

# ISEED Programming

YoungHo Kim

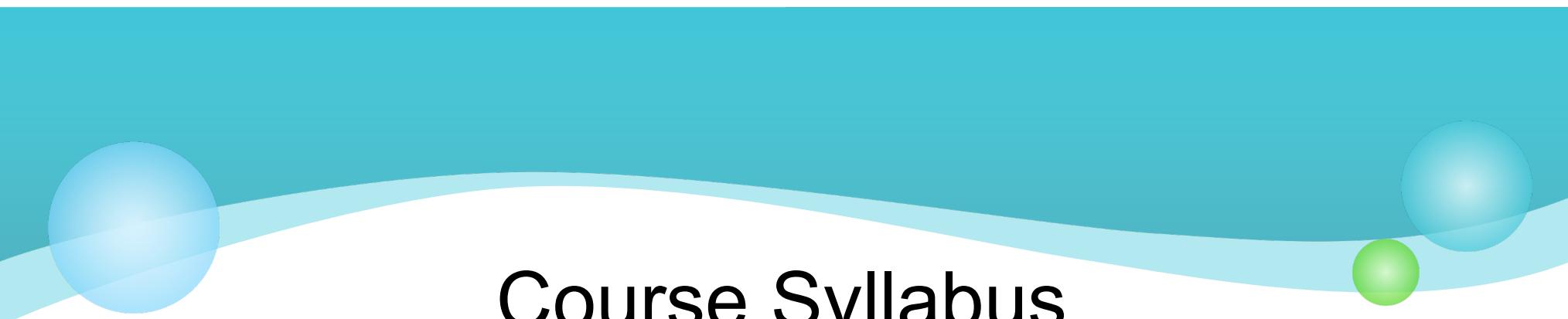
[yhokim@pknu.ac.kr](mailto:yhokim@pknu.ac.kr)

C.P.: 010-4727-8728

# Course Syllabus

1st Semester, 2022

Course Title	i-SEEDprogramming	Course Number	A10800	Class Number	001
Department/ Division	Division of Earth Environmental System Sciences	Academic Year	all	Credit–Lecture– Practice	3–3–0
Class Time	Fri.1,2,3	Classroom		On/Off-Line Option	
Instructor	Young Ho Kim	Instructor's Office		Office Hours	
		Phone		E-mail	yhokim@pknu.ac.kr
Innovative Learning	FL( ) BL( ) PBL( ) TBL( ) ME( ) AL( ) TT/CT( )				
Core Skills	/ /				
Ideal Character		Target Skills			



# Course Syllabus

## 1. Course Information

Course Objectives	Many of the AI machine learning programs, which can be said to be the driving force behind the fourth industrial revolution, have been programmed in Python. In this class, we will practice through various examples so that you can learn and apply the concept and basic grammar of Python.
Course Description	Basic concept of Python Practice by various examples Simple project
Learning Methods	On-line

## 2. Texts and Materials

Required Texts	- 파이썬 for Begginer / 우재남 / 한빛아카데미(2020)
Supplemental Texts	
Extra Materials	

# Course Syllabus

### 3. Evaluation Method

# Course Syllabus

## 4. Course Schedule

Week	Topic	Content	Assignments and Others
1		Introduction for Python	
2		Variables and data types	
3		Operator	
4		Conditional	
5		Loop	
6		List, Tuple, Dictionary	
7		Mid-term exam	
8		String	
9		Function and Module	
10		Window programming	
11		File Input/Output	
12		Object-oriented programming	
13		Database	
14		Mini Project	
15		Final exam	

# 1장 파이썬 소개

# 컴퓨터와 일상생활

- 우리는 일상생활에서 컴퓨터를 많이 사용한다.
- 컴퓨터의 최대 장점: 반복적인 작업을 잘한다.



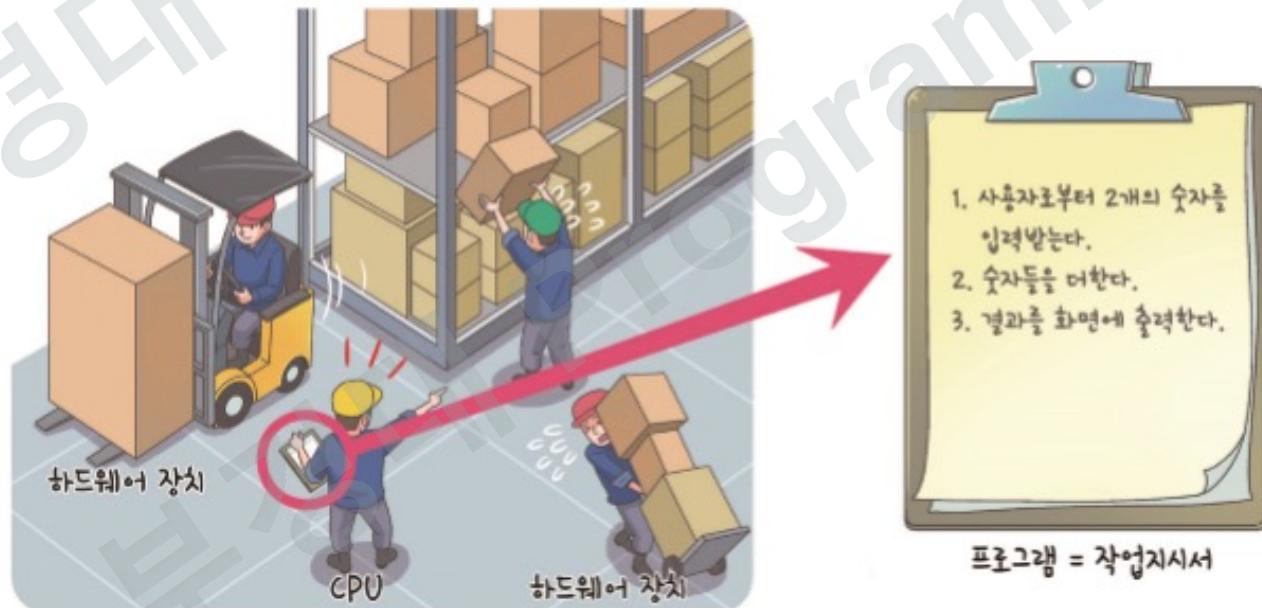
# 범용성

- 컴퓨터의 핵심: 범용성
- (예)스마트폰: 우리는 스마트폰에 다양한 기능을 하는 앱(프로그램)을 설치하여 여러가지 작업을 할 수 있다.



# 컴퓨터 프로그램

- 컴퓨터에 일을 시키려면 인간이 컴퓨터에게 자세한 명령어(instruction)들의 리스트를 주어야 한다.
- 프로그램 (program) : 컴퓨터가 수행할 명령어를 적어놓은 문서



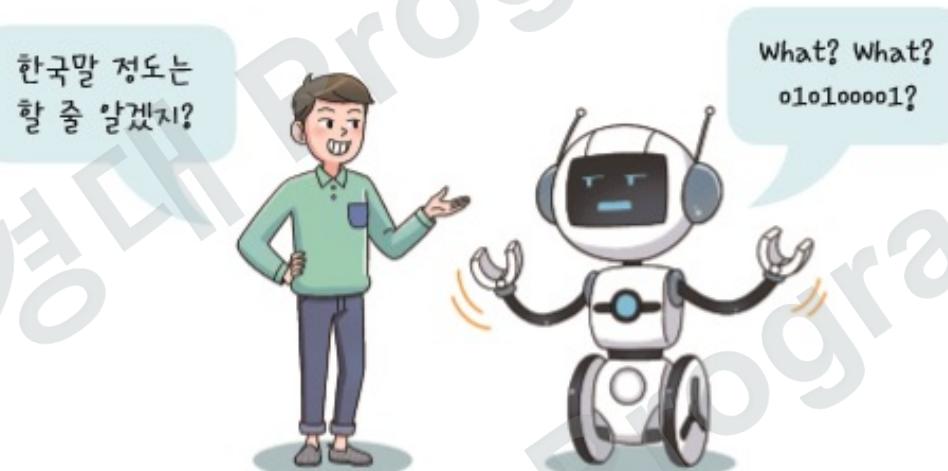
# 임베디드 프로그램

- 프로그램은 컴퓨터에만 설치되는 것이 아니다.
- 임베디드 프로그램(embedded program): 전자기기에 내장되는 프로그램



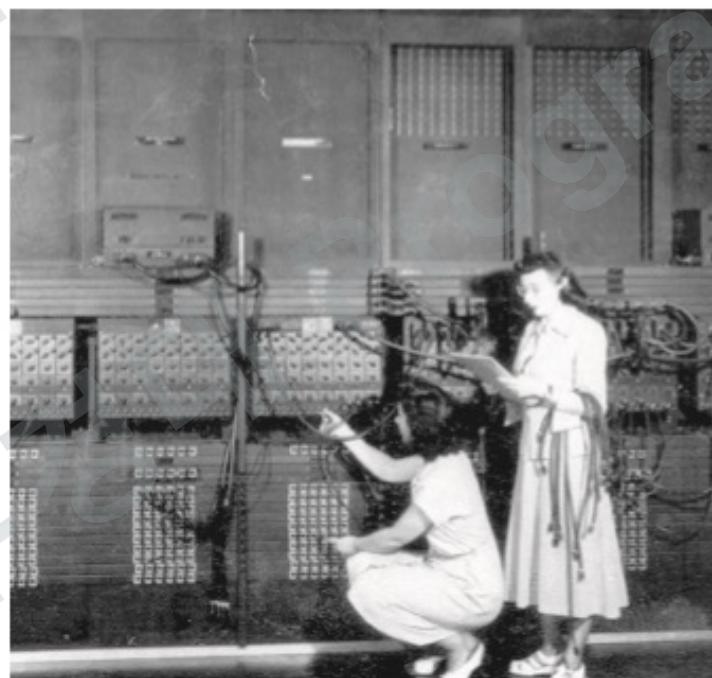
# 프로그래밍 언어

- 컴퓨터는 사람의 언어를 이해할 수 없다!



