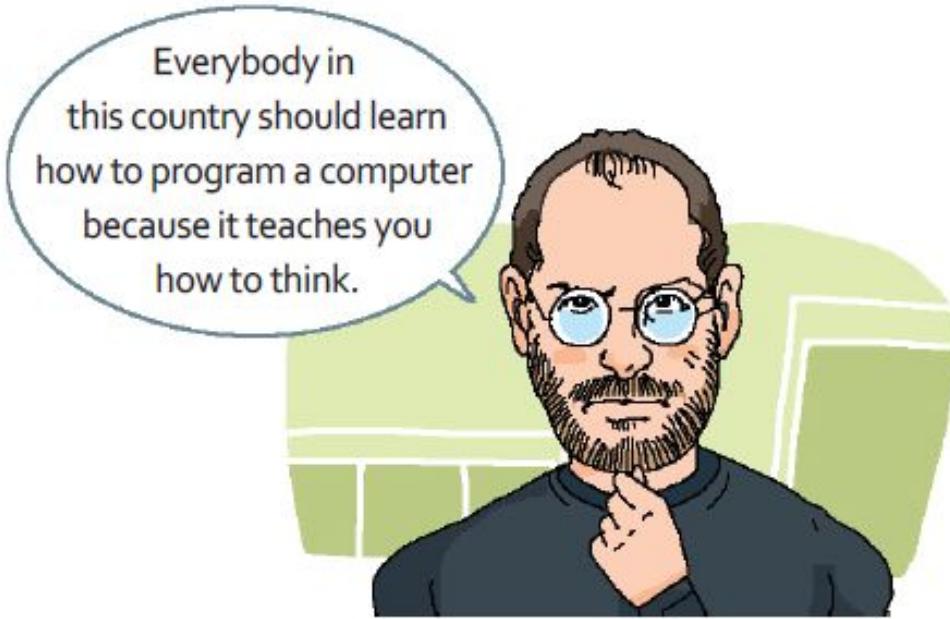




- 컴퓨터는 사람의 언어를 이해할 수 없다!
- '프로그래밍 언어'는 컴퓨터가 이해하는 언어이다.
- 컴퓨터가 알 수 있는 것은 '전기가 흐르'거나 혹은 '흐르지 않는' 것이다.

- 컴퓨터를 여러분 마음대로 제어할 수 있다.
- 자신이 해결해야 하는 일에 딱 맞는 프로그램을 작성할 수 있다.
- 프로그래밍을 하면 더 창의적인 사람이 된다(?).
- 프로그래밍을 하면 논리적으로 문제를 해결하는 능력을 배양할 수 있다.



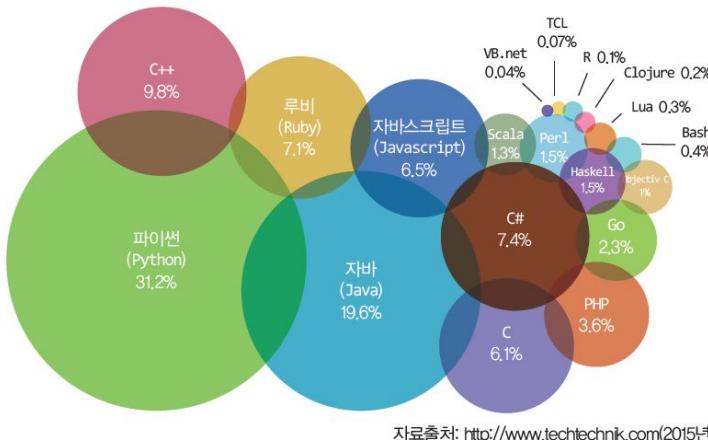


이 나라 모든 사람들이 컴퓨터 프로그래밍을  
배워야 하는 이유는 사고하는 법을 가르쳐주기 때문입니다.

– 스티브 잡스(Steve Jobs)

많이 사용되는 언어에는 '파이썬', '자바', 'C', 'BASIC' 등이 있다. 어셈블러와 기계어는 많이 사용되지 않는다.

인기있는 프로그래밍 언어



**TIOBE Programming Community Index**

Schedule a demo ≡

Rank	Rank Change	Programming Language	Ratings	Change
1	1	Python	14.51%	+0.59%
2	2	C	14.41%	+1.71%
3	3	Java	13.23%	+2.41%
4	4	C++	12.96%	+4.68%
5	5	C#	8.21%	+1.39%
6	6	Visual Basic	4.40%	-1.00%
7	7	JavaScript	2.10%	-0.31%
8	9	SQL	1.68%	-0.61%
9	10	PHP	1.36%	-0.28%
10	13	Go	1.28%	+0.20%

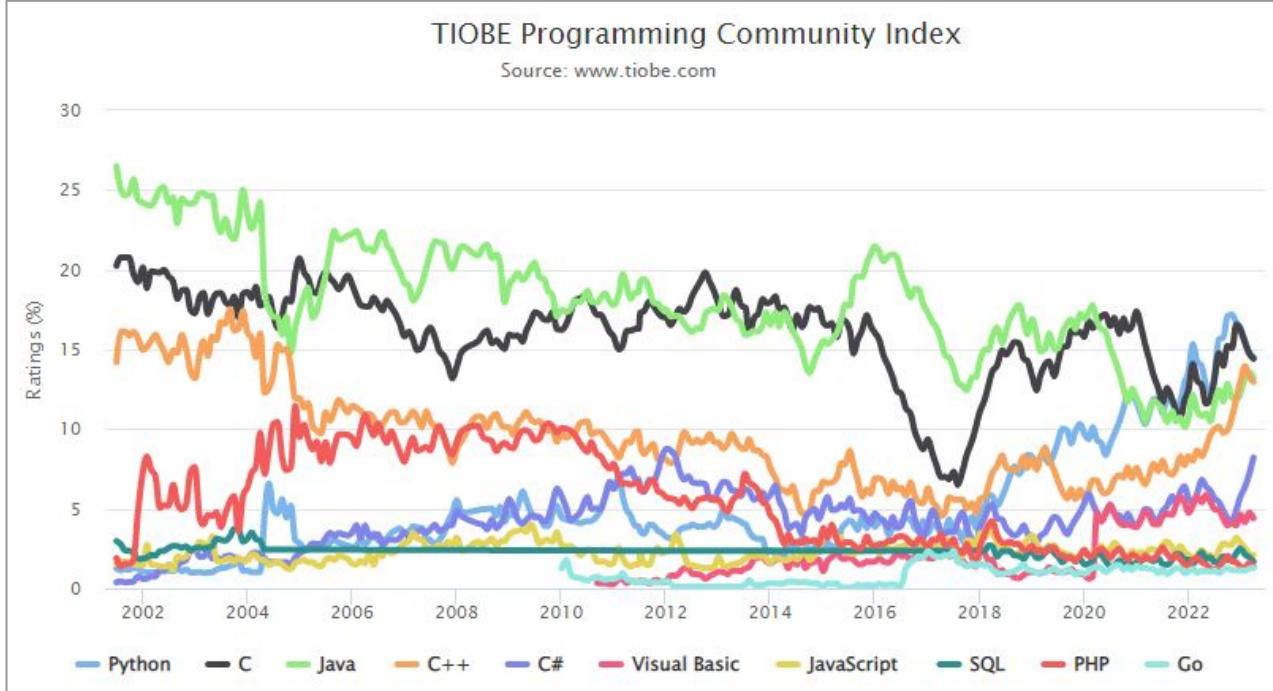
# 이론수업

## 프로그래밍 언어의 종류

TIOBE Programming Language Index

Schedule a demo

Apr 2023	Apr 2022	Change	Programming Language	Ratings	Change
1	1		Python	14.51%	+0.59%
2	2		C	14.41%	+1.71%
3	3		Java	13.23%	+2.41%
4	4		C++	12.96%	+4.68%
5	5		C#	8.21%	+1.39%
6	6		Visual Basic	4.40%	-1.00%
7	7		JavaScript	2.10%	-0.31%
8	9	▲	SQL	1.68%	-0.61%
9	10	▲	PHP	1.36%	-0.28%
10	13	▲	Go	1.28%	+0.20%
11	12	▲	Delphi/Object Pascal	1.23%	+0.05%
12	8	▼	ASM (language)	1.03%	-1.31%
13	16	▲	Classic Visual Basic	0.92%	+0.09%
14	20	▲	MATLAB	0.86%	+0.12%
15	24	▲	Scratch	0.79%	+0.13%
16	11	▼	R	0.76%	-0.79%

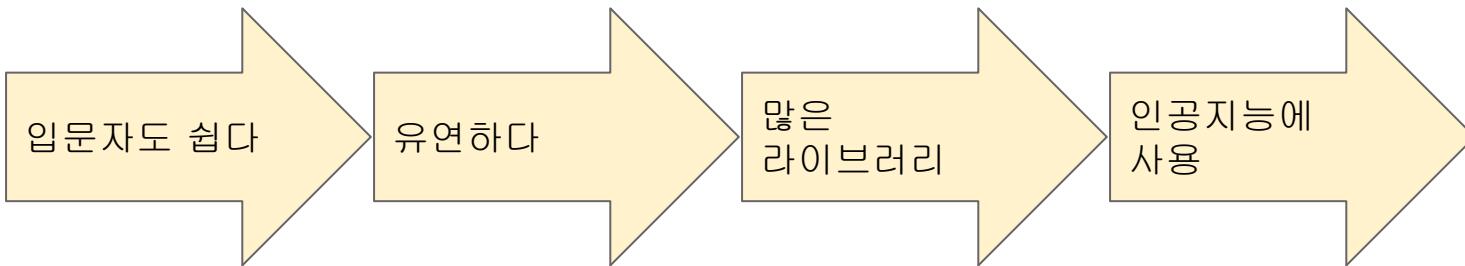




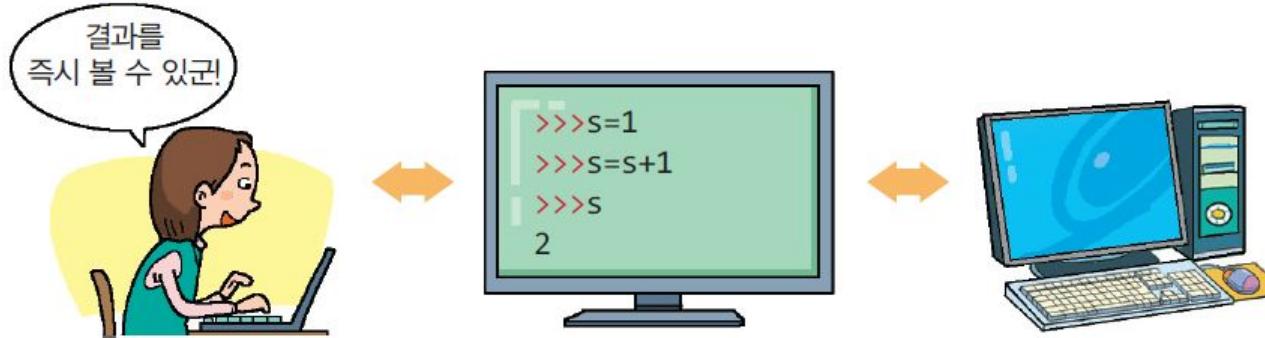
1991년에 귀도 반 로섬(Guido van Rossum)이 개발한 대화형  
프로그래밍 언어  
현재 파이썬은 속도 향상을 위한 개발 중 ...

**Life is short.  
You need Python.**

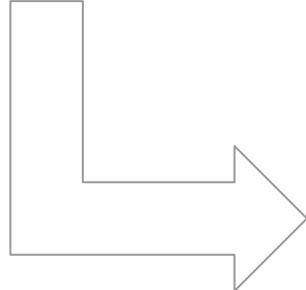
- 간결하고 읽기 쉬운 구문, 유용한 라이브러리, 쉬운 유지보수성
- 웹 개발, 시스템 자동화, 데이터 처리, 인공 지능 등 다양한 분야에서 사용
- 코드 작성이 다른 프로그래밍 언어보다 처음 시작하는 프로그래머들에게 쉽다.



- 생산성이 뛰어나다.
- 초보자한테 좋은 언어 - 인터프리터 언어 (VS 컴파일러 )



```
if "apple" in ["strawberry", "banana", "grape", "apple"]:  
    print("There is a apple")
```



- 직관적
- 다양한 플랫폼에서 사용 가능
- 많은 라이브러리
- 애니메이션, 그래픽을 쉽게 사용





YAHOO!

Google

YouTube



reddit

BitTorrent™

IBM



Dropbox



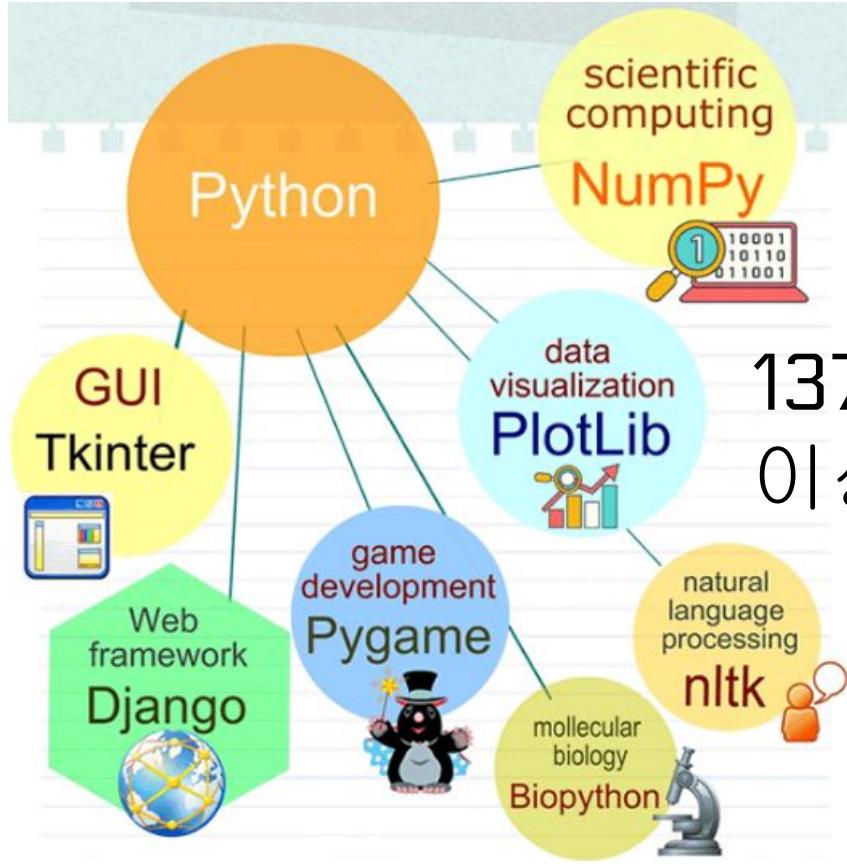
redhat

CANONICAL

NETFLIX

Quora





137,000 개 ?  
이상

## python 3.10 다운로드 &amp; 설치

## python A.B.C

- A - Major (3)
- B - Minor (10)
- C - Patch (최대)

The screenshot shows the Python Releases for Windows page. At the top, there's a navigation bar with links for About, Downloads, Documentation, Community, Success Stories, News, and Events. Below the navigation bar, the URL is shown as Python >>> Downloads >>> Windows.

## Python Releases for Windows

**Stable Releases**

- Latest Python 3 Release - Python 3.11.3
  - Download Windows embeddable package (32-bit)
  - Download Windows embeddable package (64-bit)
  - Download Windows help file
  - Download Windows installer (32-bit)
  - Download Windows installer (64-bit)
- Python 3.10.11 - April 5, 2023
  - Note that Python 3.10.11 cannot be used on Windows 7 or earlier.
  - Download Windows embeddable package (32-bit)
  - Download Windows embeddable package (64-bit)
  - Download Windows help file
  - Download Windows installer (32-bit)
  - Download Windows installer (64-bit)
- Python 3.11.3 - April 5, 2023
  - Note that Python 3.11.3 cannot be used on Windows 7 or earlier.
  - Download Windows embeddable package (32-bit)
  - Download Windows embeddable package (64-bit)
  - Download Windows embeddable package (ARM64)
  - Download Windows installer (32-bit)
  - Download Windows installer (64-bit)

**Pre-releases**

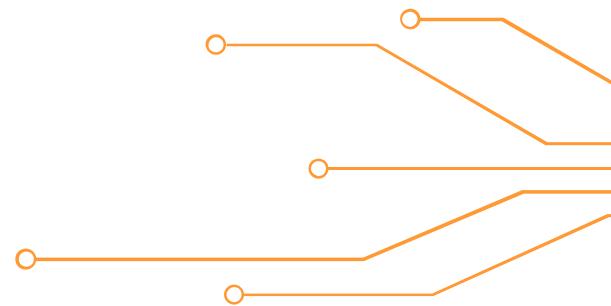
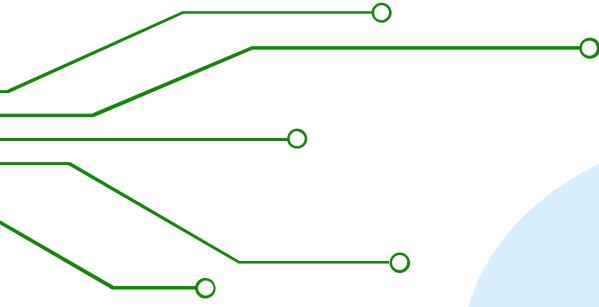
- Python 3.12.0a7 - April 4, 2023
  - Download Windows embeddable package (32-bit)
  - Download Windows embeddable package (64-bit)
  - Download Windows embeddable package (ARM64)
  - Download Windows installer (32-bit)
  - Download Windows installer (64-bit)
  - Download Windows installer (ARM64)
- Python 3.12.0a6 - March 8, 2023
  - Download Windows embeddable package (32-bit)
  - Download Windows embeddable package (64-bit)
  - Download Windows embeddable package (ARM64)
  - Download Windows installer (32-bit)
  - Download Windows installer (64-bit)
  - Download Windows installer (ARM64)
- Python 3.12.0a5 - Feb. 7, 2023

문자열 : "Hello"	숫자 : 20
문자열 : "World!"	숫자 : 30
문자열 + 문자열 = ?	숫자 + 숫자 = ?





실습



실습

파이썬 설치

# Python 3.10



실습

파이썬 기본 실행

# IDLE 실행



실습

Hello World

# Hello World!



## (도전) 한글 출력 :

- ❑ "안녕하세요?"
- ❑ "Programming에 입문하신것을 축하드립니다."



실습

파이썬을 계산기처럼 사용하기

# 간단한 계산



## (도전) 계산 :

- ❑  $3.141592 * 10.0 * 10.0$
- ❑  $(1/100) * 1234$



실습

문자열 합치기

"Hello" + "World"

실습

문자열 합치기

"Hello " \* 10

**"100" + "200"**    VS    **100 + 200**



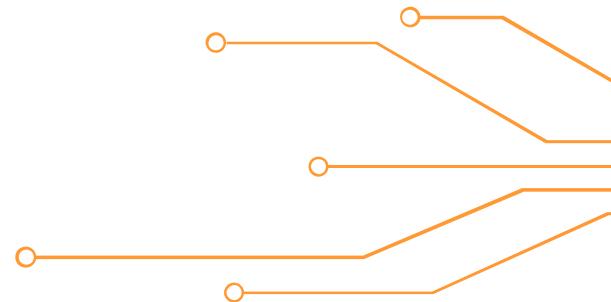
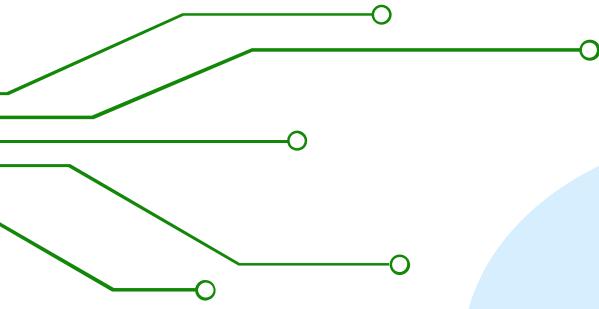
실습

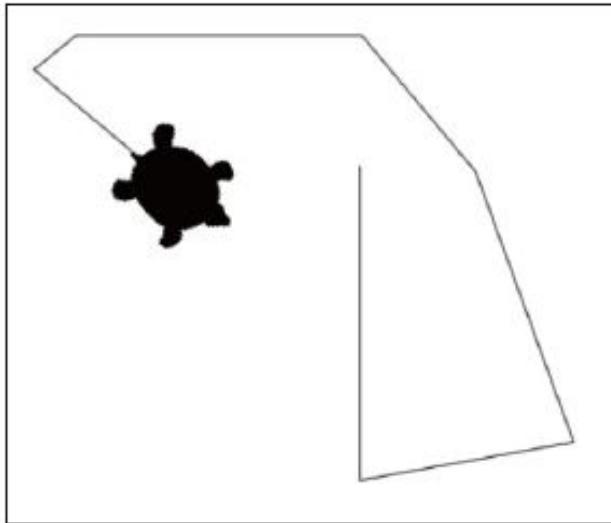
print() 함수

**print("The result is", 3.1)**



# 이론수업





터틀 그래픽은  
거북이가 화면 위에서  
지나가면서 그림을  
그리는 것이라고  
생각하세요.



- 터틀 그래픽을 시작
- 종이의 중앙에 거북이 등장
- 거북이를 움직이기 위한 명령 : "앞으로 가", "뒤로 가", "왼쪽으로 방향 전환해" 등
- 거북이에게 명령을 내리면  
거북이가 움직임.
- 거북이는 명령대로 움직이면서  
종이 위에 자취를 남김.



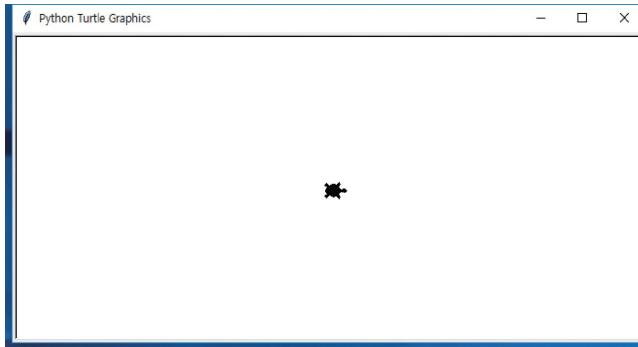
# 이론수업

## 시작

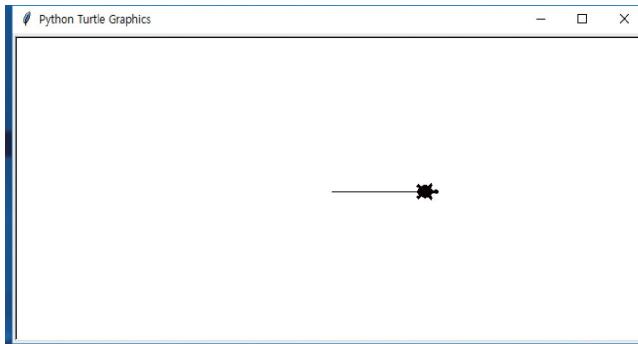
```
>>> import turtle
```

```
>>> t = turtle.Turtle()
```

```
>>> t.shape("turtle")
```



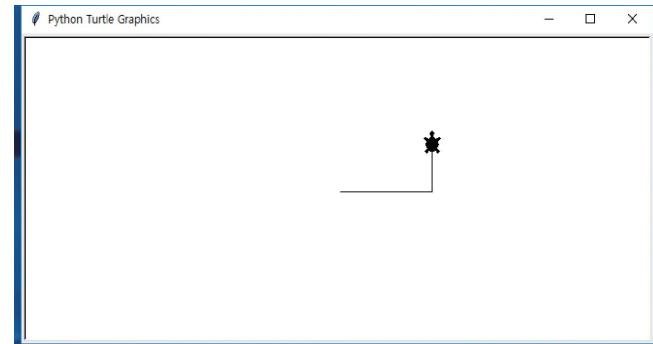
```
>>> t.forward(100)
```



```
>>> t.left(90)
```

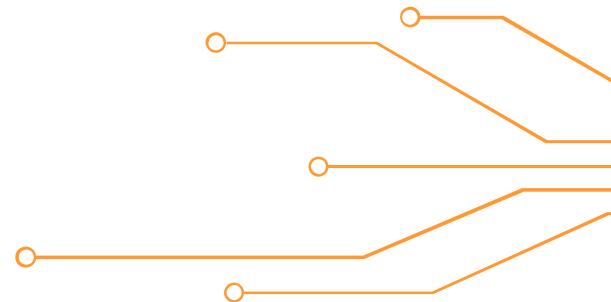
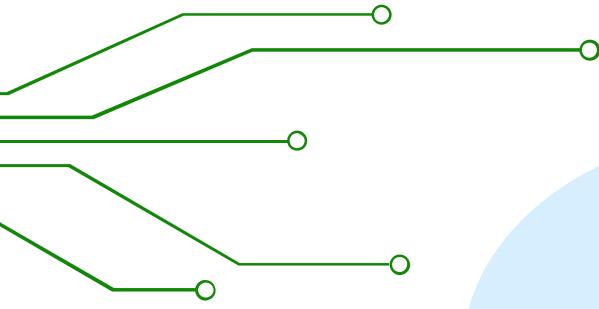
```
>>> t.forward(50)
```

```
>>> turtle.done()
```





# 이론수업



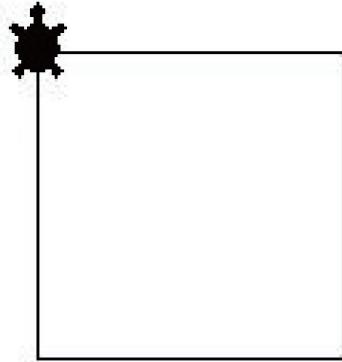
IDLE의 메뉴 중에서  
[Help]->[Turtle Demo] 를  
선택

The screenshot shows the Python IDLE environment with the "Turtle Demo" window open. On the left, the code for a digital clock is displayed in a text editor:

```
clock - a Python turtle graphics example
Examples Fontsize Help
^#! /usr/bin/env python3
# -*- coding: cp1252 -*-
"""
    turtle-example-suite:
        tdemo_clock.py
    Enhanced clock-program, showing date
    and time
    -----
    Press STOP to exit the program!
    -----
from turtle import *
from datetime import datetime
def jump(distanz, winkel=0):
    penup()
    right(winkel)
    forward(distanz)
    left(winkel)
    pendown()
def hand(laenge, spitze):
    fd(laenge*1.15)
    rt(90)
    fd(spitze/2.0)
    lt(120)
    fd(spitze)
    lt(120)
    fd(spitze)
    lt(120)
    fd(spitze/2.0)
def make_hand_shape(name, laenge, spitze):
    reset()
    jump(-laenge*0.15)
    begin_poly()
    hand(laenge, spitze)
    end_poly()
    hand_form = get_poly()
    register_shape(name, hand_form)
def clockface(radius):
    reset()
    pensize(7)
    for i in range(60):
        fd(radius)
```

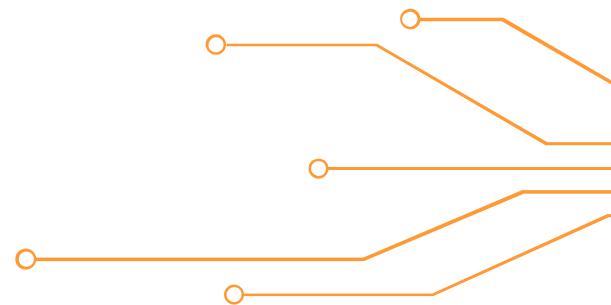
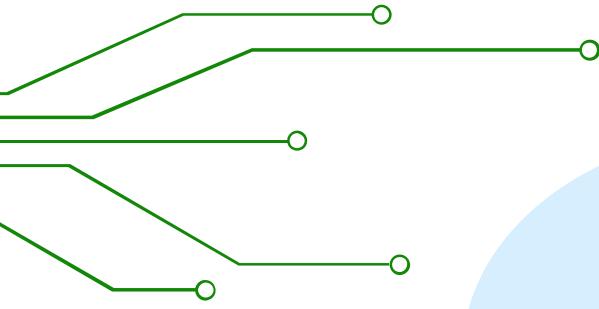
On the right, the turtle graphics window displays a digital clock interface. It features a large black circle representing the clock face. Three hands are drawn: a blue hour hand pointing towards the 12, a red minute hand pointing towards the 12, and a grey second hand pointing upwards. The text "Saturday" is written above the hands. Below the hands, the text "Dec. 31 2016" is displayed. At the bottom of the window, there are several buttons: "use mouse/keys or STOP" (in light blue), "START" (in orange), "STOP" (in red), and "CLEAR" (in orange). A small "Windows" logo is visible at the bottom right corner of the window.

Q : 터틀 그래픽에서 거북이를 전진, 회전하여 그림을 그리는 방법을 학습하였습니다. 사용되었던 명령어들을 사용해서 다음 사각형을 그려보세요. (사각형의 각 길이는 100 입니다)

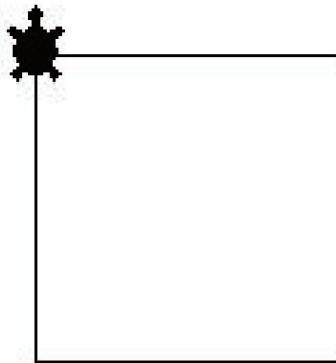




실습

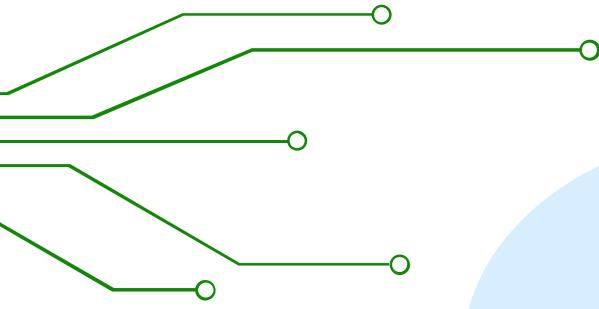


```
>>> import turtle  
>>> t = turtle.Turtle()  
>>> t.shape("turtle")  
>>> t.forward(100)  
>>> t.right(90)  
>>> t.forward(100)  
>>> t.right(90)  
>>> t.forward(100)  
>>> t.right(90)  
>>> t.forward(100)  
>>> turtle.done()
```

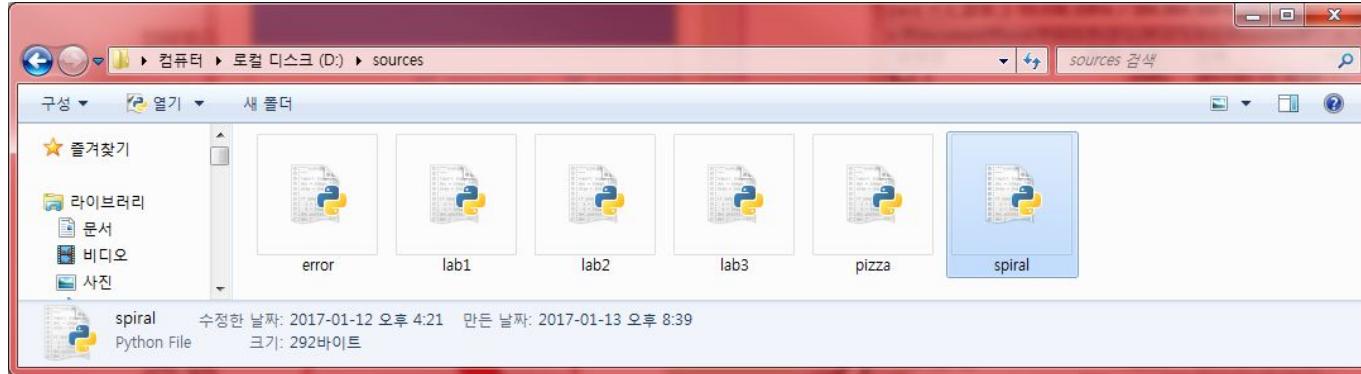




# 이론수업



- 복잡한 코드를 한줄씩 입력하는 것을 **\*비효율적\***이다.
- 텍스트 에디터를 이용, 명령어들을 파일에 저장한 후에 파일을 읽어 명령어들을 순서대로 실행하게 하자. 명령어 저장 파일을 소스파일(source file)이라고 부른다.



- ❑ IDE 메뉴 중 [File] - [New File] 선택 후 소스 코드 입력
- ❑ 입력이 마쳐진 후 [File] - [Save] 로 저장
- ❑ 실행은 [Run] - [Run Module]
- ❑ 다시 열때는 [File] - [Open]

```
>>> pront("Hello World")
SyntaxError: invalid syntax
```

```
>>> 1 +
SyntaxError: invalid syntax
```

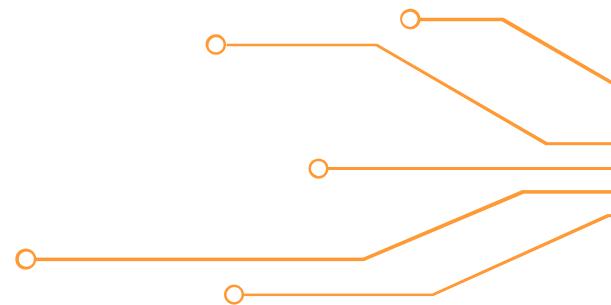
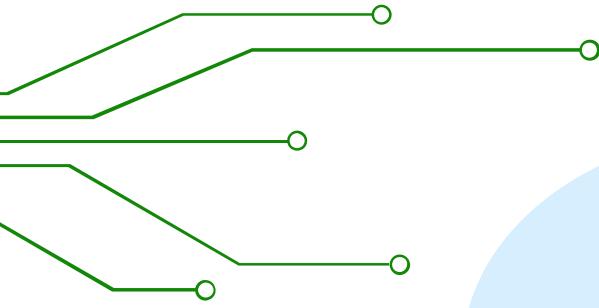
```
>>> 3 +* 2
SyntaxError: invalid syntax
```

WHY ?





