

Introduction to Python

By 박수현

Computer와 일상생활

- 일상생활에서 computer를 사용
- Computer의 최대 장점: 반복적인 작업



범용성 (general purpose)

- Computer의 핵심: 범용성

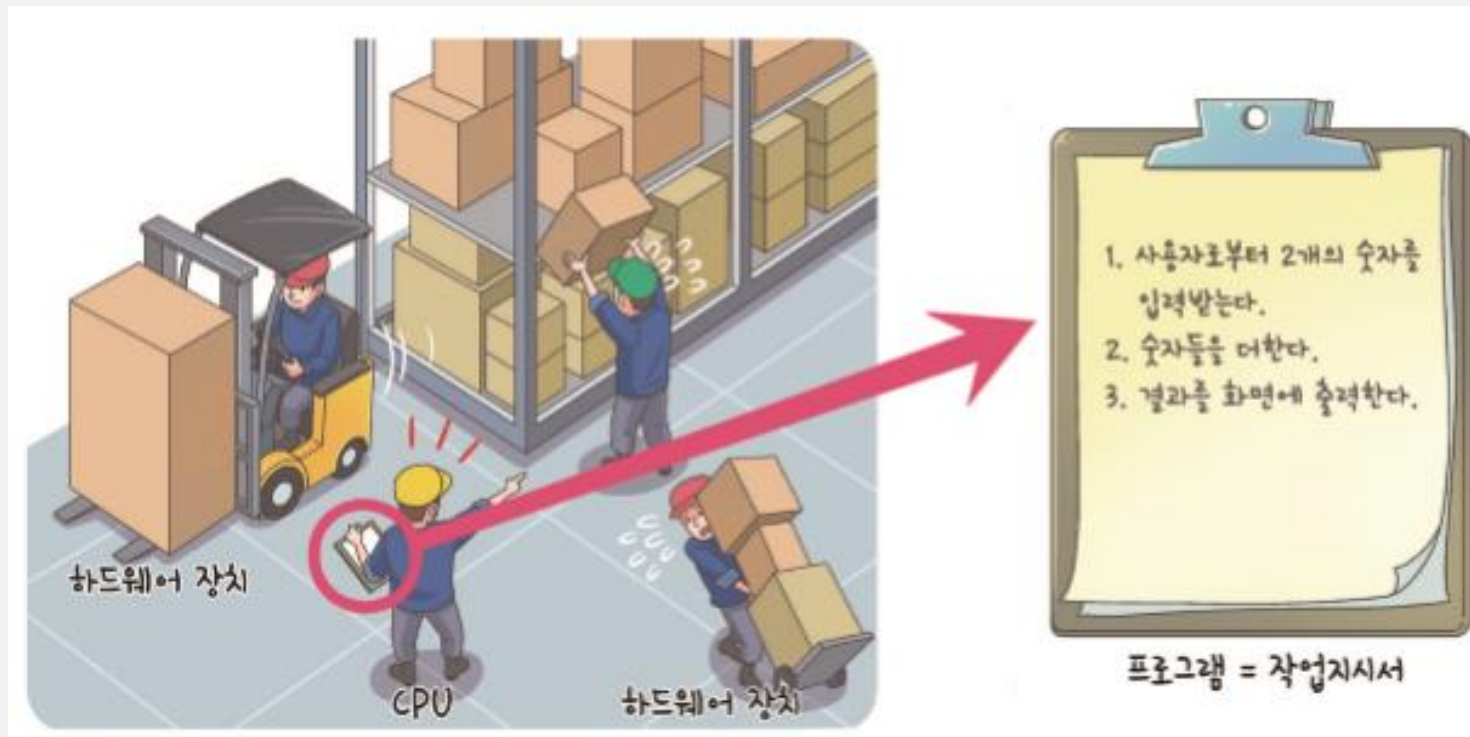
예) Smart device (phone)

- 다양한 기능을 하는 App(program)을 설치하여 여러가지 작업을 할 수 있음



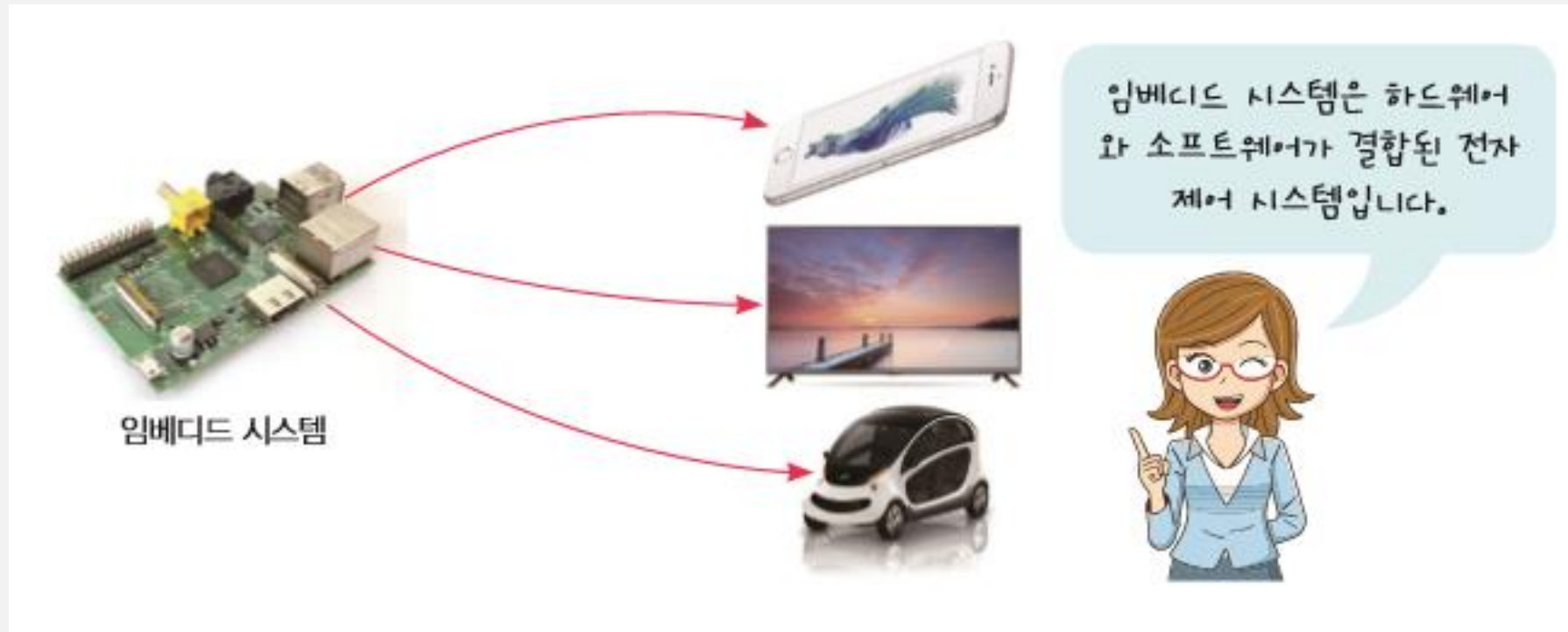
Computer program

- Computer에 일을 시키려면 인간이 computer에게 자세한 명령어 (instruction)들의 list를 주어야 함
- Program : computer가 수행할 명령어를 적어놓은 문서



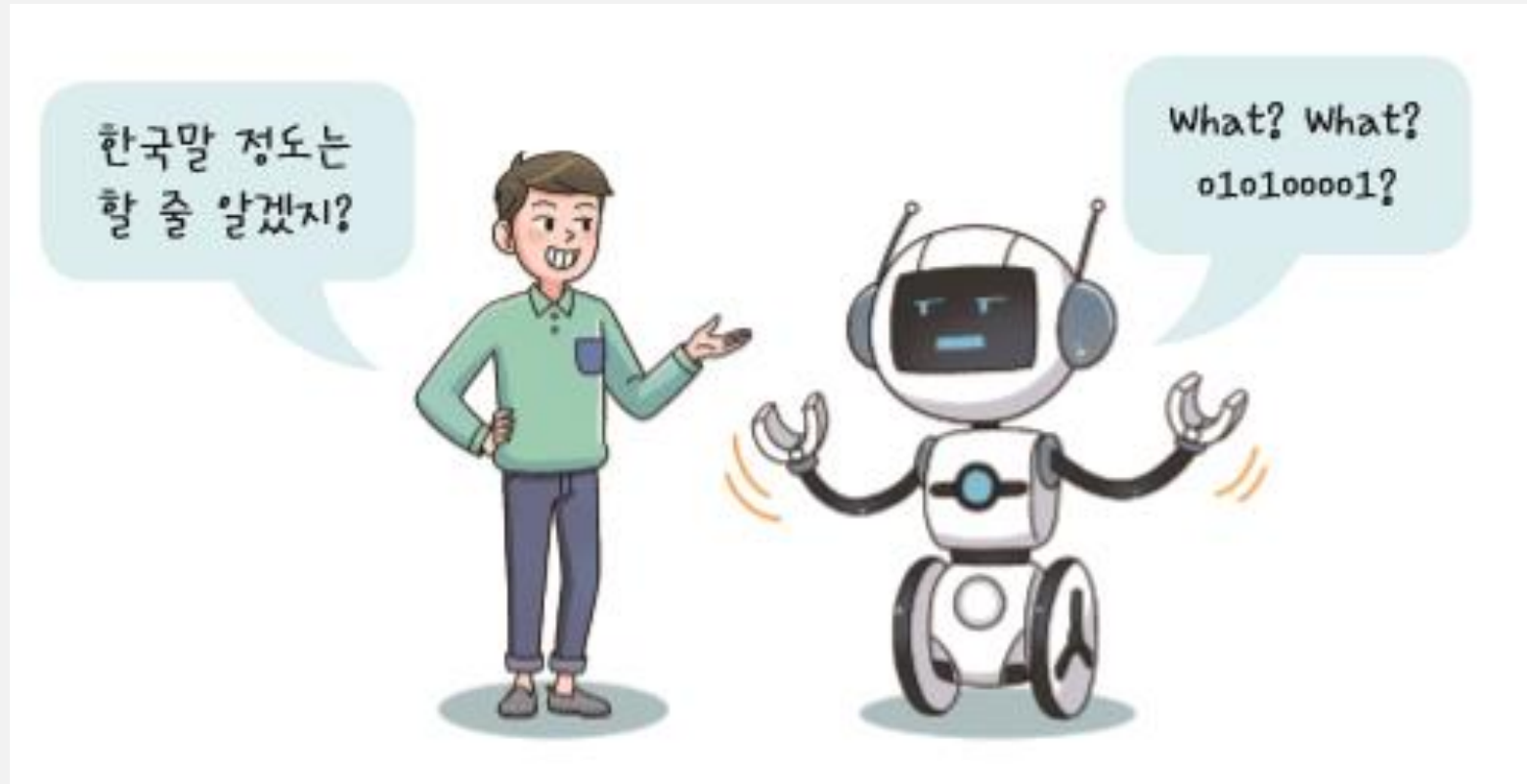
Embedded program

- Program은 computer에만 설치되는 것이 아님
- Embedded program : 전자기기에 내장되는 program



Programming language(프로그래밍 언어)