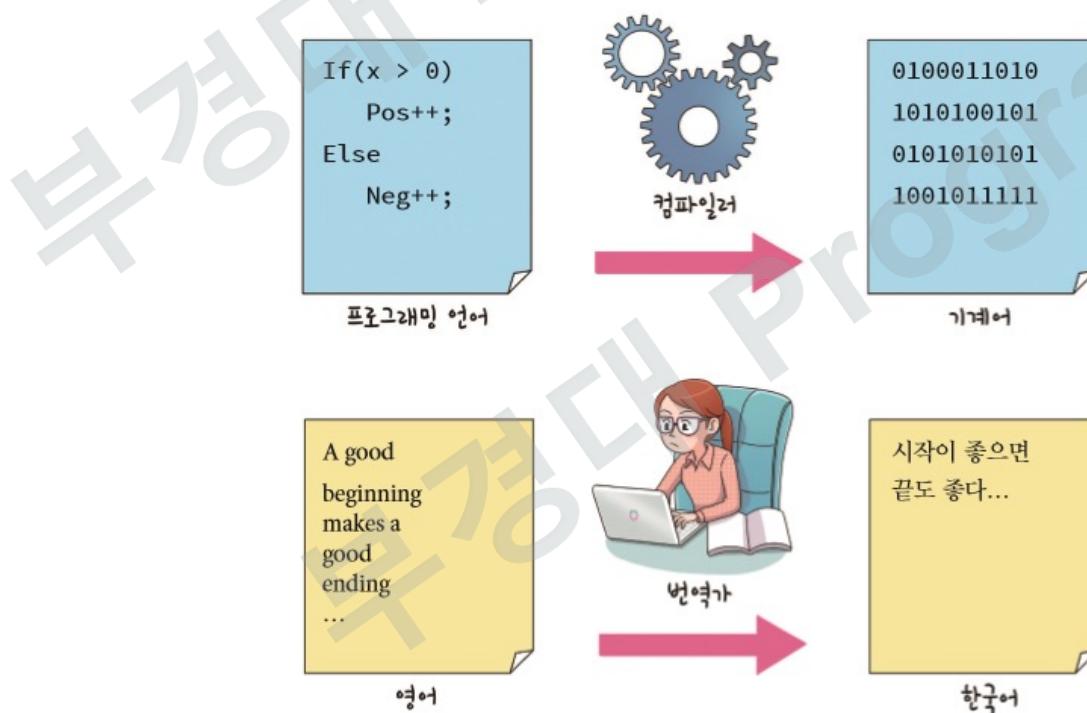
# 기계어

- 기계어 (machine language) : 컴퓨터가 알아듣는 유일한 언어
- 기계어는 0과 1로 구성된다.
- 초기의 컴퓨터에서는 기계어를 사용하여 프로그램을 했었다.



# 프로그래밍 언어란?

- 인간의 언어에 근접한 프로그래밍 언어가 개발된다.
- 인간이 프로그래밍 언어를 배워서 프로그램을 작성하면 컴파일러(compiler)라고 하는 통역을 담당하는 소프트웨어가 프로그램을 기계어로 번역한다.

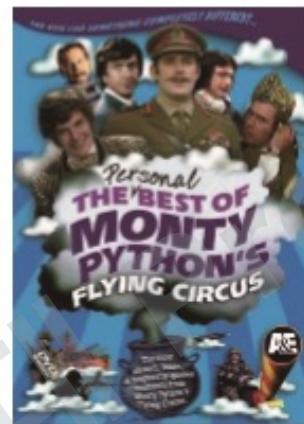


# 파이썬

- 1991년에 귀도 반 로섬(Guido van Rossum)이 개발한 대화형 프로그래밍 언어



파이썬은 제가 좋아하는 영국  
코미디 프로 이름이었어요!



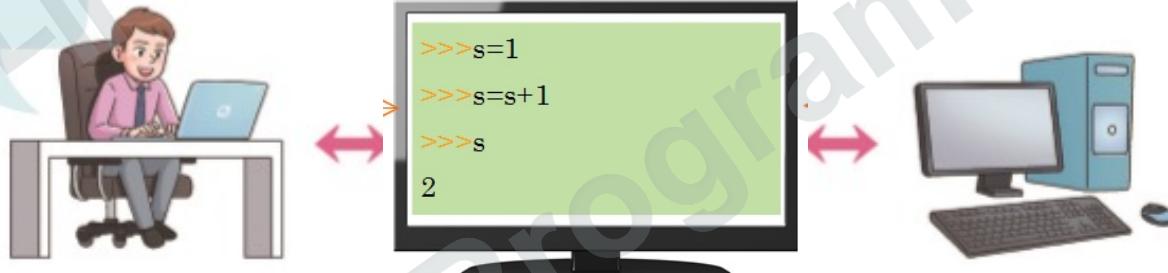
나하고는 관계가  
아니야! ^^



# 파이썬의 특징

- 생산성이 뛰어나다.
- 초보자한테 좋은 언어 - 인터프리터 언어

바로바로 결과를 볼 수 있다니!  
파이썬은 정말 대단해!



# 파이썬의 특징

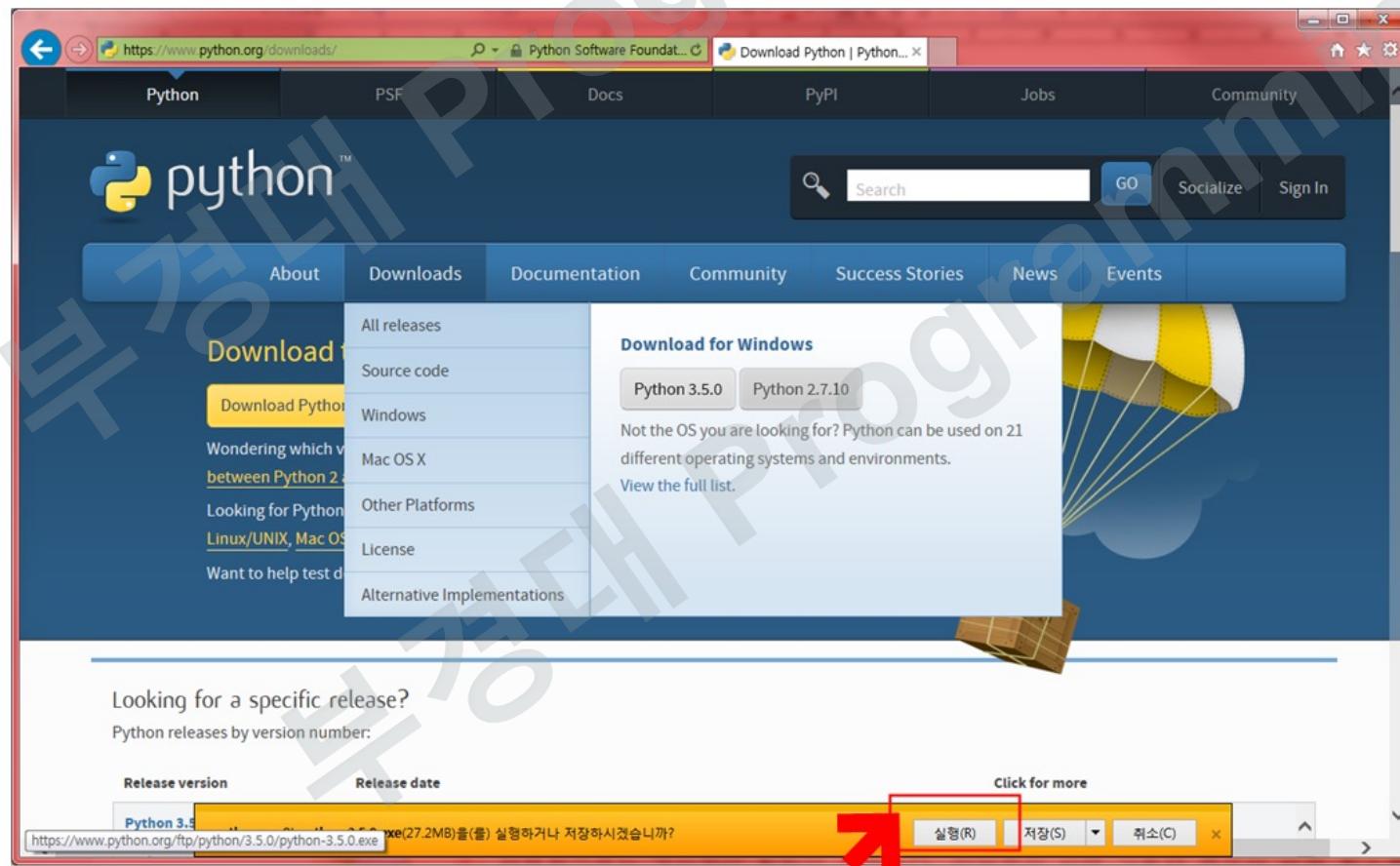
- 파이썬은 문법이 쉬워서 코드를 보면 직관적으로 알 수 있는 부분이 많다.

```
if "사과" in ["딸기", "바나나", "포도", "사과"]:  
    print("사과가 있습니다")
```

- 파이썬은 다양한 플랫폼에서 사용
- 라이브러리가 풍부
- 애니메이션이나 그래픽을 쉽게 사용

# 파이썬 설치하기

- 파이썬을 설치하려면 <http://www.python.org/>에 접속하여 Download 메뉴에서 "Python 3.7.2"를 선택한다.