실습



```
>>> print("Hello World")
```

*SyntaxError: invalid syntax*

```
>>> 1 +
```

*SyntaxError: invalid syntax*

```
>>> 3 +* 2
```

*SyntaxError: invalid syntax*

WHY ?



실습

## print() 함수 실습

다음 빈칸을 채워보세요.

```
>>> _____
```

Hello, Everybody?

```
>>> _____
```

I Love Python!

```
>>> _____
```

$9*8 = 72$

(72 를  $9*8$  의 결과가 들어가도록 코딩)



실습

컬러로 채워진 원을 그려봅시다.

t.fillcolor("blue") # 채우는 색상 지정

t.begin\_fill() # 채우기 시작

t.circle(100) # 반지름이 100인 원

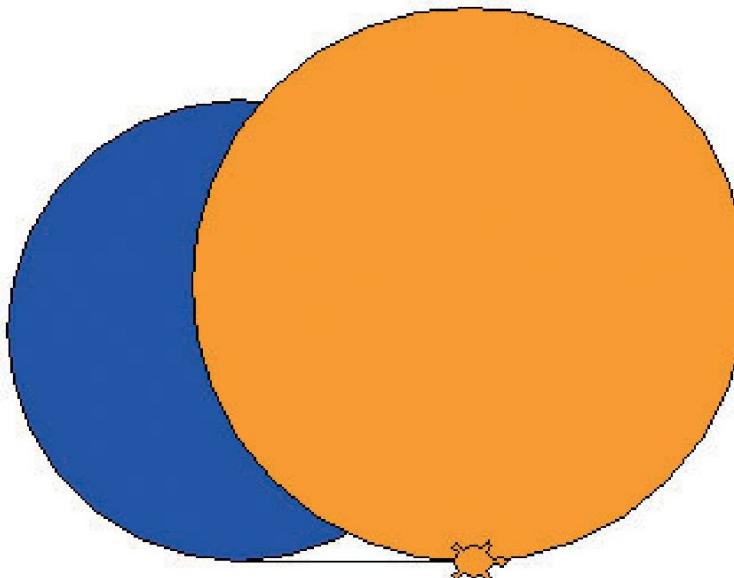
t.end\_fill() # 채우기 종료

거북이가 ...

반지름 100인 원(파란색)을 그린다.

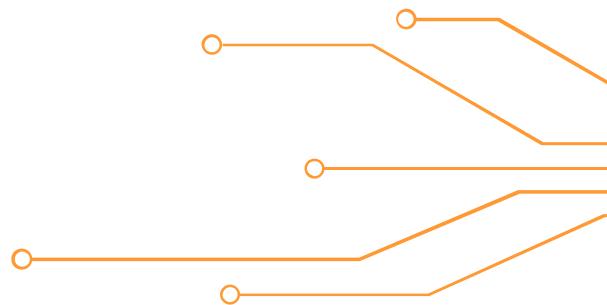
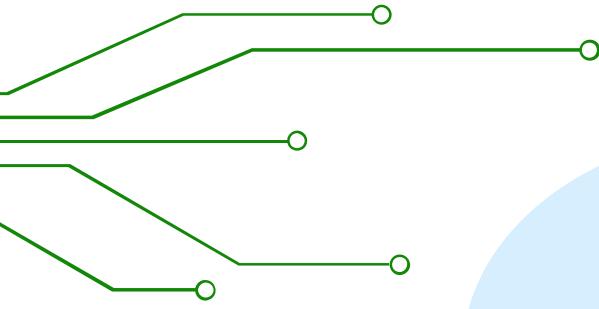
앞으로 100만큼 전진

다시 반지름 120인 원(오렌지색)을 그린다.



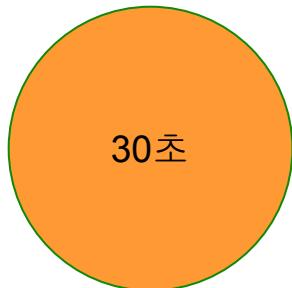


# 퀴즈



1. 다음 중 프로그래밍 언어에 대한 설명으로 틀린 것을 고르시오.

- 1) 컴퓨터에게 명령이나 연산등 인간이 원하는 작업을 할 수 있도록 지시하는 언어이다.
- 2) C언어, 자바, 파이썬, C# 등 많은 언어가 있다.
- 3) 고급언어와 저급언어로 나누어지며 고급언어는 인간 중심의 언어, 저급언어는 컴퓨터 중심의 언어이다.
- 4) 가장 인기있는 언어는 기계중심의 어셈블러언어이다.



1. 다음 중 프로그래밍 언어에 대한 설명으로 틀린 것을 고르시오. [ 4 ]
- 1) 컴퓨터에게 명령이나 연산등 인간이 원하는 작업을 할 수 있도록 지시하는 언어이다.
  - 2) C언어, 자바, 파이썬, C# 등 많은 언어가 있다.
  - 3) 고급언어와 저급언어로 나누어지며 고급언어는 인간 중심의 언어, 저급언어는 컴퓨터 중심의 언어이다.
  - 4) 가장 인기있는 언어는 기계중심의 어셈블러언어이다.**



1. 다음 중 프로그래밍 언어에 대한 설명으로 틀린 것을 고르시오. [ 4 ]

- 1) 컴퓨터에게 명령이나 연산등 인간이 원하는 작업을 할 수 있도록 지시하는 언어이다.
- 2) C언어, 자바, 파이썬, C# 등 많은 언어가 있다.
- 3) 고급언어와 저급언어로 나누어지며 고급언어는 인간 중심의 언어, 저급언어는 컴퓨터 중심의 언어이다.
- 4) 가장 인기있는 언어는 기계중심의 어셈블러언어이다.

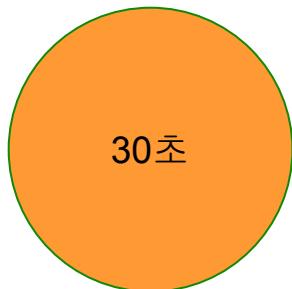
어셈블러언어는 기계중심 언어이기 때문에 배우고 사용하기에 상당한 어려움이 있어 특수한 경우를 제외하고는 잘 사용되지 않습니다.





2. 다음 중 파이썬 언어에 대한 설명으로 틀린 것을 고르시오.

- 1) 귀도 반 로썸이 만들었다.
- 2) 기계어 실행 코드를 만드는 컴파일러로 뛰어난 속도를 자랑한다.
- 3) 처음 개발 입문자가 사용하기에 적합한 쉬운 언어이다.
- 4) 많은 라이브러리가 있어 손쉽게 사용할 수 있다.



2. 다음 중 파이썬 언어에 대한 설명으로 틀린 것을 고르시오. [ 2 ]

1) 귀도 반 로썸이 만들었다.

**2) 기계어 실행 코드를 만드는 컴파일러로 뛰어난 속도를 자랑한다.**

3) 처음 개발 입문자가 사용하기에 적합한 쉬운 언어이다.

4) 많은 라이브러리가 있어 손쉽게 사용할 수 있다.



2. 다음 중 파이썬 언어에 대한 설명으로 틀린 것을 고르시오. [ 2 ]

파이썬은 컴파일러가 아닌  
인터프리터 언어입니다.

1) 귀도 반 로썸이 만들었다.

**2) 기계어 실행 코드를 만드는 컴파일러로 뛰어난 속도를 자랑한다.**

3) 처음 개발 입문자가 사용하기에 적합한 쉬운 언어이다.

4) 많은 라이브러리가 있어 손쉽게 사용할 수 있다.

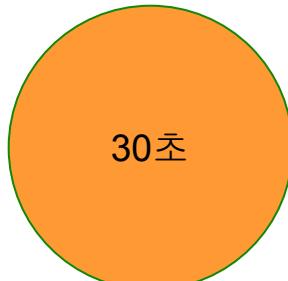




3. 다음 파이썬 명령문의 실행 결과는 어떻게 될까?

```
>>> print("Hello" + "World" + "5*20")
```

- 1) HelloWorld100
- 2) Hello World 100
- 3) HelloWorld5\*20
- 4) Hello World 520



3. 다음 파이썬 명령문의 실행 결과는 어떻게 될까? [ 3 ]

```
>>> print("Hello" + "World" + "5*20")
```

- 1) HelloWorld100
- 2) Hello World 100
- 3) HelloWorld5\*20**
- 4) Hello World 520



### 3. 다음 파이썬 명령문의 실행 결과는 어떻게 될까? [ 3 ]

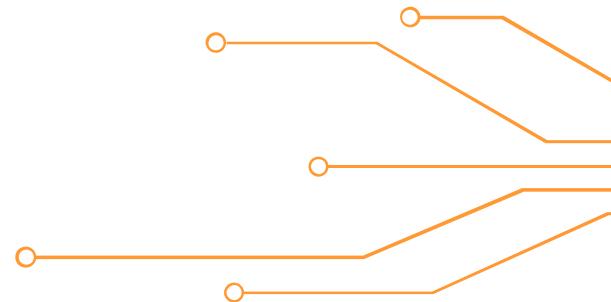
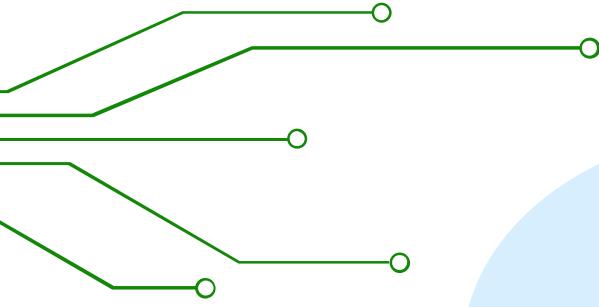
```
>>> print("Hello" + "World" + "5*20")
```

- 1) HelloWorld100
- 2) Hello World 100
- 3) HelloWorld5\*20**
- 4) Hello World 520

"Hello", "World", "5\*20"이 모두 문자열인것을 주의하세요. 문자열과 문자열 사이에 "+" 연산자가 있으면, 두 문자열은 하나로 합쳐집니다. 그래서 결과는 "**HelloWorld5\*20**" 이 됩니다.



# 핵심정리



- ❑ 프로그래밍 언어는 컴퓨터가 이해할 수 있는 언어입니다.
- ❑ 프로그래밍 언어는 개발자의 의도를 컴퓨터에 전달하는 번역기 역할을 합니다.
- ❑ 파이썬은 무료 프로그램으로 누구나 컴퓨터에 설치할 수 있습니다.
- ❑ IDLE은 파이썬으로 프로그램을 작성하기 위한 기본 개발 환경입니다.
- ❑ 파이썬 쉘에서 코드를 입력하고 [ENTER]를 눌러 한줄씩 실행할 수 있습니다.



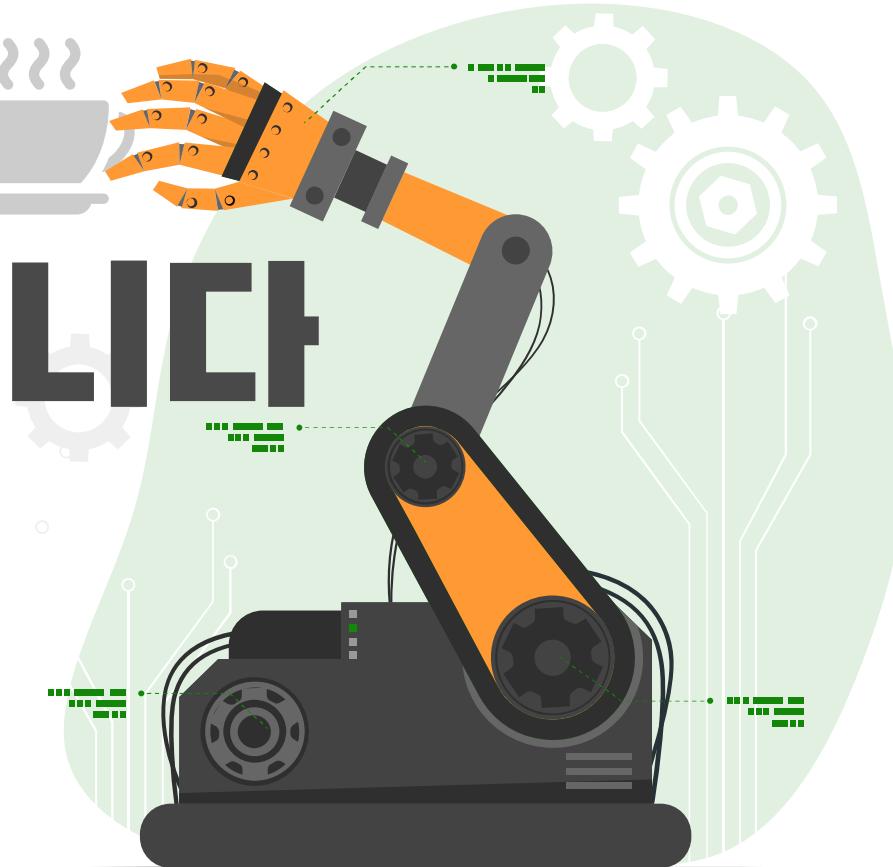
## 핵심정리

## 이번 장에서 배운 것은 ...

- 파이썬 코드를 파일에 저장하면 순차적으로 자동 실행할 수 있습니다.
- 터틀 그래픽은 거북이에 펜을 매달아서 그림을 그리는 개념입니다. 거북이를 이동시키면 자국에 그림이 그려 집니다.
- print()는 화면에 문자열이나 계산 결과를 출력할 수 있습니다.

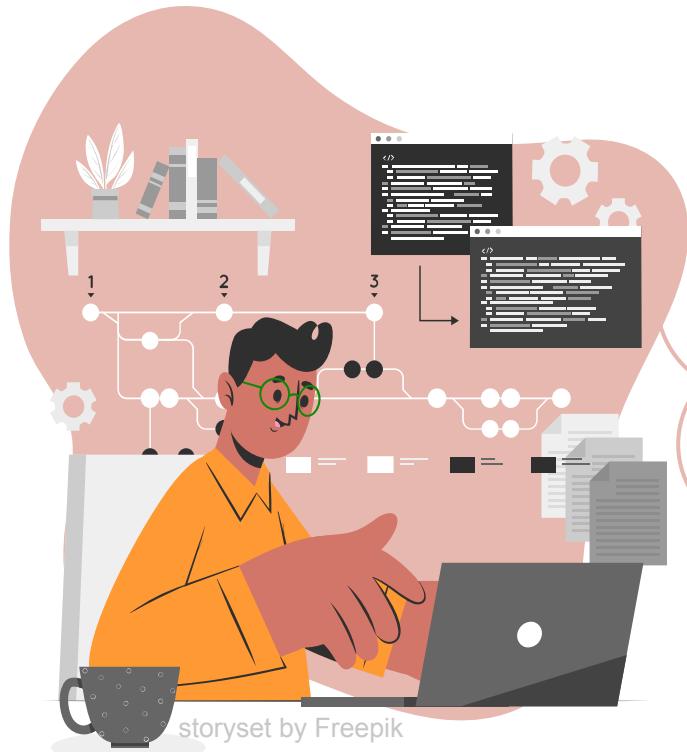


# 수고하셨습니다





# 컴퓨팅 사고와 프로그래밍



CH 02.

변수

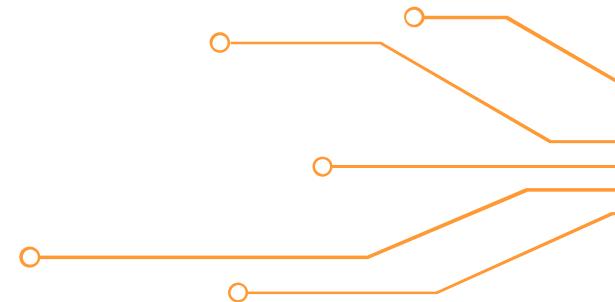
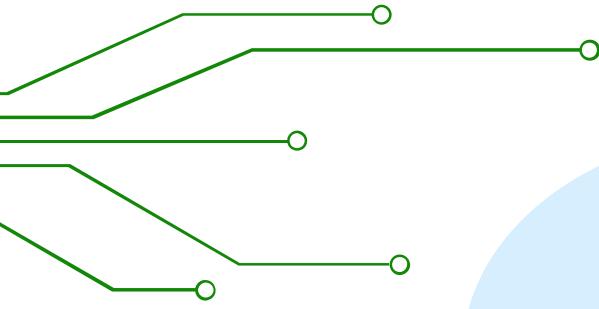
# 학습목표



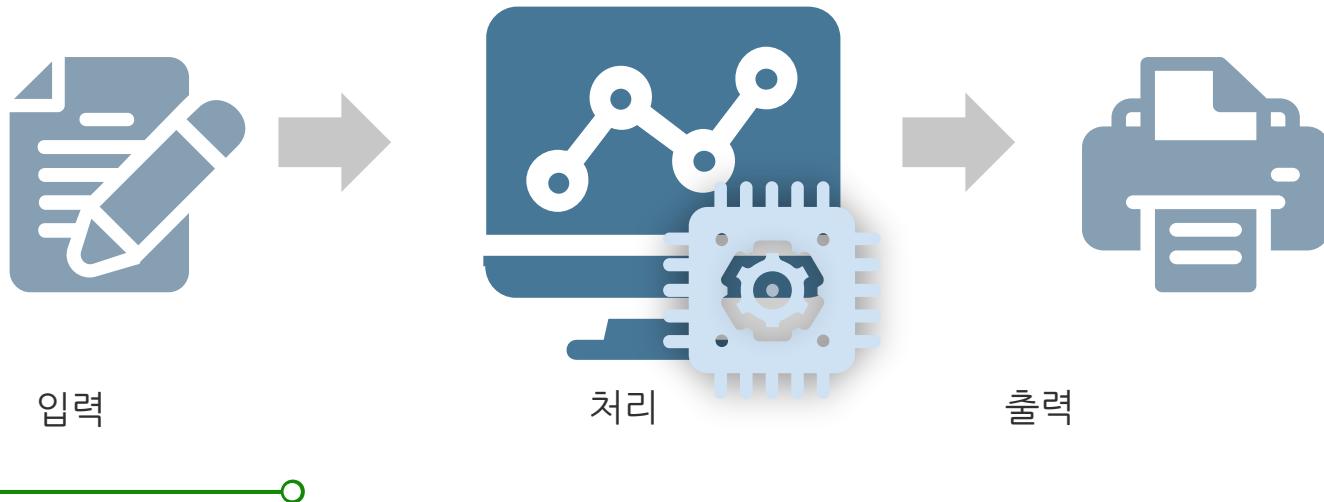
1. 변수가 무엇인지 이해한다.
2. 파이썬 콘솔에서 입력과 출력을 한다.
3. 간단한 계산기 프로그램을 작성해 본다.
4. `print()` 함수를 이용하여 원하는 내용을 원하는 포맷으로 출력한다.



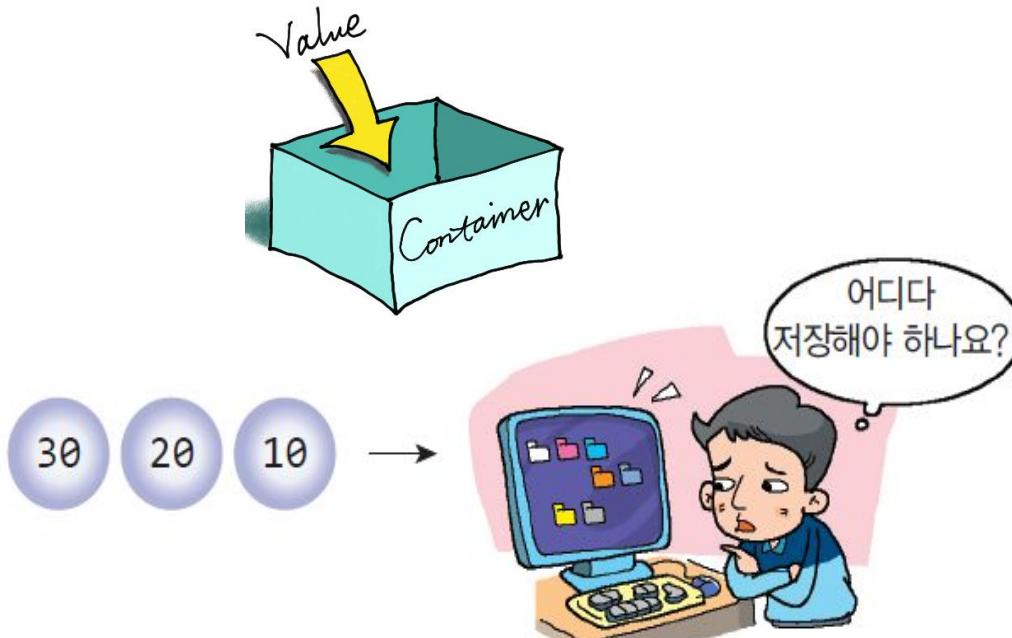
# 이론수업



1. 일반적인 프로그램은 외부로부터 데이터를 받아서(입력단계),
2. 데이터를 처리한 후에(처리단계),
3. 결과를 화면에 출력(출력단계) 합니다.

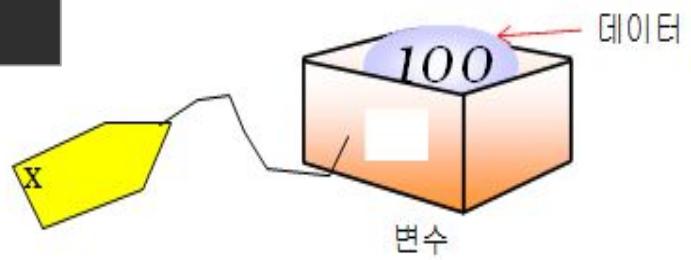


1. 변수(variable)는 값을 저장하는 상자로 생각할 수 있다.
2. 변수는 컴퓨터 메모리 공간에 만들어 진다.



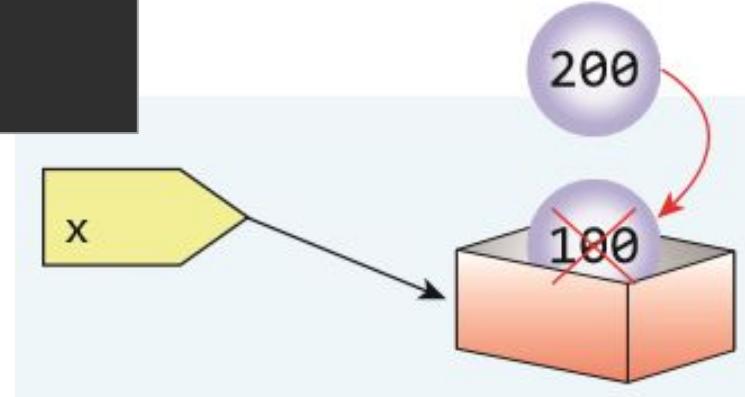
파이썬에서 변수를 생성하려면 다음과 같이 한다.

```
>>> x = 100
```



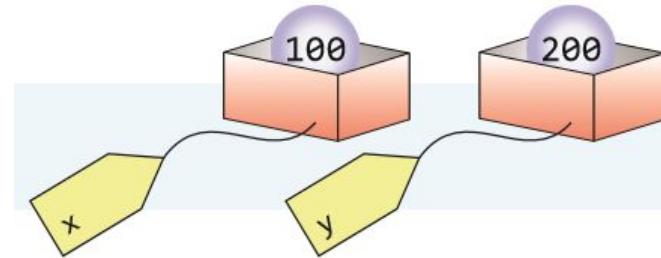
생성된 변수에는 얼마든지 다른 값을 저장할 수 있다.

```
>>> x = 100  
>>> x = 200  
>>> print(x)  
200
```



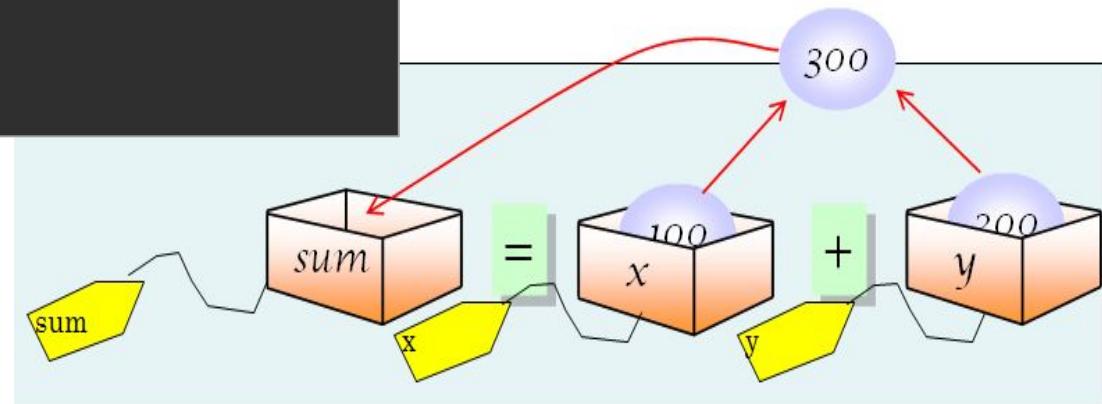
생성된 변수에는 얼마든지 다른 값을 저장할 수 있다.

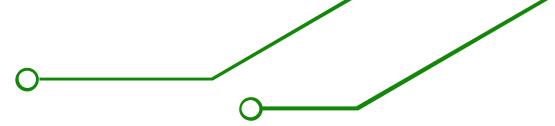
```
>>> x = 100  
>>> y = 200
```



변수에 담긴 값을 이용해서 계산할 수 있다.

```
>>> x = 100  
>>> y = 200  
>>> sum = x + y  
>>> print(sum)  
300
```





# 주의 !!!

## 경고

입문자들이 가장 많이 틀리는 부분은 '=' 기호를 '양변이 같다'고 해석하는 것입니다. 대부분의 프로그래밍 언어에서 '=' 기호는 오른쪽에 있는 값을 왼쪽에 저장한다는 뜻입니다. 좌우가 같다는 의미의 기호는 '==' 처럼 2개의 등호기호를 사용합니다.



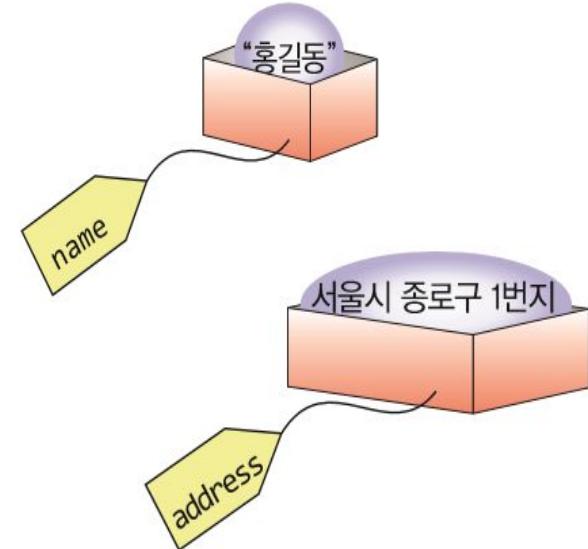
## 이론수업

## 문자열도 변수에 저장할 수 있다!

파이썬의 변수에는 정수뿐만 아니라 문자열도 저장할 수 있다.

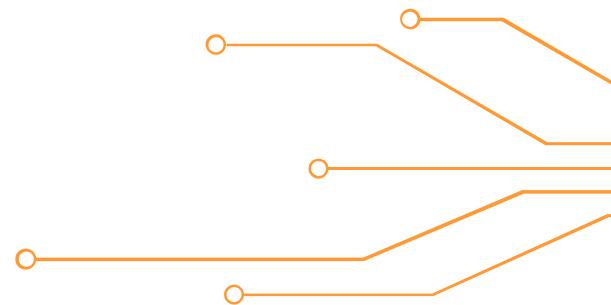
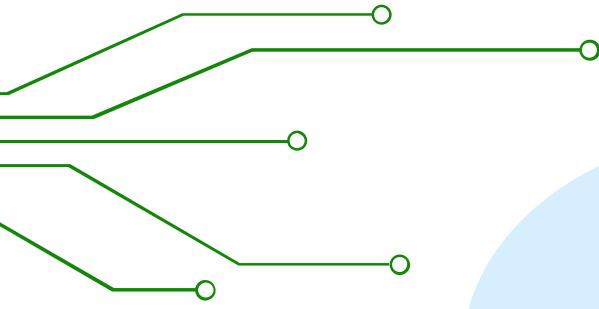
```
>>> name = "홍길동"  
>>> address = "서울시 종로구 1번지"
```

```
>>> print(name)  
홍길동  
>>> print(address)  
서울시 종로구 1번지
```





실습



실습

(도전) 아래 실행 결과를 예상하고 실행시켜보세요

```
>>> x = 7  
>>> y = 6  
>>> print(x+y)
```

```
>>> x = '7'  
>>> y = '6'  
>>> print(x+y)
```



실습

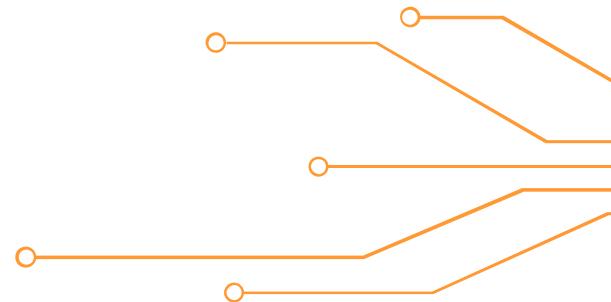
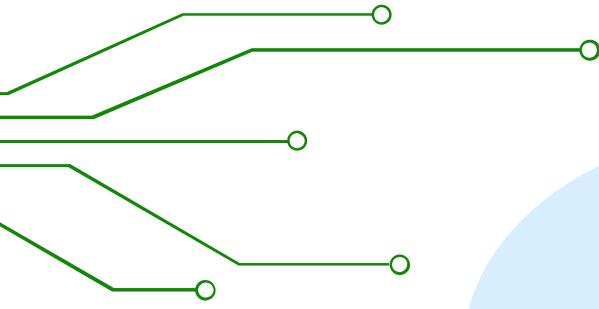
이름, 주소, 전화번호를 변수에 담아 출력해보기

이름, 주소, 전화번호를  
각각 name, address, phone 변수에 담은 후  
한줄로 출력해보세요.

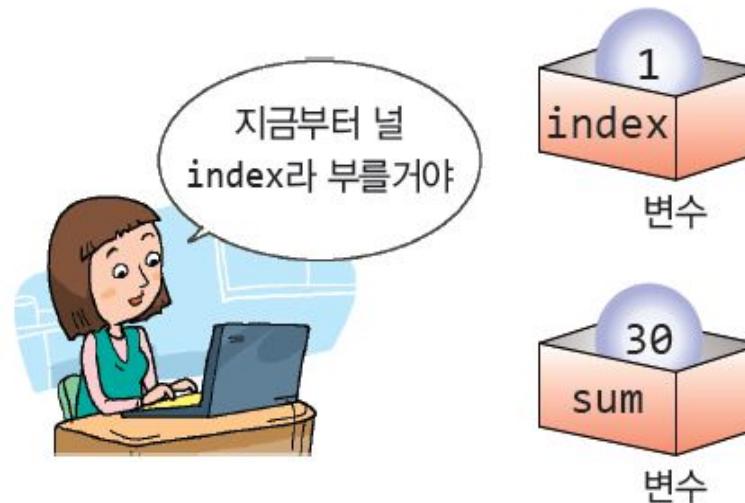
ex) 산타클로스 / 핀란드 로바니에미 / 555-5555



# 이론수업



- 의미 있는 이름을 사용
- 소문자와 대문자는 서로 다르게 취급된다.
- 변수의 이름은 영문자와 숫자, 밑줄(\_)로 이루어진다.
- 변수의 이름 중간에 공백이 들어가면 안 된다. 단어를 구분하려면 밑줄(\_)을 사용 한다.
- 첫번째 문자는 숫자가 들어갈 수 없다.



sum	# 영문 알파벳 문자로 시작
_count	# 밑줄 문자로 시작할 수 있다.
number_of_pictures	# 중간에 밑줄 문자를 넣을 수 있다.
King3	# 맨 처음이 아니라면 숫자도 넣을 수 있다.