- Computer가 이해할 수 있는 언어
- Computer는 사람의 언어(natural language)를 이해할 수 없음



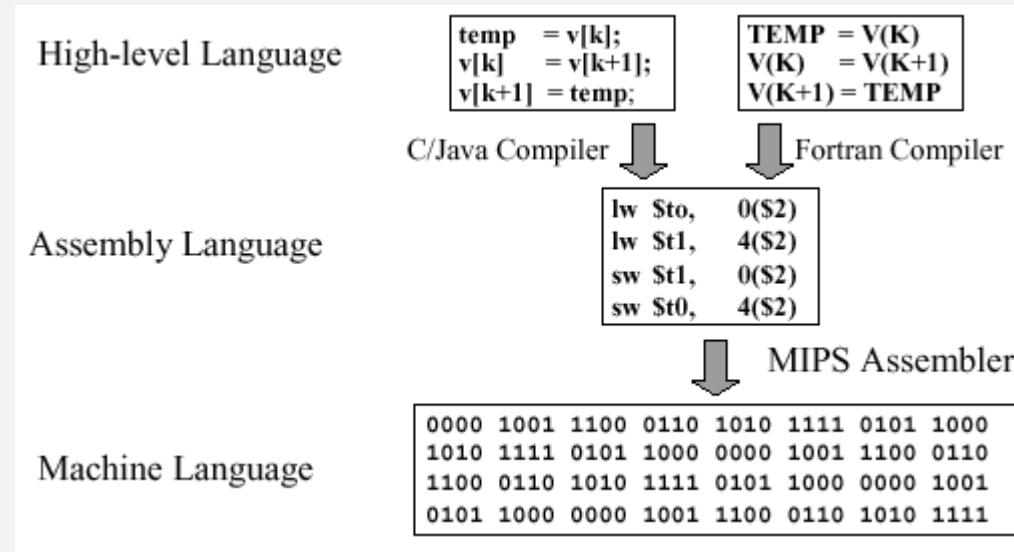
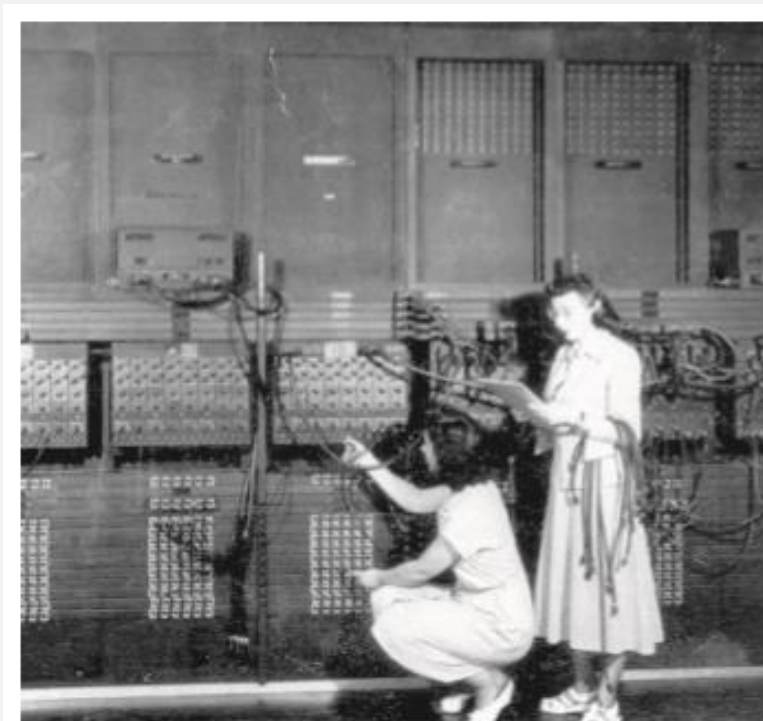
Programming language(프로그래밍 언어)

- Machine language (기계어)
- High-level language (HLL)

Machine language(기계어)

- Machine language : computer가 알아듣는 유일한 language
- Machine language는 0과 1로 구성됨
- 초기의 computer에서는 machine language 를 사용

Language가 필수적으로 가져야 할 부분
- grammar
- semantic



- Object-oriented
- Procedural

<https://www.cise.ufl.edu/~mssz/CompOrg/CDA-lang.html>

Machine Language -> 너무 휴먼 프렌들리 하지 않는다
따라서 만든게 Assembly Language
High-level -> 더 높은 프렌들리하다.

High-level language (HLL)

- 인간의 언어에 근접한 programming language
- 인간이 HLL로 program을 작성하면 컴파일러(compiler)라고 하는 통역(interpreting, compile)을 담당하는 software가 program을 machine language (machine code)로 번역

HLL : C, C++, Java

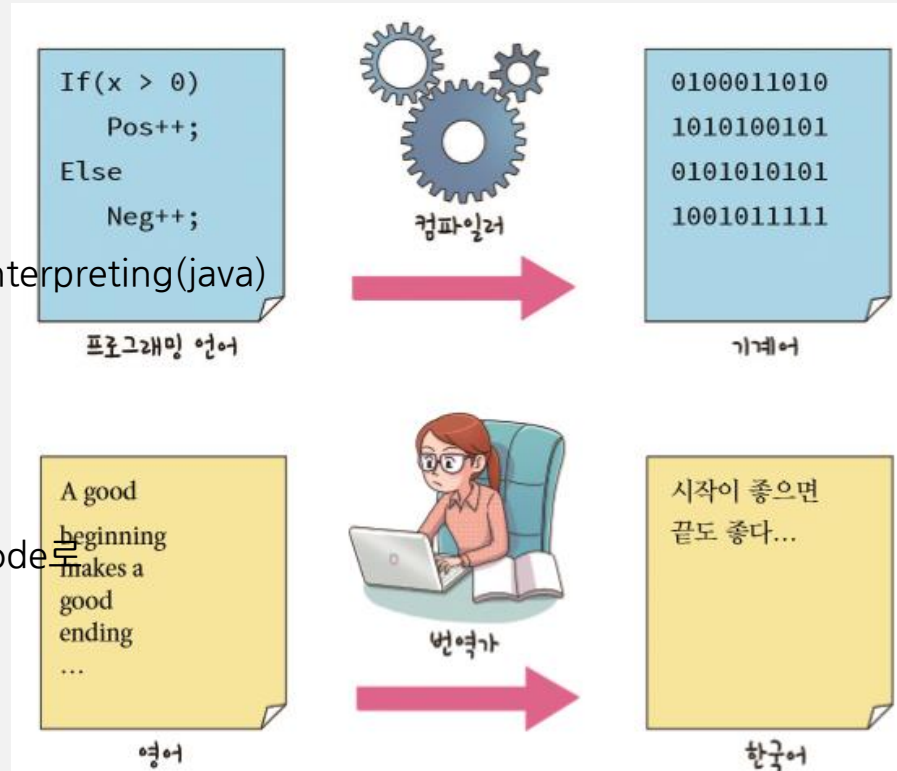


intermdiate Code



Compile(C), byte Code -> Interpreting(java)

Machine Code



C++ -> gcc -> Machine Code ~> exe(executive code)



C++는 dependent

OS(ex. Linux) -> Resource Scheduling

CPU(ex. ARM) -> platform

C가 Windows에서만 돌아가는 이유:
CPU(ARM, INTEL)별로 micro instruction이 존재
OS는 리소스 스케줄러, micro instruction을 관리

C가 너무 Platform이 따른다
그래서 HLL와 MACHine Code사이에 byte Code 만듬
이때 만들어진게 JDK

JDK의 Pros & Cous
장점 : 모든 Platform에서 작동 가능하다
단점 : 성능을 잃었다.

Python, Html, JS
compiler -> Interpreting- > execution code

Python

- 1991년 Guido van Rossum가 개발한 대화형 프로그래밍 언어 (interactive programming language)



파이썬은 제가 좋아하는 영국
코미디 프로 이름이었어요!

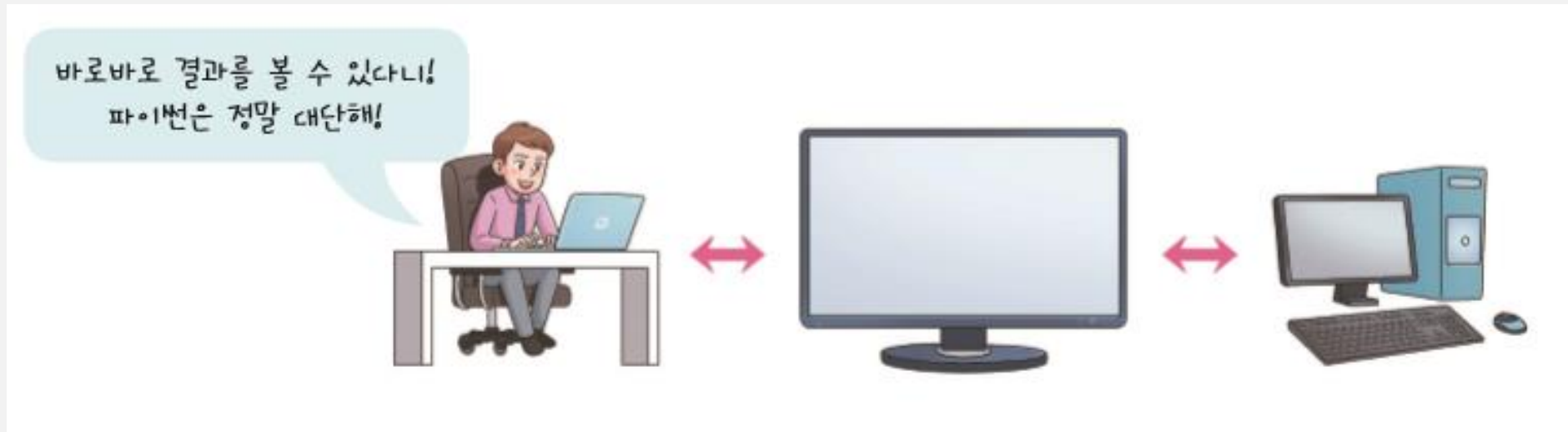


나하고는 관계가
없어!^^



Python의 특징

- 초보자한테 좋은 언어 – 인터프리터(interpreter) 언어
- 비교적 높은 생산성



Python의 특징

- Python은 문법(grammar)이 쉬워서 code를 보면 직관적으로 알 수 있는 부분이 많음

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

- Python은 다양한 platform에서 사용
- Library가 풍부
 - NumPy, Matplotlib, SciPy, Pandas(Python Data Analysis Library)..
 - Google TensorFlow, IBM Qiskit (Quantum Information Science Kit) 등
- Animation이나 graphic을 쉽게 사용

Python의 특징

Python의 특징

- interpreter이다.

- compiler code -> system dependent

- line by line

- platform neutral(independent)

- CPU, GPU, Memory

- library가 많다

- Python으로 할 수 있는 일

- 데이터 분석

- NumPy, Pandas, Matplotlib 등의 library를 활용해 data 처리, 통계 분석, 시각화를 손쉽게 수행
 - Pandas가 등장한 이래 data 분석에 R보다 Python을 사용하는 사례 증가



- Database Programming

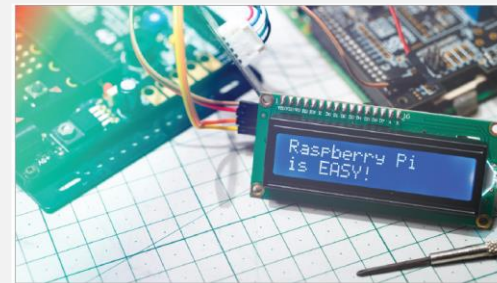
- Sybase, Infomix, Oracle, MySQL, PostgreSQL 등의 **Database**에 접근하기 위한 도구 제공
 - 자료를 변형 없이 그대로 파일에 저장하고 불러오는 Python module Pickle

- System Utility제작하기

- 운영체제(Windows, Linux 등)의 **System** 명령어를 사용하는 도구를 통한 **System Utility** 제작

Python의 특징

- Python으로 할 수 있는 일
 - **GUI(Graphic User Interface) programming**
 - 화면에 Window (창)을 만들고 program을 동작시킬 수 있는 menu나 button, 그림 등을 추가하는 것
 - GUI programming을 위한 도구들을 갖추고 있어 GUI program을 만들기 쉬움 (예 - Tkinter(티케이인터))
 - **C/C++와 결합하기** 네이티드 인터페이스
 - C나 C++로 만든 프로그램을 Python에서, Python으로 만든 program을 C나 C++에서 사용 가능
 - **사물 인터넷**
 - Raspberry-Pi를 제어하며 사물 인터넷 구현



Python의 특징

- Python으로 할 수 없는 일
 - **System과 밀접한 programming 영역** 네이티브 인터페이스
 - 운영체제, 엄청난 횟수의 반복과 연산이 필요한 program 등
매우 빠른 속도를 요구하거나 hardware를 직접 manipulating하는 program에는
적합하지 않음
 - **Mobile Programming**
 - 안드로이드 네이티브 앱(android native app) 개발에는 아직 역부족
 - 아이폰 앱 개발도 할 수 없음