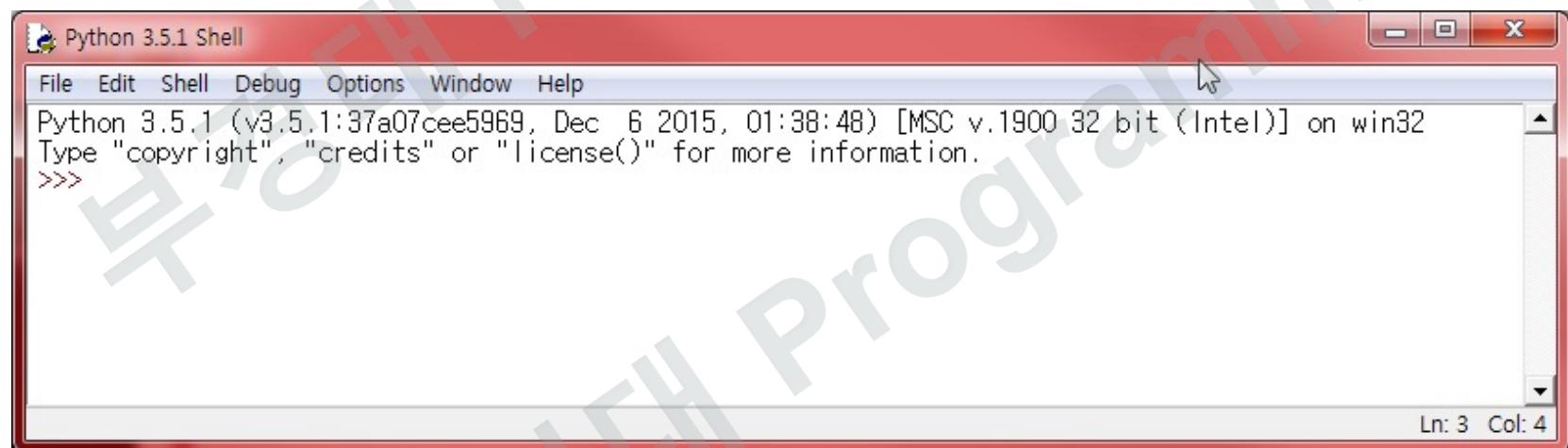
# 파이썬 설치하기

- 반드시 다음을 체크할 것!



# 파이썬 시작하기

- ① DOS 명령 프롬프트에서 "python"이라고 입력
- ② 윈도우의 시작 메뉴에서 "IDLE" 프로그램을 찾아서 실행

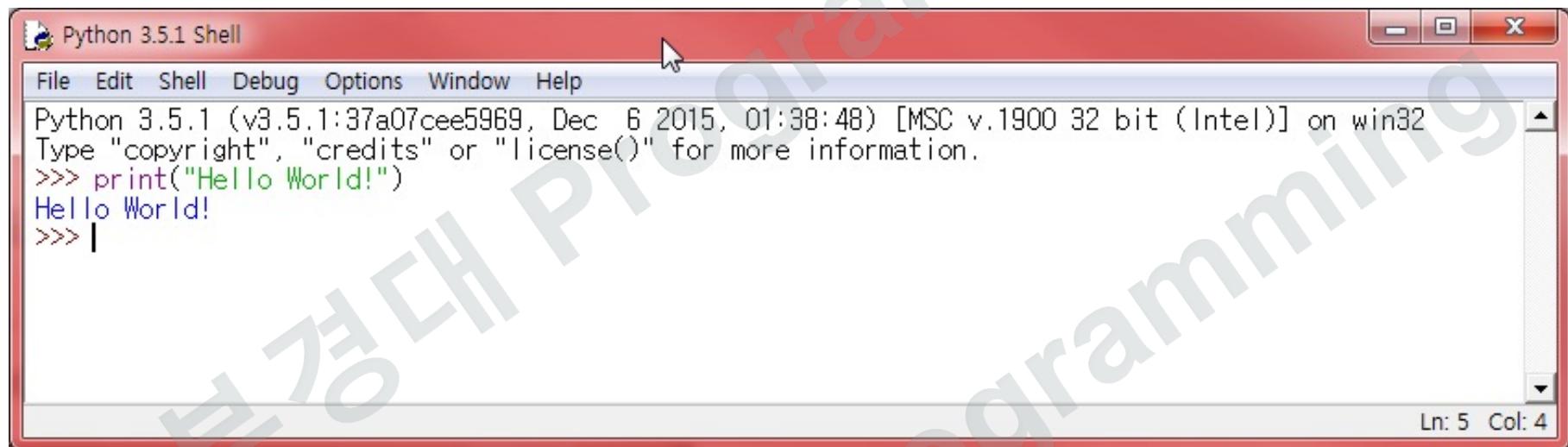


# 파이썬 쉘

- 파이썬 쉘에서는 >>> 뒤에 우리가 명령어를 입력하고 엔터키를 누르면 명령어가 실행되고 실행 결과가 화면에 출력된다.



# Hello World! 출력하기

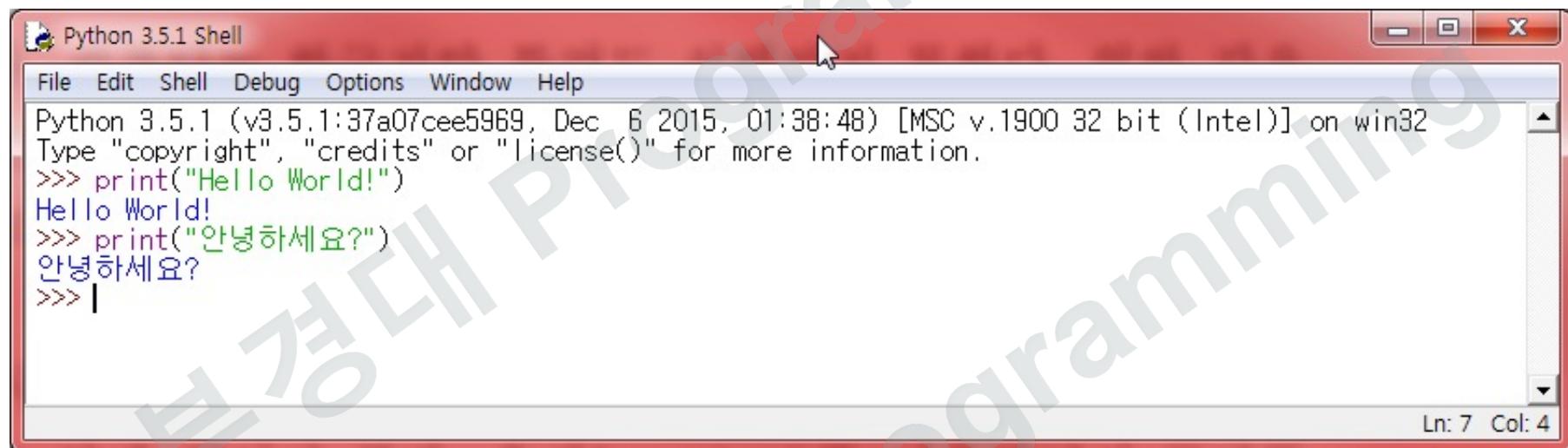


The screenshot shows the Python 3.5.1 Shell window. The title bar reads "Python 3.5.1 Shell". The menu bar includes File, Edit, Shell, Debug, Options, Window, and Help. The main window displays the following text:

```
Python 3.5.1 (v3.5.1:37a07cee5969, Dec 6 2015, 01:38:48) [MSC v.1900 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> print("Hello World!")
Hello World!
>>>
```

The status bar at the bottom right indicates "Ln: 5 Col: 4".

# 한글 출력

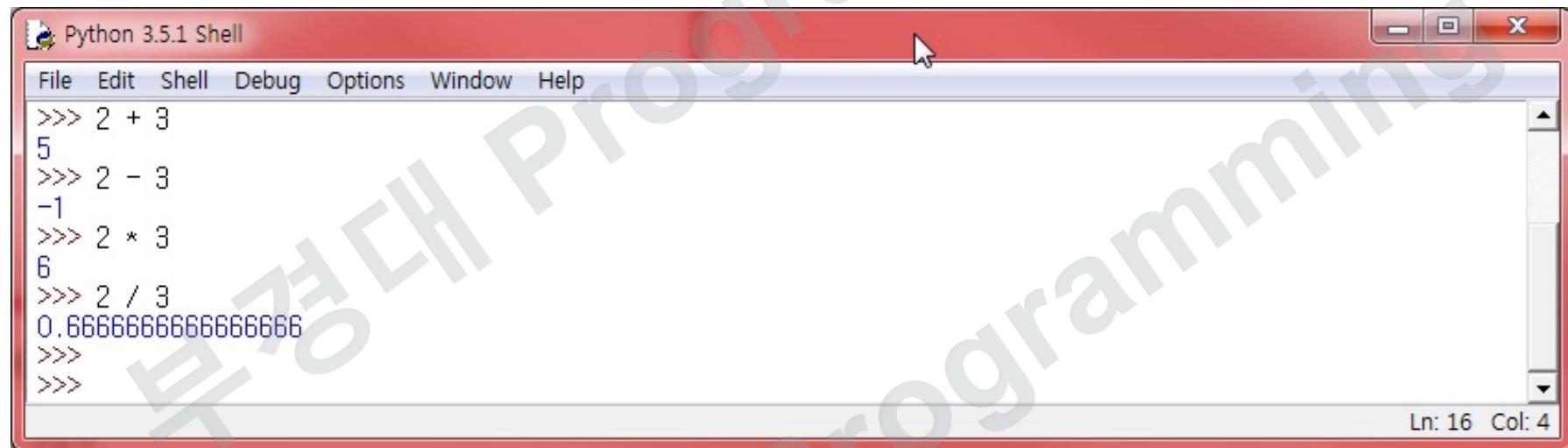


The image shows a screenshot of a Python 3.5.1 Shell window. The window title is "Python 3.5.1 Shell". The menu bar includes File, Edit, Shell, Debug, Options, Window, and Help. The main text area displays the following Python session:

```
Python 3.5.1 (v3.5.1:37a07cee5969, Dec  6 2015, 01:38:48) [MSC v.1900 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> print("Hello World!")
Hello World!
>>> print("안녕하세요?")
안녕하세요?
>>> |
```

The status bar at the bottom right indicates "Ln: 7 Col: 4".