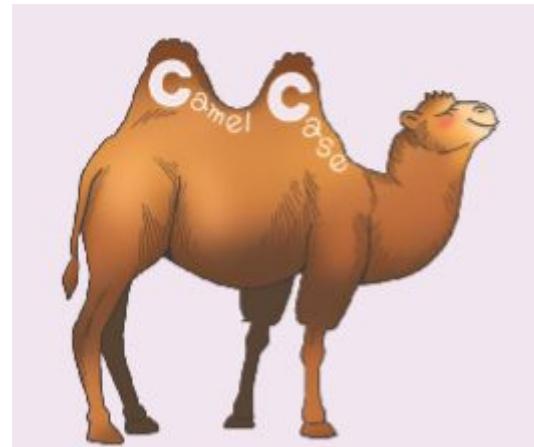
2nd_base	# (X) 숫자로 시작할 수 없다.
money#	# (X) # 같은 기호는 사용할 수 없다.



- 단어를 구분하려면 밑줄(\_)을 사용 한다.
- 낙타체는 변수의 첫 글자는 소문자로, 나머지 단어의 첫 글자는 대문자로 적는 방법이다.

my\_car

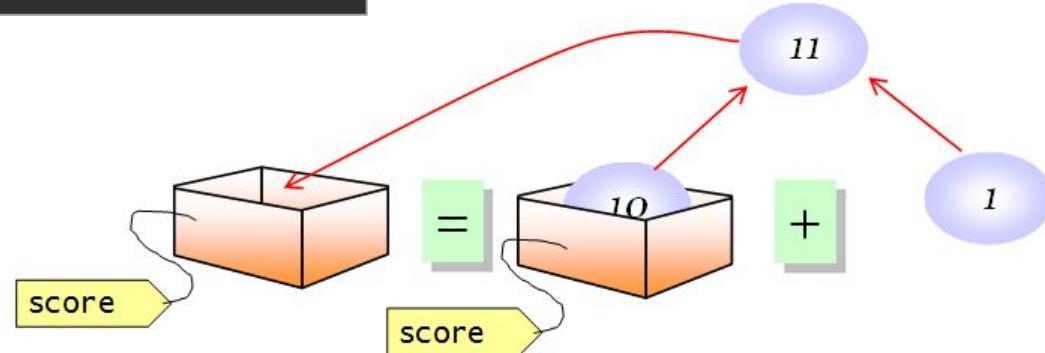
student\_name



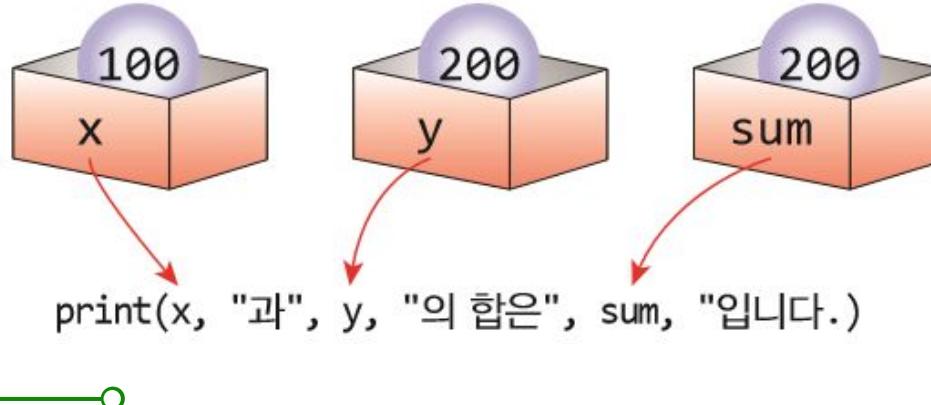
myCar

studentName

```
>>> score = 10  
>>> score = score + 1  
>>> print(score)
```

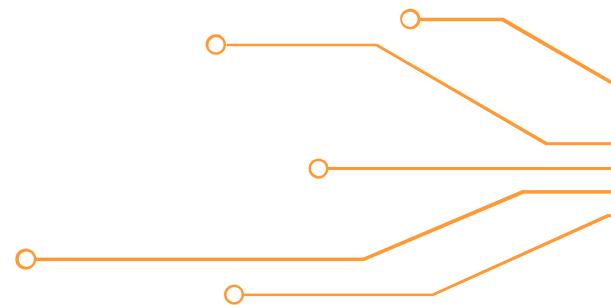
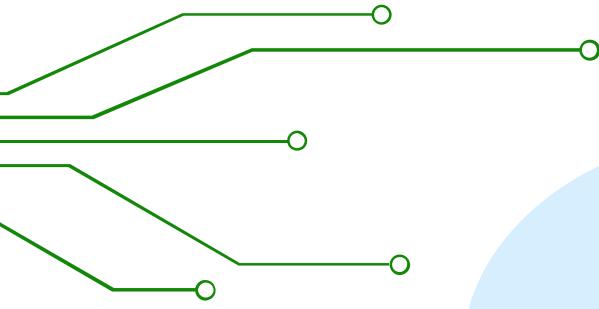


```
>>> x = 100  
>>> y = 200  
>>> sum = x + y  
>>> print(x, "과", y, "의 합은", sum, "입니다.")  
100 과 200 의 합은 300 입니다.
```

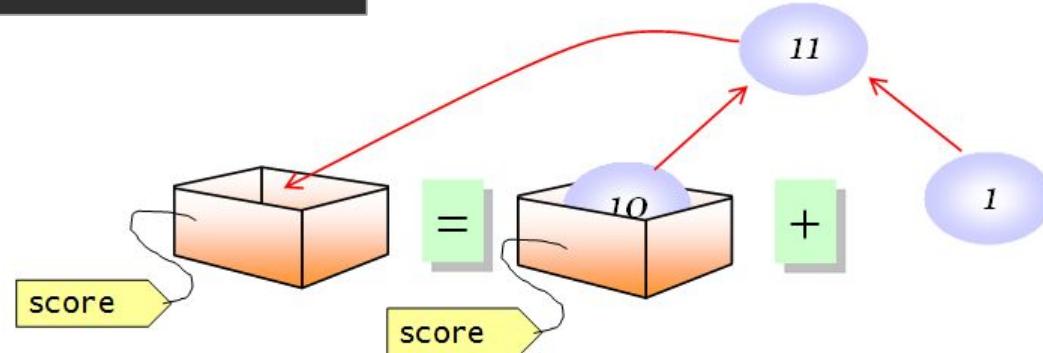




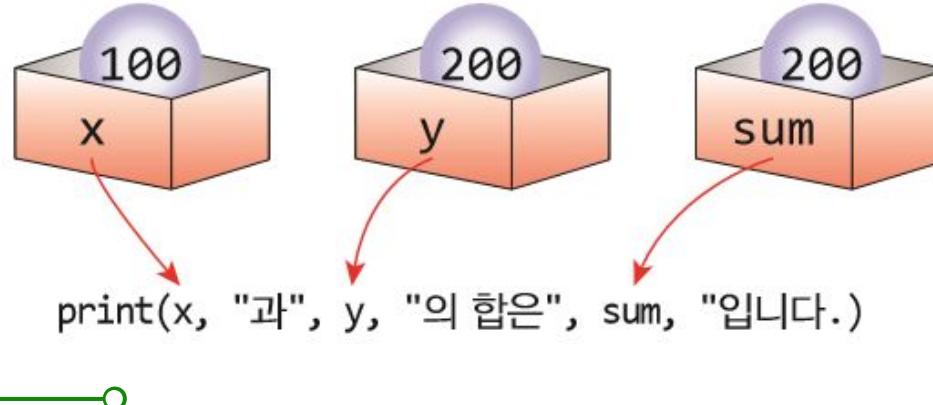
실습



```
>>> score = 10  
>>> score = score + 1  
>>> print(score)
```

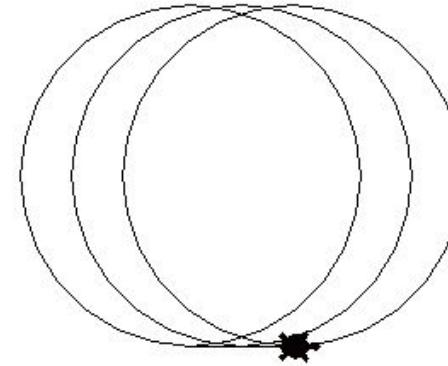


```
>>> x = 100  
>>> y = 200  
>>> sum = x + y  
>>> print(x, "과", y, "의 합은", sum, "입니다.")  
100 과 200 의 합은 300 입니다.
```



다음과 같이 터틀 그래픽을 사용하여 반지름이 100픽셀인 3개의 원을 그리는 프로그램이 있다.

```
import turtle  
t = turtle.Turtle()  
t.shape("turtle")  
  
radius = 100  
t.circle(radius) # 반지름이 100인 원이 그려 진다.  
t.fd(30)  
t.circle(radius) # 반지름이 100인 원이 그려 진다.  
t.fd(30)  
t.circle(radius) # 반지름이 100인 원이 그려 진다.
```



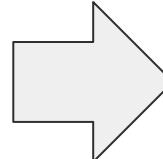
원의 반지름을 50으로 변경하여서 다시 그려야 한다면 ?

-> 원의 반지름이 변수로 표현되었기 때문에 쉬운 방법이 있다. **변수의 내용만 변경하면 된다.**



OLD

```
t.circle(100)  
t.fd(30)  
t.circle(100)  
t.fd(30)  
t.circle(100)
```



NEW

```
radius = 100  
  
t.circle(radius)  
t.fd(30)  
t.circle(radius)  
t.fd(30)  
t.circle(radius)
```

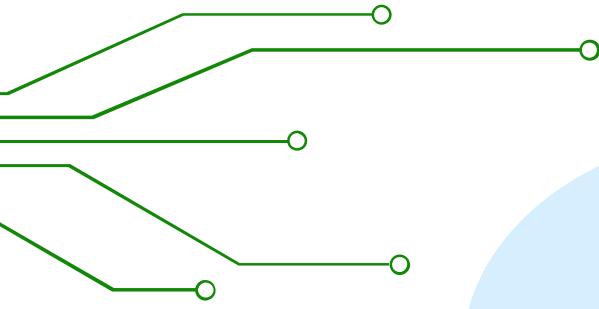


```
import turtle  
t = turtle.Turtle()  
t.shape("turtle")  
  
radius = 50  
t.circle(radius) # 반지름이 50인 원이 그려 진다.  
t.fd(30)  
t.circle(radius) # 반지름이 50인 원이 그려 진다.  
t.fd(30)  
t.circle(radius) # 반지름이 50인 원이 그려 진다.
```





# 이론수업



## 이론수업

## 사용자로부터 정수 입력받기

사용자로부터 정수 2개를  
받아서 덧셈을 한 후에  
결과를 출력

첫 번째 정수를 입력하시오: 300  
두 번째 정수를 입력하시오: 400  
300 과 400 의 합은 700 입니다.

### input() 사용법

변수

사용자가 입력한 문자열을 숫자로 변환한다.

```
x = int(input("첫 번째 정수를 입력하시오: "))
```

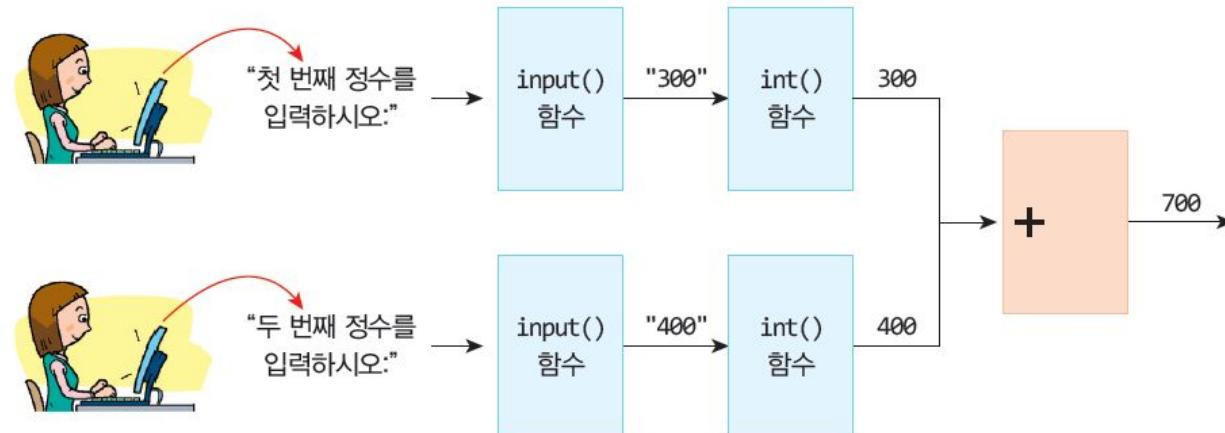
안내 메시지를 출력하고 사용자가 입력한 값을 문자열 형태로 받는다.

```
x = int(input("첫 번째 정수를 입력하시오: "))
y = int(input("두 번째 정수를 입력하시오: "))
sum = x + y
print(x, "과", y, "의 합은", sum, "입니다.")
```



첫 번째 정수를 입력하시오: 300  
두 번째 정수를 입력하시오: 400  
300 과 400 의 합은 700 입니다.

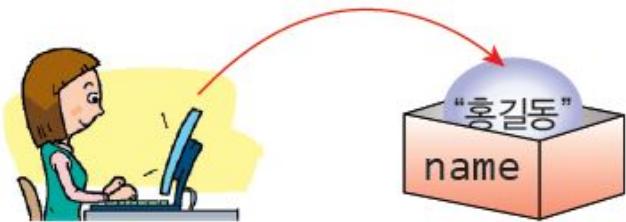
```
x = int(input("첫 번째 정수를 입력하시오: "))
y = int(input("두 번째 정수를 입력하시오: "))
sum = x + y
print(x, "과", y, "의 합은", sum, "입니다.")
```



## 이론수업

## 사용자로부터 문자열 입력받기

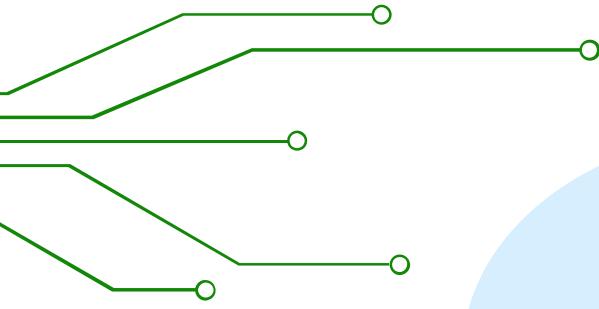
```
name = input("이름을 입력하시오: ")  
print(name, "씨, 안녕하세요?")  
print("파이썬에 오신 것을 환영합니다.")
```



이름을 입력하시오: 홍길동  
홍길동 씨, 안녕하세요?  
파이썬에 오신 것을 환영합니다.



실습



## 실습

## 사용자로부터 정수 입력받기

사용자로부터 정수 2개를  
받아서 덧셈을 한 후에  
결과를 출력

첫 번째 정수를 입력하시오: 300  
두 번째 정수를 입력하시오: 400  
300 과 400 의 합은 700 입니다.

**input() 사용법**

변수

사용자가 입력한 문자열을 숫자로 변환한다.

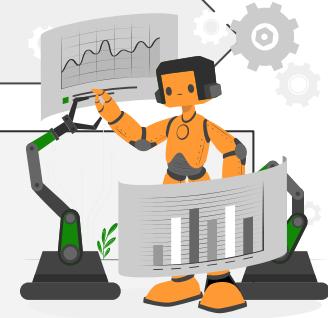
```
x = int(input("첫 번째 정수를 입력하시오: "))
```

안내 메시지를 출력하고 사용자가 입력한 값을 문자열 형태로 받는다.

## 실습

## 완전한 코드

```
x = int(input("첫 번째 정수를 입력하시오: "))
y = int(input("두 번째 정수를 입력하시오: "))
sum = x + y
print(x, "과", y, "의 합은", sum, "입니다.")
```



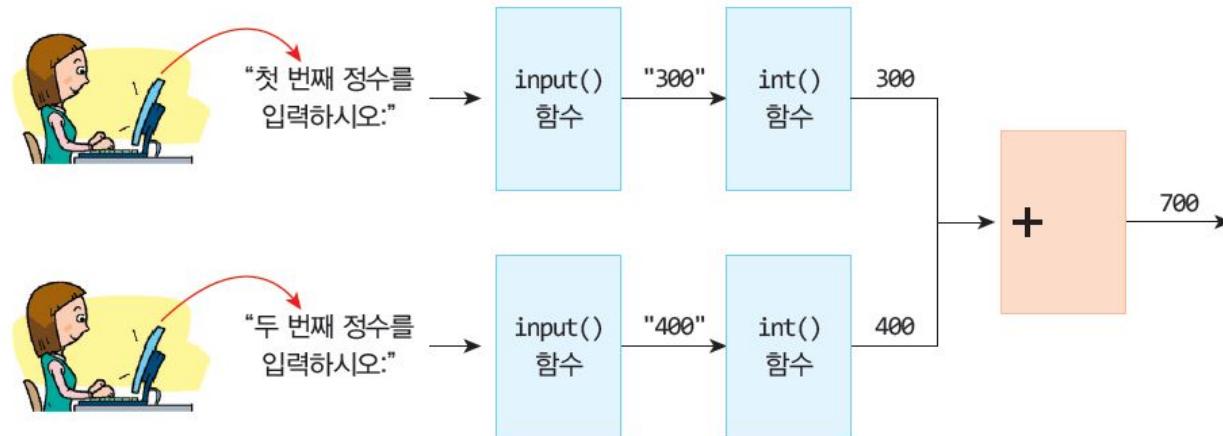
첫 번째 정수를 입력하시오: 300  
두 번째 정수를 입력하시오: 400  
300 과 400 의 합은 700 입니다.



## 실습

## 완전한 코드

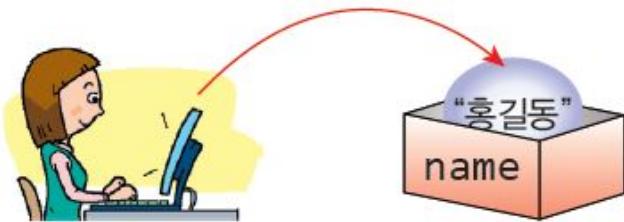
```
x = int(input("첫 번째 정수를 입력하시오: "))
y = int(input("두 번째 정수를 입력하시오: "))
sum = x + y
print(x, "과", y, "의 합은", sum, "입니다.")
```



## 실습

## 사용자로부터 문자열 입력받기

```
name = input("이름을 입력하시오: ")  
print(name, "씨, 안녕하세요?")  
print("파이썬에 오신 것을 환영합니다.")
```



이름을 입력하시오: 홍길동  
홍길동 씨, 안녕하세요?  
파이썬에 오신 것을 환영합니다.

사용자 이름을 물어보고, 이어서 2개의 정수를 받아 덧셈을 한 후 결과를 출력하는 다음과 같은 프로그램을 작성해 보세요.



### 도전문제

사용자의 이름을 물어보고 이어서 2개의 정수를 받아서 덧셈을 한 후에 결과를 출력하는 다음과 같은 프로그램을 작성해보자.

이름을 입력하시오: 홍길동

홍길동 씨, 안녕하세요?

파이썬에 오신 것을 환영합니다.

첫 번째 정수를 입력하시오: 300

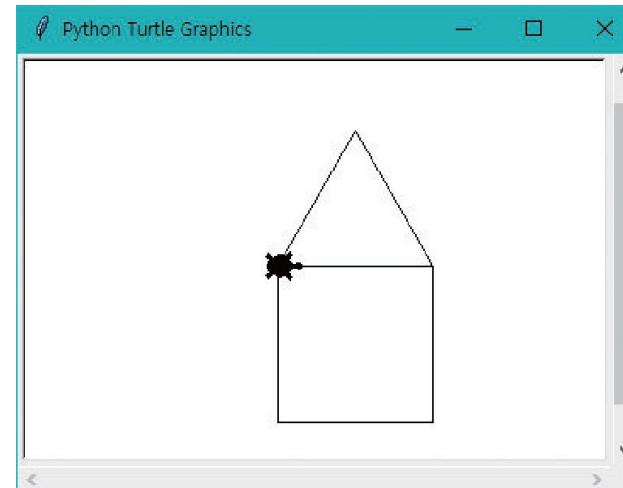
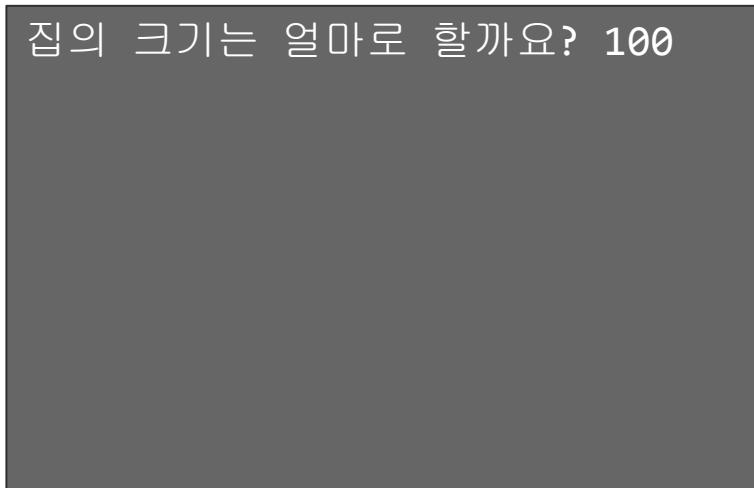
두 번째 정수를 입력하시오: 400

300 과 400 의 합은 700 입니다.

## 실습

## 집그리기

사용자에게 집의 크기를 입력받아서 크기에 맞는 집을 그리는 터틀그래픽 코드를 만들어 보세요.



## 실습

## 집그리기 SOL.

```
import turtle  
t = turtle.Turtle()  
t.shape("turtle")  
  
# 사용자로부터 집의 크기를  
# 받아서 size라는 변수에 저장한다.  
  
size = int(input("집의 크기는 얼마로 할까요? "))
```



```
t.left(60)  
t.forward(size) # 1  
t.right(120)  
t.forward(size) # 2  
t.right(30)  
t.forward(size) # 3  
t.right(90)  
t.forward(size) # 4  
t.right(90)  
t.forward(size) # 5  
t.right(90)  
t.forward(size) # 6  
  
turtle.done()
```

집을 정교하게 그려보세요. 창문과 현관도 만들어보세요. 내부를 색을 넣어 채워서 그려보세요.

ex)

```
t.color("blue")
t.begin_fill()
..
..
t.end_fill()
```



사용자에게 경기장, 점수, 이긴 팀, 진 팀, 우수 선수를 질문하고 변수에 저장한다. 이들 문자열에 문장을 붙여서 기사를 작성한다.

경기장은 어디입니까?서울  
이긴팀은 어디입니까삼성  
진팀은 어디입니까?LG  
우수선수는 누구입니까?홍길동  
스코어는 몇대몇입니까?8:7

=====  
오늘 서울에서 야구 경기가 열렸습니다.  
삼성과 LG은 치열한 공방전을 펼쳤습니다.  
홍길동이 맹활약을 하였습니다.  
결국 삼성이 LG를 8:7로 이겼습니다.  
=====

```
# 사용자의 대답을 변수에 저장한다.  
stadium = input("경기장은 어디입니까?")  
winner = input("이긴팀은 어디입니까?")  
loser = input("진팀은 어디입니까?")  
vip = input("우수선수는 누구입니까?")  
score = input("스코어는 몇대몇입니까?")
```

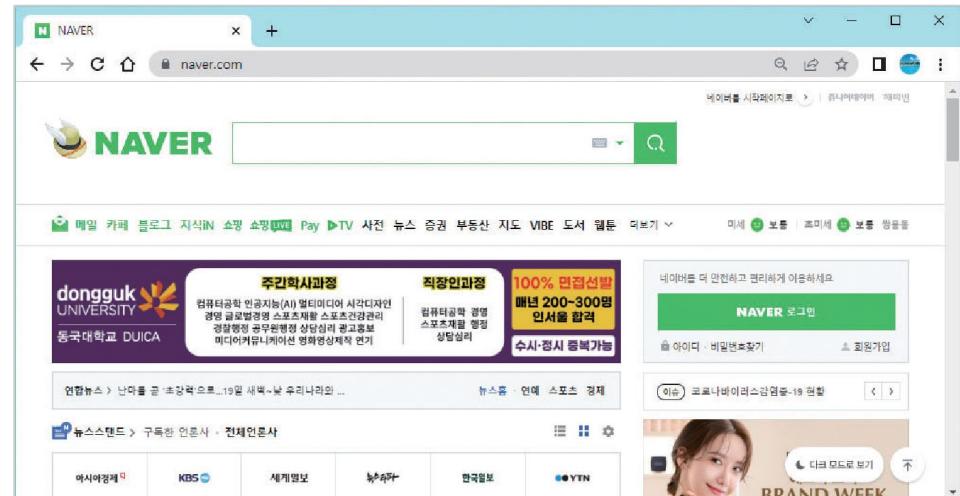
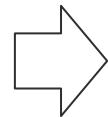
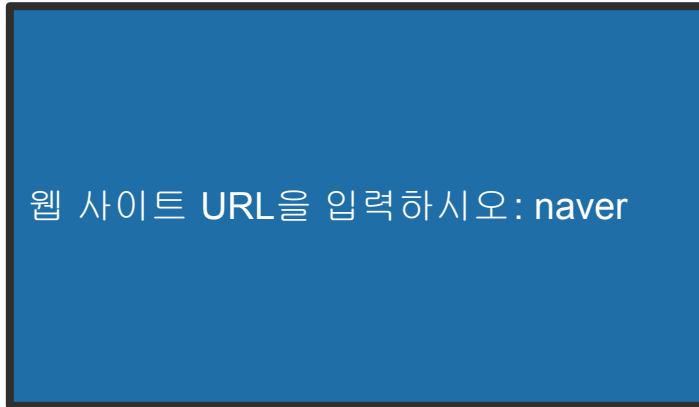
```
# 변수와 문자열을 연결하여 기사를 작성한다.
```

```
print("")  
print("=====")  
print("오늘", stadium, "에서 야구 경기가 열렸습니다.")  
print(winner, "과", loser, "은 치열한 공방전을 펼쳤습니다.")  
print(vip, "이 맹활약을 하였습니다.")  
print("결국", winner, "가", loser, "를 ", score, "로 이겼습니다.")  
print("=====")  
input()
```

실습

## 웹 사이트 오픈하기

webbrowser 모듈을 이용하여 사용자로부터 웹 사이트 주소를 받아서 특정 웹 사이트를 오픈해주는 프로그램을 만들어보자.



```
import webbrowser  
  
url = input("웹 사이트 URL을 입력하시오 : ")  
  
webbrowser.open("https://www."+ url +".com")
```



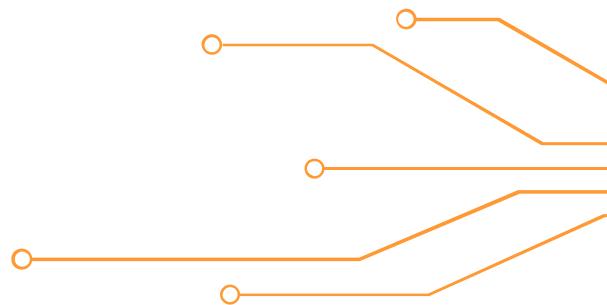
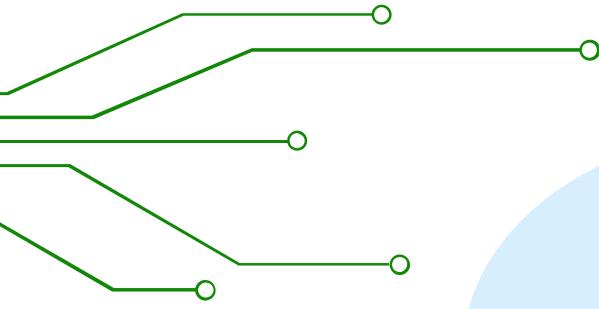
### 도전문제

“<https://translate.google.co.kr/#en/ko/>”에 번역할 영어 문장을 붙이면 자동으로 번역된 결과를 보여준다. 문자열의 + 연산을 사용해보자. 이 기능을 이용하여서 사용자가 입력한 영어 단어를 한국어로 번역해주는 사이트를 오픈해보자.

```
import webbrowser  
  
s = input("번역할 영어 문장을 입력하시오: ")  
url="https://translate.google.co.kr/#en/ko/" + s  
webbrowser.open(url)
```

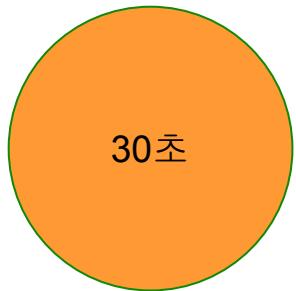


# 퀴즈



1. 다음 중 변수에 대한 설명으로 틀린 것을 고르시오.

- 1) 변수를 사용하여 데이터를 저장할 수 있다.
- 2) 메모리 공간에 불인 이름이다.
- 3) 변수의 이름에는 영문 대문자, 소문자, 숫자, \_(밑줄부호) 을 사용할 수 있다.
- 4) 변수 이름에 사용하는 첫 문자는 영어 대문자만 가능하다.



1. 다음 중 변수에 대한 설명으로 틀린 것을 고르시오. [ 4 ]

- 1) 변수를 사용하여 데이터를 저장할 수 있다.
- 2) 메모리 공간에 불인 이름이다.
- 3) 변수의 이름에는 영문 대문자, 소문자, 숫자, \_(밑줄부호) 을 사용할 수 있다.
- 4) **변수 이름에 사용하는 첫 문자는 영어 대문자만 가능하다.**



1. 다음 중 변수에 대한 설명으로 틀린 것을 고르시오. [ 4 ]

- 1) 변수를 사용하여 데이터를 저장할 수 있다.
- 2) 메모리 공간에 불인 이름이다.
- 3) 변수의 이름에는 영문 대문자, 소문자, 숫자, \_(밑줄부호) 을 사용할 수 있다.
- 4) **변수 이름에 사용하는 첫 문자는 영어 대문자만 가능하다.**

변수 이름에 사용하는 첫 문자는 숫자가 아닌  
영어 대문자, 영어 소문자, \_(밑줄부호)등을  
사용할 수 있다.