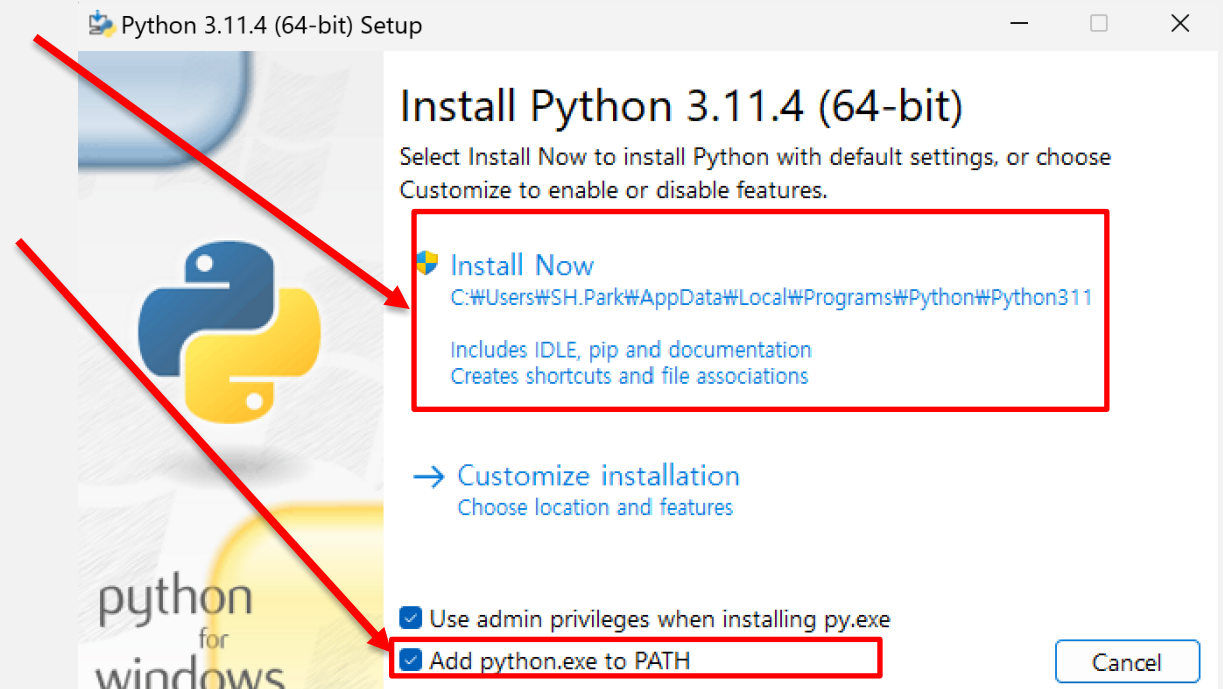
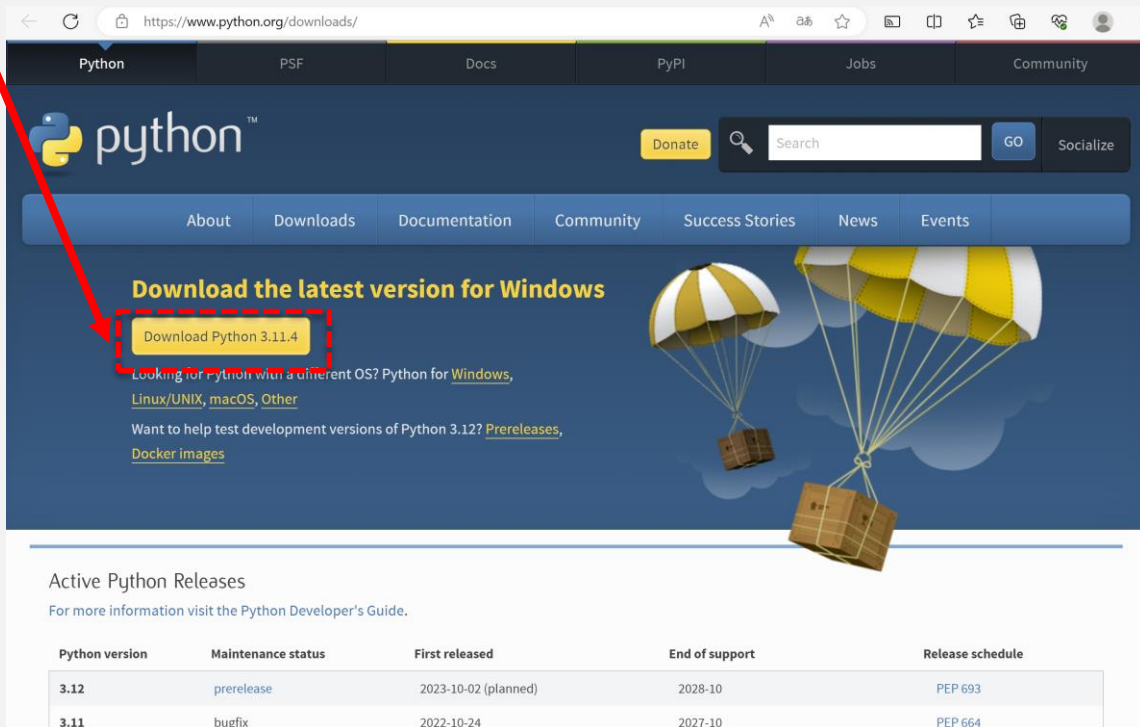
Python 설치하기

- 운영체제(Operating System) 별 설치 방법
 - Windows
 - Linux
 - macOS

Python 설치하기 - Windows

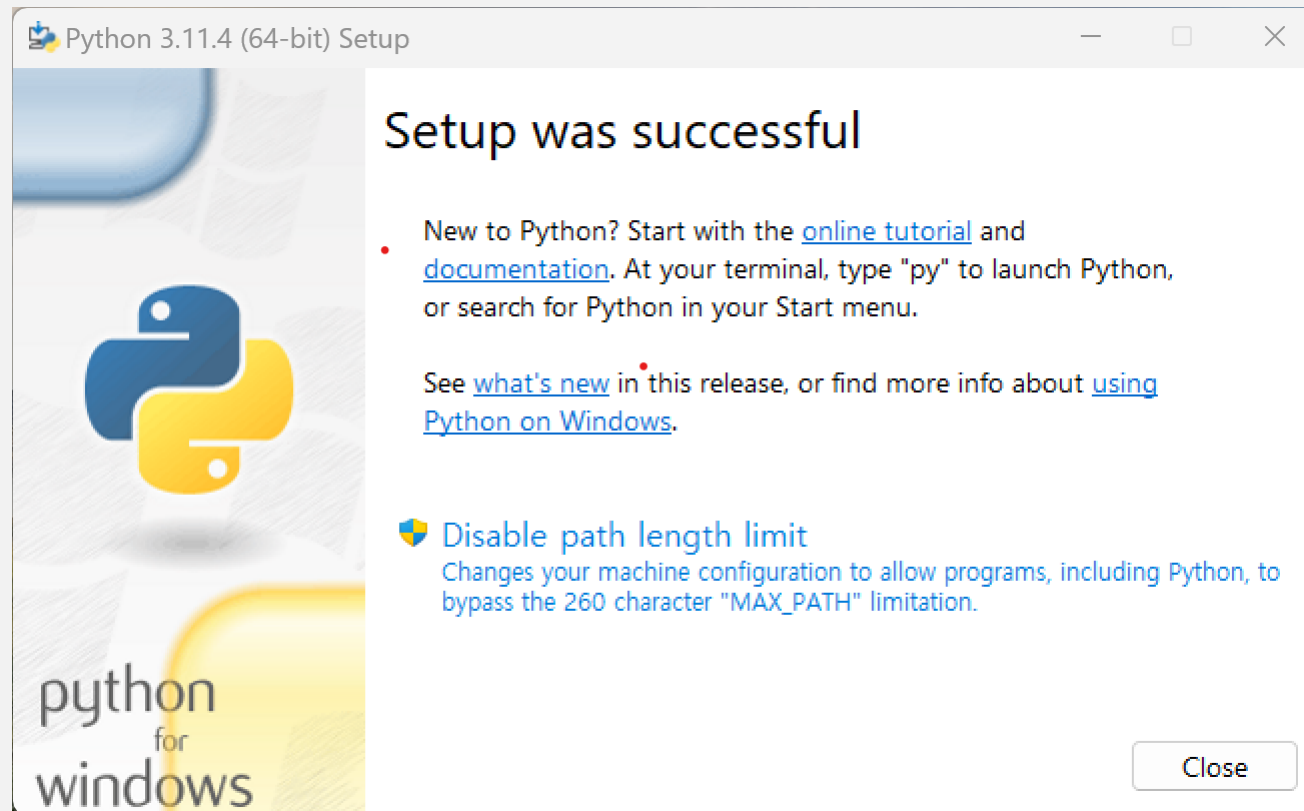
- Windows에서 Python 설치하기

1. Python 공식 홈페이지(www.python.org/downloads)에서 Python 3.x 최신 버전 다운로드
2. Python이 어느 곳에서든지 실행될 수 있도록 '**Add python.exe to PATH**' 옵션 선택



Python 설치하기 - Windows

- 필요에 따라 modify 실시



Python 설치하기 - Linux

- Linux distributions (shortly “distro”) (Linux 배포판)
 - Ubuntu
 - Ubuntu 기반
 - Zorin
 - Mint
 - Debian
 - Fedora
 - CentOS : Red Hat Enterprise Linux (RHEL) 기반
 - openSUSE / SUSE Linux Enterprise
 - Arch Linux
 - ...

Python 설치하기 - Linux

- Ubuntu Linux 최신 version에는 이미 Python 3가 install되어 있음

- Python 3.11 설치

- \$ sudo apt install python3.11

- # 설치가 잘 된 것을 확인하기 위해, 버전 확인 명령어를 실행

- \$ python3 --version

- # Python 실행

- \$ python3

- CentOS

- Python 2.x version이 install 되어있을 수도 있음

- Python 3를 install 하려면 다음의 명령을 입력

- \$ sudo yum install python

Python 설치하기 - Linux

- Linux에서 IDLE사용
 - Tkinter를 설치

- Ubuntu

```
$ sudo apt-get install python3-tk
```


- Centos

```
$ sudo yum -y epel-release
```

```
$ sudo yum install python3*-tkinter
```

Python 설치하기 - macOS

1. Python 공식 홈페이지(www.python.org)의 [Downloads] 메뉴에서 MAC용 설치 file download
2. 내려받은 **python-3.11.x-macos1.pkg** 파일을 실행하여 설치
3. Terminal에서 명령을 입력해 자신의 MAC에 설치된 Python version 확인