# 사칙 계산



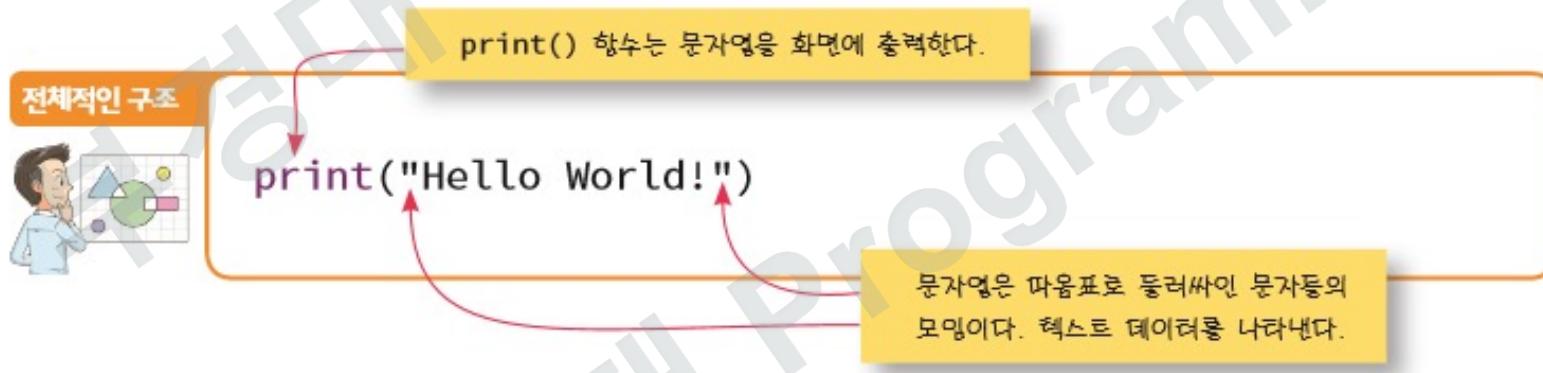
The screenshot shows a Windows-style window titled "Python 3.5.1 Shell". The menu bar includes File, Edit, Shell, Debug, Options, Window, and Help. The main window displays the following Python code and its output:

```
>>> 2 + 3
5
>>> 2 - 3
-1
>>> 2 * 3
6
>>> 2 / 3
0.6666666666666666
>>>
>>>
```

The status bar at the bottom right indicates "Ln: 16 Col: 4".

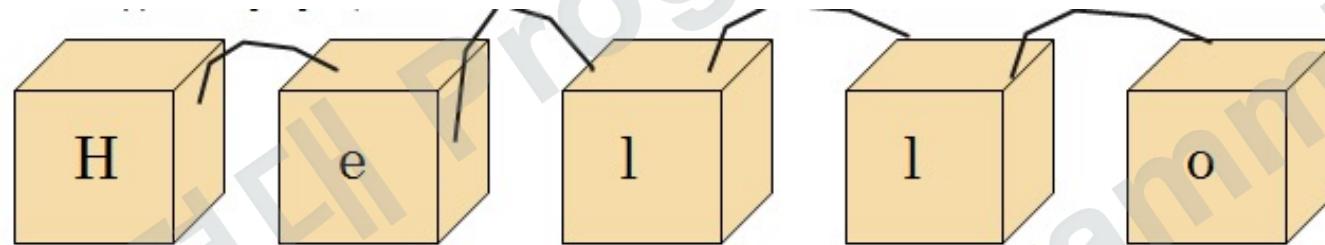
# 첫 번째 프로그램 분석하기

- 파이썬 프로그램은 여러 줄의 명령어로 이루어진다. 한 줄의 명령어를 문장(statement)이라고 부른다.
- 문장들은 파이썬 인터프리터에 의하여 순차적으로 실행된다.



# 문자열

- 문자열(string) : 큰따옴표("...")나 작은따옴표('...') 안에 들어 있는 텍스트 데이터



- 반드시 따옴표가 있어야 한다.

```
>>> print(Hello World!)
      SyntaxError: invalid syntax
```

# print() 함수

- 여러 개의 값들을 화면에 차례대로 출력할 수 있다.

```
>>> print("결과값은", 2*7, "입니다.")  
결과값은 14 입니다.
```

# 스크립트 모드

- 코드가 복잡해지면 인터프리트 모드는 번거롭다.

```
def add5(x):
    return x+5

def dotwrite(ast):
    nodename = getNodeName()
    label=symbol.sym_name.get(int(ast[0]),ast[0])
    print '%s [%s]' % (nodename, label),
    if isinstance(ast[1], str):
        if ast[1].strip():
            print '= %s';' % ast[1]
        else:
            print ''
    else:
        print ']';
    children = []
    for n, child in enumerate(ast[1:]):
        children.append(dotwrite(child))
    print '%s -> {' % nodename,
    for name in children:
        print '%s' % name,
```

이렇게 한 줄씩 입력하다간  
손가락이 남아나질 않겠어!



# 소스 파일 작성하기

- 텍스트 에디터를 이용하여 명령어들을 파일에 저장한 후에 파일을 읽어서 명령어들을 하나씩 실행하는 방법이 있다. 명령어들이 저장된 파일을 소스 파일(source file)이라고 한다.



# 온도 변환 프로그램

- 화씨온도->섭씨온도

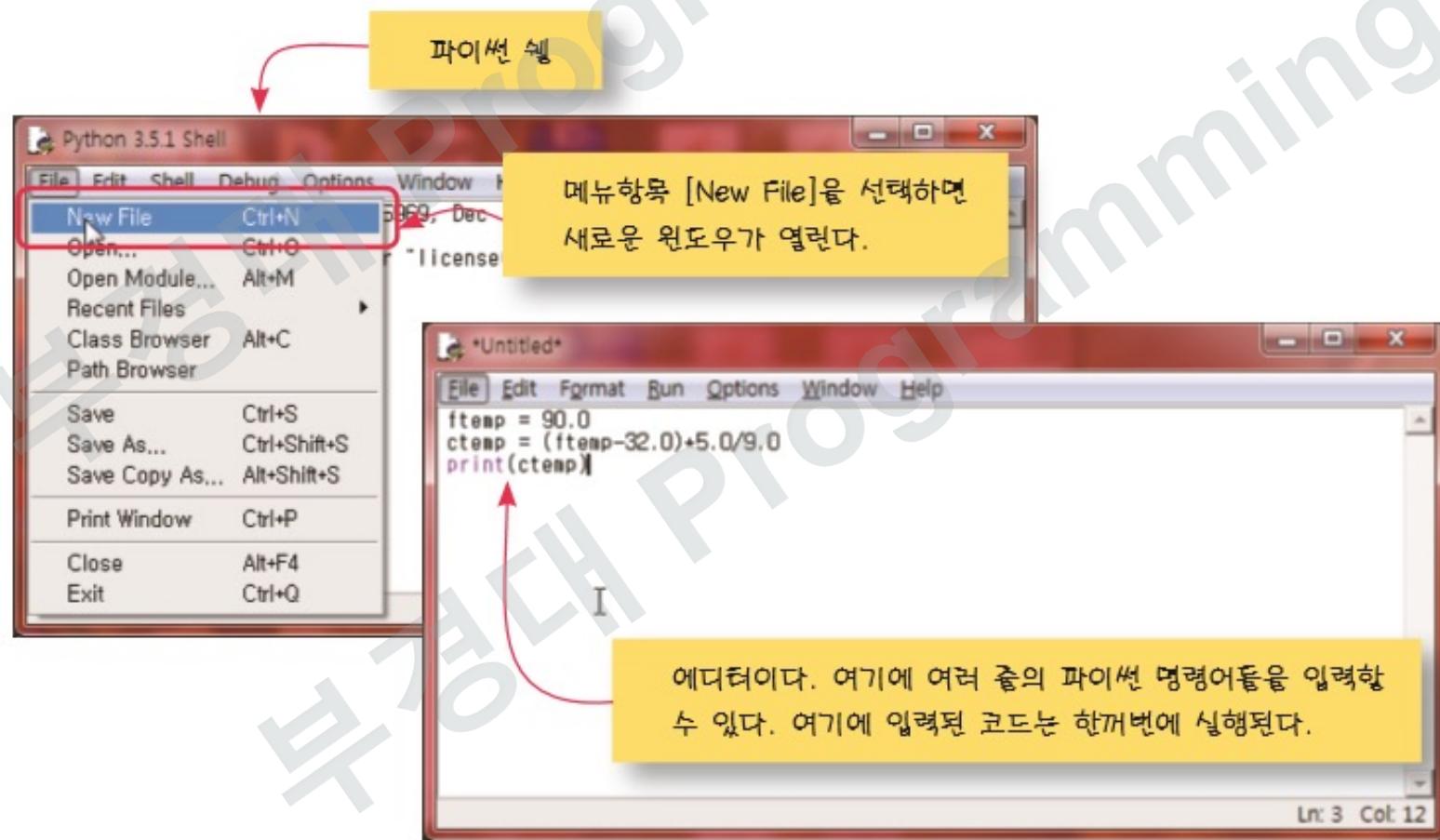
```
ftemp = 90.0  
ctemp = (ftemp-32.0)*5.0/9.0  
print(ctemp)
```



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

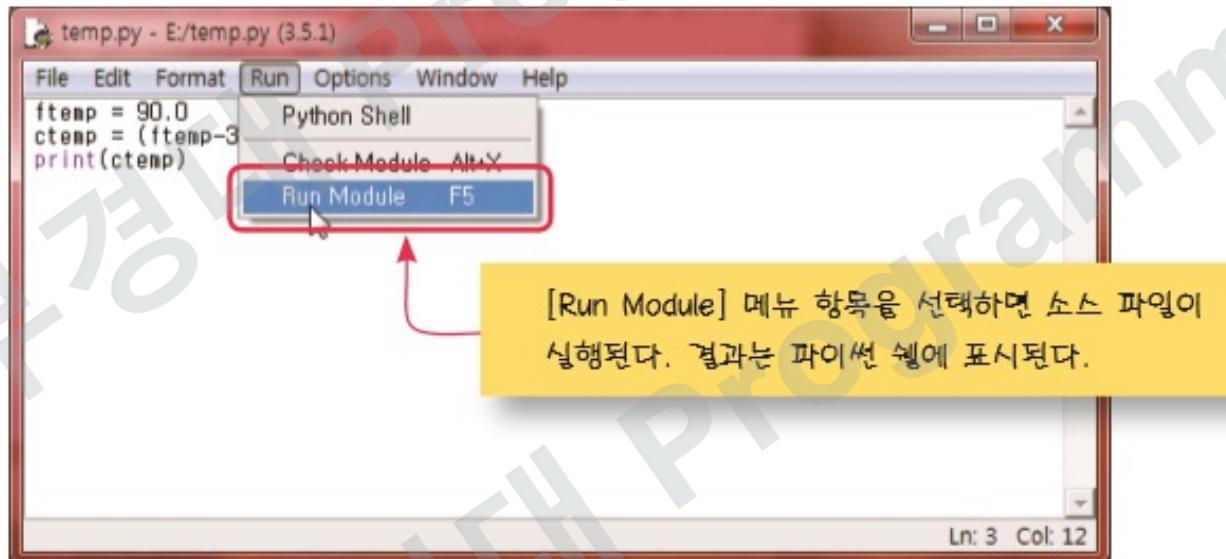
# IDLE를 이용한 소스 파일 작성

- 파이썬 셸의 메뉴 중에서 [File] -> [New File]을 선택한다.