2. 다음은 사용자로부터 3개의 값을 입력받아 평균을 계산하는 프로그램입니다.  
비워진 부분에 알맞은 것을 고르세요.

```
n1 = int(input("첫 번째 숫자를 입력하시오: "))
n2 = int(input("두 번째 숫자를 입력하시오: "))
n3 = int(input("세 번째 숫자를 입력하시오: "))
```

```
?
```

```
print(n1, n2, n3, "의 평균은", average, "입니다.")
```

- (1) `average = n1 + n2 + n3 / 3`
- (2) `average = n1 + n2 + n3`
- (3) `average = (n1 + n2 + n3 / 3)`
- (4) `average = (n1 + n2 + n3) / 3`



2. 다음은 사용자로부터 3개의 값을 입력받아 평균을 계산하는 프로그램입니다.  
비워진 부분에 알맞은 것을 고르세요. [ 4 ]

```
n1 = int(input("첫 번째 숫자를 입력하시오: "))
n2 = int(input("두 번째 숫자를 입력하시오: "))
n3 = int(input("세 번째 숫자를 입력하시오: "))
```

```
?
```

```
print(n1, n2, n3, "의 평균은", average, "입니다.")
```

- (1) `average = n1 + n2 + n3 / 3`
- (2) `average = n1 + n2 + n3`
- (3) `average = (n1 + n2 + n3 / 3)`
- (4) `average = (n1 + n2 + n3) / 3`**



2. 다음은 사용자로부터 3개의 값을 입력받아 평균을 계산하는 프로그램입니다.  
비워진 부분에 알맞은 것을 고르세요. [ 4 ]

```
n1 = int(input("첫 번째 숫자를 입력하시오: "))
n2 = int(input("두 번째 숫자를 입력하시오: "))
n3 = int(input("세 번째 숫자를 입력하시오: "))
```

?

```
print(n1, n2, n3, "의 평균은", average, "입니다.")
```

- (1) `average = n1 + n2 + n3 / 3`
- (2) `average = n1 + n2 + n3`
- (3) `average = (n1 + n2 + n3 / 3)`
- (4) `average = (n1 + n2 + n3) / 3`**

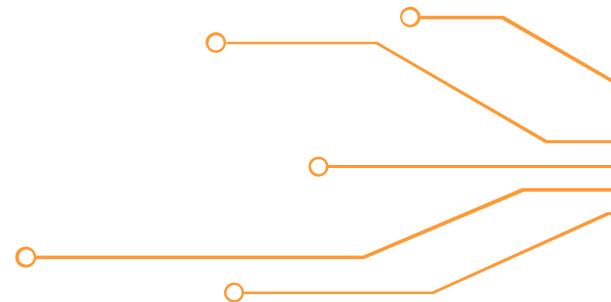
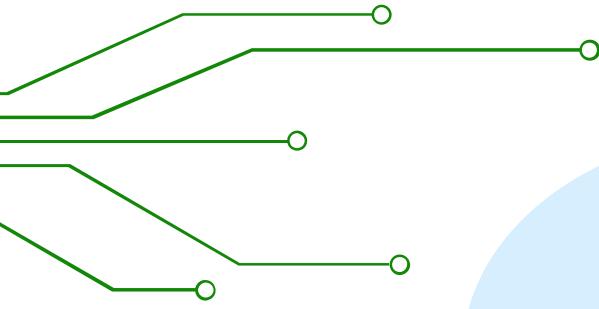
평균을 계산하기 위해서는 3개의 값을 모두 더한 후 (1)  
3으로 나누어야 합니다.  
즉  $(n1 + n2 + n3)$  로 3개의 값을 모두 더한 다음  
그 값을 3으로 나누면 됩니다.

$$(n1 + n2 + n3) / 3$$





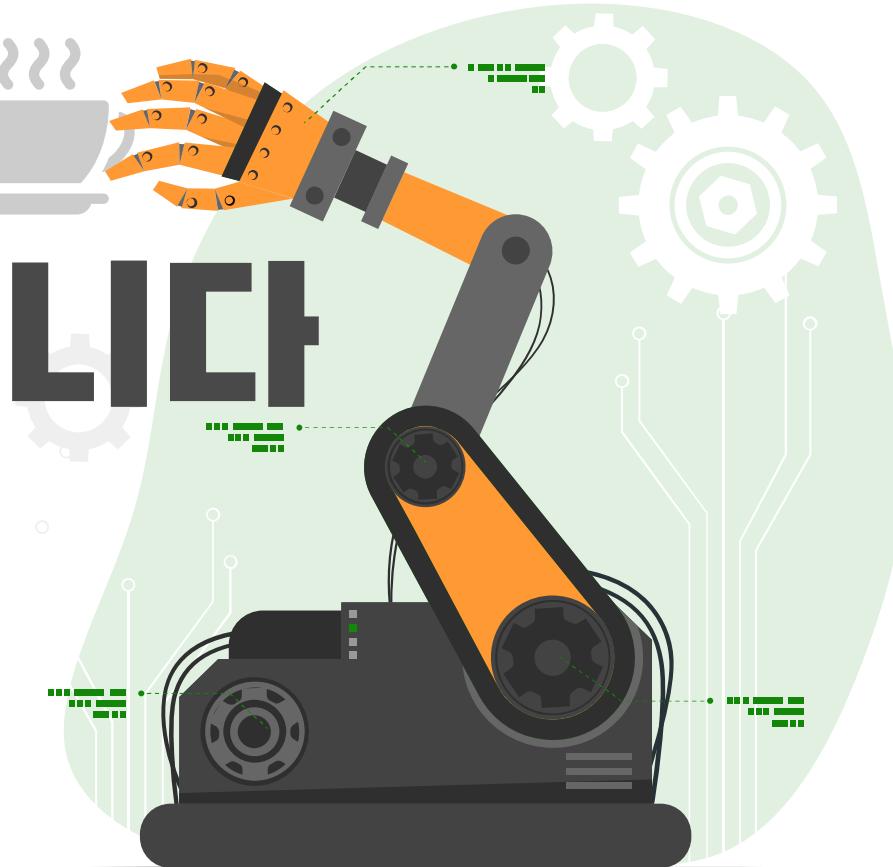
# 핵심정리



- 값을 저장하는 메모리 공간에 이름을 붙인 것을 변수라고 부릅니다.
- 변수를 사용하면 원하는 정보를 저장할 수 있습니다.
- print() 함수를 이용하여 모니터로 출력할 수 있습니다.
- input() 함수는 키보드를 이용하여 입력할 때 사용됩니다.
- int() 함수는 문자열을 정수로 변환할 때 사용됩니다.
- print() 함수를 이용하면 숫자와 문자열을 한번에 출력할 수 있습니다.

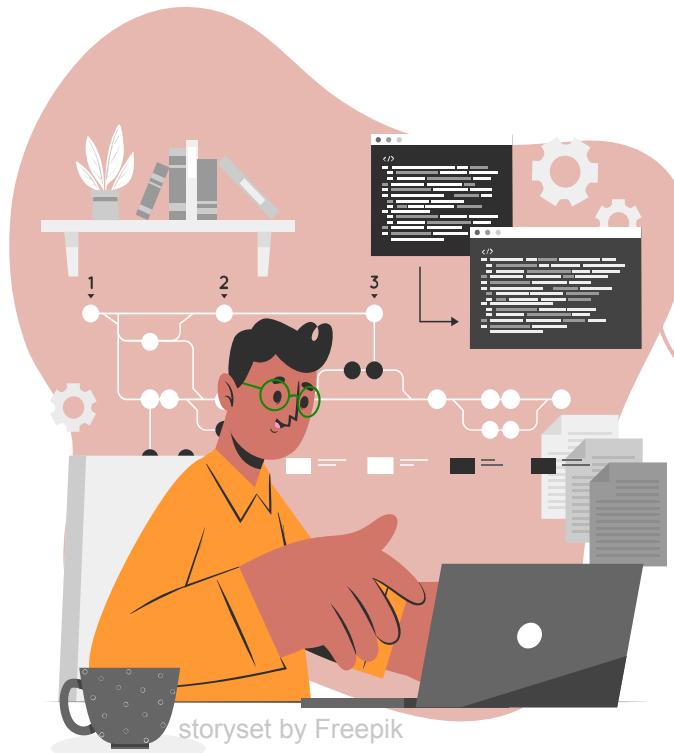


# 수고하셨습니다





# 컴퓨팅 사고와 프로그래밍



CH 03.

## 계산기능

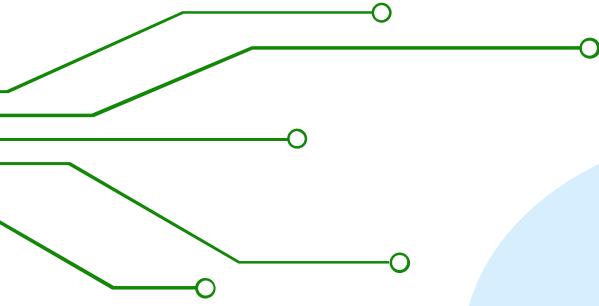
# 학습목표



1. 파이썬을 이용한 사칙연산 방법을 학습한다.
2. 나머지 계산과 지수 계산하는 연산자를 학습한다.
3. 변수에 값을 할당하는 대입연산자를 학습한다.
4. 주석의 개념을 이해한다.
5. 연산자의 우선 순위를 이해한다.



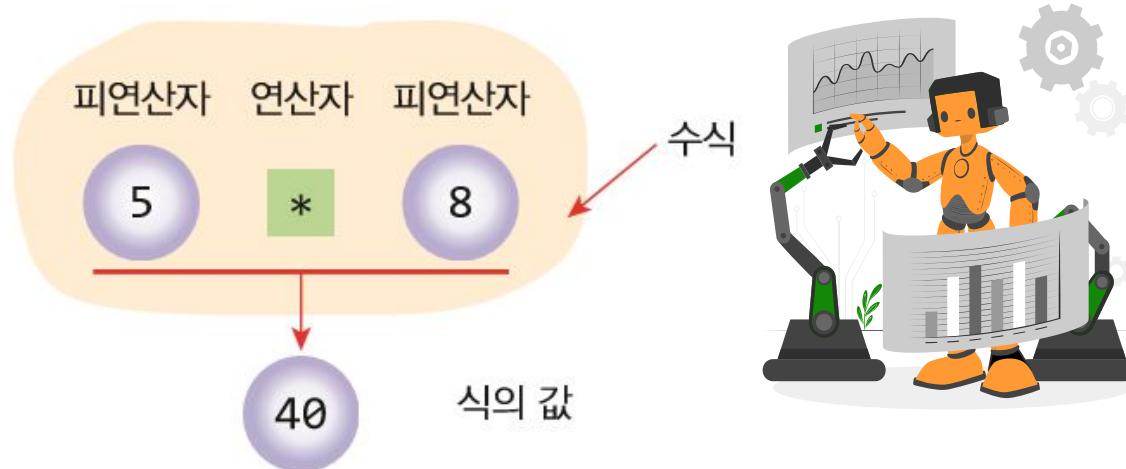
# 이론수업



우리가 즐겨보는 영화의 컴퓨터 그래픽 장면들이 컴퓨터의 계산 기능을 통하여 이루어진다는 것은 아주 흥미롭다. 예를 들어서 건물들의 폭발 장면은 물리학의 여러 가지 공식들을 이용하여 컴퓨터로 계산한 결과를 화면에 표시하는 것이다.



- 수식(expression)=피연산자들과 연산자의 조합
- 연산자(operator): 연산을 나타내는 기호
- 피연산자(operand): 연산의 대상이 되는 값



덧셈, 뺄셈, 곱셈, 나눗셈, 나머지 연산

연산자	기호	사용례	결과값
덧셈	+	$7 + 4$	11
뺄셈	-	$7 - 4$	3
곱셈	*	$7 * 4$	28
나눗셈	//	$7 // 4$	1
나눗셈	/	$7 / 4$	1.75
나머지	%	$7 \% 4$	3



생성된 변수에는 얼마든지 다른 값을 저장할 수 있다.

```
>>> 7 / 4
```

```
1.75
```

```
>>> 7 // 4
```

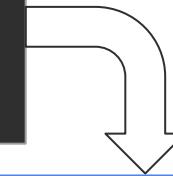
```
1
```

파이썬 버전 2.X에서는 / 연산자의 결과가 정수가 됩니다. 주의하세요!

초등학교 문제 :

10 개의 딸기를 3명에게 나눠주면 각각 몇개씩 가질수 있나요? 그때 남은 것은 몇개인가요?

```
p = int(input("딸기 갯수를 입력하시오: "))
q = int(input("인원수를 입력하시오: "))
print("일인당 딸기 개수=", p // q)
print("남은 딸기 개수=", p % q)
```



```
딸기 갯수를 입력하시오: 10
인원수를 입력하시오: 3
일인당 딸기 개수= 3
남은 딸기 개수= 1
```

```
number = int(input("정수를 입력하시오: "))
print(number%2)
```

정수를 입력하시오: 28  
0

초단위의 시간을 받아서 몇 분 몇 초인지를 계산하여 보자.

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

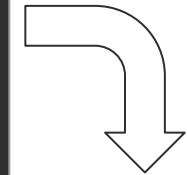
16 40

지수(power)를 계산하려면 \*\* 연산자(곱셈연산자 2개)를 사용한다.

```
>>> 2 ** 7  
128  
>>> 16 ** 0.5  
4.0
```

은행에 저금해 둔 돈이 얼마로 될까요? 계산해봅시다.

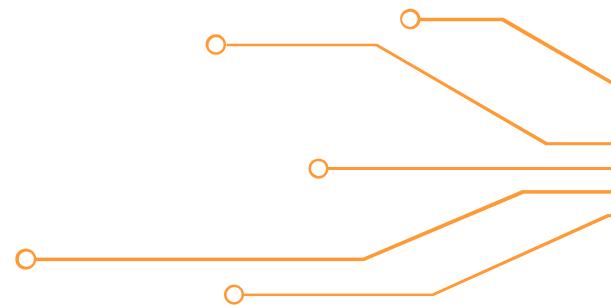
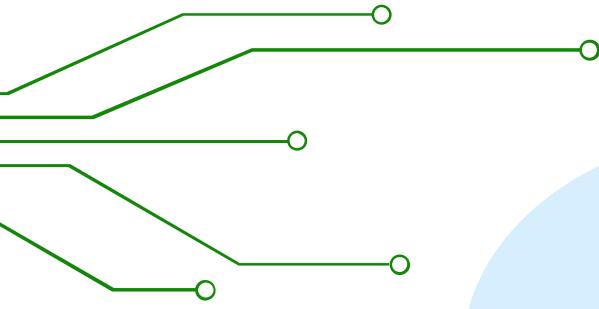
```
money = 10000000  
rate = 0.06  
n = 5  
print("5년 후 원리금 합계=", int(money*(1+rate)**n))
```



5년 후 원리금 합계= 13382255



실습



생성된 변수에는 얼마든지 다른 값을 저장할 수 있다.

```
>>> 7 / 4
```

```
1.75
```

```
>>> 7 // 4
```

```
1
```

파이썬 버전 2.X에서는 / 연산자의  
결과가 정수가 됩니다. 주의하세요!

초등학교 문제 :

10 개의 딸기를 3명에게 나눠주면 각각 몇개씩 가질수 있나요? 그때 남은 것은 몇개인가요?

```
p = int(input("딸기 갯수를 입력하시오: "))
q = int(input("인원수를 입력하시오: "))
print("일인당 딸기 개수=", p // q)
print("남은 딸기 개수=", p % q)
```

?

실습

## 짝수와 홀수

```
number = int(input("정수를 입력하시오: "))
print(number%2)
```

?

## 실습

## 나머지 연산자의 용도

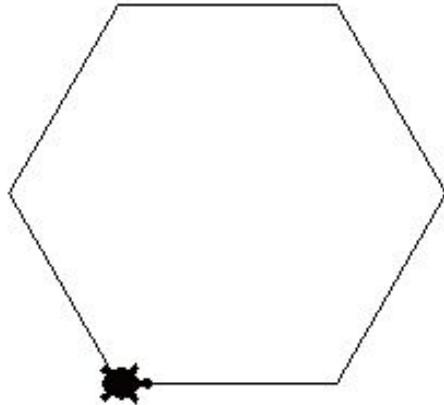
초단위의 시간을 받아서 몇 분 몇 초인지를 계산하여 보자.

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

?



몇각형을 그리시겠어요?(3-6): 6



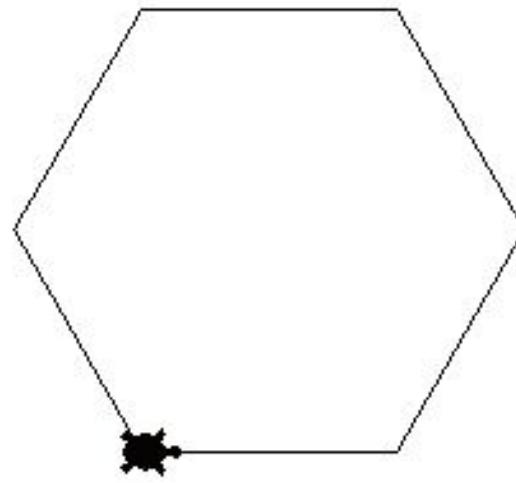
```
for i in range(6) : # 6번 반복한다.  
    t.forward(100)      # 반복하는 문장을 들여쓰기 한다  
    t.left(60)
```



실습

SOL.

```
import turtle  
  
t = turtle.Turtle()  
t.shape("turtle")  
  
n = int(input("몇각형을 그리시겠어요?(3-6): "))  
  
  
for i in range(n) :  
    t.forward(100)  
    t.left(360//n)  
  
  
turtle.done()
```



커피 전문점을 내려고 한다. 다음과 같은 커피 메뉴가 있을 때, 얼마나 많은 매출을 올릴 수 있을지 계산해보자.



아메리카노 판매 개수: 10  
카페라떼 판매 개수: 20  
카푸치노 판매 개수: 30  
총 매출은 185000 입니다.

```
americano_price = 2000  
cafelatte_price = 3000  
capucino_price = 3500  
  
americanos = int(input("아메리카노 판매 개수: "))  
cafelattes = int(input("카페라떼 판매 개수: "))  
capucinos = int(input("카푸치노 판매 개수: "))  
  
sales = americanos*americano_price  
sales = sales + cafelattes*cafelatte_price  
sales = sales + capucinos*capucino_price  
  
print("총 매출은", sales, "입니다.")
```

One More : 도전 !!

사용된 총 재료의 비용이  
100000 원이었다면,  
그때 순 이익이 얼마인지  
계산해서 출력해보세요.

화씨온도를 받아서 섭씨온도로 바꾸는 프로그램을 작성해보자.

화씨온도: 100

섭씨온도: 37.7777777777778

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



```
ftemp = int(input("화씨온도: "))  
ctemp = (ftemp-32.0)*5.0/9.0  
print("섭씨온도:", ctemp)
```

One More : 도전 !!

섭씨 온도를 화씨 온도로  
변환하는 프로그램도  
만들어보세요!!!

자동 판매기를 시뮬레이션하는 프로그램을 작성해보자. 자동 판매기는 사용자로부터 투입한 돈과 물건값을 입력받는다. 물건값은 100원 단위라고 가정한다. 프로그램은 잔돈을 계산하여 출력한다. 자판기는 동전 500원, 100원짜리만 가지고 있다고 가정하자.

프로그램은 투입한 돈과 물건값을 각각 `input()` 함수로 받아서 처리합니다.

거스름돈이 얼마인지, 500원 동전과 100원 동전을 각각 몇개씩 내보낼 것인지 결정해서 출력합니다.

```
투입한 돈: 5000  
물건값: 2600  
거스름돈: 2400  
500원 동전의 개수: 4  
100원 동전의 개수: 4
```



```
money = int(input("투입한 돈: "))
price = int(input("물건값: "))

change = money - price
print("거스름돈: ", change)
coin500s = change // 500          # 500으로 나누어서 둘이 500원짜리의 개수
change = change % 500            # 500으로 나눈 나머지를 계산한다.
coin100s = change // 100         # 100으로 나누어서 둘이 100원짜리의 개수

print("500원 동전의 개수:", coin500s)
print("100원 동전의 개수:", coin100s)
```



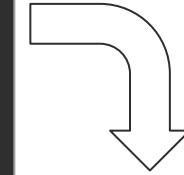
지수(power)를 계산하려면 \*\* 연산자(곱셈연산자 2개)를 사용한다.

```
>>> 2 ** 7  
128  
>>> 16 ** 0.5  
4.0
```



은행에 저금해 둔 돈이 얼마로 될까요? 계산해봅시다.

```
money = 10000000  
rate = 0.06  
n = 5  
print("5년 후 원리금 합계=", int(money*(1+rate)**n))
```



?

사용자로부터 2차 방정식의 계수를 받아서 지수 연산자 `**`을 사용하여 2개의 근을 계산한다.

근의 공식 사용 / `float("1.2")`

a: 1

b: 10

c: -24

2개의 실근이 있는 경우만 계산할 수 있습니다.

2개의 실근: 2.0 -12.0



```
a = float(input("a: "))
b = float(input("b: "))
c = float(input("c: "))

r = b**2 - 4*a*c

print("2개의 실근이 있는 경우만 계산할 수 있습니다.")
x1 = (((-b) + r**0.5)/(2*a))
x2 = (((-b) - r**0.5)/(2*a))
print("2개의 실근:", x1, x2)
```

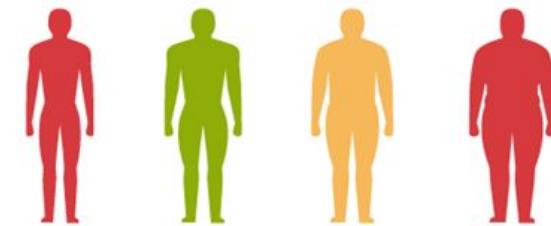


사용자로부터 신장과 체중을 입력받아서 BMI(Body Mass Index)값을 출력하는 프로그램을 작성하여 보자.

몸무게를 kg 단위로 입력하시오: 85.0

키를 미터 단위로 입력하시오: 1.83

당신의 BMI= 25.381468541909282



BMI Chart

BMI less than 18.50	Underweight
BMI 18.50 - 24.99	Healthy weight
BMI 25.00 - 29.99	Overweight
BMI 30 or more	Obese

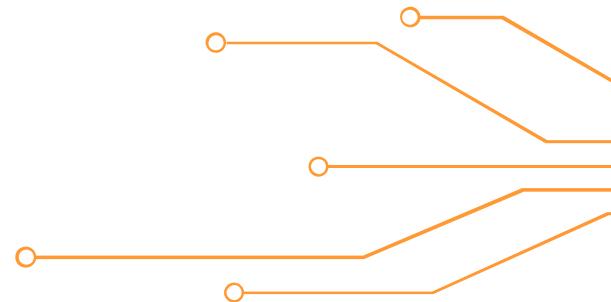
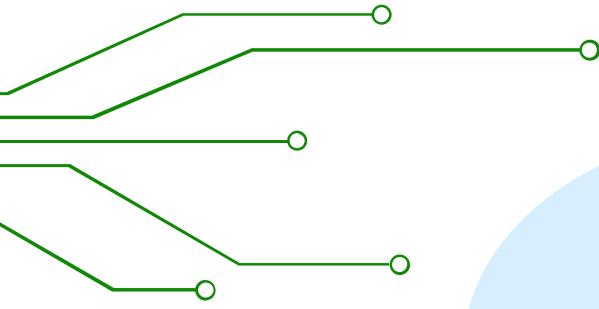
$$\text{BMI} = \frac{(\text{weight in kilograms})}{\text{height in meters}^2}$$

```
weight = float(input("몸무게를 kg 단위로 입력하시오: "))
height = float(input("키를 미터 단위로 입력하시오: "))
bmi = (weight / (height**2)) # 지수 연산자를 사용해보자.
print("당신의 BMI=", bmi)
```

? 사용자가 키를 cm 로 입력한다면  
프로그램을 어떻게 수정하면 될까 ?



# 이론수업

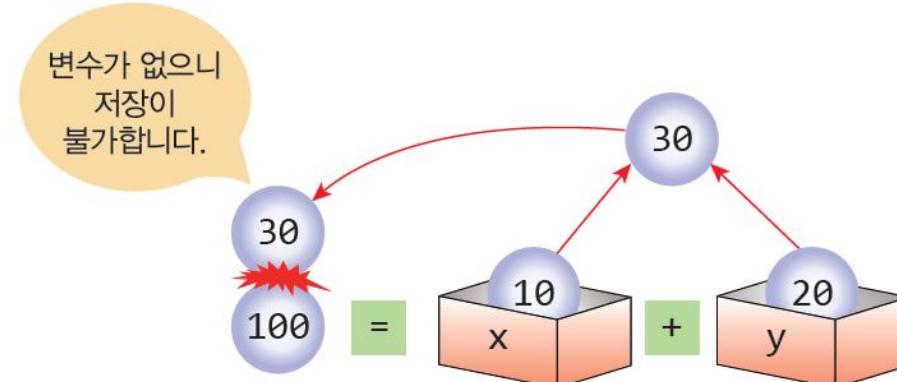


- 할당 연산자는 변수에 값을 저장하는 연산자이다. `=`의 왼쪽은 반드시 변수이어야 하고 등호의 오른쪽은 어떠한 수식이라도 가능하다.

```
x = 100 + 200
```

- 다음은 잘못된 수식이다.

```
100 = x + y
```

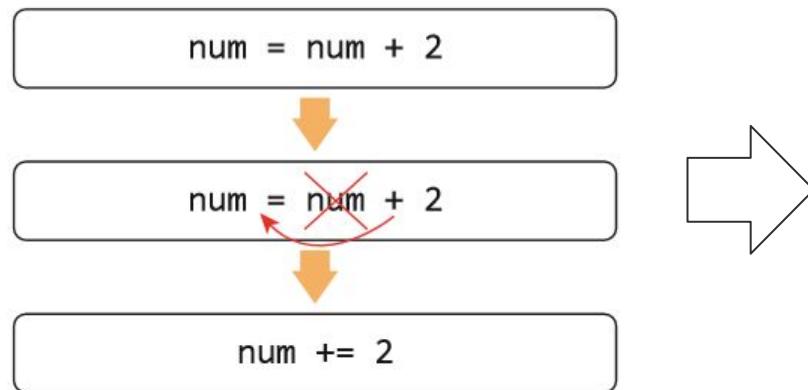


- 다음과 같이 여러 개의 변수에 동일한 값을 저장하는 것도 가능하다.
  - $x = y = 100$
- 다음과 같이 여러 개의 변수에 서로 다른 값을 저장하는 것도 가능하다.
  - $x, y = 100, 200$
  - 위의 문장이 실행되면 변수  $x$ 에는 100이 저장되고 변수  $y$ 에는 200이 저장된다.



복합 연산자(compound operator)란

$+=$ 처럼 대입 연산자와 다른 연산자를 합쳐 놓은 연산자이다.



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



관계 연산자(relational operator)는 두 개의 피연산자를 비교하는데 사용된다.

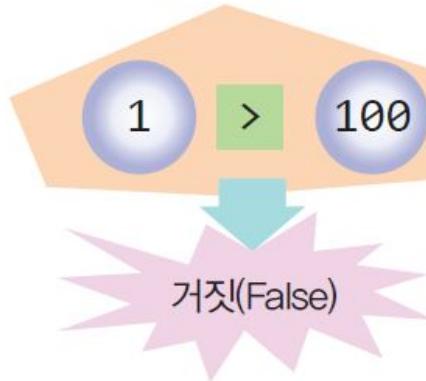
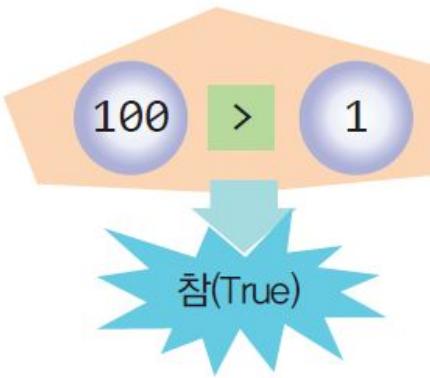
연산	의미	수학적 표기
$x == y$	x와 y가 같은가?	=
$x != y$	x와 y가 다른가?	$\neq$
$x > y$	x가 y보다 큰가?	>
$x < y$	x가 y보다 작은가?	<
$x >= y$	x가 y보다 크거나 같은가?	$\geq$
$x <= y$	x가 y보다 작거나 같은가?	$\leq$



## 이론수업

## 관계 연산자가 생성하는 값

관계 수식은 참이나 거짓이라는 값을 생성한다. 파이썬에서 참과 거짓은 **True**와 **False**로 표시된다.



왼쪽을 오른쪽에  
넣으라는 명령

=

왼쪽과 오른쪽이 같은지  
확인하라는 명령

==

비교 연산자를 사용하여 문자열도 비교할 수 있다.

```
>>> s1 = "Audrey Hepburn"  
>>> s2 = "Audrey Hepburn"  
>>> s1 == s2  
True
```

ASCII 코드

```
>>> s1 = "Audrey Hepburn"  
>>> s2 = "Grace Kelly"  
>>> s1 < s2  
True
```



## ASCII 코드

(American Standard Code for Information Interchange, 미국 정보 교환 표준 부호)

- 1963년 미국 ANSI에서 표준화한 7비트 부호체계
  - ANSI : 미국 국립 표준 협회(American National Standards Institute, ANSI)
- 초기에는 0(0x00)부터 127(0x7F)까지 총 128개의 부호 사용(7비트 부호체계)
- 이후 1개 비트를 추가로 사용, 총 8비트 부호체계로 개편됨. 이를 확장 아스키코드로 부름
- 현재 아스키코드는 8비트 체계를 사용 (그러나 7비트 부분만 공통적으로 사용됨)



# ASCII TABLE

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	@	96	60	`
1	1	[START OF HEADING]	33	21	!	65	41	A	97	61	a
2	2	[START OF TEXT]	34	22	"	66	42	B	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	C	99	63	c
4	4	[END OF TRANSMISSION]	36	24	\$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	e
6	6	[ACKNOWLEDGE]	38	26	&	70	46	F	102	66	f
7	7	[BELL]	39	27	'	71	47	G	103	67	g
8	8	[BACKSPACE]	40	28	(	72	48	H	104	68	h
9	9	[HORIZONTAL TAB]	41	29	)	73	49	I	105	69	i
10	A	[LINE FEED]	42	2A	*	74	4A	J	106	6A	j
11	B	[VERTICAL TAB]	43	2B	+	75	4B	K	107	6B	k
12	C	[FORM FEED]	44	2C	,	76	4C	L	108	6C	l
13	D	[CARRIAGE RETURN]	45	2D	.	77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E	.	78	4E	N	110	6E	n
15	F	[SHIFT IN]	47	2F	/	79	4F	O	111	6F	o
16	10	[DATA LINK ESCAPE]	48	30	0	80	50	P	112	70	p
17	11	[DEVICE CONTROL 1]	49	31	1	81	51	Q	113	71	q
18	12	[DEVICE CONTROL 2]	50	32	2	82	52	R	114	72	r
19	13	[DEVICE CONTROL 3]	51	33	3	83	53	S	115	73	s
20	14	[DEVICE CONTROL 4]	52	34	4	84	54	T	116	74	t
21	15	[NEGATIVE ACKNOWLEDGE]	53	35	5	85	55	U	117	75	u
22	16	[SYNCHRONOUS IDLE]	54	36	6	86	56	V	118	76	v
23	17	[END OF TRANS. BLOCK]	55	37	7	87	57	W	119	77	w
24	18	[CANCEL]	56	38	8	88	58	X	120	78	x
25	19	[END OF MEDIUM]	57	39	9	89	59	Y	121	79	y
26	1A	[SUBSTITUTE]	58	3A	:	90	5A	Z	122	7A	z
27	1B	[ESCAPE]	59	3B	:	91	5B	[	123	7B	{
28	1C	[FILE SEPARATOR]	60	3C	<	92	5C	\	124	7C	
29	1D	[GROUP SEPARATOR]	61	3D	=	93	5D	]	125	7D	}
30	1E	[RECORD SEPARATOR]	62	3E	>	94	5E	^	126	7E	~
31	1F	[UNIT SEPARATOR]	63	3F	?	95	5F	_	127	7F	[DEL]



아스키코드 중 전반부

0 ~ 127번

ANSI 표준으로 모든

나라에서 동일

\$ ascii										
000	0000	^@	032	0x20	!	064	0x40	@	096	0x60
001	0x01	^A	033	0x21	"	065	0x41	A	097	0x61
002	0x02	^B	034	0x22	#	066	0x42	B	098	0x62
003	0x03	^C	035	0x23	\$	067	0x43	C	099	0x63
004	0x04	^D	036	0x24	%	068	0x44	D	100	0x64
005	0x05	^E	037	0x25	&	069	0x45	E	101	0x65
006	0x06	^F	038	0x26	'	070	0x46	F	102	0x66
007	0x07	^G	039	0x27	(	071	0x47	G	103	0x67
008	0x08	^H	040	0x28	)	072	0x48	H	104	0x68
009	0x09	^I	041	0x29	*	073	0x49	I	105	0x69
010	0x0a	^J	042	0x2a	+	074	0x4a	J	106	0x6a
011	0x0b	^K	043	0x2b	-	075	0x4b	K	107	0x6b
012	0x0c	^L	044	0x2c	.	076	0x4c	L	108	0x6c
013	0x0d	^M	045	0x2d	/	077	0x4d	M	109	0x6d
014	0x0e	^N	046	0x2e	0	078	0x4e	N	110	0x6e
015	0x0f	^O	047	0x2f	1	079	0x4f	O	111	0x6f
016	0x10	^P	048	0x30	2	080	0x50	P	112	0x70
017	0x11	^Q	049	0x31	3	081	0x51	Q	113	0x71
018	0x12	^R	050	0x32	4	082	0x52	R	114	0x72
019	0x13	^S	051	0x33	5	083	0x53	S	115	0x73
020	0x14	^T	052	0x34	6	084	0x54	T	116	0x74
021	0x15	^U	053	0x35	7	085	0x55	U	117	0x75
022	0x16	^V	054	0x36	8	086	0x56	V	118	0x76
023	0x17	^W	055	0x37	9	087	0x57	W	119	0x77
024	0x18	^X	056	0x38	:	088	0x58	X	120	0x78
025	0x19	^Y	057	0x39	;	089	0x59	Y	121	0x79
026	0x1a	^Z	058	0x3a	<	090	0x5a	Z	122	0x7a
027	0x1b	^_	059	0x3b	=	091	0x5b	_	123	0x7b
028	0x1c	^`	060	0x3c	?	092	0x5c	`	124	0x7c
029	0x1d	^]	061	0x3d	>	093	0x5d	]	125	0x7d
030	0x1e	^^	062	0x3e	-	094	0x5e	^	126	0x7e
031	0x1f	^_	063	0x3f	-	095	0x5f	_	127	0x7f