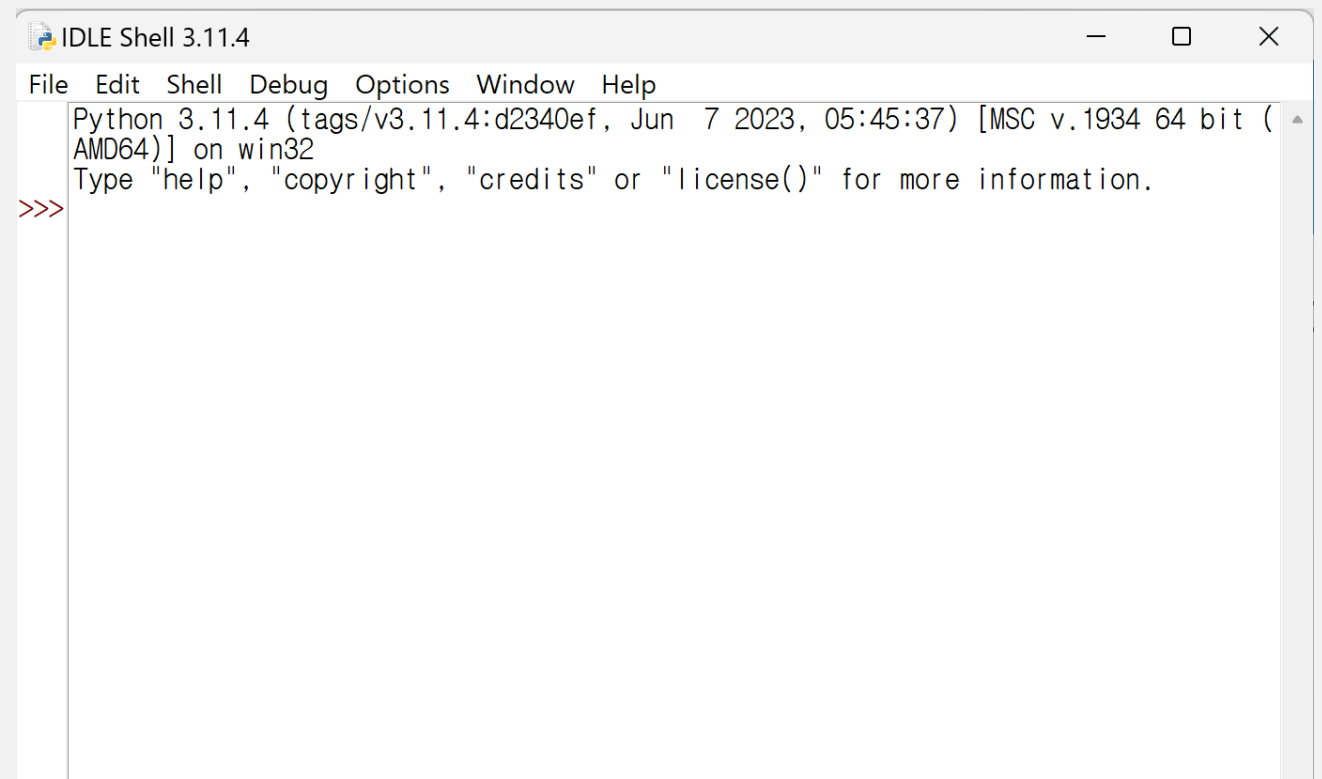
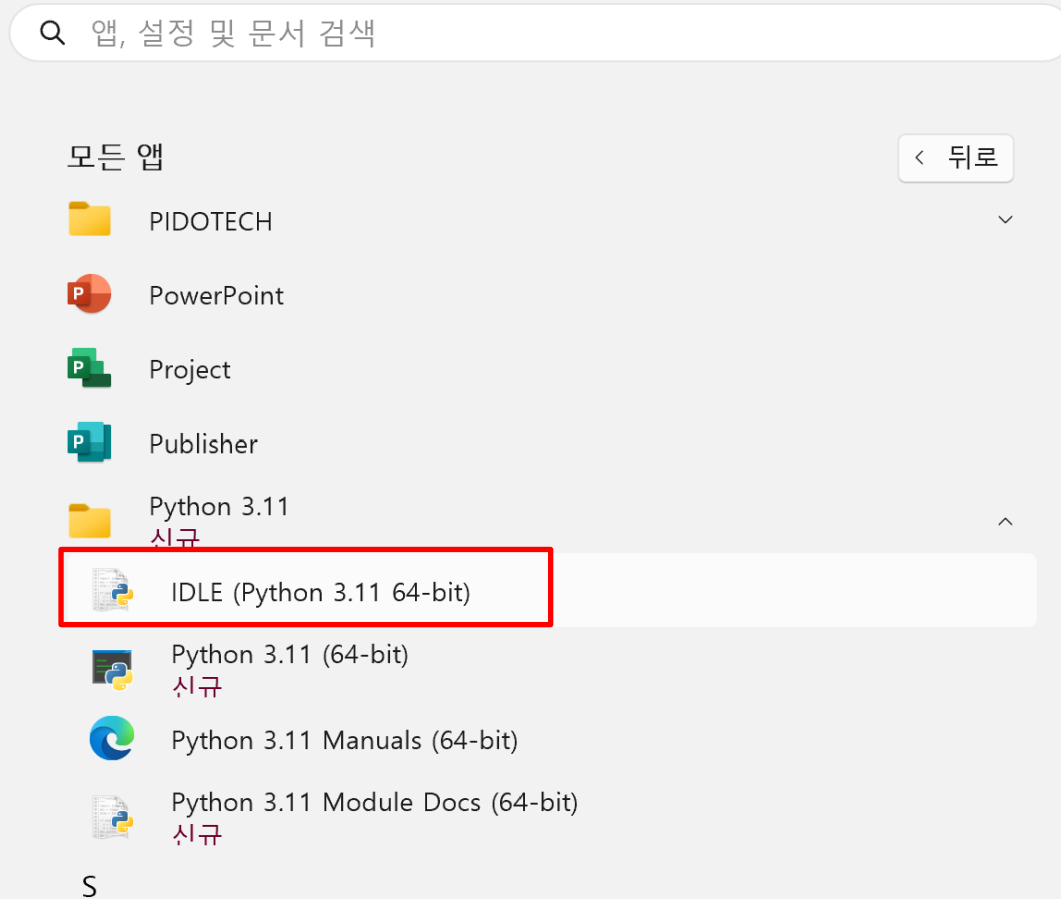
Python Release Python 3.11.4 x +						
python.org/downloads/release/python-3114/						
Version	Operating System	Description	MD5 Sum	File Size	GPG	Sigstore
Gzipped source tarball	Source release		bf6ec50f2f3bfa6ffbdb385286f2c628	26526163	SIG	.sigstore
XZ compressed source tarball	Source release		fb7f7eae520285788449d569e45b6718	19954828	SIG	.sigstore
macOS 64-bit universal2 installer	macOS	for macOS 10.9 and later	91498b67b9c4b5ef33d1b7327e401b17	43120982	SIG	.sigstore
Windows embeddable package (32-bit)	Windows		81b0acfcdd31a73d1577d6e977acbd6	9596761	SIG	.sigstore
Windows embeddable package (64-bit)	Windows		d0e85bf50d2adea597c40ee28e774081	10591509	SIG	.sigstore
Windows embeddable package (ARM64)	Windows		bdce328de19973012123dc62c1cfa7e9	9965162	SIG	.sigstore
Windows installer (32-bit)	Windows		9ec180db64c074e57bdcca8374e9ded6	24238000	SIG	.sigstore
Windows installer (64-bit)	Windows	Recommended	e4413bb7448cd13b437dffffba294ca0	25426160	SIG	.sigstore
Windows installer (ARM64)	Windows	Experimental	60785673d37c754ddceb5788b5e5baa9	24714240	SIG	.sigstore

Editor

- Visual Studio Code(code.visualstudio.com)
- PyCharm (www.jetbrains.com/pycharm/download)
- IDLE

Python 시작하기

- Windows의 시작 메뉴에서 "IDLE" program을 찾아서 실행



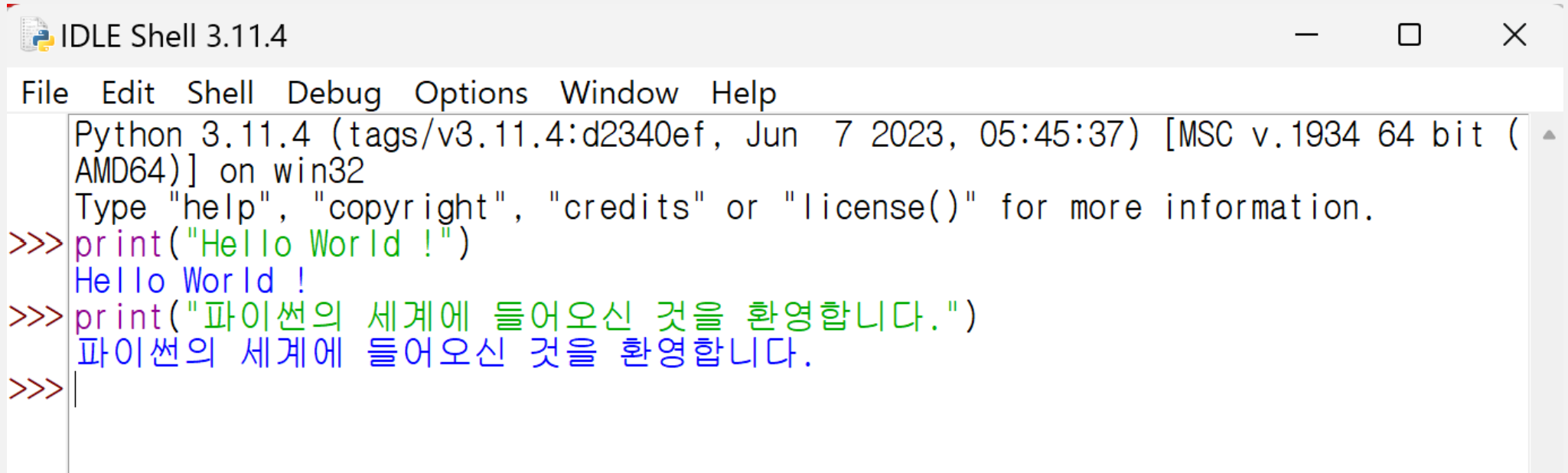
Python Shell

- Python Shell에서는 >>> 뒤에 명령어를 입력하고 <enter>키를 누르면 명령어가 실행되고 실행결과가 화면에 출력



```
IDLE Shell 3.11.4
File Edit Shell Debug Options Window Help
Python 3.11.4 (tags/v3.11.4:d2340ef, Jun 7 2023, 05:45:37) [MSC v.1934 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> print("Hello World!")
Hello World !
>>>
```

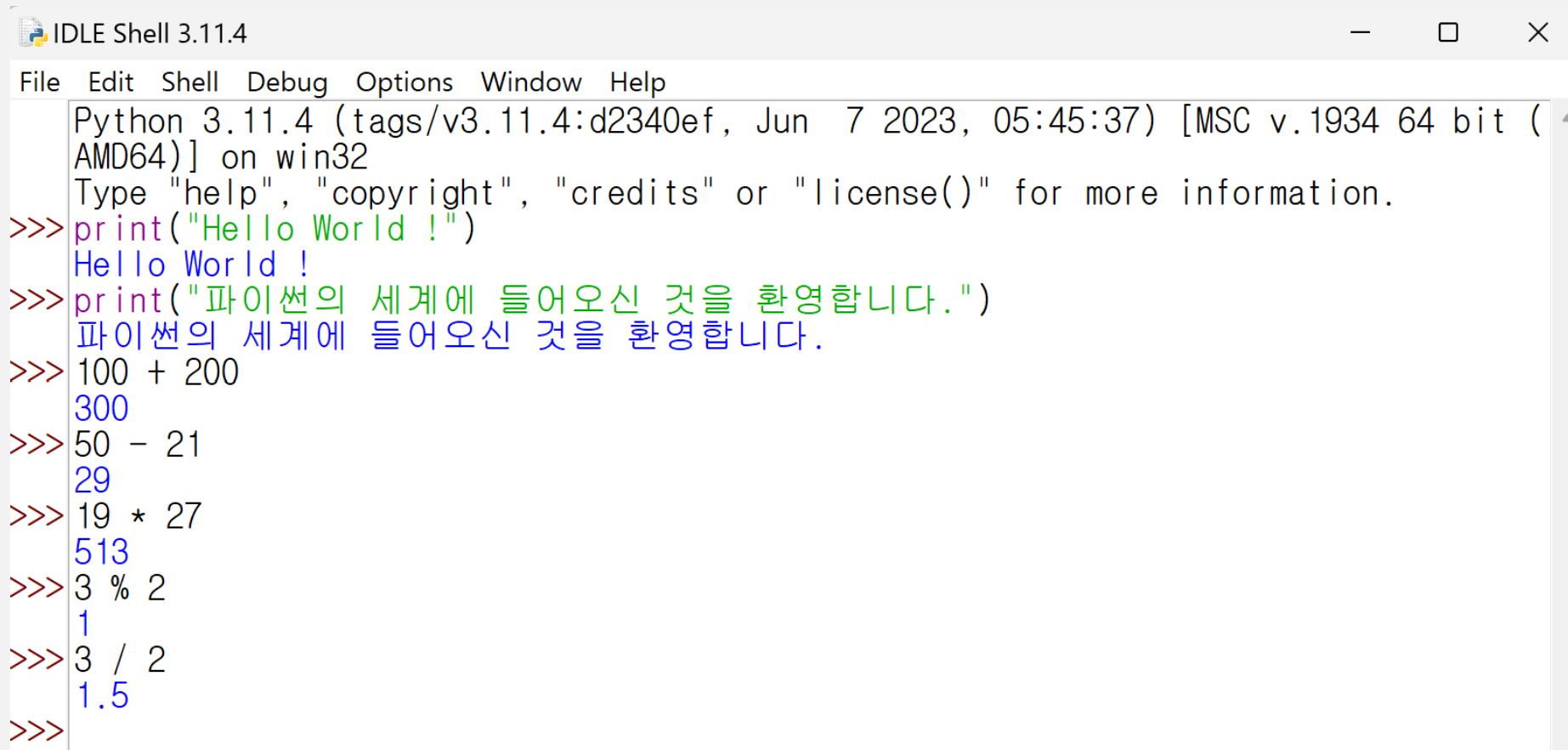
한글 출력



The screenshot shows the IDLE Shell 3.11.4 window. The title bar reads "IDLE Shell 3.11.4". The menu bar includes "File", "Edit", "Shell", "Debug", "Options", "Window", and "Help". The shell displays the following text:

```
Python 3.11.4 (tags/v3.11.4:d2340ef, Jun 7 2023, 05:45:37) [MSC v.1934 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> print("Hello World !")
Hello World !
>>> print("파이썬의 세계에 들어오신 것을 환영합니다.")
파이썬의 세계에 들어오신 것을 환영합니다.
>>> |
```