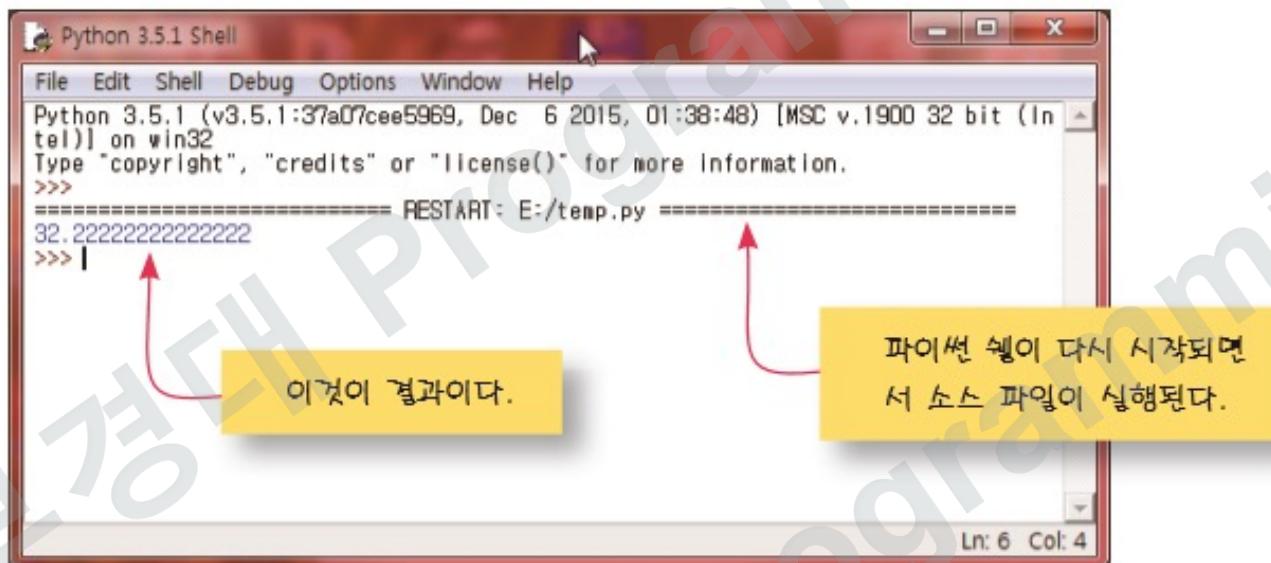


# IDLE를 이용한 소스 파일 실행

- 파이썬 셸의 메뉴 중에서 메뉴 [Run]->[Run Module]을 선택한다.



# 실행 결과



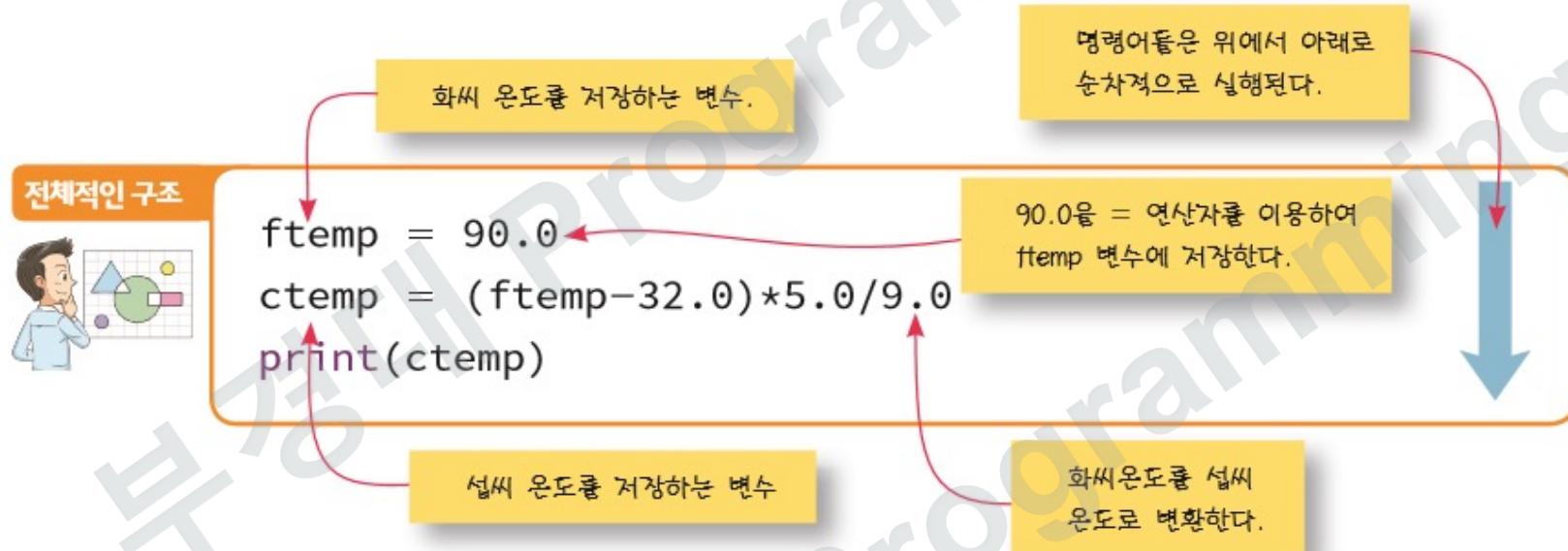
The screenshot shows the Python 3.5.1 Shell window. The command `>>> 32/1000` was entered, resulting in the output `0.032`. A yellow callout box points to the output with the text "이것이 결과이다." (This is the result). Another yellow callout box points to the command line with the text "파이썬 셸이 다시 시작되면 서 소스 파일이 실행된다." (When the Python shell starts again, the source file is executed).

```
Python 3.5.1 (v3.5.1:37a07cee5969, Dec  6 2015, 01:38:48) [MSC v.1900 32 bit (In tel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
=====
RESTART: E:/temp.py =====
0.032
>>> |
```

이것이 결과이다.

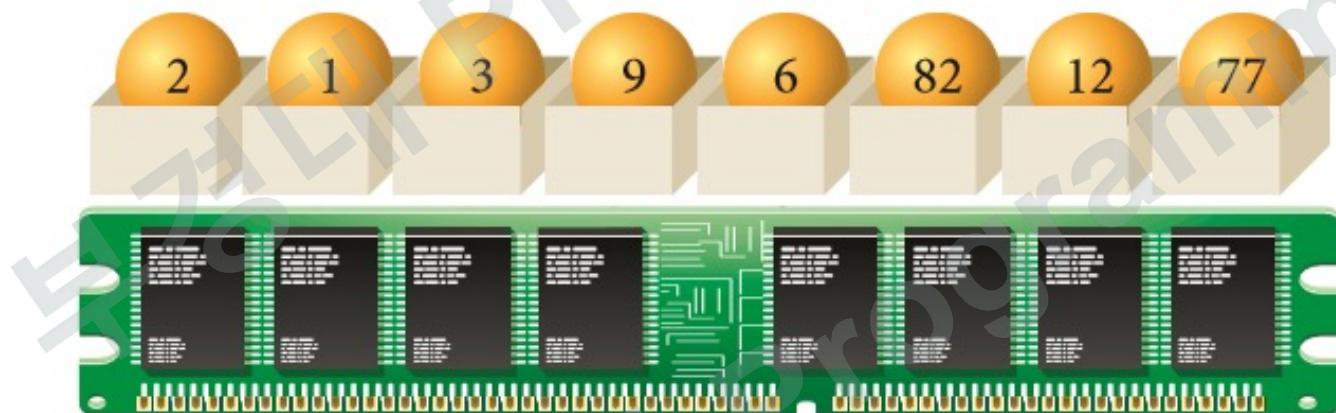
파이썬 셸이 다시 시작되면  
서 소스 파일이 실행된다.

# 프로그램의 간단한 분석



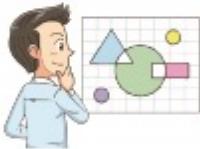
# 변수

- 변수는 컴퓨터의 메모리 안에 만들어지는 공간으로 우리는 여기에 숫자나 문자를 저장할 수 있다.



# 변수에 값을 저장할 때

전체적인 구조



왼쪽에는 변수를 적어준다.

`ftemp = 90.0`

오른쪽에는 변수에 저장할 값을 적어준다.