아스키코드 중 후반부

128 ~ 255번

ANSI 표준이 아니므로

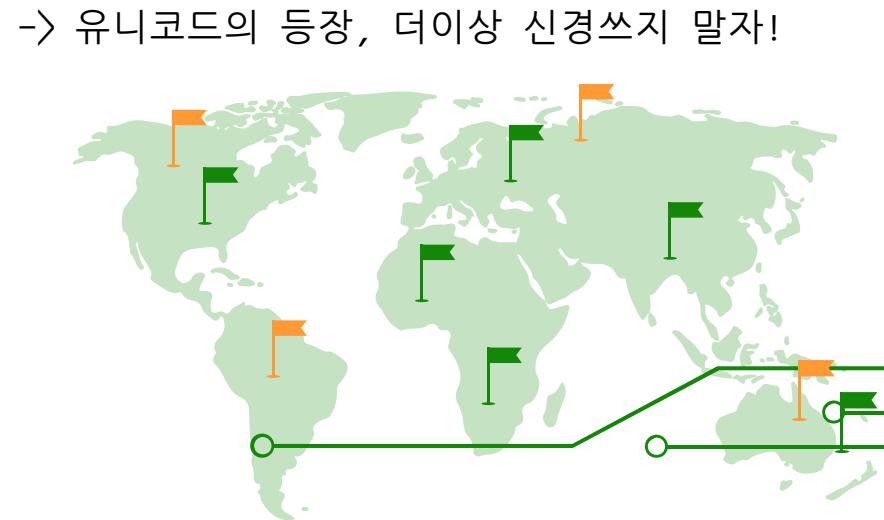
개별 나라마다 다르게

사용

128	0x80	?	160	0xa0	;	192	0xc0	À	224	0xe0	à
129	0x81	?	161	0xa1	à	193	0xc1	Á	225	0xe1	á
130	0x82	,	162	0xa2	ç	194	0xc2	Â	226	0xe2	â
131	0x83	ƒ	163	0xa3	£	195	0xc3	Ã	227	0xe3	ã
132	0x84	,	164	0xa4	*	196	0xc4	Ä	228	0xe4	ä
133	0x85	.	165	0xa5	¥	197	0xc5	Å	229	0xe5	å
134	0x86	+	166	0xa6	_	198	0xc6	Æ	230	0xe6	æ
135	0x87	½	167	0xa7	§	199	0xc7	Ç	231	0xe7	ç
136	0x88	^	168	0xa8	"	200	0xc8	È	232	0xe8	é
137	0x89	٪	169	0xa9	ç	201	0xc9	É	233	0xe9	ê
138	0x8a	S	170	0xaa	œ	202	0xca	E	234	0xea	ë
139	0x8b	<	171	0xab	«	203	0xcb	E	235	0xeb	ë
140	0x8c	0	172	0xac	»	204	0xcc	I	236	0xec	í
141	0x8d	?	173	0xad	-	205	0xcd	I	237	0xed	í
142	0x8e	Z	174	0xae	r	206	0xce	I	238	0xee	í
143	0x8f	?	175	0xaf	ó	207	0xcf	I	239	0xef	í
144	0x90	?	176	0xb0	ö	208	0xd0	D	240	0xf0	d
145	0x91	‘	177	0xb1	±	209	0xd1	Ñ	241	0xf1	ñ
146	0x92	’	178	0xb2	²	210	0xd2	Ó	242	0xf2	ò
147	0x93	”	179	0xb3	³	211	0xd3	Ó	243	0xf3	ó
148	0x94	”	180	0xb4	’	212	0xd4	Ó	244	0xf4	ô
149	0x95		181	0xb5	µ	213	0xd5	Ó	245	0xf5	ó
150	0x96	-	182	0xb6	¶	214	0xd6	Ö	246	0xf6	ö
151	0x97	-	183	0xb7	·	215	0xd7	×	247	0xf7	÷
152	0x98	~	184	0xb8	,	216	0xd8	Ó	248	0xf8	ó
153	0x99	T	185	0xb9	í	217	0xd9	Ú	249	0xf9	ú
154	0x9a	s	186	0xba	œ	218	0xda	Ú	250	0xfa	ú
155	0x9b	>	187	0xbb	»	219	0xdb	Ú	251	0xfb	û
156	0x9c	o	188	0xbc	¾	220	0xdc	Ü	252	0xfc	ü
157	0x9d	?	189	0xbd	½	221	0xdd	Ý	253	0xfd	y
158	0x9e	z	190	0xbe	-	222	0xde	-	254	0xfe	-
159	0x9f	Y	191	0xbf	ç	223	0xdf	Ø	255	0xff	ÿ

0	00	1	01	2	02	3	03	4	04	5	05	6	06	7	07	8	08	9	09	10	0A	11	0B	12	0C	13	0D	14	0E	15	0F			
NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	TAB	LF	VT	FF	CR	SO	SI																			
16	10	17	11	18	12	19	13	20	14	21	15	22	16	23	17	24	18	25	19	26	1A	27	1B	28	1C	29	1D	30	1E	31	1F			
DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US																			
32	20	33	21	34	22	35	23	36	24	37	25	38	26	39	27	40	28	41	29	42	2A	43	2B	44	2C	45	2D	46	2E	47	2F			
SP	!	"	#	\$	₩	&	'	(	)	*	+	,	-	.	/																			
48	30	49	31	50	32	51	33	52	34	53	35	54	36	55	37	56	38	57	39	58	3A	59	3B	60	3C	61	3D	62	3E	63	3F			
0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?																			
64	40	65	41	66	42	67	43	68	44	69	45	70	46	71	47	72	48	73	49	74	4A	75	4B	76	4C	77	4D	78	4E	79	4F			
@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O																			
80	50	81	51	82	52	83	53	84	54	85	55	86	56	87	57	88	58	89	59	90	5A	91	5B	92	5C	93	5D	94	5E	95	5F			
P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^																				
96	60	97	61	98	62	99	63	100	64	101	65	102	66	103	67	104	68	105	69	106	6A	107	6B	108	6C	109	6D	110	6E	111	6F			
`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o																			
112	70	113	71	114	72	115	73	116	74	117	75	118	76	119	77	120	78	121	79	122	7A	123	7B	124	7C	125	7D	126	7E	127	7F			
p	q	r	s	t	u	v	w	x	y	z	{		}	~	DEL																			
128	80	129	81	130	82	131	83	132	84	133	85	134	86	135	87	136	88	137	89	138	8A	139	8B	140	8C	141	8D	142	8E	143	8F			
Ç	ü	é	â	ä	à	å	ç	ê	ë	è	ï	î	ì	Ä	Å																			
144	90	145	91	146	92	147	93	148	94	149	95	150	96	151	97	152	98	153	99	154	9A	155	9B	156	9C	157	9D	158	9E	159	9F			
É	æ	Æ	Ô	Ö	Ò	Ù	Ù	Ý	Ö	Ü	¢	£	¥	₱	f																			
160	A0	161	A1	162	A2	163	A3	164	A4	165	A5	166	A6	167	A7	168	A8	169	A9	170	AA	171	AB	172	AC	173	AD	174	AE	175	AF			
á	í	ó	ú	ñ	Ñ	a	º	¿	¬	¬	¬	¬	¬	¬	¬	¬	¬	¬	¬	¬	¬	¬	¬	¬	¬	¬	¬	¬	¬	¬	¬	¬	¬	
176	B0	177	B1	178	B2	179	B3	180	B4	181	B5	182	B6	183	B7	184	B8	185	B9	186	BA	187	BB	188	BC	189	BD	190	BE	191	BF			
192	C0	193	C1	194	C2	195	C3	196	C4	197	C5	198	C6	199	C7	200	C8	201	C9	202	CA	203	CB	204	CC	205	CD	206	CE	207	CF			
L	†	‡	†	‡	-	+	†	‡	†	‡	†	‡	†	‡	†	‡	†	‡	†	‡	†	‡	†	‡	†	‡	†	‡	†	‡	†	‡		
208	D0	209	D1	210	D2	211	D3	212	D4	213	D5	214	D6	215	D7	216	D8	217	D9	218	DA	219	DB	220	DC	221	DD	222	DE	223	DF			
224	E0	225	E1	226	E2	227	E3	228	E4	229	E5	230	E6	231	E7	232	E8	233	E9	234	EA	235	EB	236	EC	237	ED	238	EE	239	EF			
α	β	Γ	Π	Σ	σ	μ	τ	Φ	Θ	Ω	δ	∞	∅	ε	∩																			
240	F0	241	F1	242	F2	243	F3	244	F4	245	F5	246	F6	247	F7	248	F8	249	F9	250	FA	251	FB	252	FC	253	FD	254	FE	255	FF			

8번째 비트를 포함시킨 128~255 까지의 문자는  
앞으로 어떻게 될까?



→ 유니코드의 등장, 더이상 신경쓰지 말자!

다음과 같은 프로그램을 실행시키면 결과가 어떻게 나올까?

```
>>> a=0.1 + 0.2  
>>> a==0.3  
False
```



```
>>> import math          # math 모듈 포함  
>>> a=0.1 + 0.2  
>>> abs(a-0.3) < 0.00000001 # a와 0.3의 차이가 작으면  
True
```

십진수 0.1 을  
이진수로 어떻게  
표현할까?



## 이론수업

## 0.1 을 이진수로 ...

[십진수] 12.34

$$12.34 = 1 * 10^1 + 2 * 10^0 + 3 * 10^{-1} + 4 * 10^{-2}$$

[이진수]

$$\begin{aligned} 101.011 &= 1 * 2^2 + \Rightarrow 1*4 = 4 \\ &\quad 0 * 2^1 + \Rightarrow 0*2 = 0 \\ &\quad 1 * 2^0 + \Rightarrow 1*1 = 1 \quad + ==> 5.375 \\ &\quad 0 * 2^{-1} + \Rightarrow 0*0.5 = 0 \\ &\quad 1 * 2^{-2} + \Rightarrow 1*0.25 = 0.25 \\ &\quad 1 * 2^{-3} \Rightarrow 1*0.125 = 0.125 \end{aligned}$$



[이진수]

$$0.?????? \Rightarrow ? * 2^{-1} \quad 0.5$$
$$+ ? * 2^{-2} \quad 0.25$$
$$+ ? * 2^{-3} \quad 0.125$$
$$+ ? * 2^{-4} \quad 0.0625$$
$$+ ? * 2^{-5} \quad 0.03125$$
$$+ ? * 2^{-6} \quad 0.015625$$
$$+ ? * 2^{-7} \quad 0.0078125$$
$$+ ? * 2^{-8} \quad 0.00390625$$



주석(comment)은 소스 코드에 붙이는 설명글과 같은 것이다.

주석은 프로그램이 하는 일을 설명한다.

주석은 프로그램의 실행 결과에 영향을 끼치지 않는다.

```
# 사용자로부터 화씨온도를 입력받음  
  
ftemp = int(input("화씨온도: "))  
  
ctemp = (ftemp-32.0)*5.0/9.0 # 화씨온도->섭씨온도  
  
print("섭씨온도:", ctemp)      # 섭씨온도를 화면에 출력
```

```
>>> 10 + 20 /2  
20.0
```

```
>>> (10 + 20) /2  
15.0
```

$$x + \underbrace{y * z}_{\textcircled{②}}$$

$$\underbrace{(x + y)}_{\textcircled{①}} * z$$

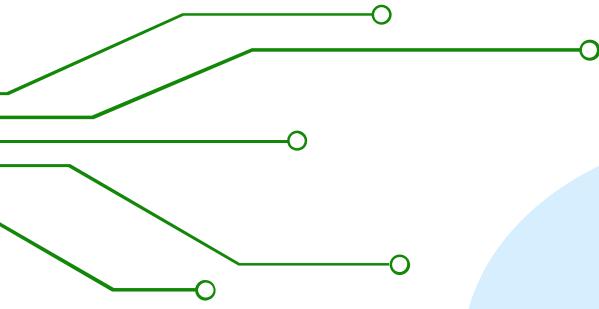


연산자	설명
**	지수 연산자
* / % //	곱셈, 나눗셈, 나머지 연산자
+ -	덧셈, 뺄셈
<= < > >=	비교 연산자
== !=	동등 연산자
=	할당 연산자
in not in	소속 연산자
not or and	논리 연산자





실습

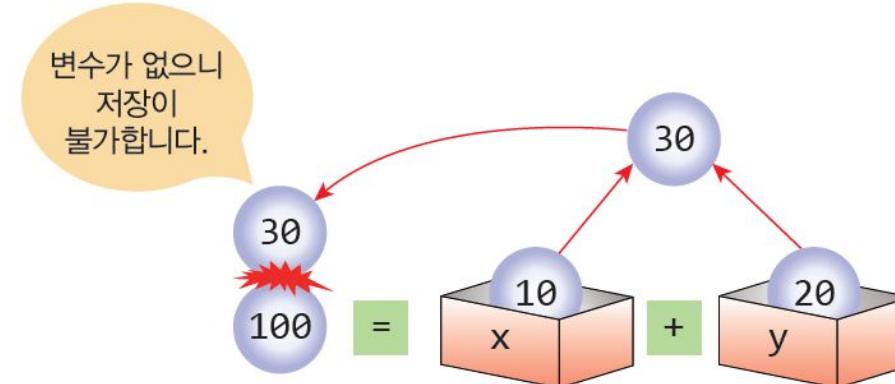


- 할당 연산자는 변수에 값을 저장하는 연산자이다. `=`의 왼쪽은 반드시 변수이어야 하고 등호의 오른쪽은 어떠한 수식이라도 가능하다.

```
x = 100 + 200
```

- 다음은 잘못된 수식이다.

```
100 = x + y
```



## 실습

## 문자열의 비교

비교 연산자를 사용하여서 문자열도 비교할 수 있다.

```
>>> s1 = "Audrey Hepburn"  
>>> s2 = "Audrey Hepburn"  
>>> s1 == s2  
True
```

ASCII 코드

```
>>> s1 = "Audrey Hepburn"  
>>> s2 = "Grace Kelly"  
>>> s1 < s2  
True
```



다음과 같은 프로그램을 실행시키면 결과가 어떻게 나올까?

```
>>> a=0.1 + 0.2  
>>> a==0.3  
False
```

```
>>> import math          # math 모듈 포함  
>>> a=0.1 + 0.2  
>>> abs(a-0.3) < 0.00000001 # a와 0.3의 차이가 작으면  
True
```



평균을 구하고자 한다. 잘못된 부분은 어디일까?

```
x = int(input("첫 번째 수를 입력하시오: "))
y = int(input("두 번째 수를 입력하시오: "))
z = int(input("세 번째 수를 입력하시오: "))

avg = x + y + z / 3
print("평균 =", avg)
```



0부터 9까지의 숫자를 이용하여 간단한 산수 퀴즈를 출제하는 프로그램을 만들어보자.

산수 퀴즈에 오신 것을 환영합니다 .

$2 + 5 = 7$

True

$7 - 6 = 1$

True

$2 ** 3 = 8$

True



숫자가 고정되면 답이 정해져 있어서 재미가 없어집니다. 숫자들이 랜덤하게 바뀔 수 있도록 아래 코드를 참고해서 만들어 봅시다.

```
import random  
n = random.randint(0,10)
```

## 실습

## math 모듈 사용하기

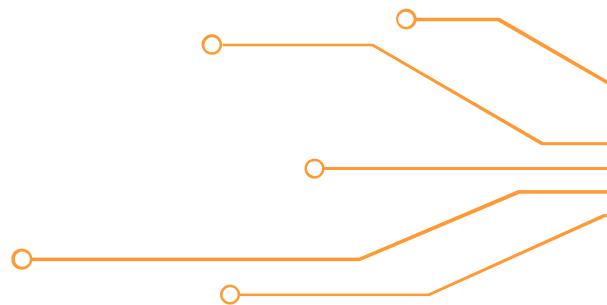
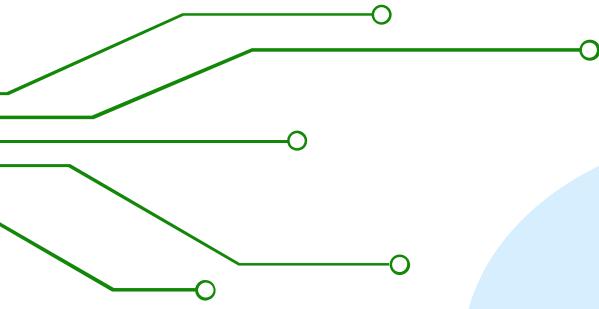
파이썬에는 여러 가지 수학 함수를 가지고 있는 math 내장 모듈이 제공된다.

이 모듈을 사용하여 여러 가지 산술 계산을 해보자.





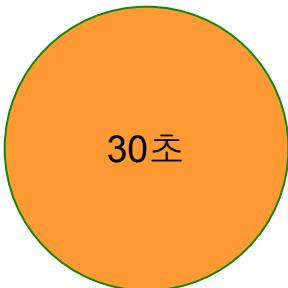
# 퀴즈



1. 다음 수식을 파이썬의 수식형태로 바꾼것 중 올바른 것을 고르시오.

$$x^4 - 9x^3 + x^2$$

- 1)  $x(4)-9*x(3)+x(2)$
- 2)  $x^4-9*x^3+x^2$
- 3)  $x**4-9*x**3+x**2$
- 4)  $x**4-9x**3+x**2$



1. 다음 수식을 파이썬의 수식형태로 바꾼것 중 올바른 것을 고르시오. [ 3 ]

$$x^4 - 9x^3 + x^2$$

- 1)  $x(4)-9*x(3)+x(2)$
- 2)  $x^4-9*x^3+x^2$
- 3)  $x**4-9*x**3+x**2$**
- 4)  $x**4-9x**3+x**2$



1. 다음 수식을 파이썬의 수식형태로 바꾼 것 중 올바른 것을 고르시오. [ 3 ]

$$x^4 - 9x^3 + x^2$$

1)  $x(4)-9*x(3)+x(2)$

2)  $x^4-9*x^3+x^2$

3)  $x^{**4}-9*x^{**3}+x^{**2}$

4)  $x^{**4}-9x^{**3}+x^{**2}$

지수승을 할 때는 \*\* 연산자를 사용합니다. 즉 x의 4승은  $x^{**4}$  으로 표현할 수 있습니다.

4)  $x^{**4}-9x^{**3}+x^{**2}$  에서 틀린 부분은  $9x^{**3}$  입니다.  $9*x^{**2}$  으로 표시해야 합니다.



2. 다음 계산식의 결과값이 얼마가 되는지 옳은 것을 고르세요.

```
n1 = 40  
n2 = 20  
n3 = 10  
  
r = n1 / n2 * 2 - 2 * n3  
  
print(r)
```

- (1) -16
- (2) -19
- (3) -8
- (4) 20



2. 다음 계산식의 결과값이 얼마가 되는지 옳은 것을 고르세요. [ 1 ]

```
n1 = 40  
n2 = 20  
n3 = 10  
  
r = n1 / n2 * 2 - 2 * n3  
  
print(r)
```

- (1) -16
- (2) -19
- (3) -8
- (4) 20



## 2. 다음 계산식의 결과값이 얼마가 되는지 옳은 것을 고르세요. [ 1 ]

```
n1 = 40  
n2 = 20  
n3 = 10  
  
r = n1 / n2 * 2 - 2 * n3  
  
print(r)
```

- (1) -16
- (2) -19
- (3) -8
- (4) 20

\* , / 는 +, - 보다 우선순위가 높습니다.

$(n1 / n2 * 2) - (2 * n3)$  이렇게 괄호로 표시한 부분을 먼저 계산합니다.

$(n1 / n2 * 2)$  에서는 / 와 \* 는 우선순위가 같으므로 왼쪽부터 오른쪽으로 순서대로 계산합니다.

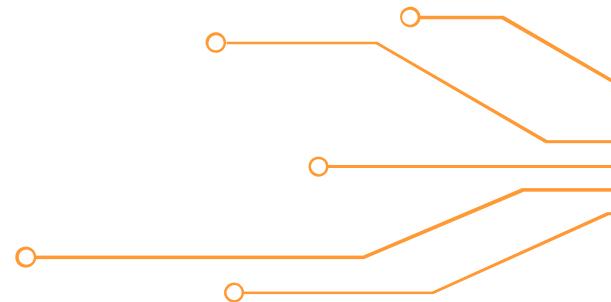
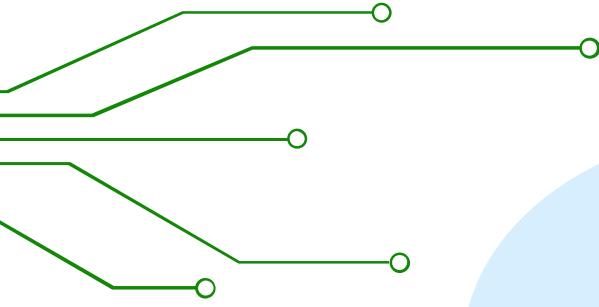
그래서

- 1)  $t1 = n1 / n2$  ( $2=40/20$ )
- 2)  $t2 = t1 * 2$  ( $4=2*2$ )
- 3)  $t3 = 2 * n3$  ( $20=2*10$ )
- 4)  $t4 = t2 - t3$  ( $-16=4-20$ )

순서대로 계산이 되어 결과 값은 -16이 됩니다.

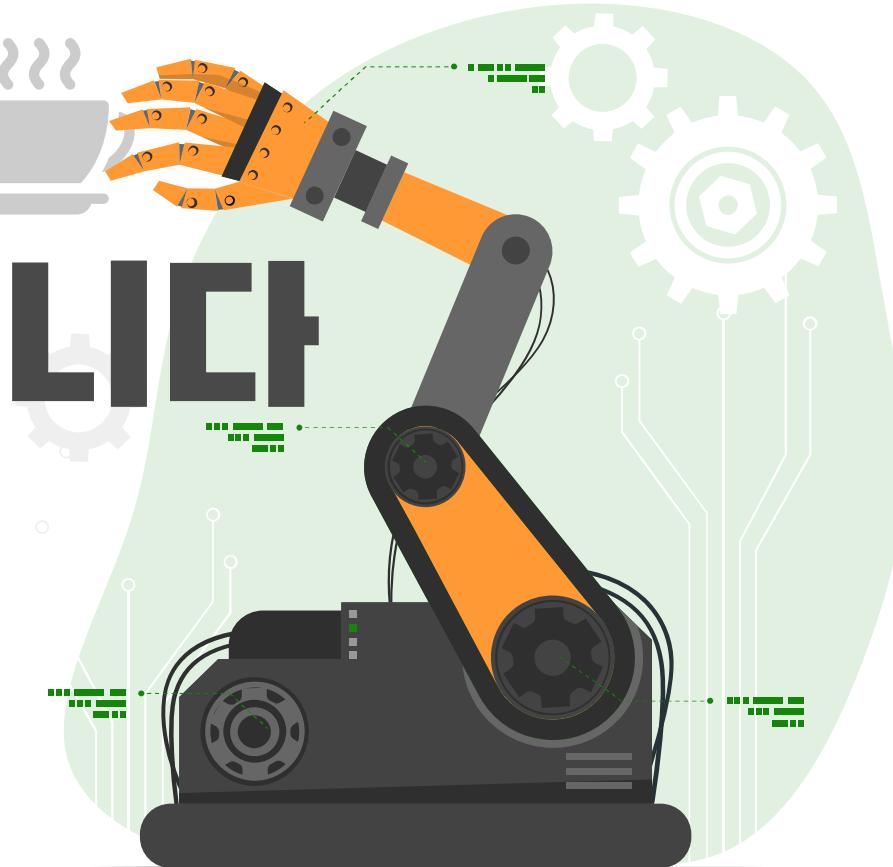


# 핵심정리



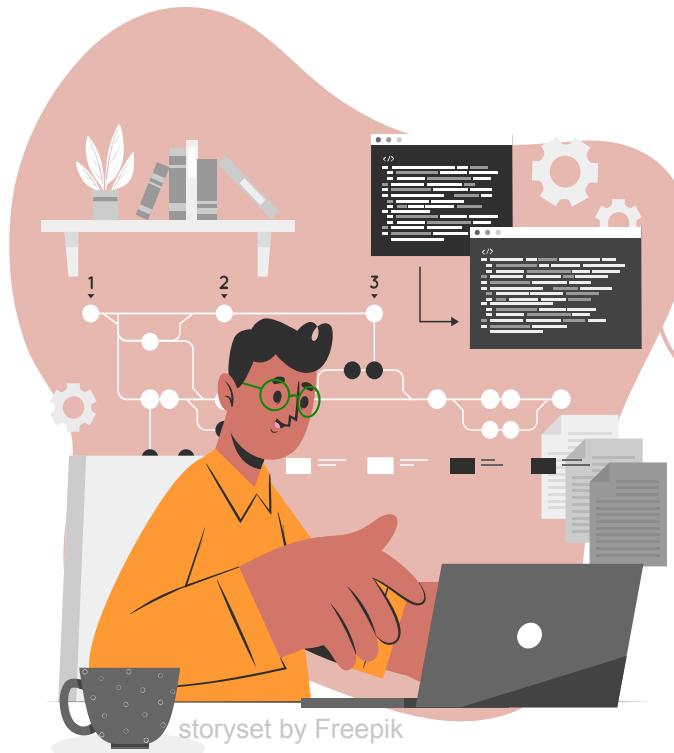
- 파이썬에서는 덧셈, 뺄셈, 곱셈, 나눗셈을 위하여 +, -, \*, / 기호를 사용합니다.
- 나머지 계산을 위해 %, 지수 계산을 위해 \*\*, 나눗셈의 몫 계산은 // 연산자를 사용합니다.
- 할당연산자를 이용하여 변수에 값을 저장할 수 있습니다.
- 할당연산자와 다른 연산자가 결합된 단축된 형태의 연산자(+=)도 쓰입니다.
- 주석은 코드에 대한 설명으로 코드의 실행에는 영향을 주지 않습니다.
- 우선순위가 높은 연산자가 먼저 계산됩니다. \*와 /가 +와 -보다 우선순위가 높습니다. 연산자의 우선순위를 변경하려면 괄호를 사용한다.

# 수고하셨습니다





# 컴퓨팅 사고와 프로그래밍



CH 04.

## 자료의 종류

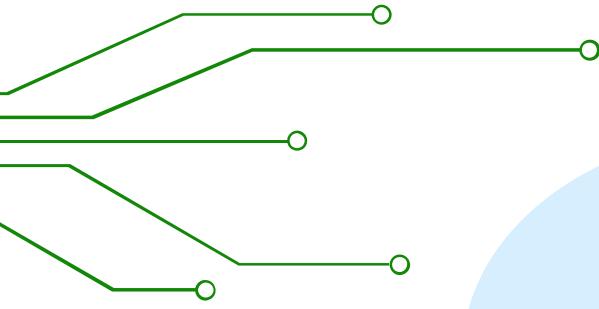
# 학습목표



1. 파이썬에서 사용하는 자료형을 설명할 수 있다.
2. 정수를 문자열로, 문자열을 정수로 변환할 수 있다.
3. 문자열과 관련된 연산을 이해한다.
4. 리스트를 만들수 있고, 사용법을 익힌다.
5. 인덱스와 슬라이스를 이해한다.



# 이론수업



자료형 : 프로그램에서 사용할 수 있는 데이터의 종류

자료형	예
정수	..., -2, -1, 0, 1, 2, ...
실수	3.2, 3.14, 0.12
문자열	'Hello World!', "123"



변수에 어떤 종류의 자료도 저장할 수 있다

```
x = 10  
print("x =", x)
```

```
x = 3.14  
print("x =", x)
```

```
x = "Hello World!"  
print("x =", x)
```

```
x = 10  
x = 3.14  
x = Hello World!
```

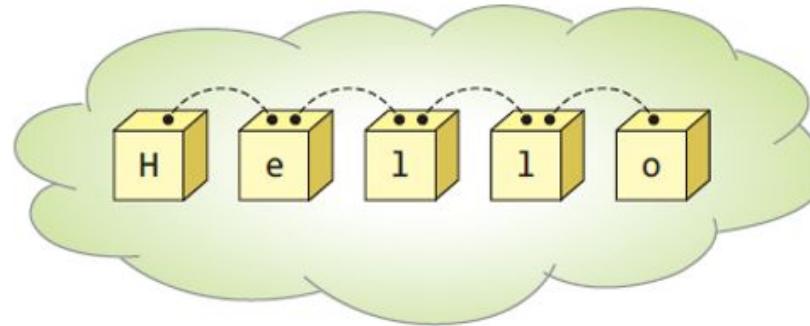
컴퓨터에게는 숫자가 중요하지만 인간에게는 텍스트(text)가 중요하다.

(예) 문자 메시지, 인터넷 도메인 이름

컴퓨터를 이용한 텍스트의 처리도 무척 중요하다.



문자열(string)은 문자들의 나열(sequence of characters)이다.

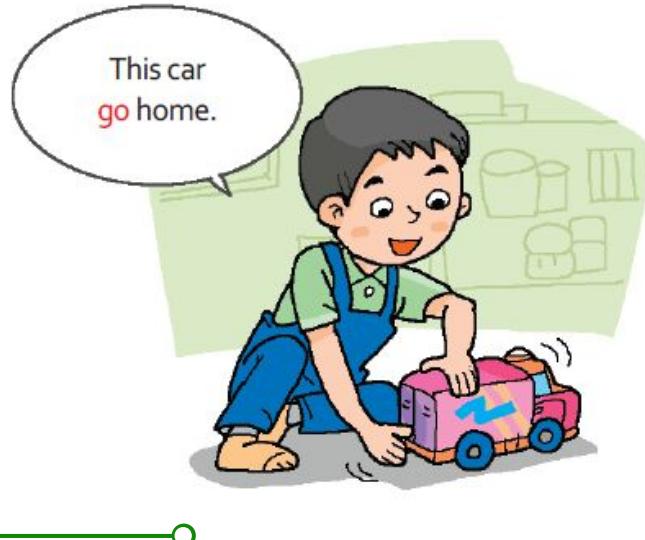


- 큰따옴표
- 작은 따옴표

```
>>> "Hello"  
'Hello'  
  
>>> msg = 'Hello'  
>>> msg  
'Hello'  
>>> print(msg)  
Hello
```

큰따옴표(“)로 시작했다가 작은따옴표(')로 끝내면 문법적인 오류이다.

```
>>> msg = "Hello'  
SyntaxError: EOL while scanning string literal
```

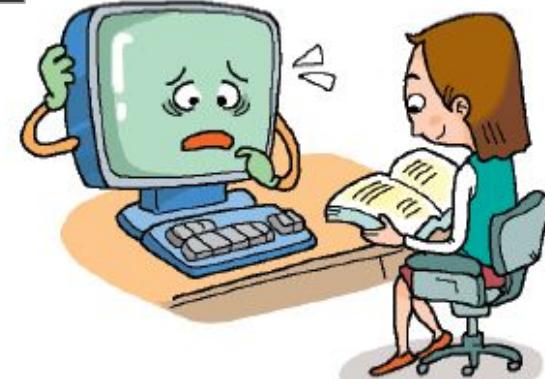


이론수업

큰따옴표와 작은따옴표를 동시에 사용하는 이유

문자열 안에 따옴표가 들어가는 경우를 처리하기 위해서이다. 예를 들어서 “철수가 “안녕”이라고 말했습니다.” 문장을 문자열 안에 저장할 수 있을까?

```
>>> message="철수가 \"안녕\"이라고 말했습니다."  
SyntaxError: invalid syntax
```



```
>>> message="철수가 '안녕'이라고 말했습니다."  
>>> print(message)  
철수가 '안녕'이라고 말했습니다.
```



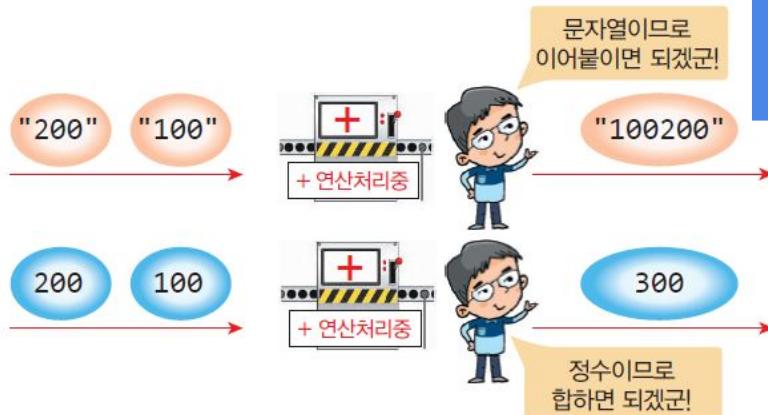
```
>>> print(100+200)
```

```
300
```

```
>>> print("100"+"200")
```

```
100200
```

100+200을 하면 (정수+정수) 형태가 되어서 덧셈이 가능하다. 하지만 "100"+"200"은 텍스트와 텍스트끼리 합하는 것이기 때문에 2개의 텍스트가 이어진다.



자료형을 알려면 type() 함수를 사용한다.

```
>>> type("Hello World!")
```

```
<class 'str' >
```

```
>>> type(3.2)
```

```
<class 'float' >
```

```
>>> type(17)
```

```
<class 'int' >
```

```
>>> x=10
```

```
>>> type(x)
```

```
<class 'int' >
```

## 이론수업

## 문자열 → 숫자

int(): 문자열을 정수로 변환

float(): 문자열을 실수로 변환

```
t = input("정수를 입력하시오: ")  
x = int(t)
```

```
t = input("정수를 입력하시오: ")  
y = int(t)
```

```
print(x+y)
```

```
정수를 입력하시오: 100  
정수를 입력하시오: 200  
300
```

다음 코드에 오류가 발생하는 이유는?

```
>>> print('나는 현재 ' + 21 + '살이다.')
```

Traceback (most recent call last):

File "<pyshell#1>", line 1, in <module>

print('나는 현재 ' + 21 + '살이다.')