사칙계산

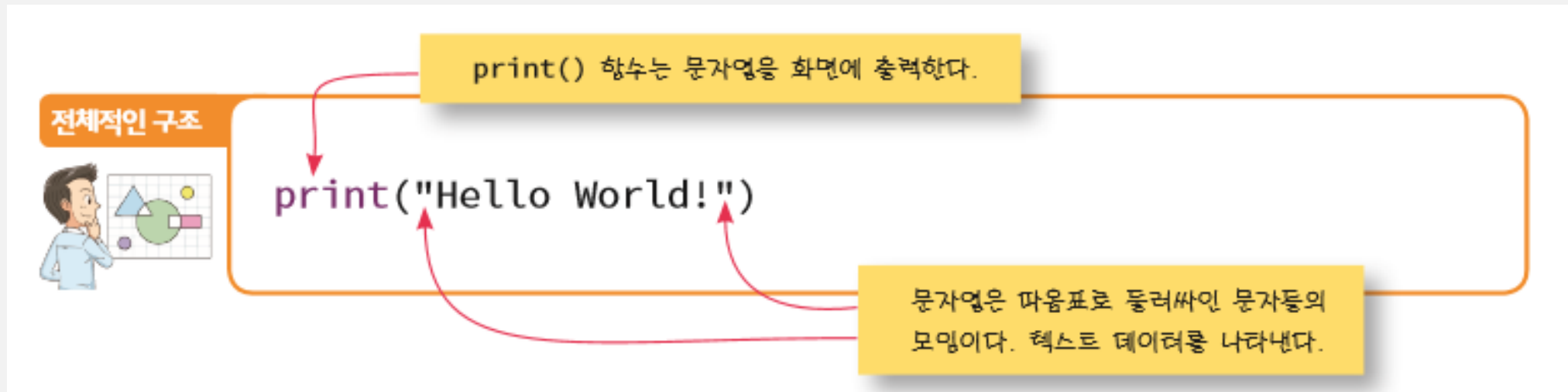


```
IDLE Shell 3.11.4
File Edit Shell Debug Options Window Help
Python 3.11.4 (tags/v3.11.4:d2340ef, Jun 7 2023, 05:45:37) [MSC v.1934 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> print("Hello World !")
Hello World !
>>> print("파이썬의 세계에 들어오신 것을 환영합니다.")
파이썬의 세계에 들어오신 것을 환영합니다.
>>> 100 + 200
300
>>> 50 - 21
29
>>> 19 * 27
513
>>> 3 % 2
1
>>> 3 / 2
1.5
>>>
```

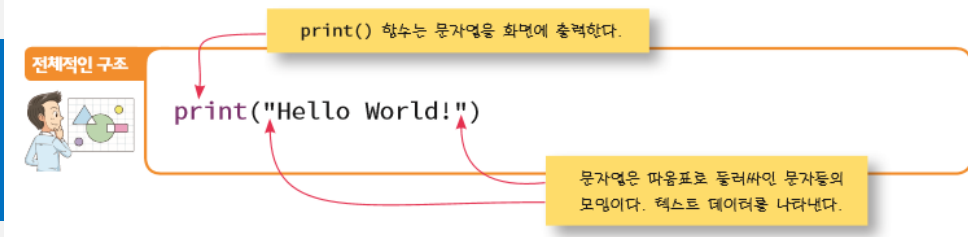
첫 번째 program 분석하기

- Python program은 여러 줄의 명령어로 이루어 짐
- 한 줄의 명령어를 문장(statement) 이라고 부름
- 문장들은 Python interpreter에 의하여 순차적으로 실행

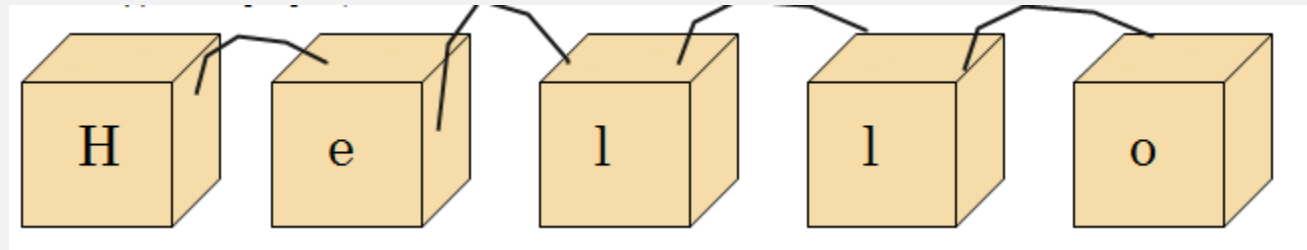
- system defined func
- user defined func



문자열 (string)



- 문자열(string) : 큰 따옴표("...")나 작은 따옴표('...') 안에 들어 있는 text data



- 반드시 따옴표가 있어야 함

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

print() 함수



- 여러 개의 값들을 화면에 차례대로 출력할 수 있음

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

Script mode(스크립트 모드)

- Code가 복잡해지면 interpret mode는 번거로움

```
web_crawling_3.py - C:\소사\과소사-강의예제\web_crawling\web_crawling_3.py (3.11.4)
File Edit Format Run Options Window Help
# 참조 : http://hleecaster.com/python-web-crawling-with-beautifulsoup/
# "11가지 프로젝트로 시작하는 생활프로그래밍", 이창현 저, 이지스퍼블리싱, 2020
#
D = True
D_1 = False

import os, re
import usecsv
import requests
import urllib.request as ur

from bs4 import BeautifulSoup as bs

try:
    os.chdir(r'C:\소사\과소사-강의예제\Web_crawling')

    news = "https://news.daum.net/"

    webpage = requests.get(news)
    if D:
        print("Wn1 >> webpage : ", webpage)

    soup = bs(webpage.content, 'html.parser')
    if D:
        print("Wn2 >> soup : ", soup)

    # 기사 제목 추출하기
    # find_all로 <div> 내용 추출
    # class 속성이 'item_issue' 인 div 안에 존재
    if D:
        print("Wn4 >> 기사 제목 추출하기")

    headline = soup.find_all('div', {"class": "item_issue"})

    for i in headline:
        print(i.text, "Wn")

    # find_all로 <a> tag 추출하기
    if D:
```



Source file(소스 파일) 작성하기

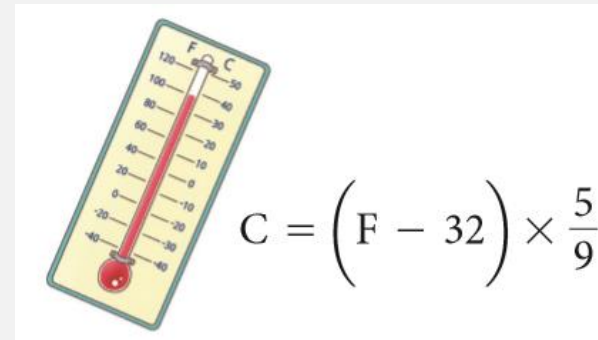
- Text editor를 이용하여 명령어들을 file에 저장한 후에 file을 읽어서 명령어들을 하나씩 실행하는 방법
 - Visual Studio Code(code.visualstudio.com)
 - PyCharm (www.jetbrains.com/pycharm/download)
 - IDLE
- 본 수업에서는 IDLE 사용
- 명령어들이 저장된 file을 **source file**이라고 함



온도 변환 program

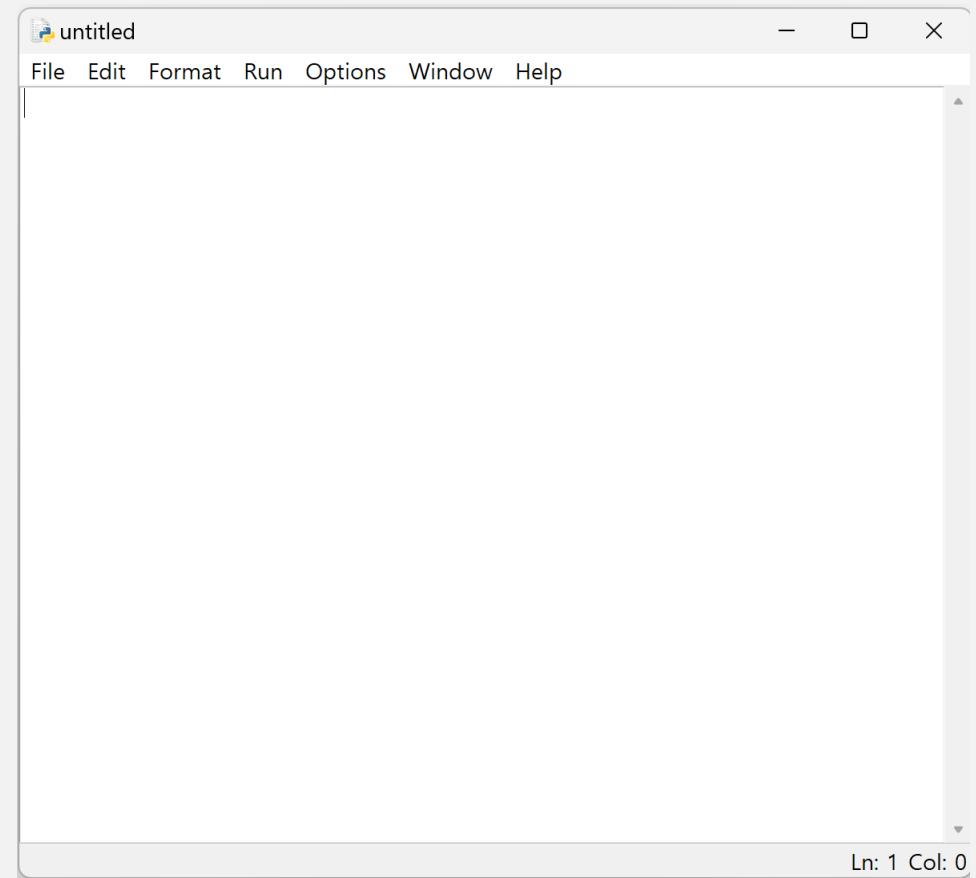
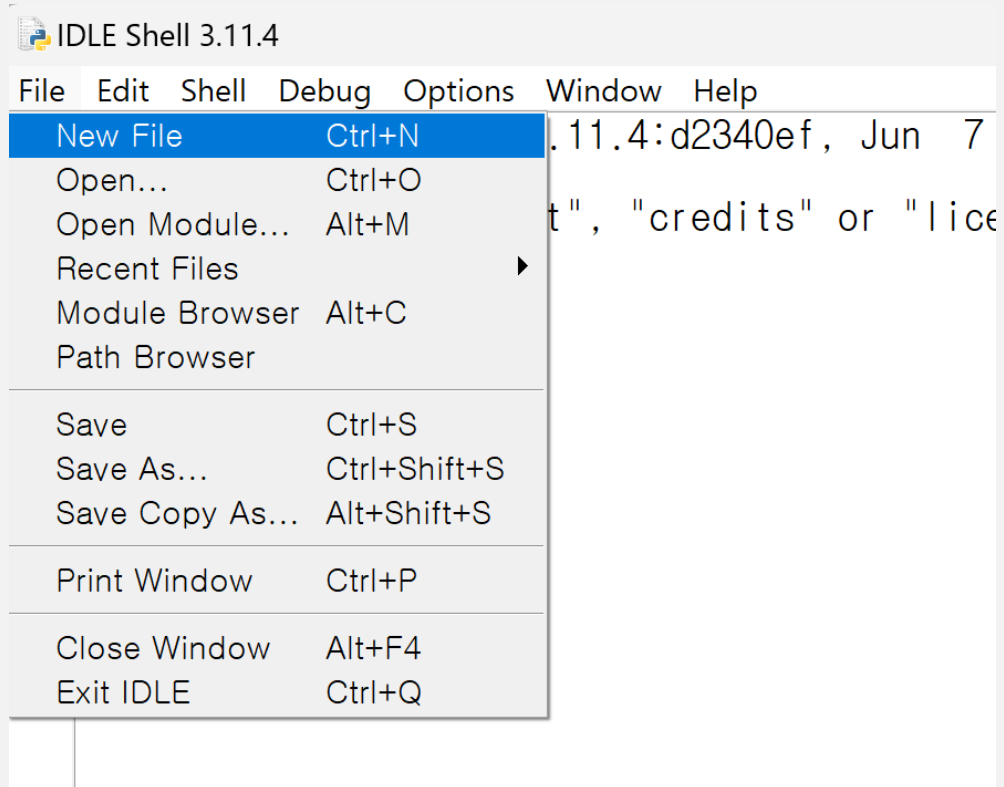
- 화씨온도->섭씨온도