# Lab: print() 함수 실습

- 다음과 같이 출력하는 소스를 작성해보자.

```
안녕하세요? 여러분  
저는 파이썬을 무척 좋아합니다.  
9*8은 72 입니다.  
안녕히 계세요.
```

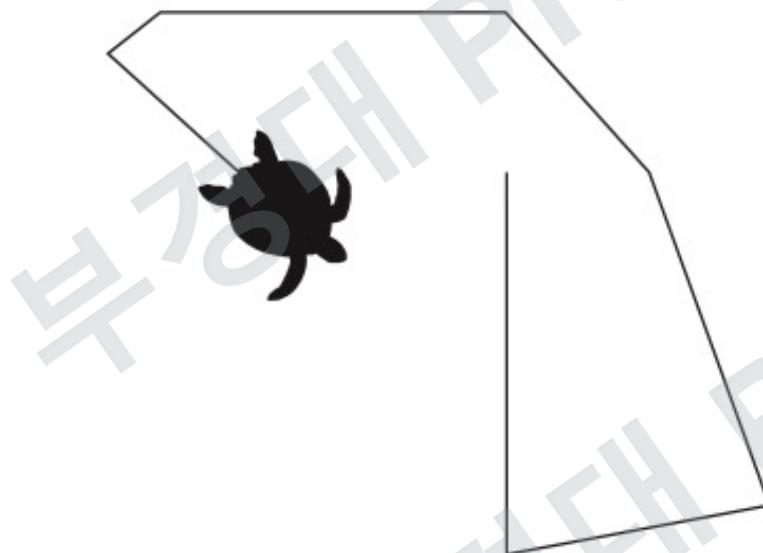


# Solution

```
print("안녕하세요? 여러분")
print("저는 파이썬을 무척 좋아합니다.")
print("9*8은", 9*8, "입니다.")
print("안녕히 계세요.")
```

# 터틀 그래픽

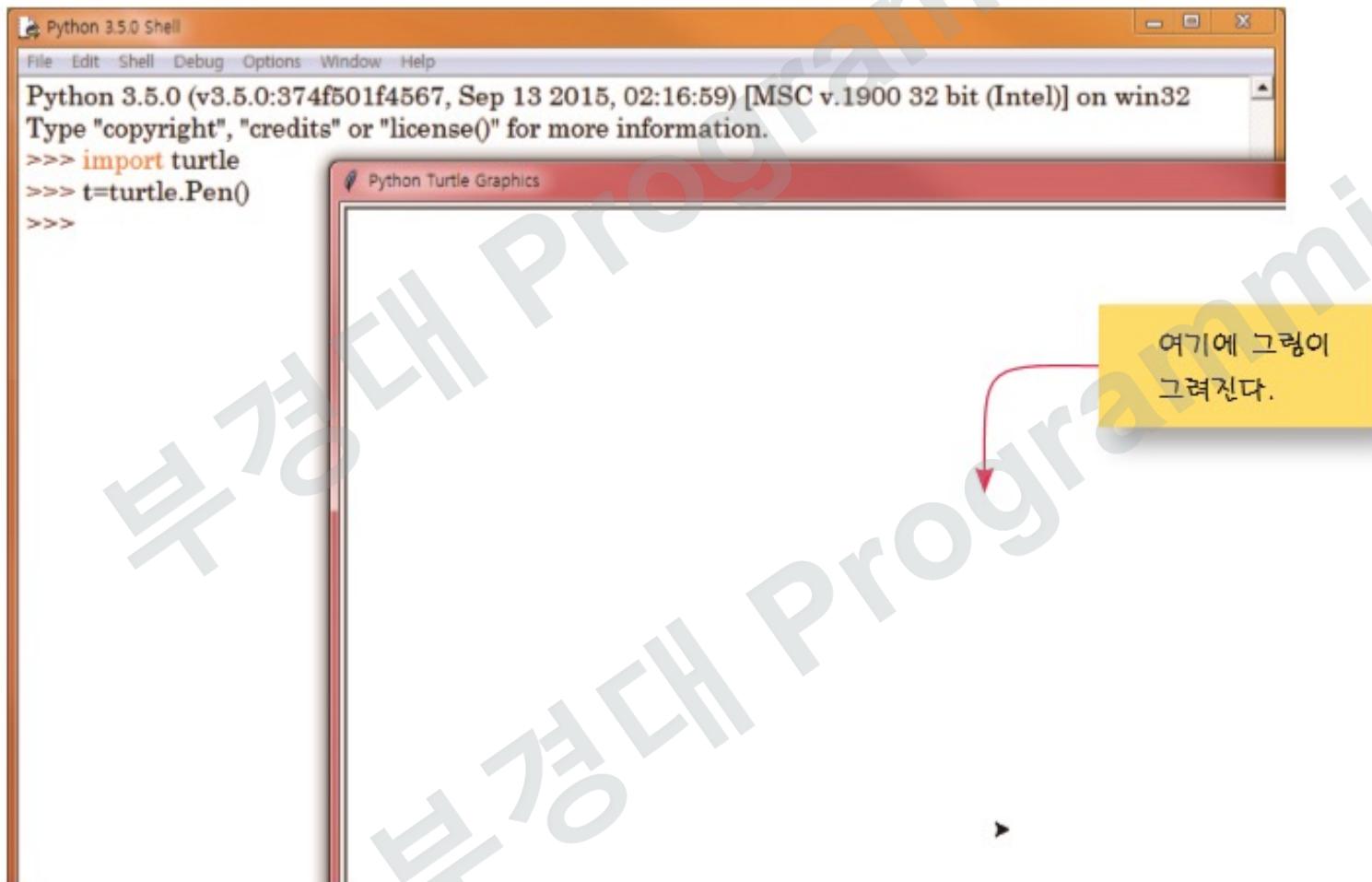
- 카테시안 공간에서 커서(터틀)를 이용하여서 그림을 그리는 기능



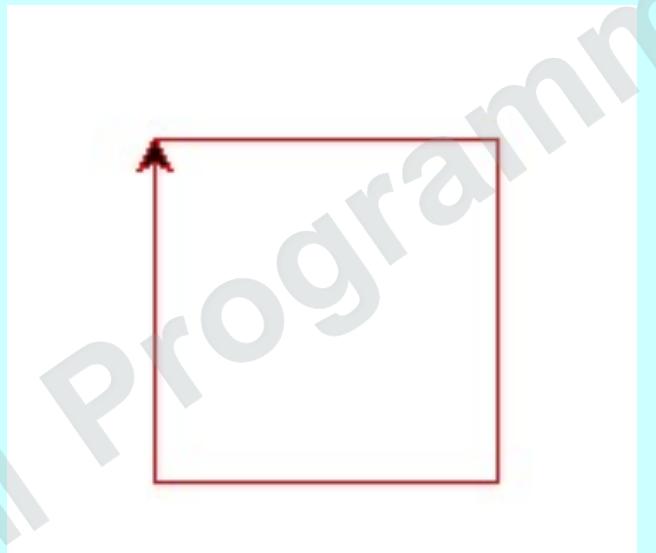
터틀 그래픽은 거북이가 화면 위에서 지나가면서 그림을 그리는 것이라고 생각하세요.