TypeError: Can't convert 'int' object to str implicitly

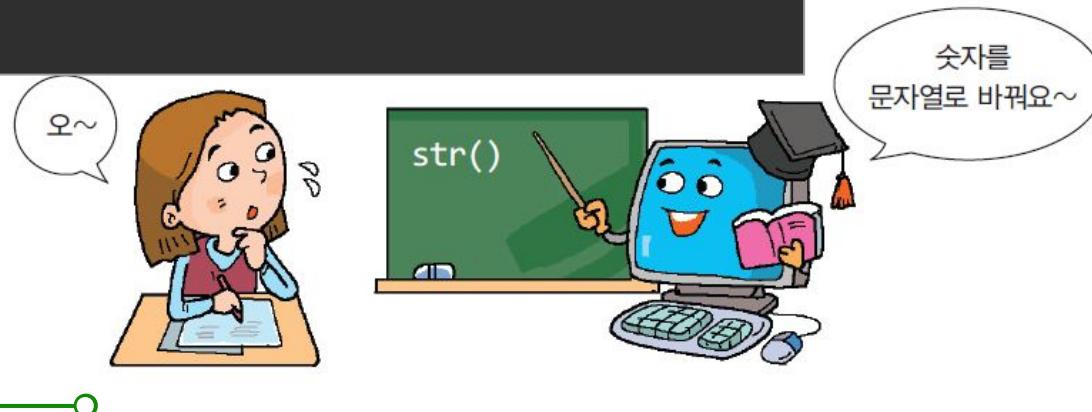
문자열과 숫자를 합칠 수 없는 의미입니다.



str() 함수 사용

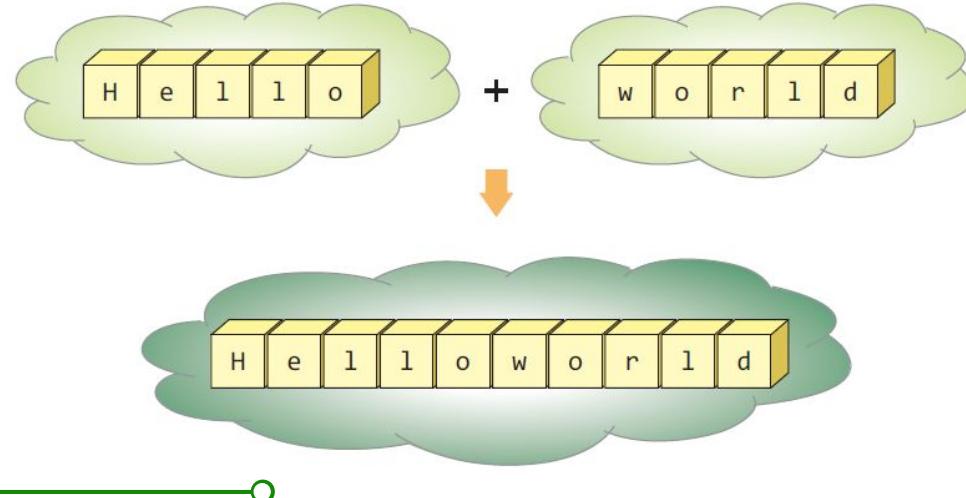
```
>>> print('나는 현재 ' + str(21) + '살이다.')  
나는 현재 21살이다.
```

```
>>> print('원주율은 ' + str(3.14) + '입니다.')  
원주율은 3.14입니다.
```



2개의 문자열을 합치려면 -> + 연산자

```
>>> 'Hello' + 'World!'  
'Hello World!'
```



문자열을 반복하려면 -> \* 연산자

```
>>> "="*50
```

```
=====
```

```
>>> len("Hello World")      # 문자열의 길이  
11
```

```
>>> 'Hello' in "Hello World"    # 서브 문자열이 있으면 True  
True
```



파이썬에서 문자열은 객체(객체=변수+함수)

```
>>> text = " Hello World "
```

```
>>> text.count("o") # 서브 문자열이 나오는 횟수  
2
```

```
>>> text.lower()           # 소문자로 만든다.  
' hello world '
```

```
>>> text.strip()          # 좌우 공백 문자 제거  
'Hello World'
```



문자열에 변수의 값을 삽입하여 출력하고 싶으면 -> f-문자열

```
>>> price = 10000  
>>> print(f"상품의 가격은 {price}원입니다.")  
상품의 가격은 10000원입니다.
```

```
product = "coffee"  
count = 3  
price = 10000  
print(f"상품 {product} {count}개의 가격은 {count*price}원입니다.")
```

상품 coffee 3개의 가격은 30000원입니다.



## 이론수업

## 개별 문자 추출

문자열에서 개별 문자들을  
추출하려면 -> 인덱스라는 번호를  
사용한다.

```
s = "Monty Python"  
print(s[6:10])
```

Pyth

0	1	2	3	4	5	6	7	8	9	10	11
M	o	n	t	y		P	y	t	h	o	n
-12	-11	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1

[6:10]

[-12:-7]

## 이론수업

## 특수 문자열

문자 앞에 \(역슬래시)가 붙으면 특수 문자열이 된다.

다음 표에서 파이썬에서 많이 사용되는 특수 문자열을 나타내었다.

특수 문자열	의미
\n	줄 바꿈 문자
\t	탭 문자
\\"	역슬래시 자체
\“	큰따옴표 자체
\‘	작은따옴표 자체

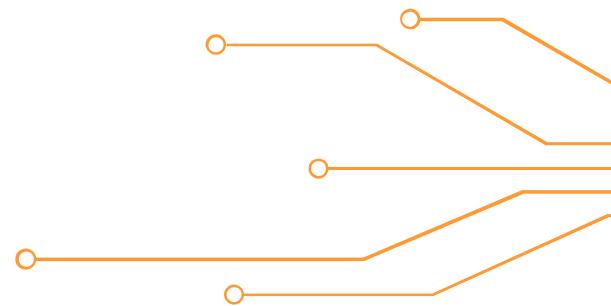
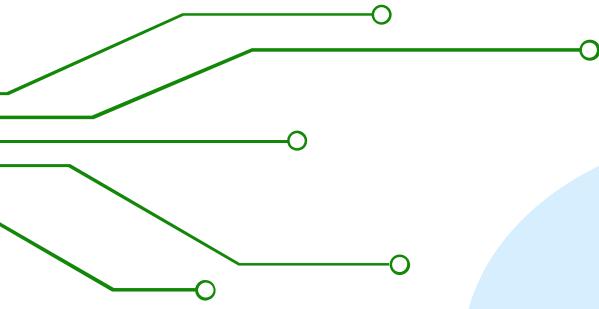
```
>>> print("말 한마디로\n천 냥 빛을 갚는다")
말 한마디로
천 냥 빛을 갚는다
```

특수 문자열을 이용하여 홀따옴표를 화면에 출력해보면 다음과 같다

```
>>> message= 'doesn\'t'  
>>> print(message)  
doesn't
```



실습



변수를 사용하여 사용자의 이름과 나이를 문자열 형태로 기억했다가 출력할 때 사용하는 프로그램을 작성해보자.

안녕하세요?  
이름이 어떻게 되시나요? 홍길동

만나서 반갑습니다. 홍길동씨  
이름의 길이는 다음과 같군요:3  
나이가 어떻게 되나요? 21  
내년이면 22가 되시는군요.



```
print('안녕하세요?')
```

```
name = input('이름이 어떻게 되시나요? ')
```

```
print('\n만나서 반갑습니다.' + name + "씨")
```

```
print('이름의 길이는 다음과 같군요:' +str(len(name)))
```

```
age = int(input("나이가 어떻게 되나요? "))
```

```
print("내년이면 "+str(age+1)+"이 되시는군요.")
```



사용자로부터 영어 이름을 받아서 첫 글자를 모아서 영어 이니셜을 만들어보자. 예를 들면 다음과 같다.

이름의 첫 번째 글자를 입력하시오: Kim

이름의 두 번째 글자를 입력하시오: Jung

이름의 세 번째 글자를 입력하시오: Soon

영어 이니셜은 KJS입니다.



```
x = input("이름의 첫 번째 글자를 입력하시오: ")  
y = input("이름의 두 번째 글자를 입력하시오: ")  
z = input("이름의 세 번째 글자를 입력하시오: ")  
  
initial = x[0] + y[0] + z[0]  
print("\n영어 이니셜은"+initial +"입니다.")
```

J.S.KIM 과 같이 출력되도록 코드를 변경해보세요.

파이썬을 이용하면 텍스트를 소리 내 읽어주는 TTS 프로그램을 쉽게 만들 수 있다.

- 구글 TTS 설치

```
C:\>pip install gtts  
...
```

```
from gtts import gTTS  
import os  
  
text ="안녕하세요, 여러분. 파이썬은 재미있습니다. "  
  
tts = gTTS(text=text, lang='ko')  
tts.save("ttt.mp3")  
os.system("ttt.mp3")
```

도전 : 챗봇 프로그램에 TTS 를 적용해 보세요.

단답형 퀴즈를 출제하고 사용자가 답변을 하면 정답인지 오답인지를 출력하는 프로그램을 작성해보자.

세상에서 가장 쉬운 프로그래밍 언어는? 파이썬

True

파이썬에서 제곱 연산자는? \*\*

True

프로그래밍 언어에서 텍스트를 무엇이라고 부르는가? 글자열

False



```
s = input("세상에서 가장 쉬운 프로그래밍 언어는? ")  
print(s=="파이썬")  
s = input("파이썬에서 제곱 연산자는? ")  
print(s=="**")  
s = input("프로그래밍 언어에서 텍스트를 무엇이라고 부르는가? ")  
print(s=="문자열")
```

도전 : 다른 문제를 추가해보세요.

## 실습

## 날짜 차이 계산하기

두근이와 두순이는 2022년 9월 1일에 처음 만났다. 현재 날짜로 만난 지 며칠이나 되었을까?

만난 일: 2022-09-01 00:00:00  
현재: 2022-09-18 20:35:03.911234  
만난 지 17일 되었습니다.



```
import datetime

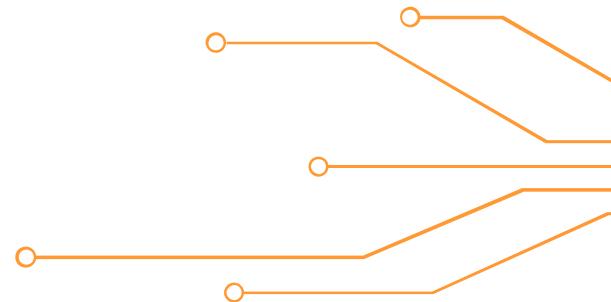
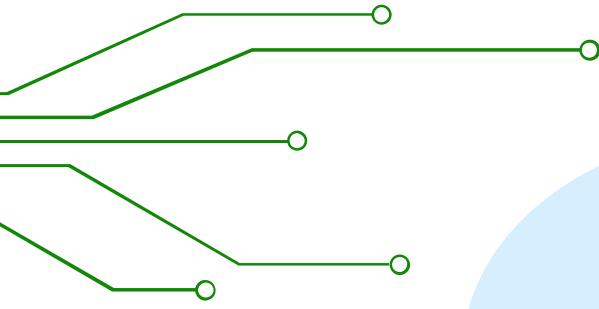
x = datetime.datetime(2022, 9, 1)
y = datetime.datetime.now()

delta = y - x

print("만난 일: "+str(x))
print("현재: "+str(y))
print("만난 지 "+ str(delta.days)+"일 되었습니다.")
```

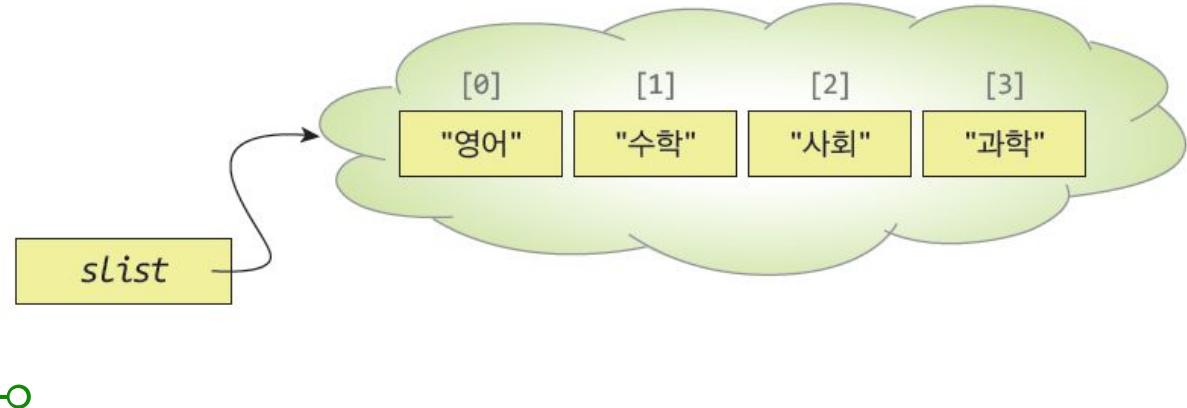


# 이론수업



리스트(list): 여러 개의 자료들을 모아서 하나의 묶음으로 저장하는 것

```
slist = [ '영어', '수학', '사회', '과학' ]
```



## 이론수업

## 리스트에 항목을 동적으로 추가

공백 리스트를 생성한 후에 코드로 리스트에 값을 추가하는 것

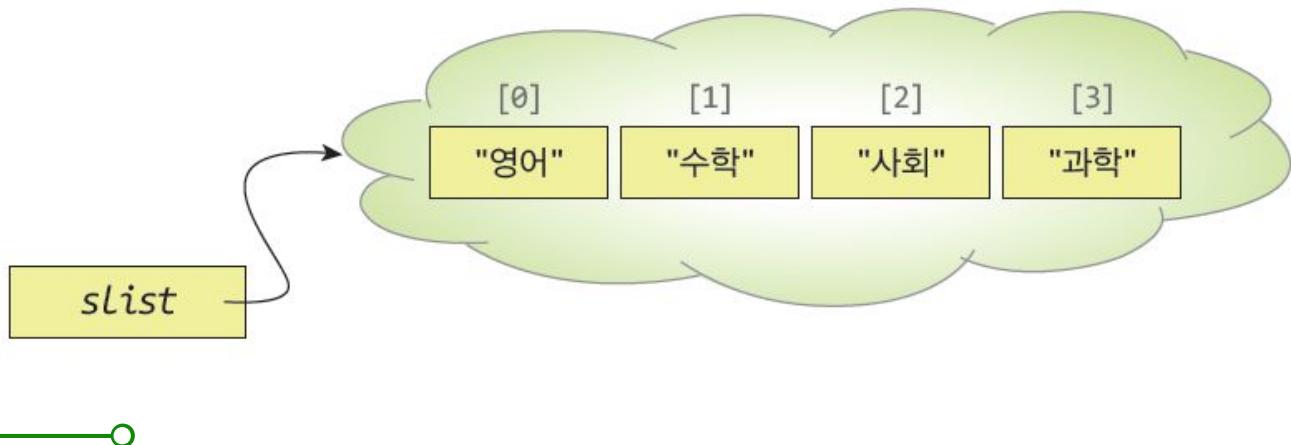
```
list = []
list.append(1)
list.append(2)
list.append(6)
list.append(3)

print(list)
```

[1, 2, 6, 3]

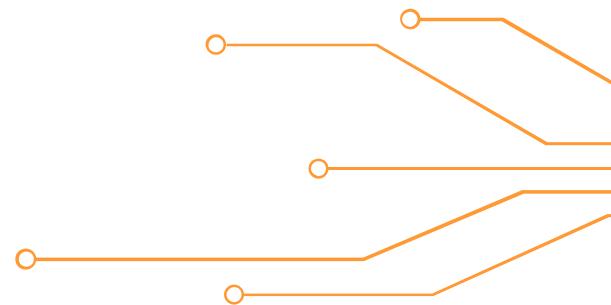
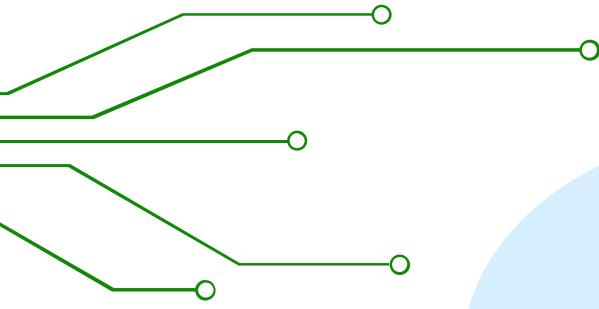
```
slist = [ '영어', '수학', '사회', '과학' ]  
print(slist[0])
```

영어





실습



## 실습

## 친구들의 리스트 생성하기

제일 친한 친구 5명의 이름을 리스트에 저장했다가 출력하는 프로그램을 작성하자

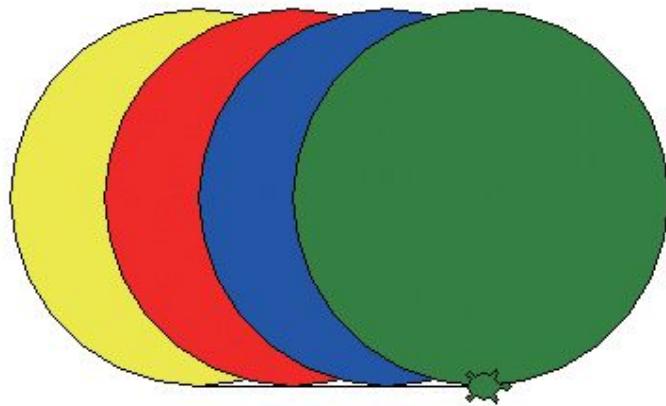
```
친구의 이름을 입력하시오: 홍길동  
친구의 이름을 입력하시오: 강감찬  
친구의 이름을 입력하시오: 이순신  
친구의 이름을 입력하시오: 권율  
친구의 이름을 입력하시오: 정약용  
['홍길동', '강감찬', '이순신', '권율', '정약용']
```

```
friend_list = [ ]  
  
friend = input("친구의 이름을 입력하시오: ")  
friend_list.append(friend)  
  
print(friend_list)
```

## 실습

## 리스트에 저장된 색상으로 원그리기

리스트에 색상을 문자열로 저장하였다가 하나씩 꺼내서 거북이의 채우기 색상으로 설정하고 원을 그려 보자.



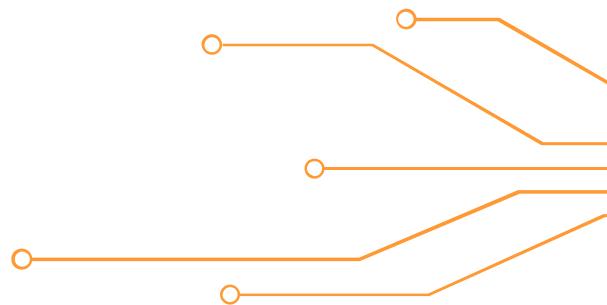
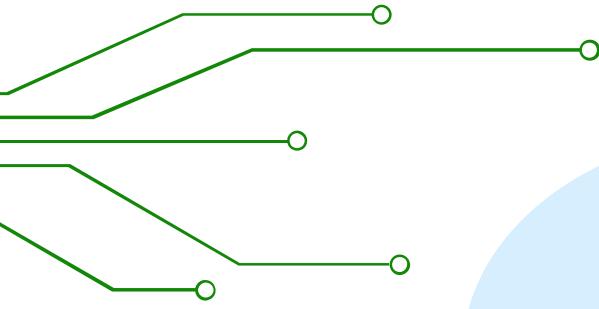
```
import turtle  
t = turtle.Turtle()  
t.shape("turtle")  
  
# 리스트를 사용하여 색상을 문자열로 저장한다.  
color_list = ["yellow", "red", "blue", "green"]  
  
t.fillcolor(color_list[0]) # 채우기 색상을 설정한다.  
t.begin_fill() # 채우기를 시작한다.  
t.circle(100) # 속이 채워진 원이 그려진다.  
t.end_fill() # 채우기를 종료한다.  
t.forward(50)  
  
t.fillcolor(color_list[1]) # 채우기 색상을 설정한다.  
t.begin_fill() # 채우기를 시작한다.  
t.circle(100) # 속이 채워진 원이 그려진다.  
t.end_fill() # 채우기를 종료한다.
```

```
t.forward(50)  
t.fillcolor(color_list[2]) # 채우기 색상을 설정한다.  
t.begin_fill() # 채우기를 시작한다.  
t.circle(100) # 속이 채워진 원이 그려진다.  
t.end_fill() # 채우기를 종료한다.  
  
t.forward(50)  
t.fillcolor(color_list[3]) # 채우기 색상을 설정한다.  
t.begin_fill() # 채우기를 시작한다.  
t.circle(100) # 속이 채워진 원이 그려진다.  
t.end_fill() # 채우기를 종료한다.  
  
turtle.done()
```



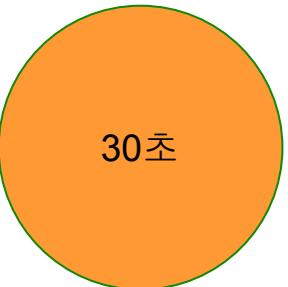


# 퀴즈



1. 다음 수식 중 오류를 발생시키지 않는 바른 것을 고르시오.

- 1) `print('나는' + 12 + '개의 '사과'를 먹었다.')`
- 2) `print('나는' + str(12) + '개의 '사과'를 먹었다.)`
- 3) `print('나는' + str(12) + '개의 "사과"를 먹었다.')`
- 4) `print("나는" + 12 + "개의 '사과'를 먹었다.")`



30초



1. 다음 수식 중 오류를 발생시키지 않는 바른 것을 고르시오. [ 3 ]

- 1) `print('나는' + 12 + '개의 '사과'를 먹었다.')`
- 2) `print('나는' + str(12) + '개의 '사과'를 먹었다.')`
- 3) `print('나는' + str(12) + '개의 "사과"를 먹었다.')`
- 4) `print("나는" + 12 + "개의 '사과'를 먹었다.")`



1. 다음 수식 중 오류를 발생시키지 않는 바른 것을 고르시오. [ 3 ]

- 1) `print('나는' + 12 + '개의 '사과'를 먹었다.')`
- 2) `print('나는' + str(12) + '개의 '사과'를 먹었다.')`
- 3) **`print('나는' + str(12) + '개의 "사과"를 먹었다.')`**
- 4) `print("나는" + 12 + "개의 '사과'를 먹었다.")`

1), 4) 에서 '+' 연산자에는 문자열과 숫자가 함께 오면 안됩니다.

1), 2) 에서 '' 문자열 내부의 '기호는 문법오류를 일으키므로 \' 로 사용되어야 합니다. 수정 : '개의 \'사과\'를 먹었다.'

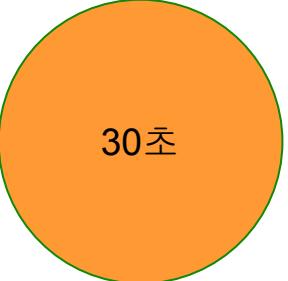
'' 로 묶여진 문자열안의 "" 기호는 단순 문자로 취급됩니다.



2. 리스트 name 의 마지막 원소로 '홍길동' 이 들어가게 하는 코드를 고르세요.

```
name = ['홍부', '놀부', '산신령']
```

- 1) name.insert(0, '홍길동')
- 2) del name[0, '홍길동']
- 3) name.append('홍길동')
- 4) add(name, '홍길동')



30초

2. 리스트 name 의 마지막 원소로 '홍길동' 이 들어가게 하는 코드를 고르세요. [ 3 ]

```
name = ['홍부', '놀부', '산신령']
```

- 1) name.insert(0, '홍길동')
- 2) del name[0, '홍길동']
- 3) name.append('홍길동')**
- 4) add(name, '홍길동')



2. 리스트 name 의 마지막 원소로 '홍길동' 이 들어가게 하는 코드를 고르세요. [ 3 ]

```
name = ['홍부', '놀부', '산신령']
```

- 1) name.insert(0, '홍길동')
- 2) del name[0, '홍길동']
- 3) name.append('홍길동')**
- 4) add(name, '홍길동')

리스트에 원소를 추가할 때는 `insert()` 혹은 `append()`를 사용합니다. `insert(인덱스번호, 요소값)` 으로

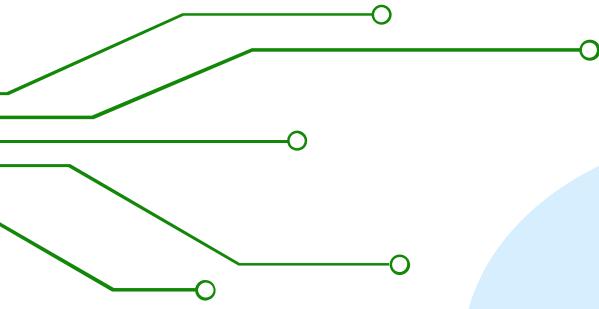
원하는 위치에 원소를 추가하거나 `append(요소값)`으로  
만 마지막 위치에 원소를 추가합니다.

1)은 마지막 원소가 아닌 첫번째 원소로 '홍길동'이  
들어갑니다.





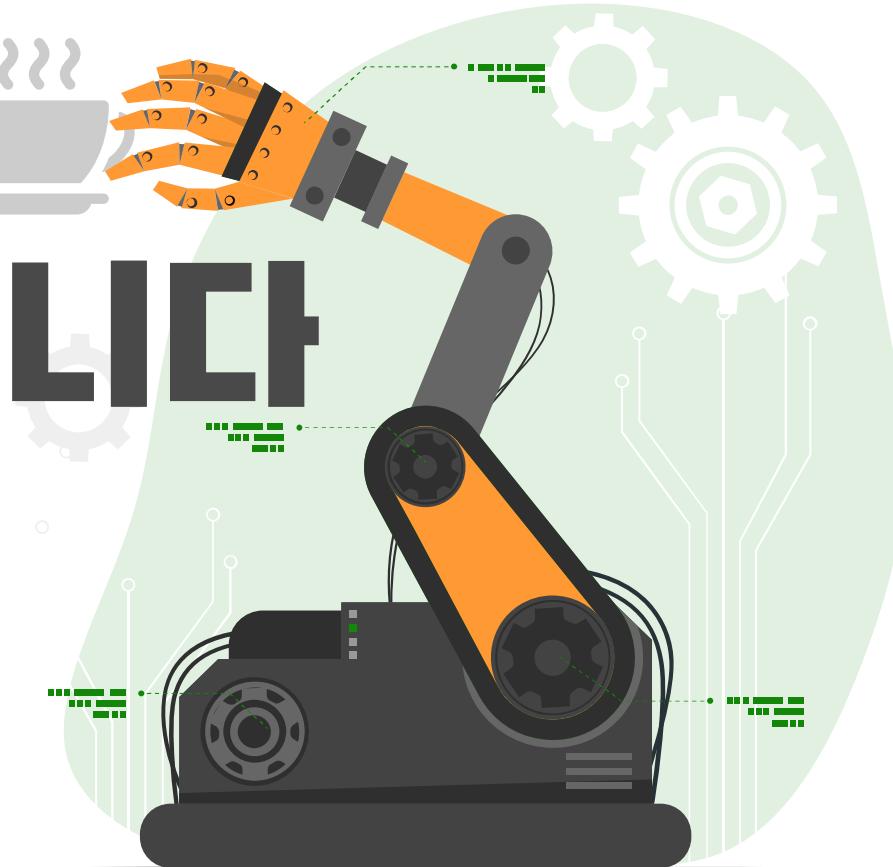
# 핵심정리



- 파이썬에서는 자료형은 데이터의 유형입니다.
- 파이썬에서 사용가능한 자료형으로 정수형, 실수형, 문자열등이 있습니다.
- 문자열은 큰따옴표("...")나 작은따옴표('...')을 사용할 수 있습니다.
- 문자열을 정수로 변경하려면 int()를, 실수로 변경하려면 float()를 사용한다.
- 문자열은 + 연산자로 합칠 수 있습니다.
- 리스트는 자료들을 모아서 저장할 수 있다.

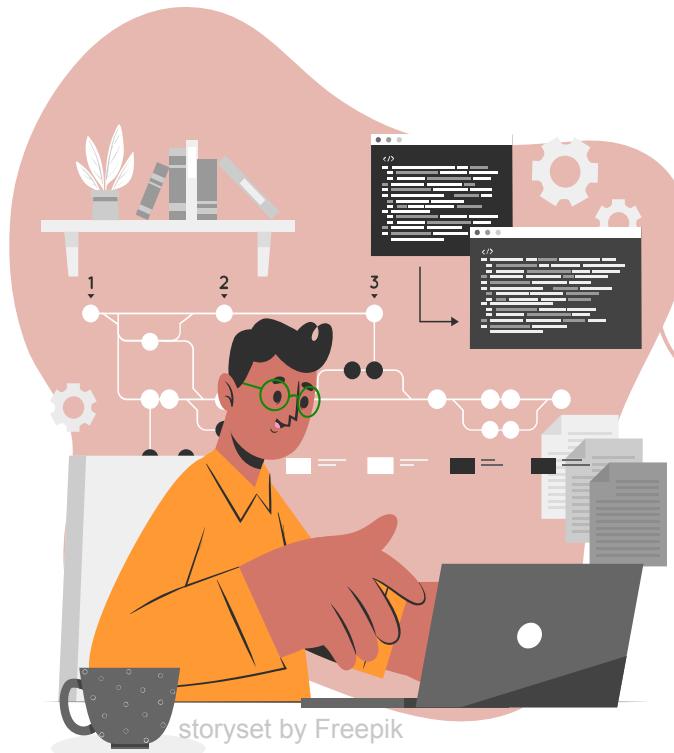


# 수고하셨습니다





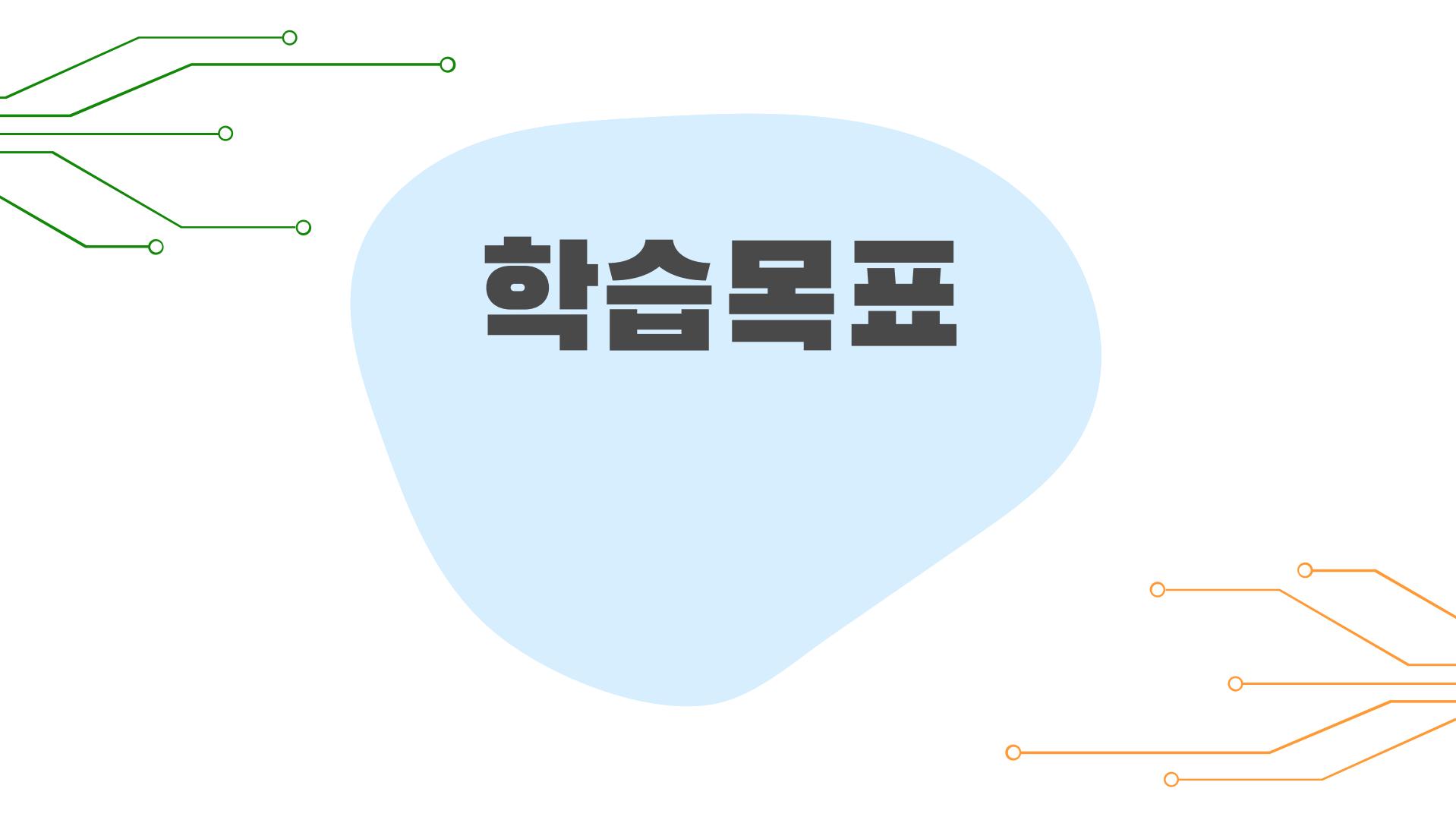
# 컴퓨팅 사고와 프로그래밍



CH 05.