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

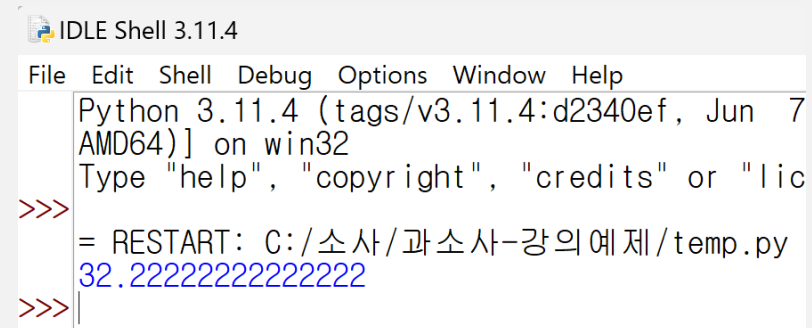
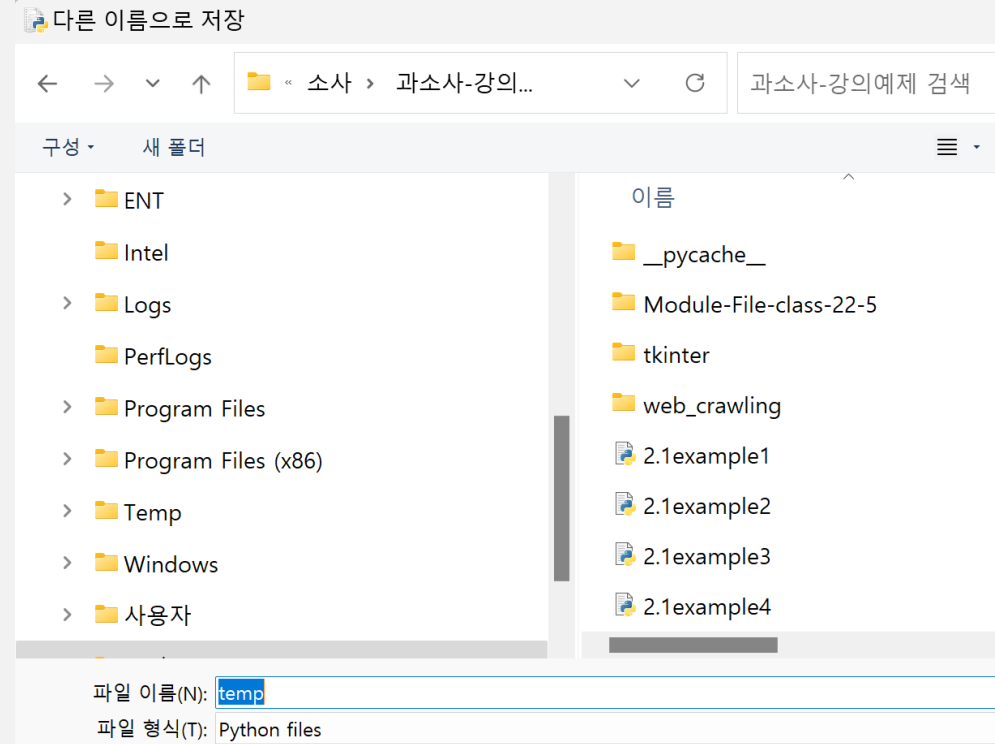
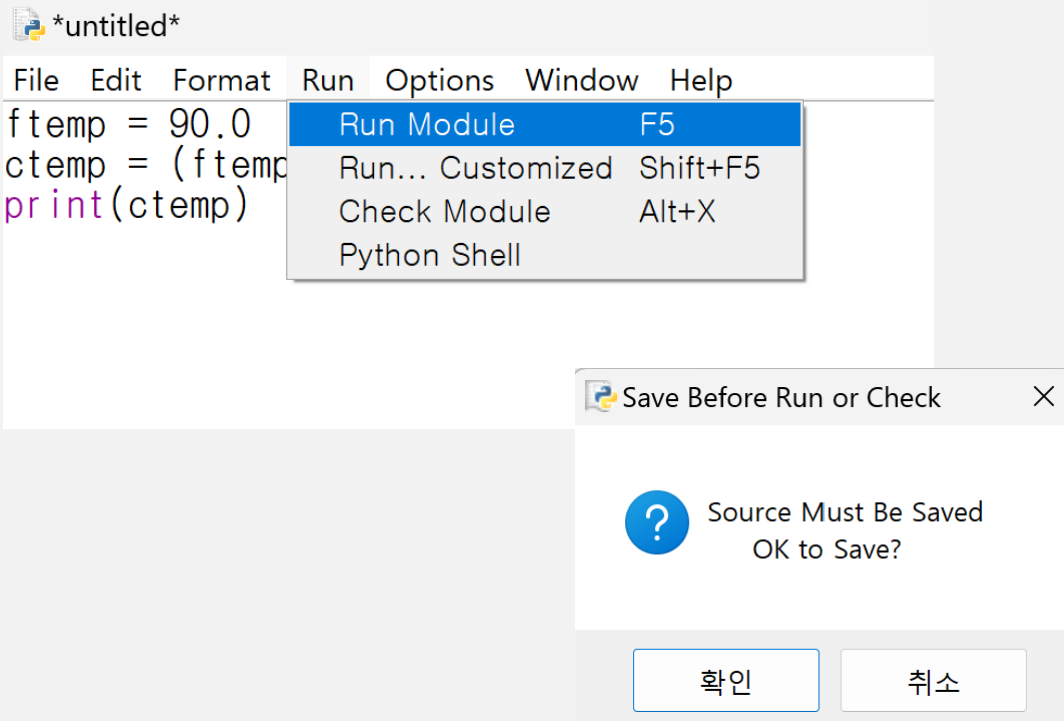


IDLE를 이용한 source file 작성

- Python shell의 menu 중에서 [File] -> [New File]을 선택한다.



IDLE를 이용한 source file 작성 - 실행결과



IDLE를 이용한 source file 작성 - 실행결과

temp.py - C:/소사/과소사-강의예제/temp.py (3.11.4)

File	Edit	Format	Run	Options	Window	Help
ftemp = 90.0				Run Module	F5	
ctemp = (ftemp				Run... Customized	Shift+F5	
print(ctemp)				Check Module	Alt+X	
				Python Shell		

IDLE Shell 3.11.4

File	Edit	Shell	Debug	Options	Window	Help
Python 3.11.4 (tags/v3.11.4:d2340ef, Jun 7 2023, 0 AMD64)] on win32						
Type "help", "copyright", "credits" or "license()"						
>>>	= RESTART: C:/소사/과소사-강의예제/temp.py					
	32.22222222222222					
>>>	===== RESTART: C:/소사/과소사-강					
	=====					
	32.22222222222222					
>>>						

Program의 간단한 분석

전체적인 구조



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

화씨 온도를 저장하는 변수.

명령어들은 위에서 아래로
순차적으로 실행된다.

90.0을 = 연산자를 이용하여
ftemp 변수에 저장한다.

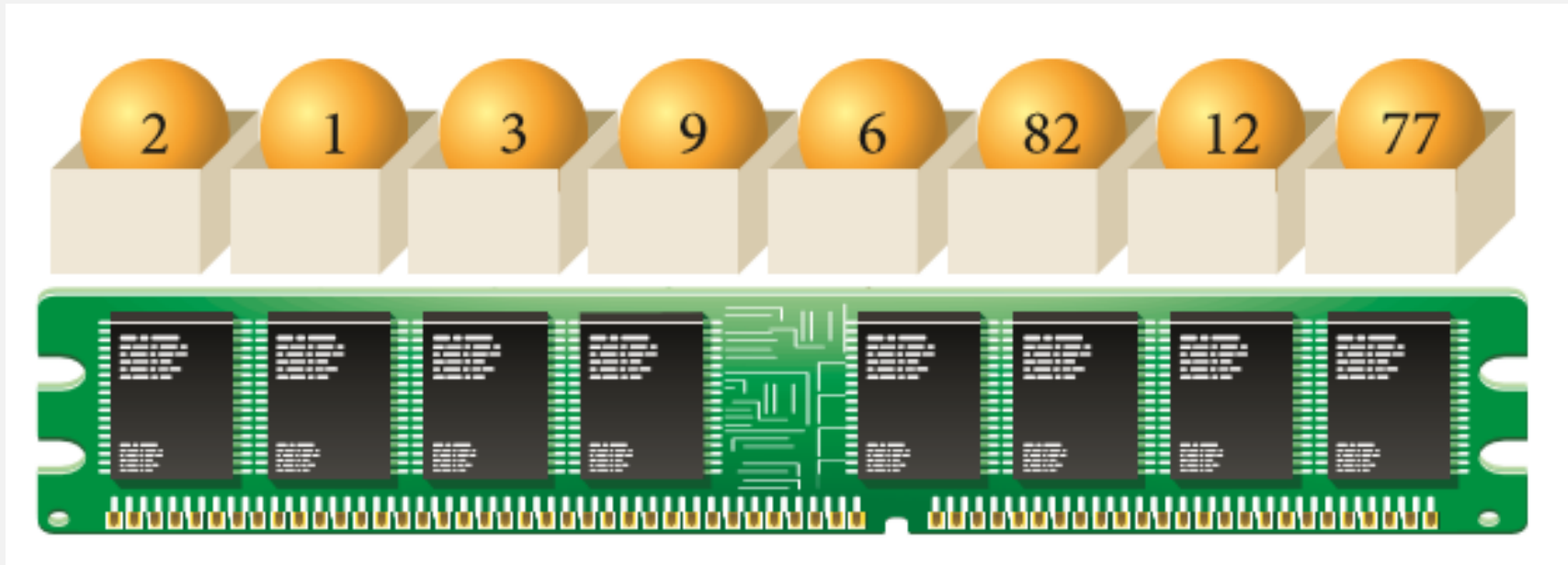
섭씨 온도를 저장하는 변수

화씨온도를 섭씨
온도로 변환한다.

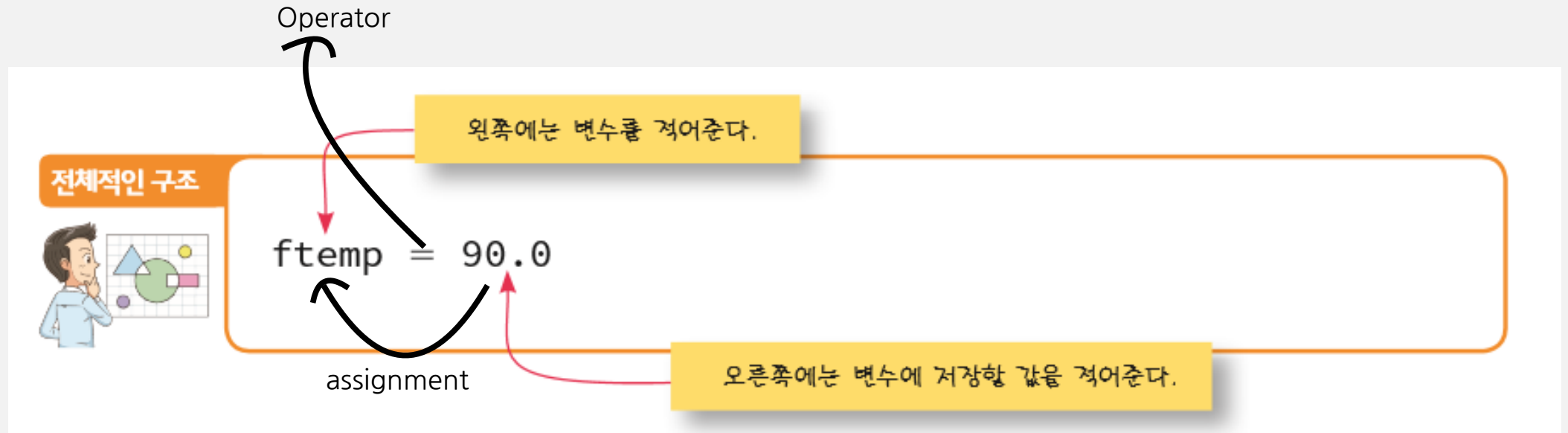
변수(變數 : variable)