# 터틀 그래픽 윈도우

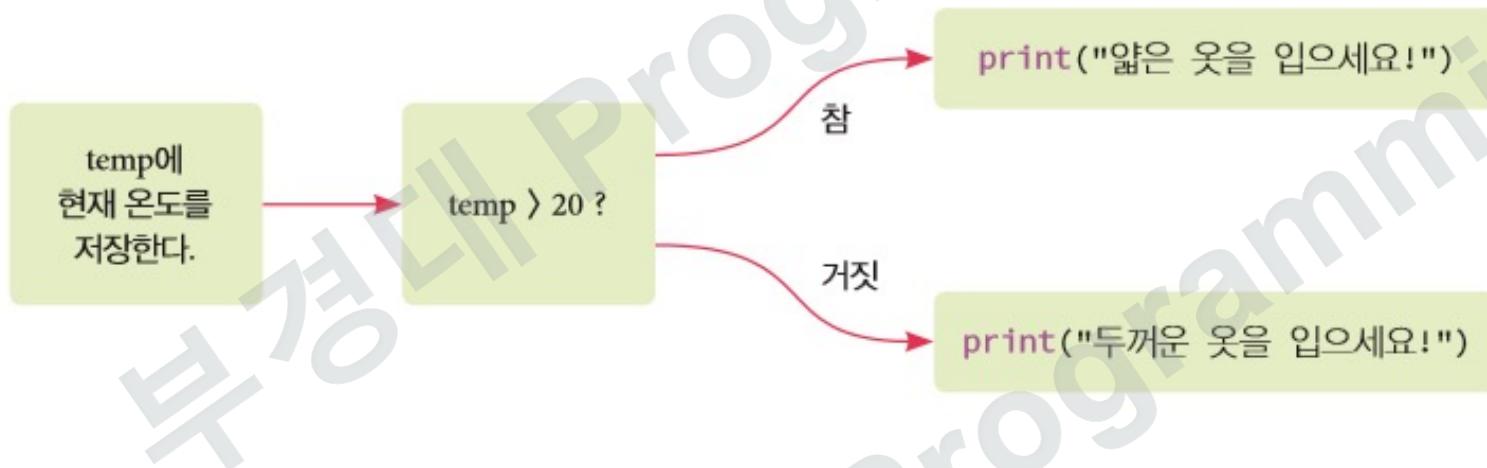


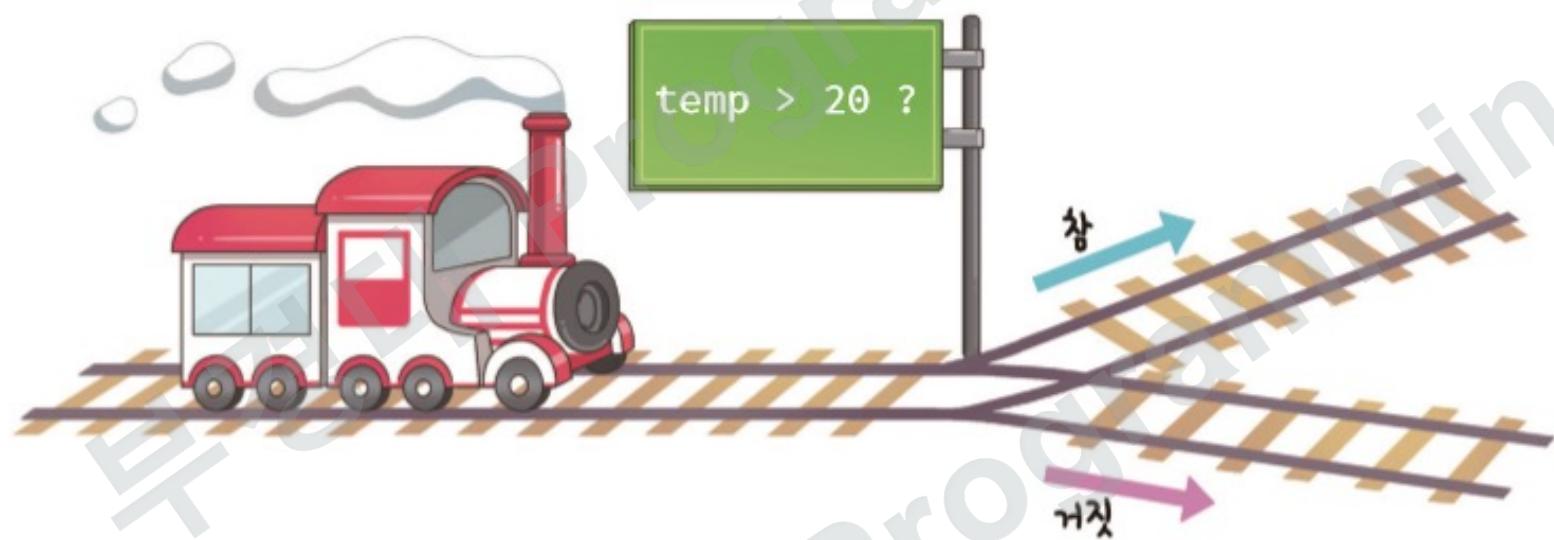
```
import turtle
t=turtle.Pen()
t.pencolor("red")
t.forward(100)
t.right(90)
t.forward(100)
t.right(90)
t.forward(100)
t.right(90)
t.forward(100)
```



# 조건문

- 예를 들면 날씨에 따라서 옷을 선택해주는 프로그램



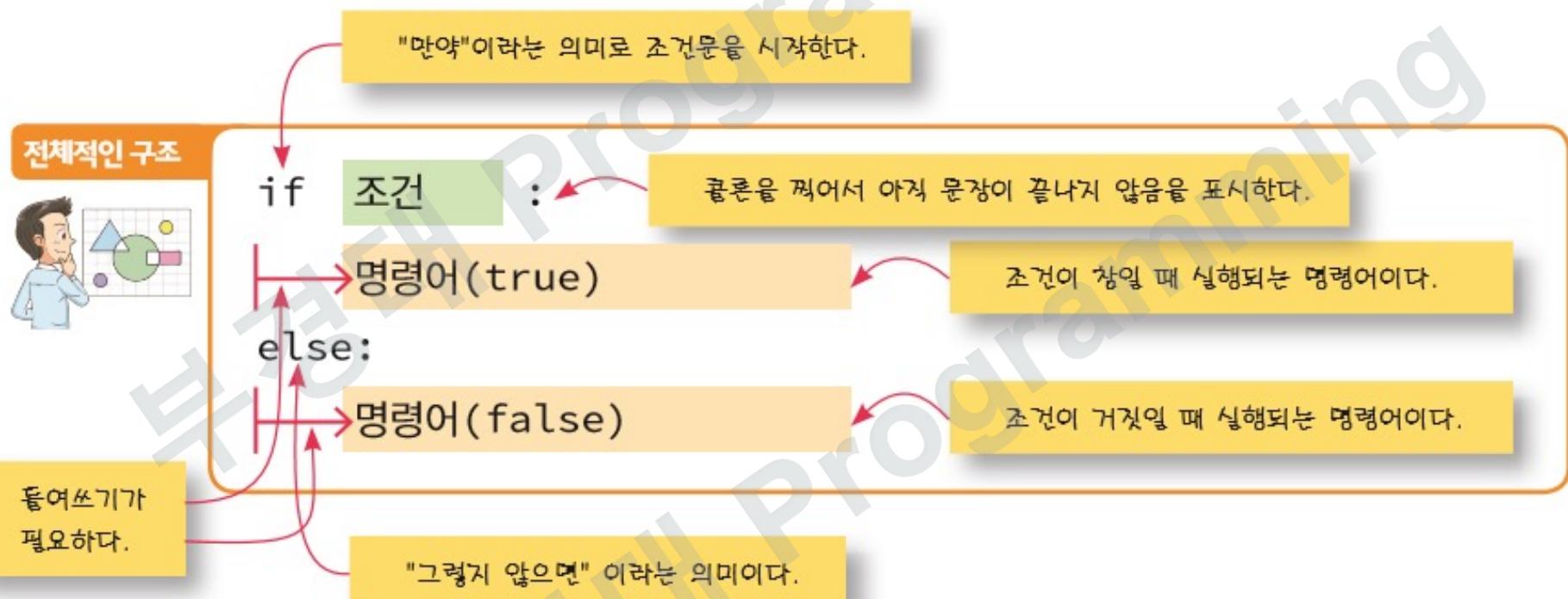


temp > 20 ?

창

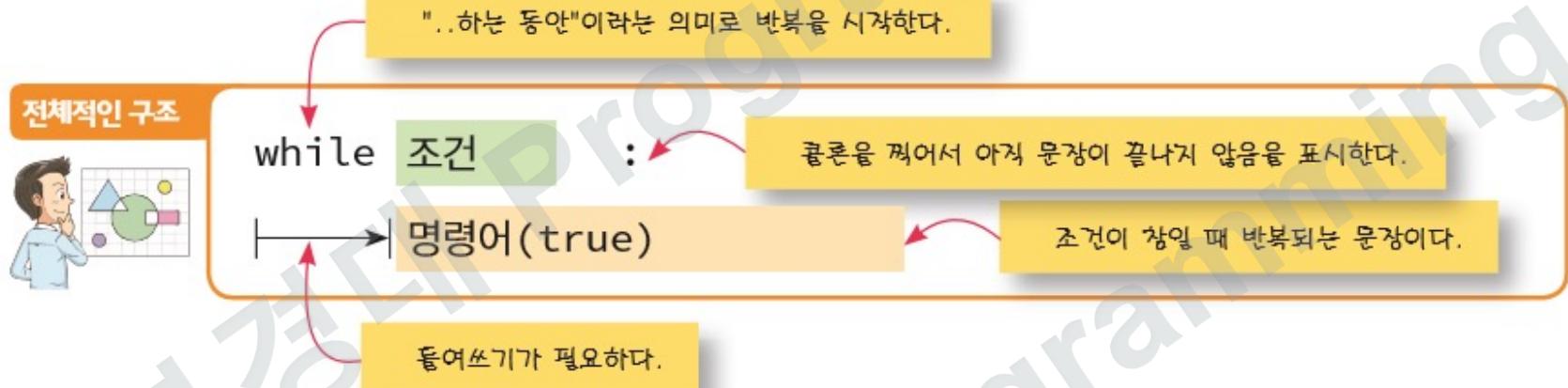
거짓

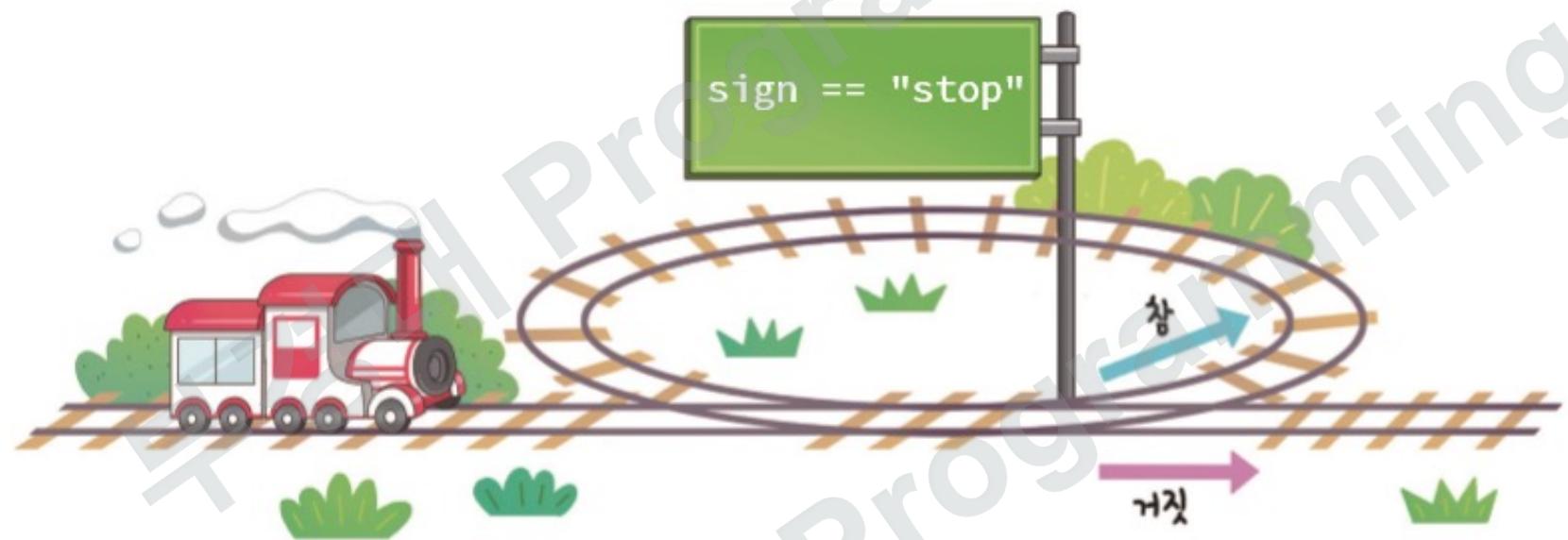
# 조건문의 구조



```
temp = 10
if temp > 20 :
    print("얇은 옷을 입으세요!")
else:
    print("두꺼운 옷을 입으세요!")
```