## 조건

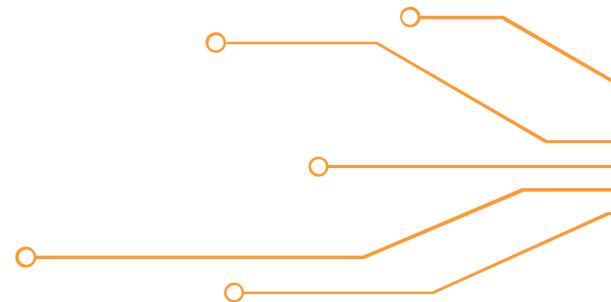
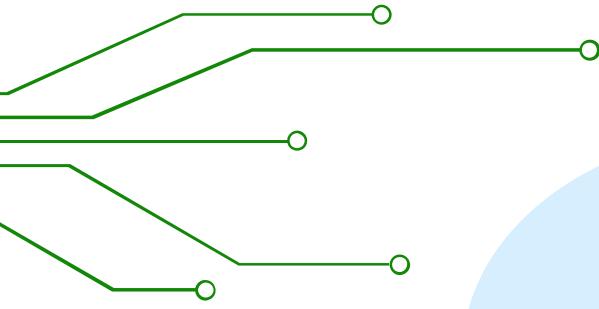
# 학습목표



1. 프로그래밍에서 제어문을 설명할 수 있다.
2. if 문을 이용한 조건문과 if-else 를 이용한 조건문을 사용할 수 있다.
3. 조건문에서 관계연산자와 논리연산자를 사용할 수 있다.
4. 중첩된 조건문을 이용하여 프로그래밍할 수 있다.

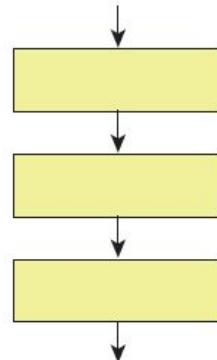


# 이론수업

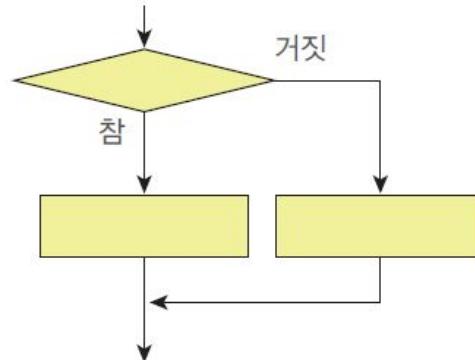


- 순차 구조(sequence) - 명령들이 순차적으로 실행되는 구조이다.
- 선택 구조(selection) - 둘 중의 하나의 명령을 선택하여 실행되는 구조이다.
- 반복 구조(iteration) - 동일한 명령이 반복되면서 실행되는 구조이다.

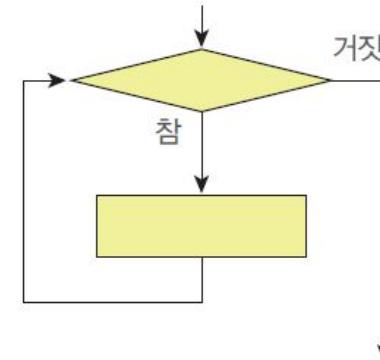
순차구조



선택구조



반복구조



- 선택 구조가 없다면 프로그램은 항상 동일한 동작만을 되풀이할 것이다.
- (예) 자율 주행 자동차 프로그램이 신호등이나장애물에 따라서 동작을 다르게 하지 않는다면 어떻게 될까?



### 만약 ~ 하면, ==> ~ 한다

- 게임에서 정답을 맞혔으면 ==> 점수를 1 증가한다.
- 포탄이 우주선을 맞추었으면 ==> 폭발음을 낸다.
- 파일이 폴더에 없으면 ==> 오류 메시지를 출력한다.

## if-else 문

```
만약      조건식  
if score >= 60 :  
    print("합격입니다.")  
else :  
    print("불합격입니다.")
```

콜론은 아직 문장이 종료되지 않았다는 것을 의미한다.

그렇지 않으면

조건식이 참일 때 실행되는 문장, then 절이라고 한다.

조건식이 참이 아닐 때 실행되는 문장, else 절이라고 한다.

```
score = int(input("성적을 입력하시오: "))

if score >= 60:          # score가 60 이상이면
    print("합격입니다.")  # 합격이다.
else:                   # 그렇지 않으면
    print("불합격입니다.") # 불합격이다.
```

성적을 입력하시오: 80  
합격입니다.

```
num = int(input("정수를 입력하시오: "))

if num % 2 == 0 :          # num을 2로 나눈 나머지가 0이면
    print("짝수입니다.")    # 짝수이다.
else:                      # 그렇지 않으면
    print("홀수입니다.")    # 홀수이다.
```

정수를 입력하시오: 10  
짝수입니다.

조건이 참이어서 여러 개의 문장이 실행되어야 한다면 어떻게 해야 하는가?

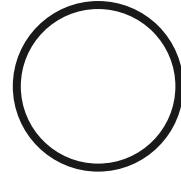
- 들여쓰기를 이용한 블록 이용

### 블록문

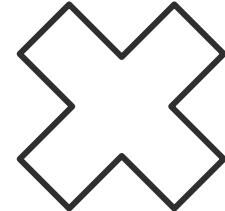
```
if score > 90 :  
    print("합격입니다.")  
    print("장학금도 받을 수 있습니다.")
```

블록: 여러 문장들을 묶은 것이다.

```
score = int(input("점수를 입력하시오: "))
if score >= 90 :
    print("합격입니다.")
    print("장학금도 받을 수 있습니다.")
```

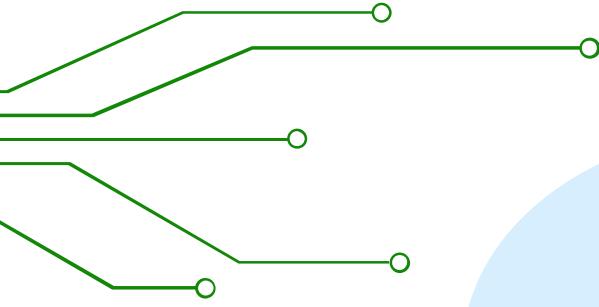


```
score = int(input("점수를 입력하시오: "))
if score >= 90 :
    print("합격입니다.")
    print("장학금도 받을 수 있습니다.")
```





실습



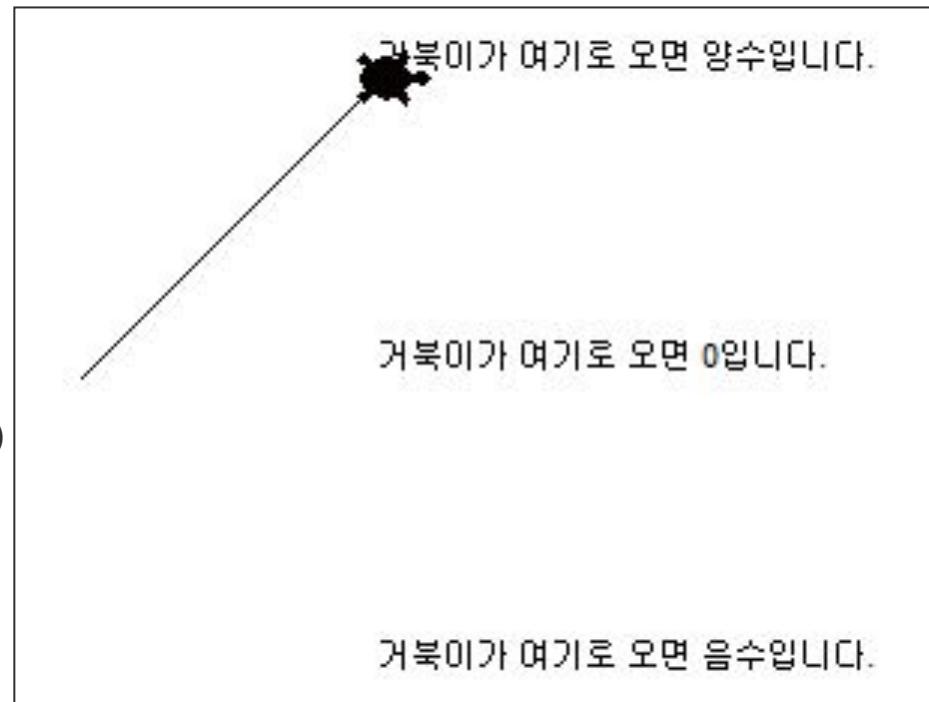
## 실습

### Lab: 부호에 따라 거북이를 움직이자

사용자로부터 정수를 받아서 정수의 부호에 따라서 거북이를  $(100, 100)$ ,  $(100, 0)$ ,  $(100, -100)$ 으로 움직이는 프로그램을 작성해보자.

터틀그래픽에서 숫자를 입력받을 때  
사용할 수 있는 명령문장

```
s = turtle.textinput("", "숫자를 입력하시오: ")
```



```

import turtle
t = turtle.Turtle()
t.shape("turtle")

t.penup()
t.goto(100, 100)
t.write("거북이가 여기로 오면 양수입니다.")
t.goto(100, 0)
t.write("거북이가 여기로 오면 0입니다.")
t.goto(100, -100)
t.write("거북이가 여기로 오면 음수입니다.")

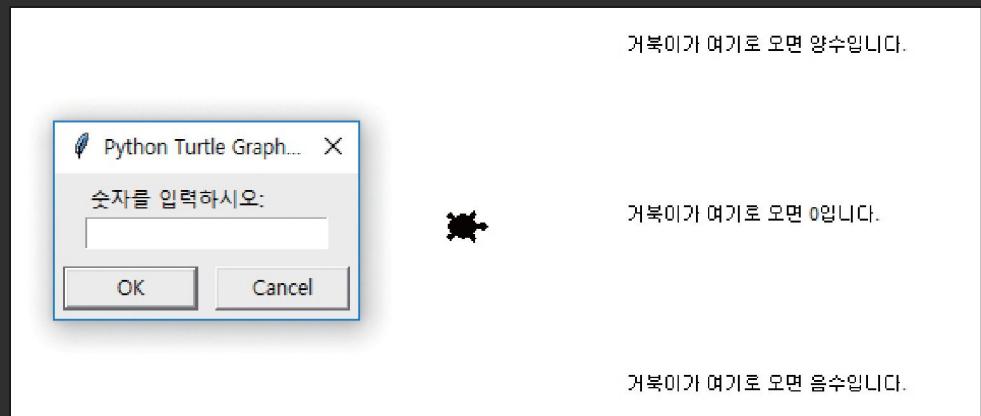
t.goto(0, 0)
t.pendown()
s = turtle.textinput("", "숫자를 입력하시오: ")
n=int(s)

if n > 0 :
    t.goto(100, 100)
if n == 0 :
    t.goto(100, 0)
if n < 0 :
    t.goto(100, -100)
turtle.done()

```

# 펜을 올려서 그림이 그려지지 않게 한다.  
# 거북이를 (100, 100)으로 이동시킨다.

# (0, 0) 위치로 거북이를 이동시킨다.  
# 펜을 내려서 그림이 그려지게 한다.



## 실습

## Lab: 영화 나이 제한 검사

안녕하세요? 두근두근 영화관입니다.

나이를 입력하시오: 19

이 영화를 보실 수 있습니다.

안녕하세요? 두근두근 영화관입니다.

나이를 입력하시오: 14

이 영화를 보실 수 없습니다.



실습

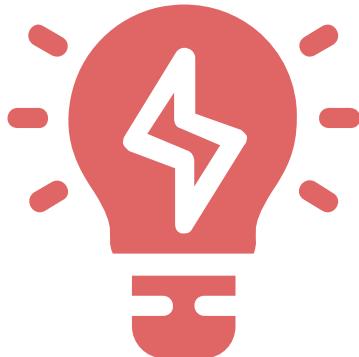
SOL.

```
print("안녕하세요? 두근두근 영화관입니다.")
age = int(input("나이를 입력하시오: "))

if age >= 15:
    print("이 영화를 보실 수 있습니다.")
else:
    print("이 영화를 보실 수 없습니다.")
```

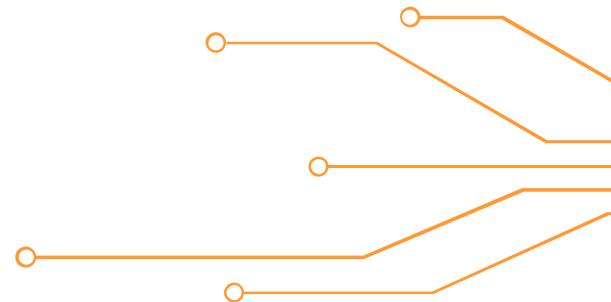
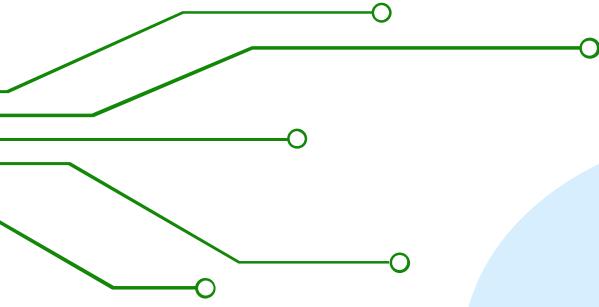


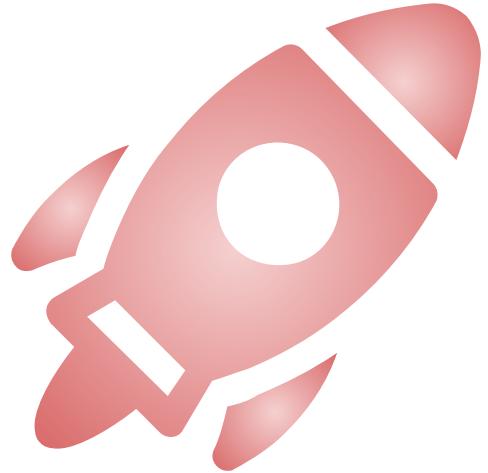
15세 이상이면, "이 영화를 보실 수 있습니다." 메시지에 추가로 "영화의 가격은 10000원 입니다." 를 출력하고, 15세 미만이면 "이 영화를 보실 수 없습니다." 메시지에 추가로 "다른 영화를 보시겠어요?" 를 출력해 보세요.





# 이론수업





복잡한 조건을 표현하는 **"논리연산자"**

(나이가 10살 이상이다) **and** (키가 165cm 이상이다)  
→ 놀이기구를 탈 수 있다.

그리고

age  $\geq 10$  and height  $\geq 165$

나이가 10살 이상

키가 165cm 이상

연산	의미
x and y	AND 연산, x와 y가 모두 참이면 참, 그렇지 않으면 거짓
x or y	OR 연산, x나 y중에서 하나만 참이면 참, 모두 거짓이면 거짓
not x	NOT 연산, x가 참이면 거짓, x가 거짓이면 참

그리고  
age  $\geq 10$  and height  $\geq 165$   
나이가 10살 이상      키가 165cm 이상

성적이

- ✓ 90점 이상이면 ==> A
- ✓ 80점 이상이면 ==> B
- ✓ 70점 이상이면 ==> C
- ✓ 60점 이상이면 ==> D
- ✓ 60점 미만이면 ==> F

<b>if</b> (조건1)	실행1		
<b>else</b>	<b>if</b> (조건2)		실행2
	<b>else</b>	<b>if</b> (조건3)	실행3
		<b>else</b> (기타)	실행4



IF 성적이 90 이상이면,

학점은 A

ELSE IF 성적인 80 이상이면,

학점은 B

ELSE IF 성적인 70 이상이면,

학점은 C

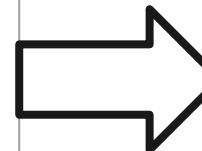
ELSE IF 성적인 60 이상이면,

학점은 D

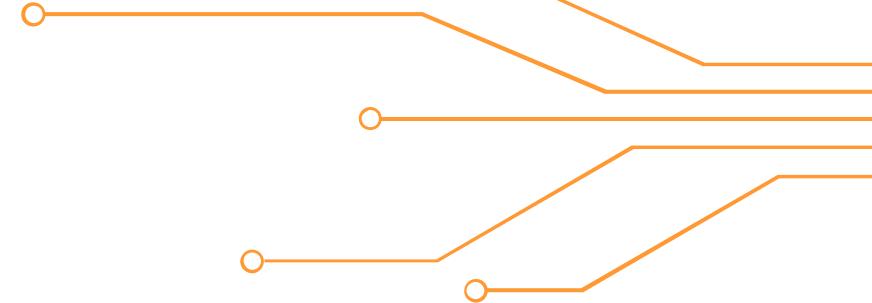
ELSE

학점은 F

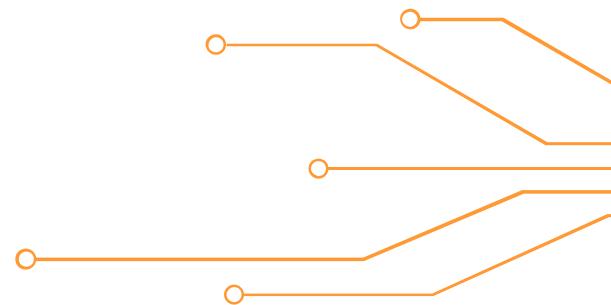
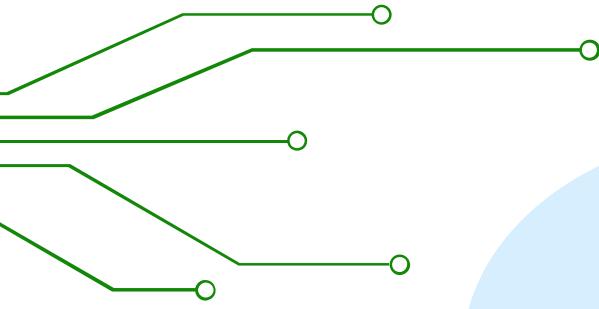
```
if (조건1):  
    실행1  
else :  
    if (조건2):  
        실행2  
    else :  
        if (조건3):  
            실행3  
        else :  
            실행4
```



```
if (조건1):  
    실행1  
elif (조건2):  
    실행2  
elif (조건3):  
    실행3  
else:  
    실행4
```



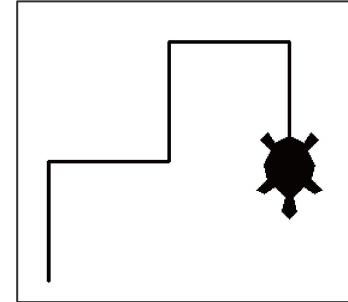
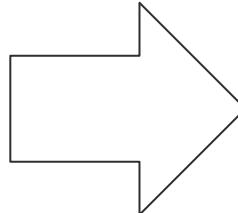
실습



- 파이썬 쉘에서 “l”을 입력하면 거북이가 왼쪽으로 “r”을 입력하면 거북이가 오른쪽으로 100픽셀 이동
- 계속 반복 ==> 무한반복 구조 사용

```
while True:  
    # 실행문  
    # 실행문
```

명령을 입력하시오: r  
명령을 입력하시오: l  
명령을 입력하시오: r  
명령을 입력하시오: r  
명령을 입력하시오:



```
import turtle

# 거북이를 만든다.
t = turtle.Turtle()
# 커서의 모양을 거북이로 한다.
t.shape("turtle")

# 거북이가 그리는 선의 두께를 3으로 한다.
t.width(3)

# 거북이를 3배 확대한다.
t.shapesize(3, 3)

# 무한 루프이다.
while True:
    command = input("명령을 입력하시오: ")
    if command == "l":          # 사용자가 "l" 을 입력하였으면
        t.left(90)
        t.forward(100)
    if command == "r":          # 사용자가 "r" 을 입력하였으면
        t.right(90)
        t.forward(100)
turtle.done()
```

실습

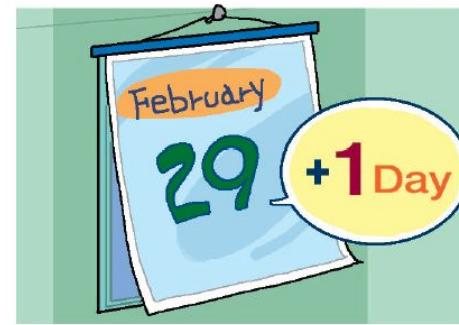
도전

1. 사용자가 l 또는 left 를 입력하면 왼쪽으로, r 또는 right 를 입력하면 오른쪽으로 움직이게 해보자.



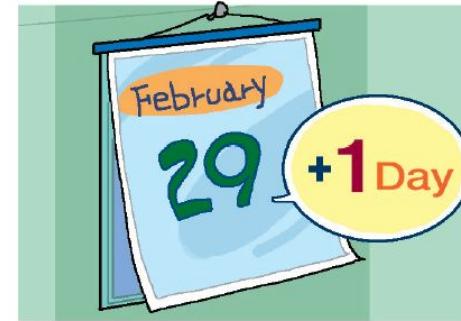
입력한 연도가 윤년인지 아닌지를 판단하는 프로그램을 만들어 보자.

연도를 입력하시오: 2012  
2012년은 윤년입니다.



- 연도가 4로 나누어 떨어지면 윤년이다.
- 100으로 나누어 떨어지는 연도는 제외한다.
- 400으로 나누어 떨어지는 연도는 윤년이다.

- 연도가 4로 나누어 떨어지면 윤년이다.
- 100으로 나누어 떨어지는 연도는 제외한다.
- 400으로 나누어 떨어지는 연도는 윤년이다.


$$(\text{year \% 4} == 0) \text{ and } (\text{year \% 100} != 0) \text{ or } (\text{year \% 400} == 0)$$

연도가 4로 나누어떨어진다.

100으로 나누어떨어지는  
연도는 제외한다.

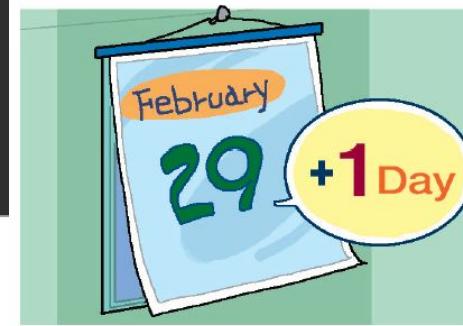
400으로 나누어떨어지는  
연도는 윤년이다.

실습

SOL.

```
year = int(input("연도를 입력하시오: "))

if ( (year % 4 ==0 and year % 100 != 0) or year % 400 == 0):
    print(year, "년은 윤년입니다.")
else :
    print(year, "년은 윤년이 아닙니다.")
```



- 동전을 던지기 게임을 작성해 보자.
- import random한 후에 random.randrange(2) 하면 0이나 1을 랜덤하게 생성할 수 있다.

동전 던지기 게임을 시작합니다.  
뒷면입니다.  
게임이 종료되었습니다.



```
import random

print("동전 던지기 게임을 시작합니다.")
coin = random.randrange(2)

if coin == 0 :
    print("앞면입니다.")
else :
    print("뒷면입니다.")
print("게임이 종료되었습니다.")
```

도전문제 : 주사위 던지기 게임으로 변환 - `elif` 사용해서 1부터 6까지 출력하게



## 실습

## Lab: 동전 던지기 게임(그래픽 버전)

동전을 던지기 게임을 그래픽 버전으로 만들어보자.

이미지위치 :

[https://github.com/MLBasic/CS\\_Programming/tree/main/img](https://github.com/MLBasic/CS_Programming/tree/main/img)

<https://github.com/MLBasic>에서 검색



또는



이미지를 불러오려면

```
screen = turtle.Screen()

image1 = "front.gif"
screen.addshape(image1)      # 이미지를 추가한다.
t1.shape(image1)            # 거북이의 모양을 설정한다.
t1.stamp()                  # 현재 위치에 이미지를 찍는다.
```

```
import turtle      # 터틀 그래픽 모듈을 불러온다.
import random     # 난수 모듈을 불러온다.
```

```
screen = turtle.Screen()
image1 = "front.gif"
image2 = "back.gif"
screen.addshape(image1)
screen.addshape(image2)

t1 = turtle.Turtle()        # 첫 번째 거북이를 생성한다.
```

```
coin = random.randrange(2)

if coin == 0 :
    t1.shape(image1)
    t1.stamp()
else :
    t1.shape(image2)
    t1.stamp()
```

SOL.

사용자로부터 아이디를 받아서 프로그램에 저장된 아이디와 일치하는지 여부를 출력하는 프로그램을 작성해보자.

아이디를 입력하시오: ilovepython  
환영합니다.

아이디를 입력하시오: iloveruby  
아이디를 찾을 수 없습니다.



```
id = "ilovepython"  
s = input("아이디를 입력하시오 : ")  
  
if s == id:  
    print("환영합니다 .")  
else:  
    print("아이디를 찾을 수 없습니다 .")
```



### 도전문제

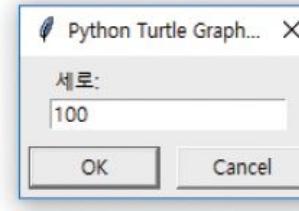
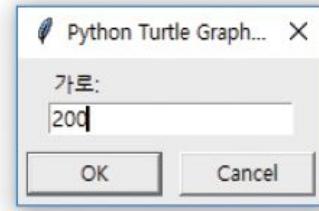
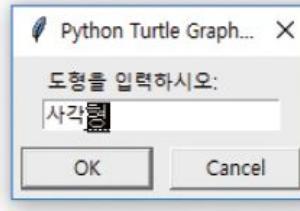
아이디 검사가 종료되면 바로 패스워드 검사를 해보자. 즉 다음과 같은 출력을 가지는 프로그램을 작성한다.

아이디를 입력하시오: ilovepython

패스워드를 입력하시오: 123456

환영합니다.

터틀 그래픽을 이용하여 사용자가 선택하는 도형을 화면에 그리는 프로그램을 작성해보자. 도형은 “사각형”, “삼각형”, “원” 중의 하나이다. 각 도형의 치수는 사용자에게 물어보도록 하자.



## 실습

## draw\_figure.py

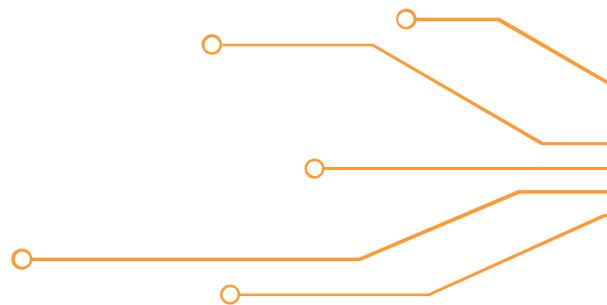
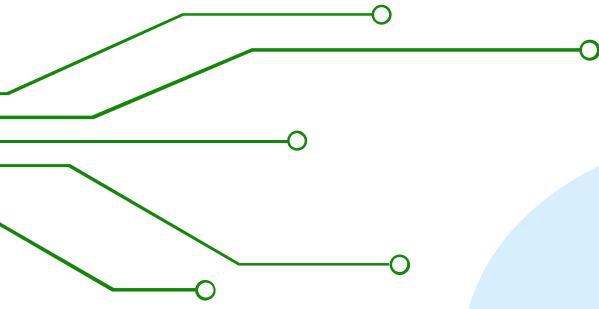
```
import turtle  
t = turtle.Turtle()  
t.shape("turtle")  
  
s = turtle.textinput("", "도형을 입력하시오: ")  
if s == "사각형" :  
    w = turtle.numinput("", "가로: ")  
    h = turtle.numinput("", "세로: ")  
    t.forward(w)  
    t.left(90)  
    t.forward(h)  
    t.left(90)  
    t.forward(w)  
    t.left(90)  
    t.forward(h)  
  
    elif s == "삼각형" :  
        pass  
  
    else :  
        pass  
  
    turtle.done()
```



```
elif s == "삼각형" :  
    w = turtle.numinput("", "길이: ")  
    t.forward(w)  
    t.left(120)  
    t.forward(w)  
    t.left(120)  
    t.forward(w)  
  
elif s == "원" :  
    r = turtle.numinput("", "반지름: ")  
    t.circle(r)  
  
else :  
    turtle.write("사각형, 삼각형, 원 중에 하나를 입력하세요")  
  
turtle.done()
```



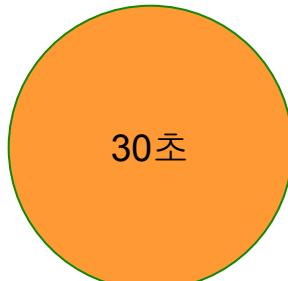
# 퀴즈



1. 다음 프로그램의 출력 결과를 고르세요.

```
age = 20
if age < 20:
    print("20살 미만")
else:
    print("20살 이상")
```

- 1) 20살 미만
- 2) 20살 이상



30초

1. 다음 프로그램의 출력 결과를 고르세요. [ 2 ]

```
age = 20
if age < 20:
    print("20살 미만")
else:
    print("20살 이상")
```

- 1) 20살 미만
- 2) 20살 이상



# 1. 다음 프로그램의 출력 결과를 고르세요. [ 2 ]

```
age = 20
if age < 20:
    print("20살 미만")
else:
    print("20살 이상")
```

age < 20 은 변수 age 의 값이 20 보다  
작은 경우에만 참이 됩니다.

1) 20살 미만

2) 20살 이상

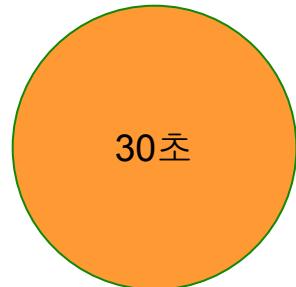




2. 다음 프로그램은 학생의 시험성적이 90점 이상이면 A, 80점 이상이면 B, 70점 이상이면 C, 60점 이상이면 D, 60점 보다 작으면 F를 학점으로 주는 프로그램입니다. 빈 칸에 알맞은 것을 적어보세요.

```
score = int(input("성적을 입력하시오: "))

[ ] score >= 90:
    print("A학점입니다.")
[ ] score >= 80:
    print("B학점입니다.")
[ ] score >= 70:
    print("C학점입니다.")
[ ] score >= 60:
    print("D학점입니다.")
[ ] :
    print("F학점입니다.")
```



2. 다음 프로그램은 학생의 시험성적이 90점 이상이면 A, 80점 이상이면 B, 70점 이상이면 C, 60점 이상이면 D, 60점 보다 작으면 F를 학점으로 주는 프로그램입니다. 빈 칸에 알맞은 것을 적어보세요.

```
score = int(input("성적을 입력하시오: "))

if score >= 90:
    print("A학점입니다.")
elif score >= 80:
    print("B학점입니다.")
elif score >= 70:
    print("C학점입니다.")
elif score >= 60:
    print("D학점입니다.")
else :
    print("F학점입니다.")
```



2. 다음 프로그램은 학생의 시험성적이 90점 이상이면 A, 80점 이상이면 B, 70점 이상이면 C, 60점 이상이면 D, 60점 보다 작으면 F를 학점으로 주는 프로그램입니다. 빈 칸에 알맞은 것을 적어보세요.

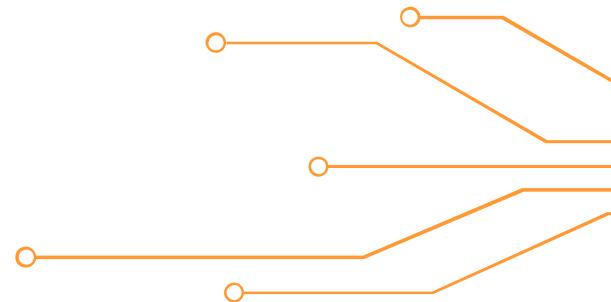
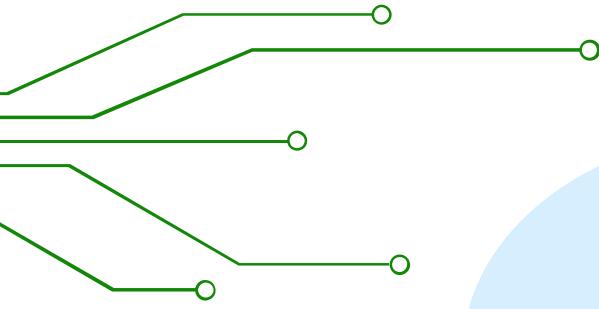
```
score = int(input("성적을 입력하시오: "))

if score >= 90:
    print("A학점입니다.")
elif score >= 80:
    print("B학점입니다.")
elif score >= 70:
    print("C학점입니다.")
elif score >= 60:
    print("D학점입니다.")
else :
    print("F학점입니다.")
```

if .. elif .. else 구문을 이용합니다.