- 변수 :
 - Computer의 memory 안에 만들어지는 공간으로 여기에 숫자나 문자를 저장

Variable vs Constant
-> C에는 Constant 존재




변수에 값을 저장할 때



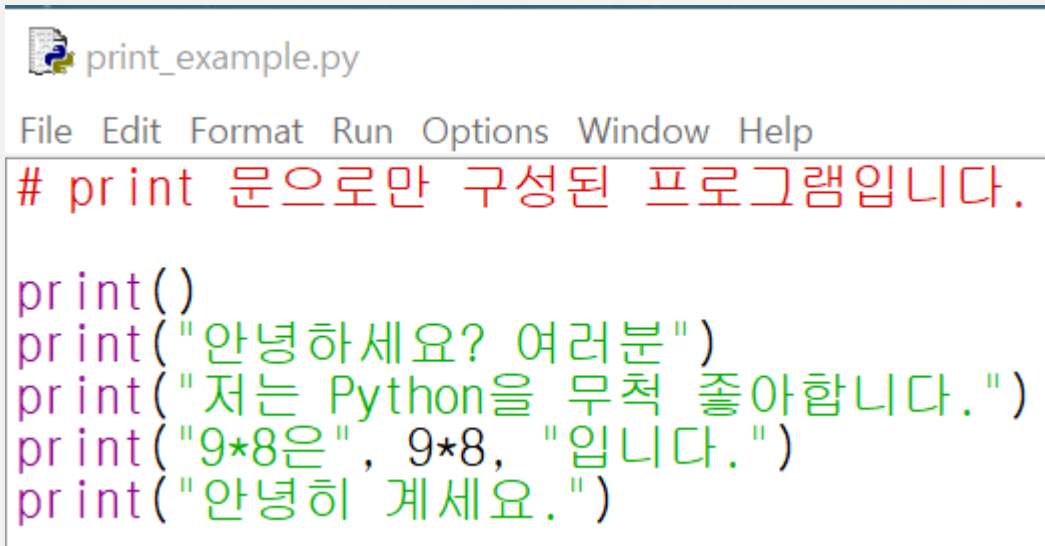
Lab: print() 함수 실습

- 다음과 같이 출력하는 source를 작성



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

Solution

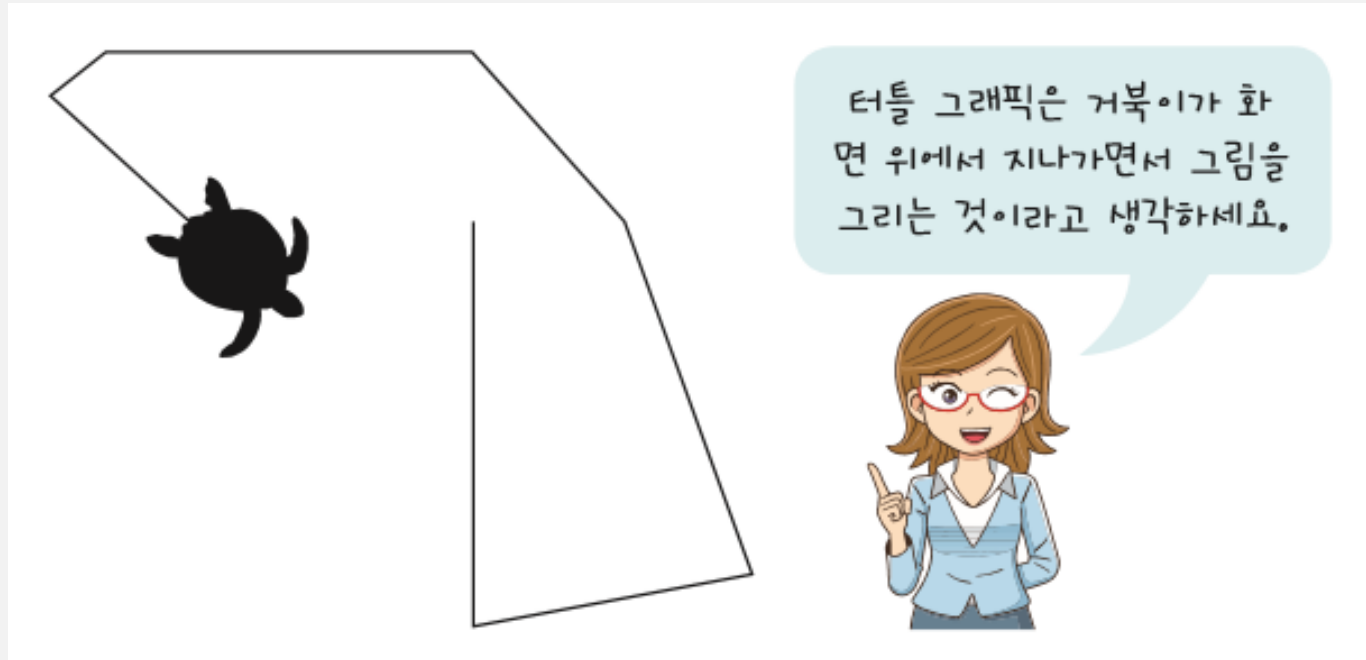


```
# print 문으로만 구성된 프로그램입니다.  
  
print()  
print("안녕하세요? 여러분")  
print("저는 Python을 무척 좋아합니다.")  
print("9*8은", 9*8, "입니다.")  
print("안녕히 계세요.")
```

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

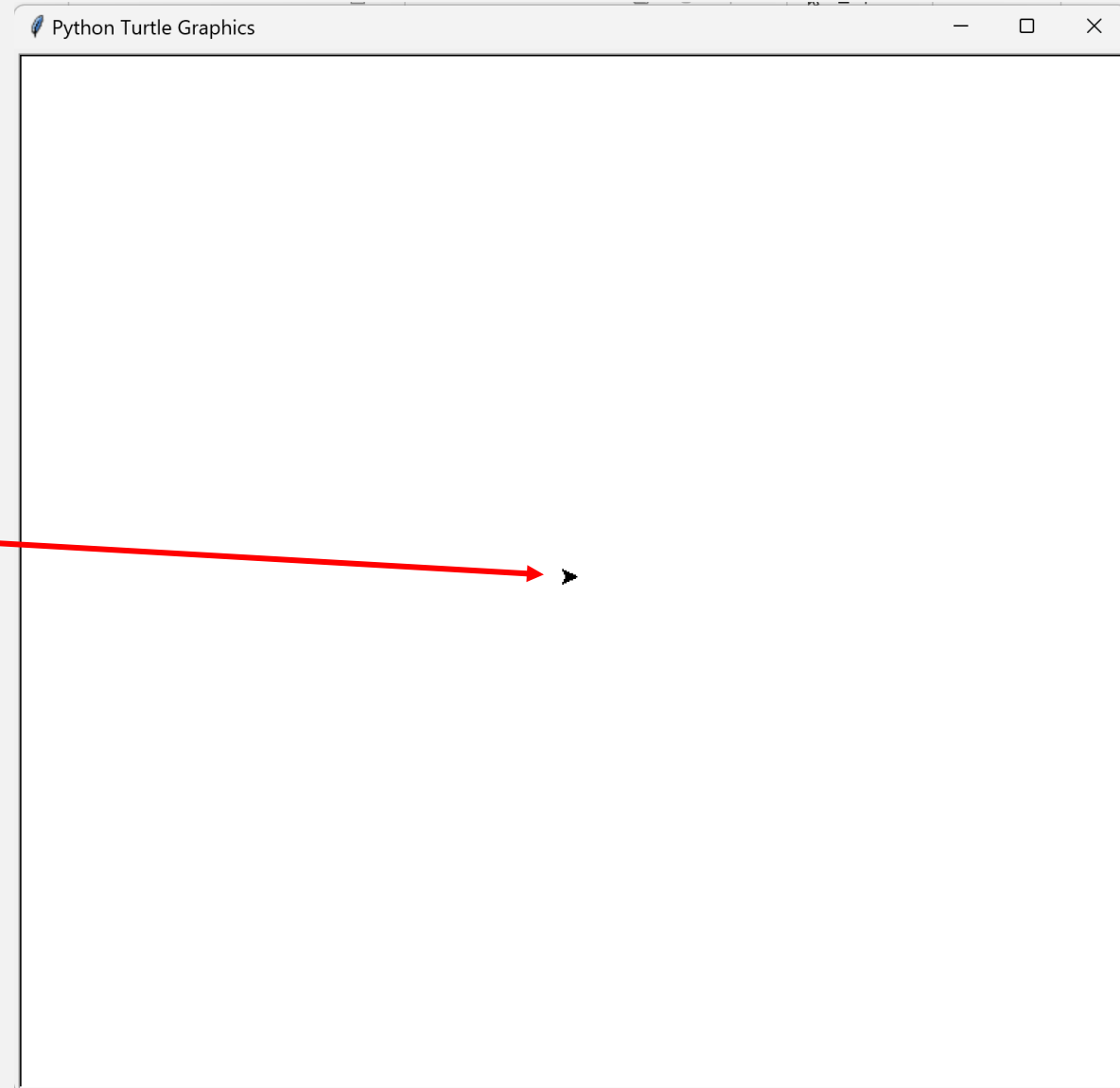
터틀 그래픽 (Turtle Graphic)

- Cartesian 공간에서 cursor(turtle)를 이용하여서 그림을 그리는 기능



Turtle Graphic Window

```
IDLE Shell 3.11.4
File Edit Shell Debug Options
Python 3.11.4 (tags/v
AMD64)] on win32
Type "help", "copyrig
>>>
>>>
>>> import turtle
>>> t = turtle.Pen()
>>>
```



Turtle Graphic Window

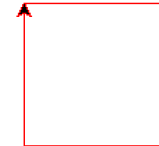
turtle_ch1.py - C:\소사\과소사-강의예제\turtle_ch1.py (3.11.4)

File Edit Format Run Options Window Help

```
import turtle
```

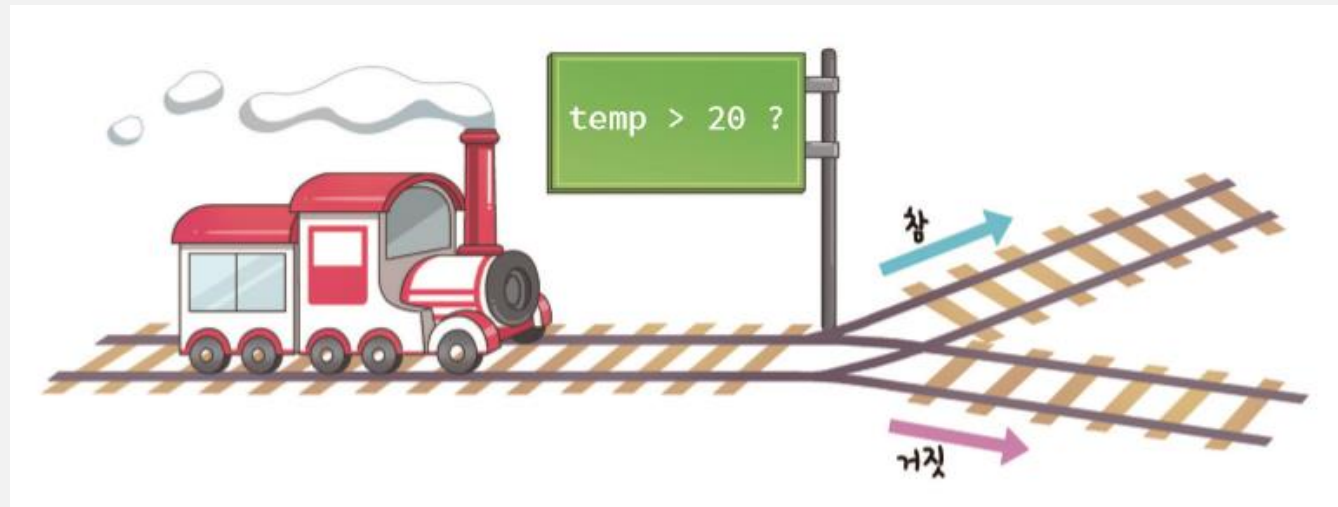
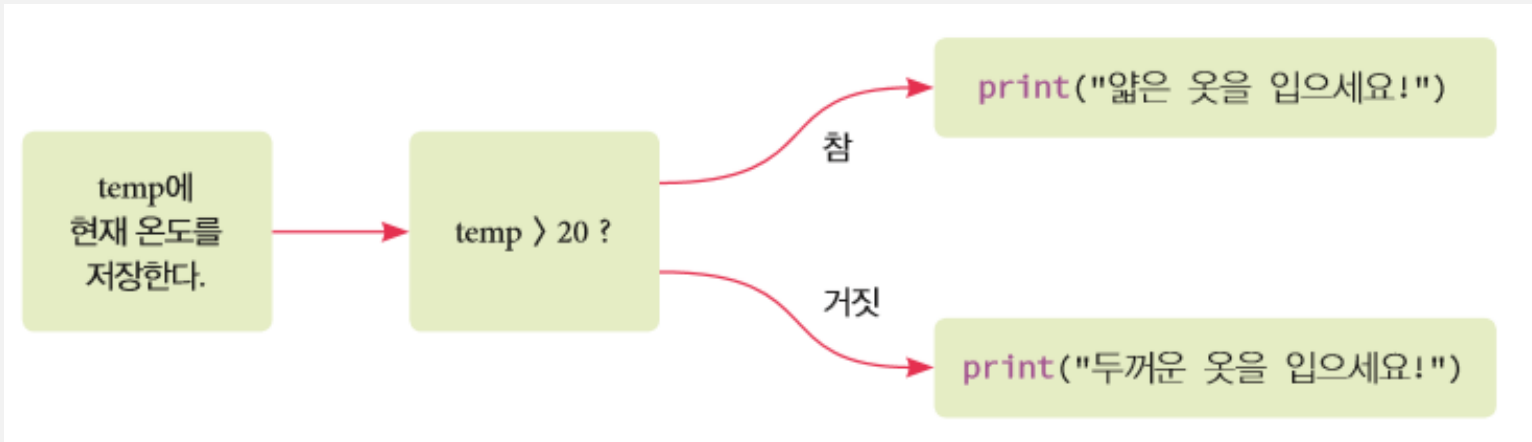
```
t=turtle.Pen()                # 또는 t=turtle.Turtle()  
t.pencolor("red")  
t.forward(100)  
t.right(90)  
t.forward(100)  
t.right(90)  
t.forward(100)  
t.right(90)  
t.forward(100)
```