# 반복문





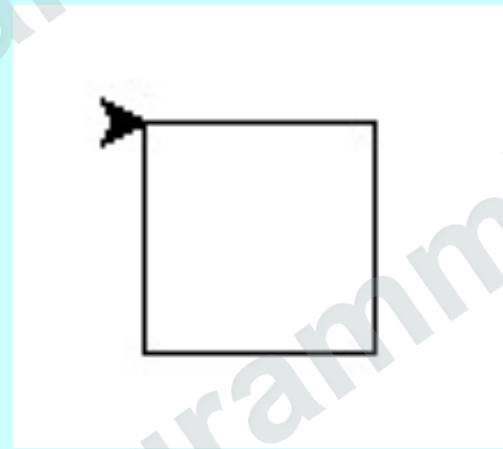
```
sign = "stop"
```

```
while sign == "stop":  
    sign = input("현재 신호를 입력하시오: ")  
print("OK! 진행합니다.")
```

```
현재 신호를 입력하시오: stop  
현재 신호를 입력하시오: go  
OK! 진행합니다.
```

# 반복문을 사용하는 터틀 그래픽 #1

```
import turtle  
  
t = turtle.Turtle()  
  
for i in [ 0, 1, 2, 3 ]:  
    t.forward(50)  
    t.right(90)
```

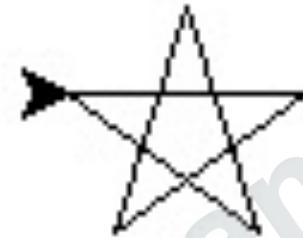


# 반복문을 사용하는 터틀 그래픽 #2

```
import turtle
```

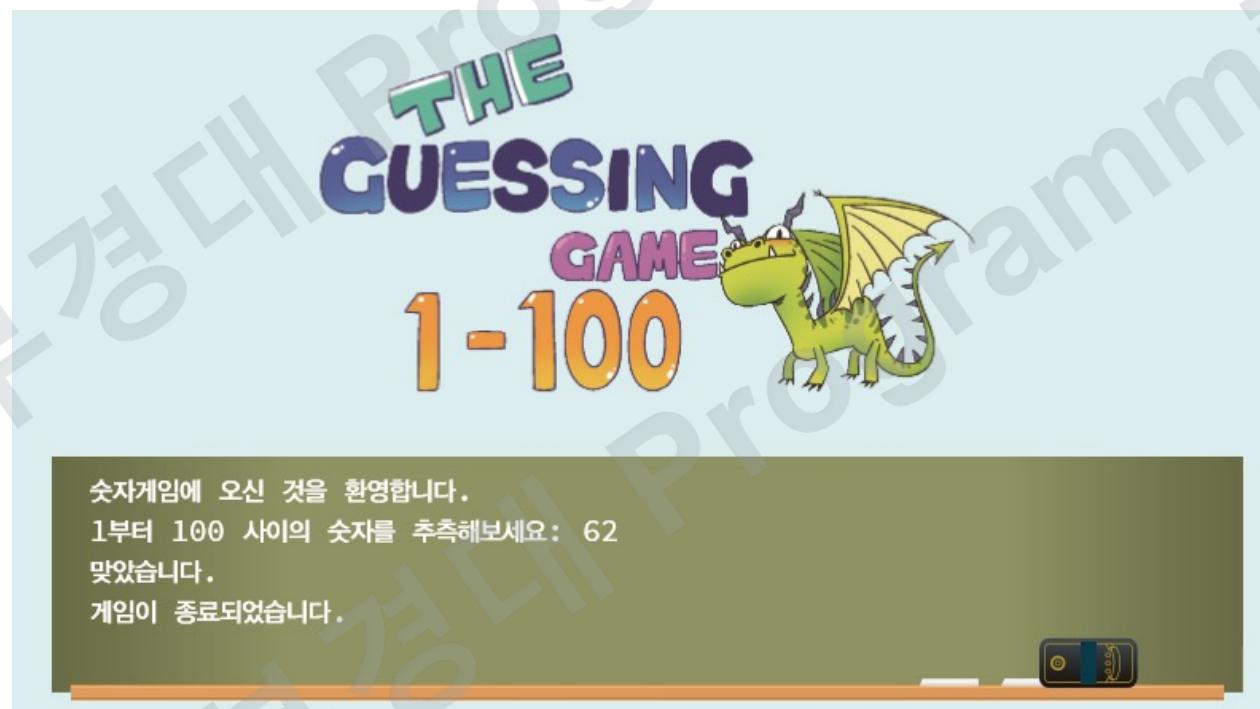
```
t = turtle.Turtle()
```

```
for i in [ 0, 1, 2, 3, 4 ]:  
    t.forward(50)  
    t.right(144)
```



# Lab: 숫자 추측 게임

- 우리가 학습한 조건문을 사용하여 간단한 숫자 맞추기 게임을 작성해보자.

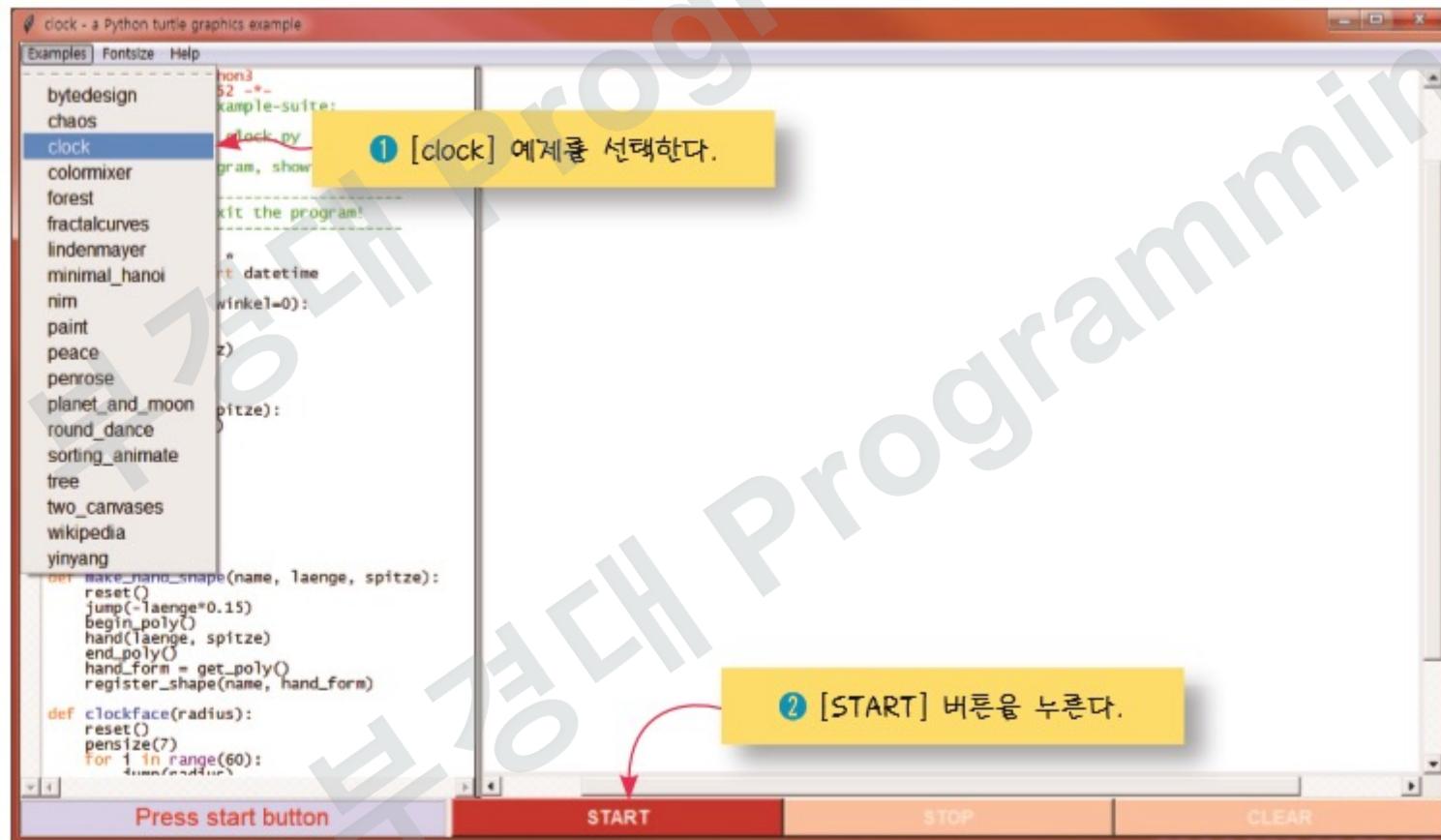


# Solution

```
print("숫자게임에 오신 것을 환영합니다.")  
number = 62  
s = input("1부터 100 사이의 숫자를 추측해보세요: ")  
guess = int(s)  
if guess == number:  
    print("맞았습니다.")  
else:  
    print("틀렸습니다.")  
print("게임이 종료되었습니다.")
```

# 파이썬과 놀아보자.

- 파이썬 셸에서 [Help] ->[Turtle Demo]를 선택한다.



# 헥심 정리

- 프로그램은 명령어들로 이루어진 텍스트 파일 형태로 작성된다. 이것을 소스파일이라고 한다.
- 파이썬 인터프리터는 소스 파일을 해석하여서 컴퓨터가 이해할 수 있는 기계어 파일로 변환하여 생성한다.
- 문장들은 기본적으로 순차적으로 실행되지만 조건에 따라서 서로 다른 경로로 실행되거나 반복될 수 있다.

# Q & A