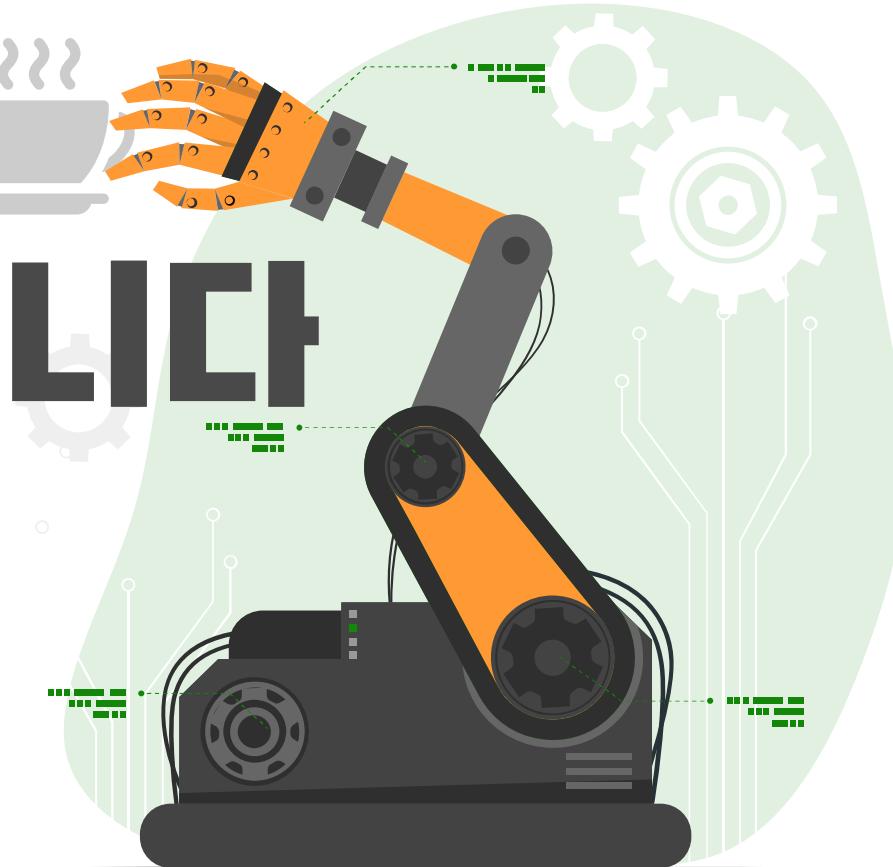


# 핵심정리



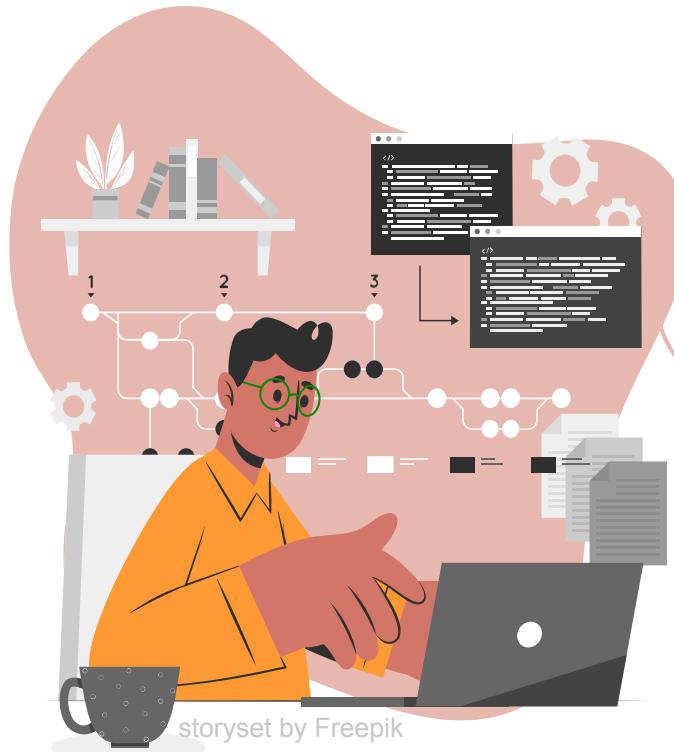
- 파이썬의 제어문으로 문장들의 실행 순서를 제어할 수 있습니다.
- if - else 문에서 if 뒤의 조건식이 참이면 if 아래 블록에 있는 내용이 실행됩니다.  
그렇지 않으면 else 아래의 블록에 있는 내용이 실행됩니다.
- if .. elif .. else 구문을 사용하면 다양한 경우를 제어할 수 있습니다.
- 논리연산자에는 and, or, not 이 있으며 and 는 앞뒤의 모든 내용이 참일때만  
결과가 참이 되고, or 는 앞뒤의 내용중 하나라도 참이면 결과가 참이 되며, not 은  
뒤의 내용에 반대되는 결과를 돌려줍니다.
- if 뒤에는 조건이 있고 그 뒤에 : 이 반드시 있어야하며, 그 다음줄은 일정한  
들여쓰기가 된 블록이 위치합니다.

# 수고하셨습니다





# 컴퓨팅 사고와 프로그래밍



CH 06.

반복

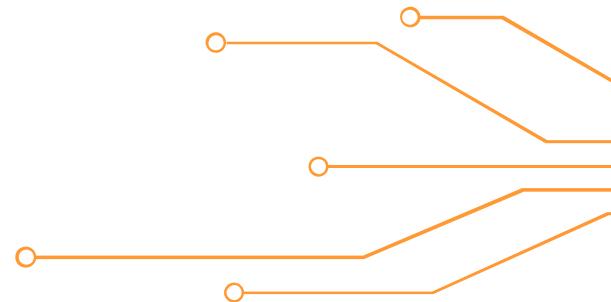
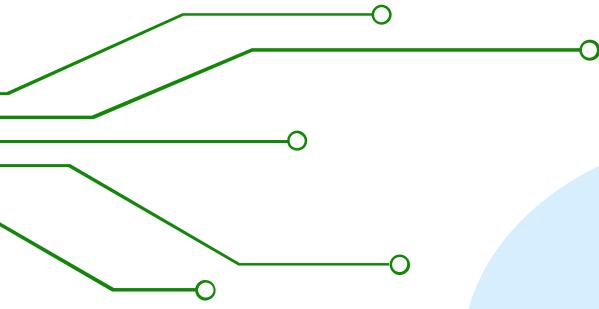
# 학습목표



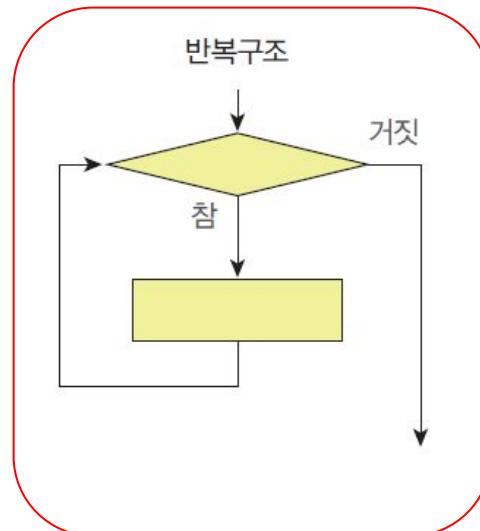
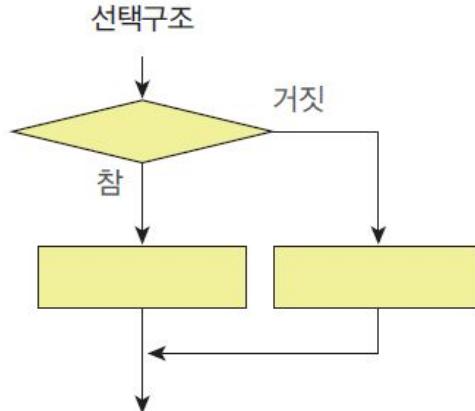
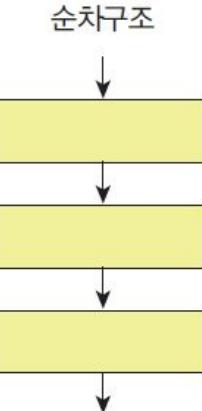
1. while 문을 사용하여 반복문을 작성할 수 있다.
2. for 문을 사용하여 정해진 횟수만큼 반복할 수 있다.
3. 리스트에 저장된 항목들에 대해서 저장된 개수만큼 반복할 수 있다.



# 이론수업



- 순차 구조(sequence) - 명령들이 순차적으로 실행되는 구조이다.
- 선택 구조(selection) - 둘 중의 하나의 명령을 선택하여 실행되는 구조이다.
- 반복 구조(iteration) - 동일한 명령이 반복되면서 실행되는 구조이다.

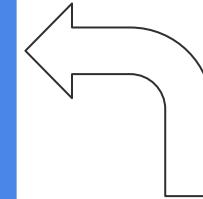


- 반복(iteration)은 동일한 문장을 여러 번 반복시키는 구조
- 컴퓨터는 인간과 다르게 반복적인 작업을 실수 없이 빠르게 할 수 있다.  
이것이 컴퓨터의 가장 큰 장점



- 반복(iteration)은 동일한 문장을 여러 번 반복시키는 구조
- 컴퓨터는 인간과 다르게 반복적인 작업을 실수 없이 빠르게 할 수 있다.  
이것이 컴퓨터의 가장 큰 장점

방문을 환영합니다!  
방문을 환영합니다!  
방문을 환영합니다!  
방문을 환영합니다!  
방문을 환영합니다!



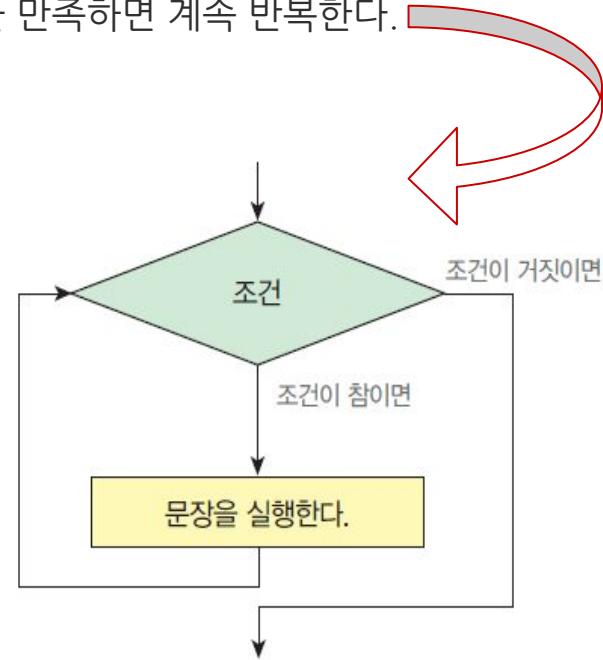
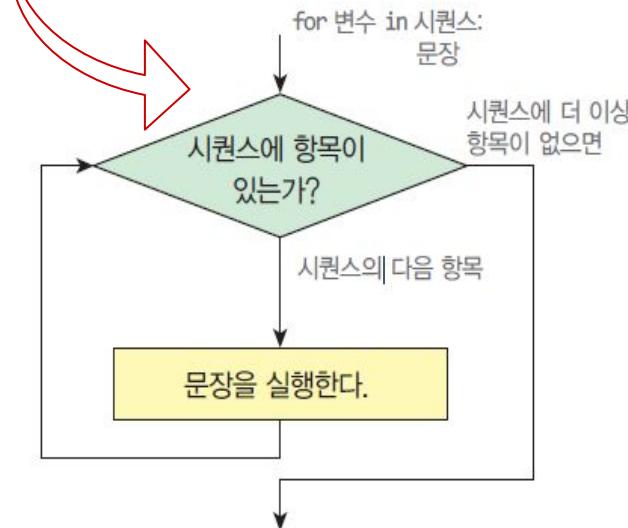
```
print("방문을 환영합니다!")
print("방문을 환영합니다!")
print("방문을 환영합니다!")
print("방문을 환영합니다!")
print("방문을 환영합니다!")
```

```
print("방문을 환영합니다!")
print("방문을 환영합니다!")
print("방문을 환영합니다!")
print("방문을 환영합니다!")
print("방문을 환영합니다!")
...
...
...
...
...
...
...
...
print("방문을 환영합니다!")
...
...
```

1000 번

```
for i in range(1000):
    print("방문을 환영합니다!")
```

- 횟수 제어 반복(for 문): 정해진 횟수만큼 반복한다.
- 조건 제어 반복(while 문): 특정한 조건을 만족하면 계속 반복한다.

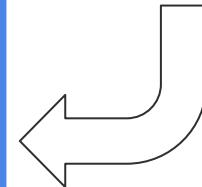


파이썬에서 횟수 제어 반복은 for 루프라고도 한다.

```
for i in [1, 2, 3, 4, 5]:  
    print("방문을 환영합니다.")
```

# 끝에 :이 있어야 함  
# 들여쓰기하여야 함

방문을 환영합니다!  
방문을 환영합니다!  
방문을 환영합니다!  
방문을 환영합니다!  
방문을 환영합니다!



## 이론수업

## 횟수 제어 반복의 진행 과정

```
for i in [ 1, 2, 3, 4, 5 ] :  
    print("방문을 환영합니다.")
```

```
for i in [ 1, 2, 3, 4, 5 ] :  
    print("방문을 환영합니다.")
```

```
for i in [ 1, 2, 3, 4, 5 ] :  
    print("방문을 환영합니다.")
```

```
for i in [ 1, 2, 3, 4, 5 ] :  
    print("방문을 환영합니다.")
```

```
for i in [ 1, 2, 3, 4, 5 ] :  
    print("방문을 환영합니다.")
```



```
for i in [1, 2, 3, 4, 5]:  
    print("i=", i)
```

```
i= 1  
i= 2  
i= 3  
i= 4  
i= 5
```



```
sum = 0
for i in [1, 2, 3, 4, 5]:
    print(f"{i}를 더하는 중")
    sum += i
print("합계는=", sum)
```

1를 더하는 중  
2를 더하는 중  
3를 더하는 중  
4를 더하는 중  
5를 더하는 중  
합계는= 15

## for 문

```
for 변수 in range(종료 값) :  
    문장
```

0에서 (종료 값-1)까지의 숫자를 반환한다.

반복되는 문장으로  
들여쓰기 하여야 한다.

```
for i in range(5):  
    print("방문을 환영합니다!")
```

방문을 환영합니다!  
방문을 환영합니다!  
방문을 환영합니다!  
방문을 환영합니다!  
방문을 환영합니다!

## range() 함수

시작값이다.

종료 값이지만 stop은 포함되지 않는다.

한 번에 증가되는 값이다.

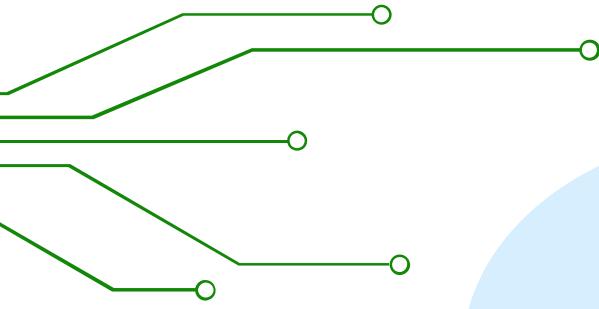
`range(start=0, stop, step=1)`

```
for i in range(1, 6, 1):  
    print(i, end=" ")
```

1 2 3 4 5



실습



## 실습

### Lab. range() 함수 실습 - range\_demo.py

```
for i in range(5, 10):
    print(f"변수 i의 값={i}")
print()

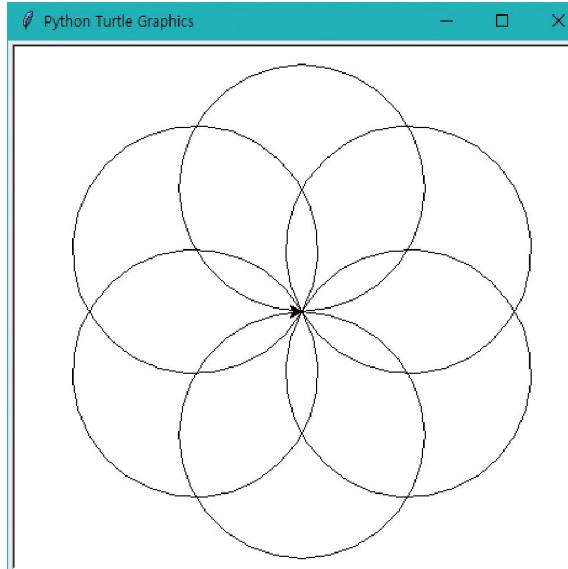
for i in range(1, 10, 3):
    print(f"변수 i의 값={i}")
print()

for i in range(10, 7, -1):
    print(f"변수 i의 값={i}")
```



실습

## Lab: 6개의 원 그리기



```
import turtle  
t = turtle.Turtle()  
  
t.circle(100) # 반지름이 100인 원을 그린다.  
t.left(60) # 60도만큼 왼쪽으로 회전시킨다.  
t.circle(100)  
t.left(60)  
t.circle(100)  
t.left(60)  
t.circle(100)  
t.left(60)  
t.circle(100)  
t.left(60)  
t.circle(100)
```

원을 N 개 그리는 프로그램으로 소스를 고쳐보자!



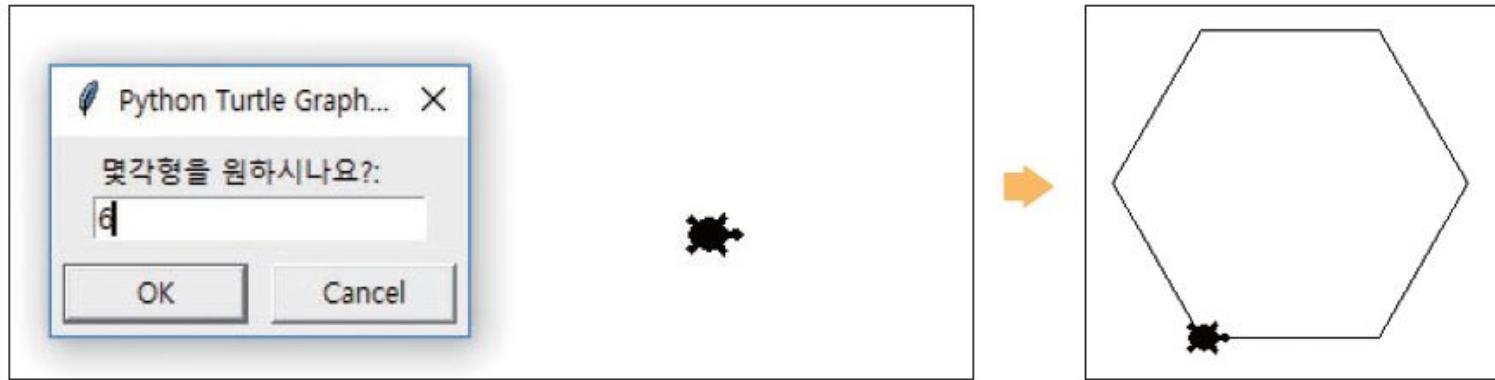
```
import turtle  
t = turtle.Turtle()
```

```
for count in range(6):  
    t.circle(100)  
    t.left(360/6)
```

## 실습

## Lab: n-각형 그리기

사용자로부터 정수  $n$ 을 받아서  $n$ -각형을 그리는 프로그램을 작성할 수 있는가?



```
import turtle  
t = turtle.Turtle()  
t.shape("turtle")  
  
s = turtle.textinput("", "몇각형을 원하시나요 ?:")  
n = int(s)  
  
for i in range(n):  
    t.forward(100)  
    t.left(360/n)  
  
turtle.done()
```



## 실습

## Lab: 카운트 다운 프로그램 만들기

1초에 한 번씩 숫자를 역순으로 출력하고 숫자가 0이 되면 “발사!”라고 출력하는 프로그램을 만들어보자

```
10  
9  
8  
...  
2  
1  
발사!
```



```
import time

for i in range(10, 0, -1):
    print(i)
    time.sleep(1)
print("발사!")
```



사용자가 5초를 입력하면 5초 후에 알람을 울려주는 프로그램을 작성해보자.

초단위의 시간을 입력하시오: 5

5초 남았습니다.

4초 남았습니다.

3초 남았습니다.

2초 남았습니다.

1초 남았습니다.



```
import time
import winsound

seconds = int(input("초단위의 시간을 입력하시오: "))

for i in range(seconds, 0, -1):
    print(f"{i}초 남았습니다.")
    time.sleep(1)

winsound.Beep(2000, 3000)
```

구구단 중에서 사용자가 원하는 단을 반복문을 이용하여 출력하여 보자.

원하는 단은: 9

$9*1=9$

$9*2=18$

$9*3=27$

$9*4=36$

$9*5=45$

$9*6=54$

$9*7=63$

$9*8=72$

$9*9=81$



```
dan = int(input("원하는 단은: "))

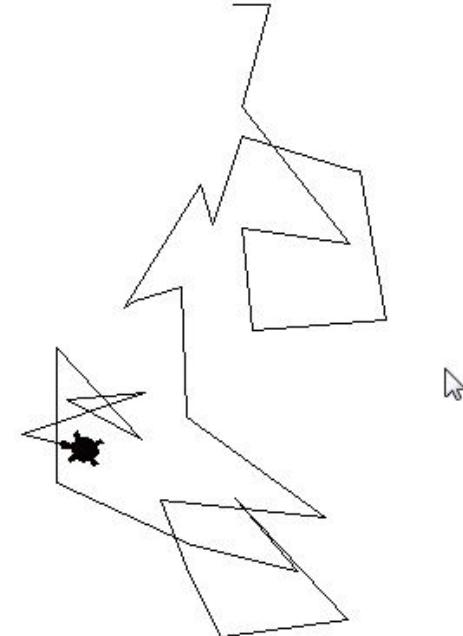
for i in range(1, 10, 1):
    print(f"{dan}*{i}={dan*i}")
```



터틀 그래픽에서 거북이가 술에 취한 것처럼 랜덤하게 움직이게 해보자.

### 30번 반복

- \* [1, 100] 사이의 난수를 발생하여 변수 length에 저장한다.
- \* 거북이를 length만큼 움직인다.
- \* [-180, 180] 사이의 난수를 발생하여 변수 angle에 저장한다.
- \* 거북이를 angle만큼 회전시킨다.



```
import turtle
import random

t = turtle.Turtle()
t.shape("turtle")

for i in range(30):
    length = random.randint(1, 100)
    t.forward(length)
    angle = random.randint(-180, 180)
    t.right(angle)

turtle.done()
```



for문을 이용하여서 팩토리얼을 계산해보자.

팩토리얼  $n!$ 은 1부터  $n$ 까지의 정수를 모두 곱한 것을 의미한다. 즉,  $n! = 1 \times 2 \times 3 \times \dots \times (n-1) \times n$ 이다.

정수를 입력하시오: 10

10!은 3628800이다.



```
n = int(input("정수를 입력하시오: "))
```

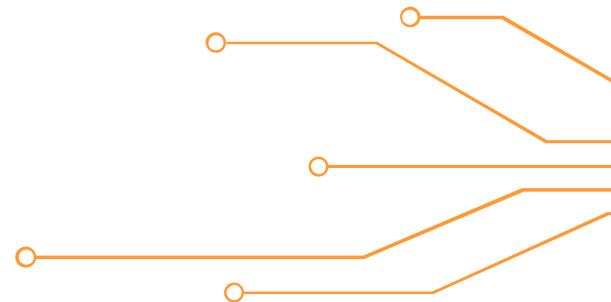
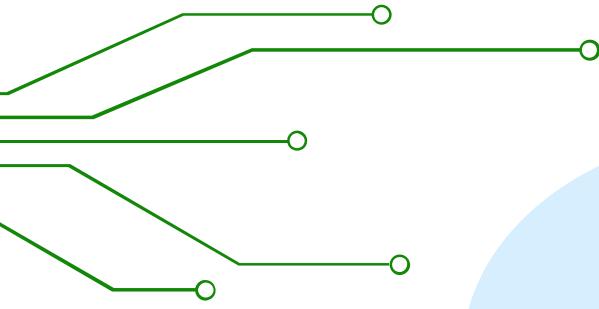
```
fact = 1
```

```
for i in range(1, n+1):
    fact = fact * i
print(n, "!은", fact, "이다.")
```





# 이론수업



조건 제어 반복은 어떤 조건이 만족되는 동안 반복하는 구조

### while 루프

while 조건 :

반복 문장

반복을 하는 조건이다. 조건이 참이면 반복을 계속한다.

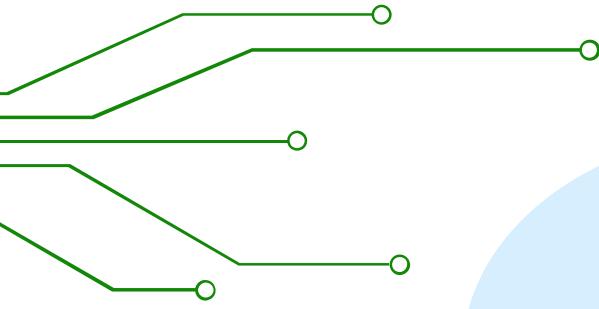
반복되는 문장이다.

```
response = "NO"  
while response != "YES":  
    response = input("ARE YOU READY ?");  
print("LET'S GO")
```





실습



사용자가 암호를 입력하고 프로그램에서 암호가 맞는지를 체크합니다.

```
암호를 입력하시오: idontknow  
암호를 입력하시오: 12345678  
암호를 입력하시오: password  
암호를 입력하시오: pythonisfun  
로그인 성공
```



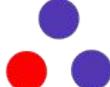
```
password = ""  
  
while password != "pythonisfun":  
    password = input("암호를 입력하시오: ")  
print("로그인 성공")
```

1부터 10까지의 합을 계산하는 예제를 while 루프로 작성해 보자.

합계는 55



1



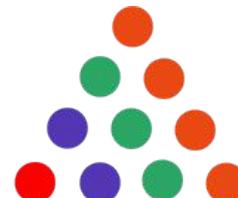
1 + 2

= 3



1 + 2 + 3

= 6



1 + 2 + 3 + 4

= 10

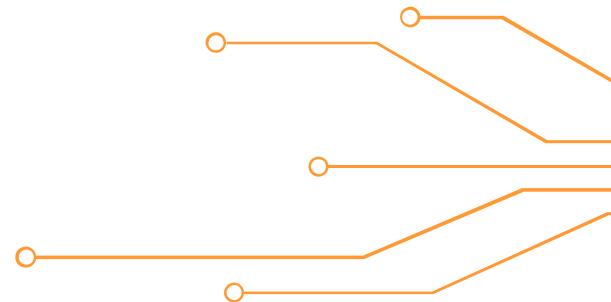
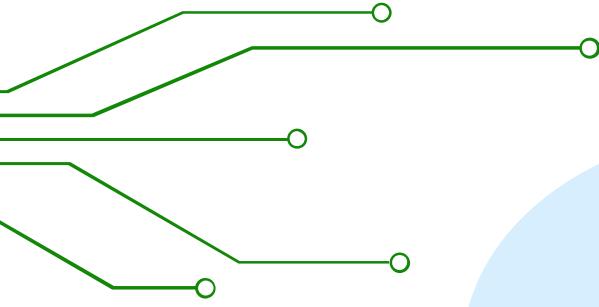


```
count = 1
sum = 0
while count <= 10 :
    sum = sum + count
    count = count + 1
print("합계는", sum)
```





# 이론수업



조건 제어 루프에서 무한히 반복하는 루프를 사용하는 경우도 있다. 이것을 무한 루프(infinite loop)라 한다.  
break 문장은 루프를 강제적으로 빠져 나올 때 사용하는 문장이다.

### break 문

```
while True :
```

```
    반복 문장
```

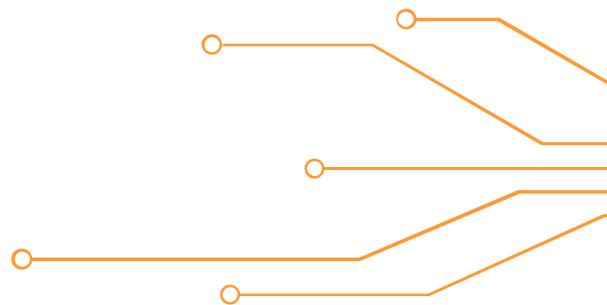
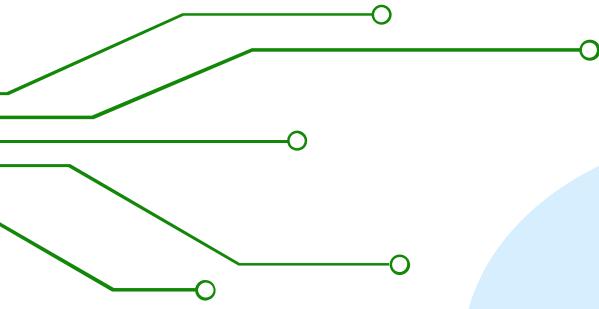
```
    반복 문장
```

```
        if 조건 :  
            break;
```



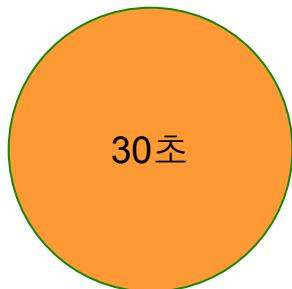


# 퀴즈



## 1. 파이썬의 while 문과 for 문의 차이점은 무엇인가요?

- 1) while 문은 주어진 조건을 만족하는 동안 계속해서 반복하며, for 문은 주어진 시퀀스나 반복 가능한 객체를 순회하며 반복합니다.
- 2) while 문은 주어진 시퀀스나 반복 가능한 객체를 순회하며 반복하며, for 문은 주어진 조건을 만족하는 동안 계속해서 반복합니다.
- 3) while 문과 for 문은 완전히 다르게 사용되고 반복문에는 while만 사용합니다.
- 4) while 문과 for 문은 완전히 동일한 기능을 수행하며, 사용 방식에 차이가 없습니다.



1. 파이썬의 while 문과 for 문의 차이점은 무엇인가요? [ 1 ]

1) while 문은 주어진 조건을 만족하는 동안 계속해서 반복하며, for 문은 주어진 시퀀스나 반복 가능한 객체를 순회하며 반복합니다.

2) while 문은 주어진 시퀀스나 반복 가능한 객체를 순회하며 반복하며, for 문은 주어진 조건을 만족하는 동안 계속해서 반복합니다.

3) while 문과 for 문은 완전히 다르게 사용되고 반복문에는 while만 사용합니다.

4) while 문과 for 문은 완전히 동일한 기능을 수행하며, 사용 방식에 차이가 없습니다.



1. 파이썬의 while 문과 for 문의 차이점은 무엇인가요? [ 1 ]

1) while 문은 주어진 조건을 만족하는 동안 계속해서 반복하며, for 문은 주어진 시퀀스나 반복 가능한 객체를 순회하며 반복합니다.

2) while 문은 주어진 시퀀스나 반복 가능한 객체를 순회하며 반복하며, for 문은 주어진 조건을 만족하는 동안 계속해서 반복합니다.

**while** 은 조건중심, **for** 는 횟수와 시퀀스 중심으로 반복문을 구성합니다 .

3) while 문과 for 문은 완전히 다르게 사용되고 반복문에는 while 만 사용합니다.

**while** 과 **for** 는 모두 반복문에 사용됩니다 .

4) while 문과 for 문은 완전히 동일한 기능을 수행하며, 사용 방식에 차이가 없습니다.

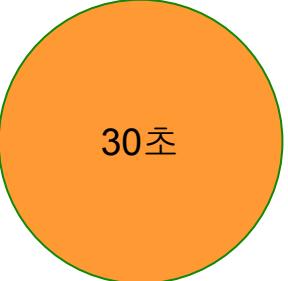
**while** 과 **for** 는 모두 반복문에 사용되지만 사용방식에는 차이가 있습니다 .





2. 아래 코드는 2에서 100사이의 모든 짝수를 출력하는 반복 루프입니다. 빈 부분을 채워넣으세요.

```
[ ] i [ ] range(2, 101):
    if i%2==0 :
        print(i,end = " ")
```



30초



2. 아래 코드는 2에서 100사이의 모든 짝수를 출력하는 반복 루프입니다. 빈 부분을 채워넣으세요.

```
for i in range(2, 101):
    if i%2==0 :
        print(i,end = " ")
```



2. 아래 코드는 2에서 100사이의 모든 짝수를 출력하는 반복 루프입니다. 빈 부분을 채워넣으세요.

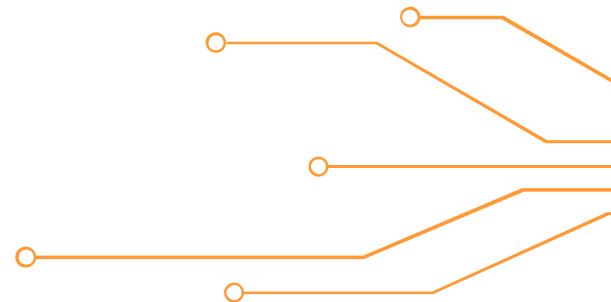
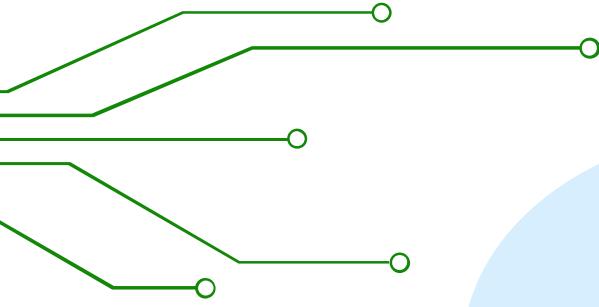
```
for i in range(2, 101):
    if i%2==0 :
        print(i,end = " ")
```

for .. in .. 을 이용하면 정해진 횟수 혹은 요소의 숫자만큼 반복할 수 있습니다.





# 핵심정리



- ❑ while 문은 루프의 첫 부분에서 반복 조건을 검사합니다. while 문은 반복 조건이 정해져 있을 때 유용합니다.
- ❑ for 문은 시퀀스에서 항목들을 하나씩 가지고 와서 반복합니다. 반복 횟수가 정해져 있을 때 유용합니다.
- ❑ 리스트에 저장된 항목들은 for in 형식의 반복문을 사용하여 이용할 수 있습니다.



# 수고하셨습니다