Python Turtle Graphics



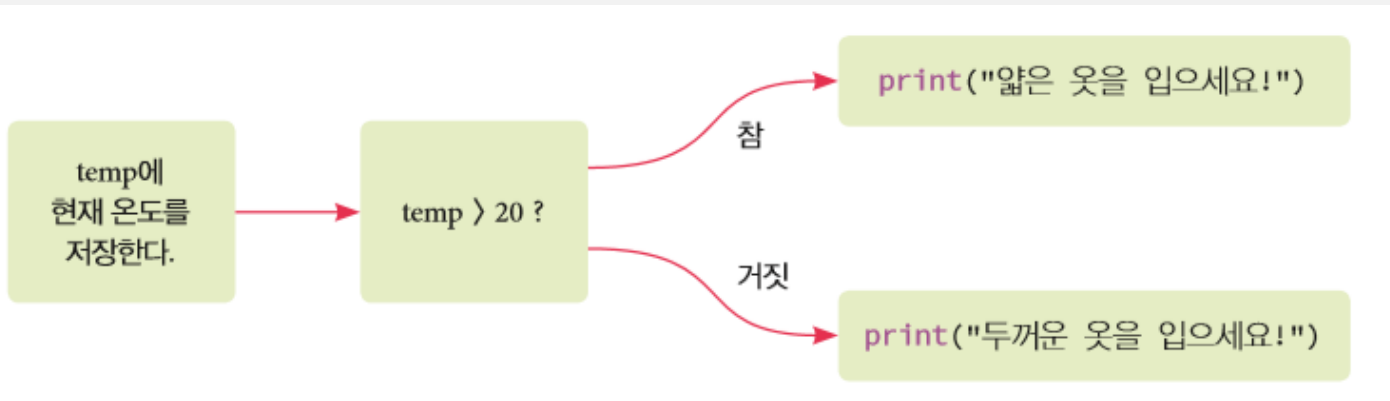
조건문(condition statement)

- 날씨에 따라서 옷을 선택해주는 program



조건문(condition statement)

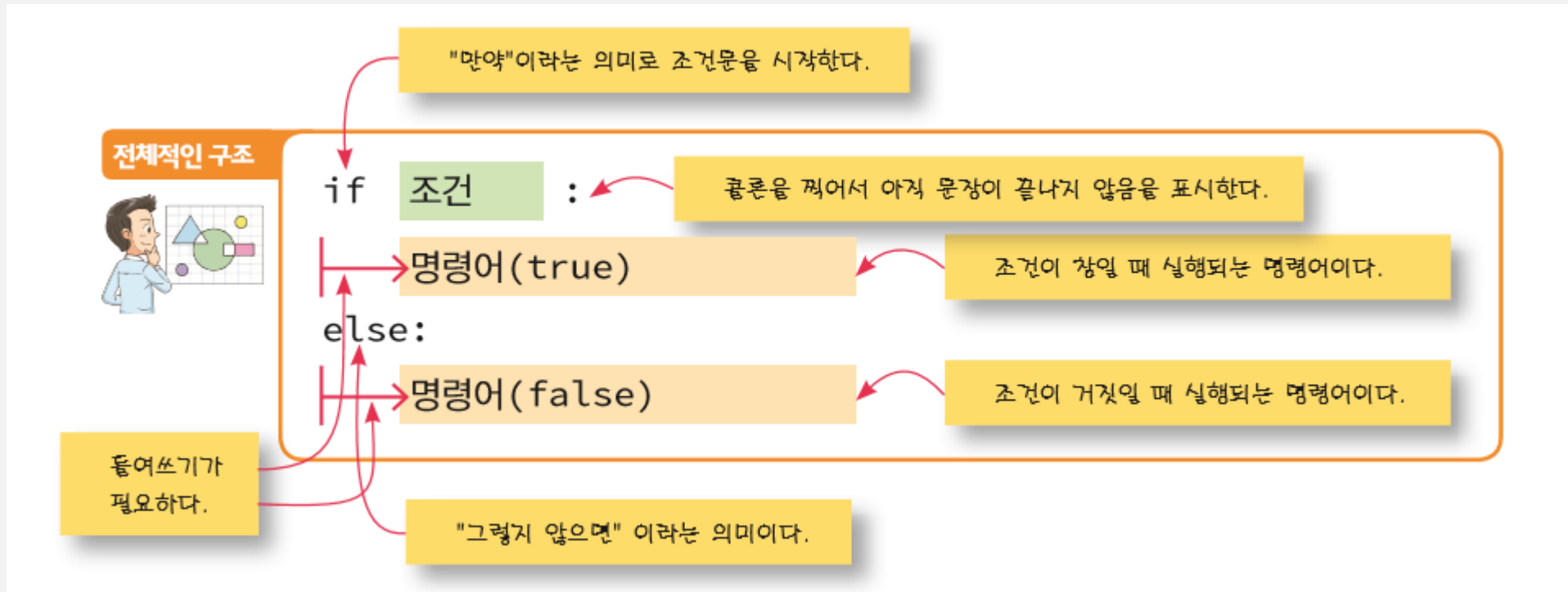
- 날씨에 따라서 옷을 선택해주는 program



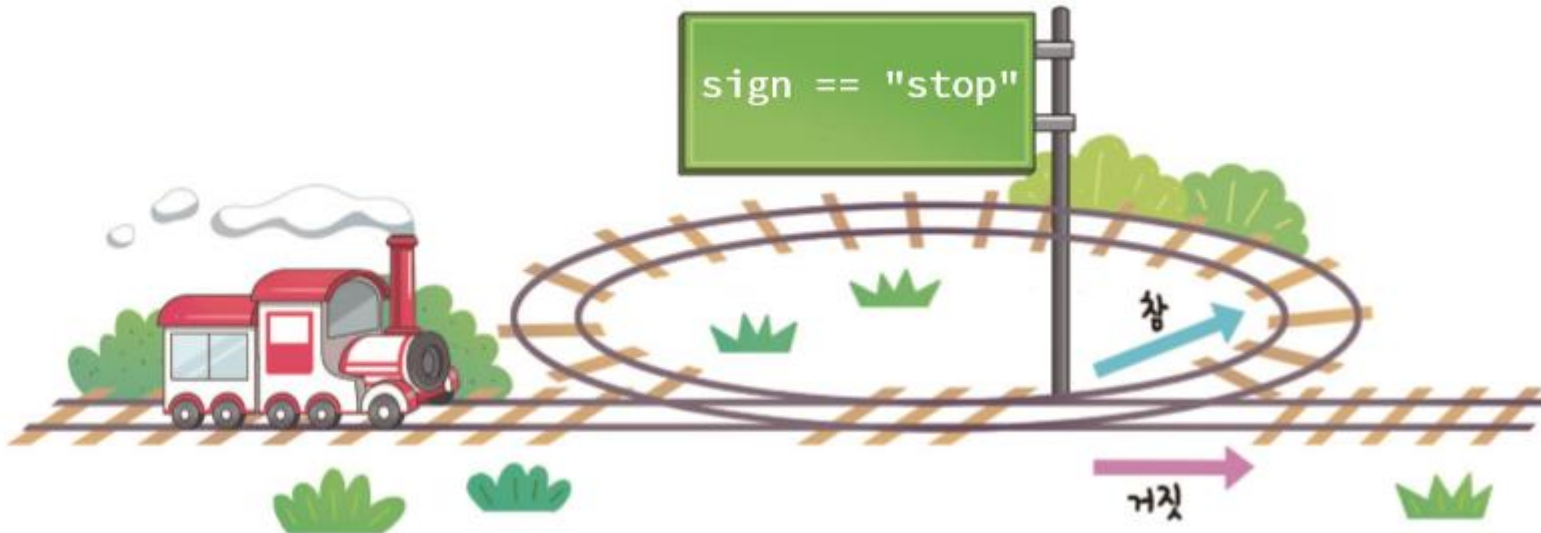
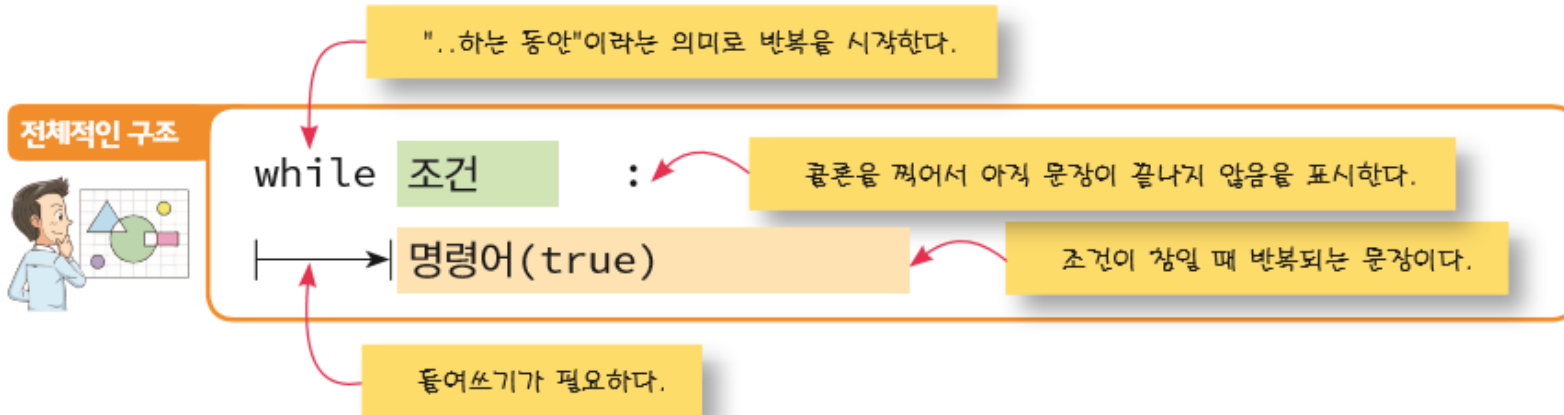
```
if_chap1.py
File Edit Format Run Options Window Help
1 temp = 10
2 if temp > 20 :
3     print("얇은 옷을 입으세요!")
4 else:
5     print("두꺼운 옷을 입으세요!")
6
7
```

```
=====
=====
두꺼운 옷을 입으세요!
>>> |
```

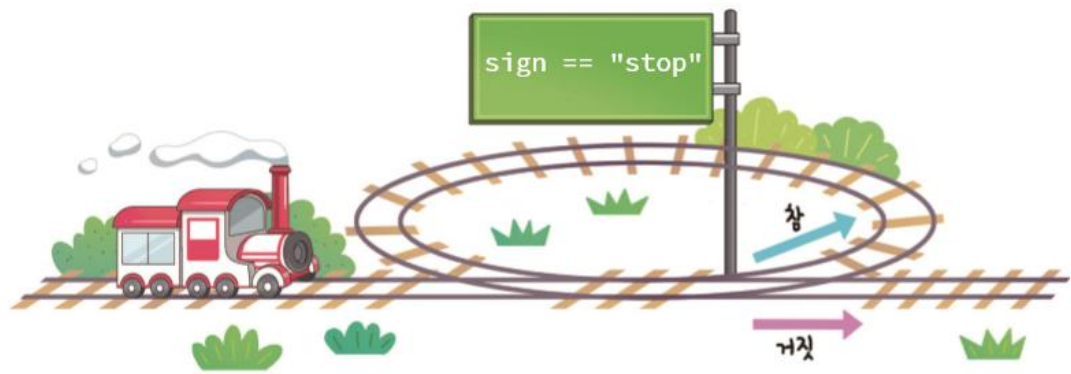
조건문(condition statement)



반복문 (iteration statement)



반복문 (iteration statement)



while_ch1.py

File Edit Format Run Options Window Help

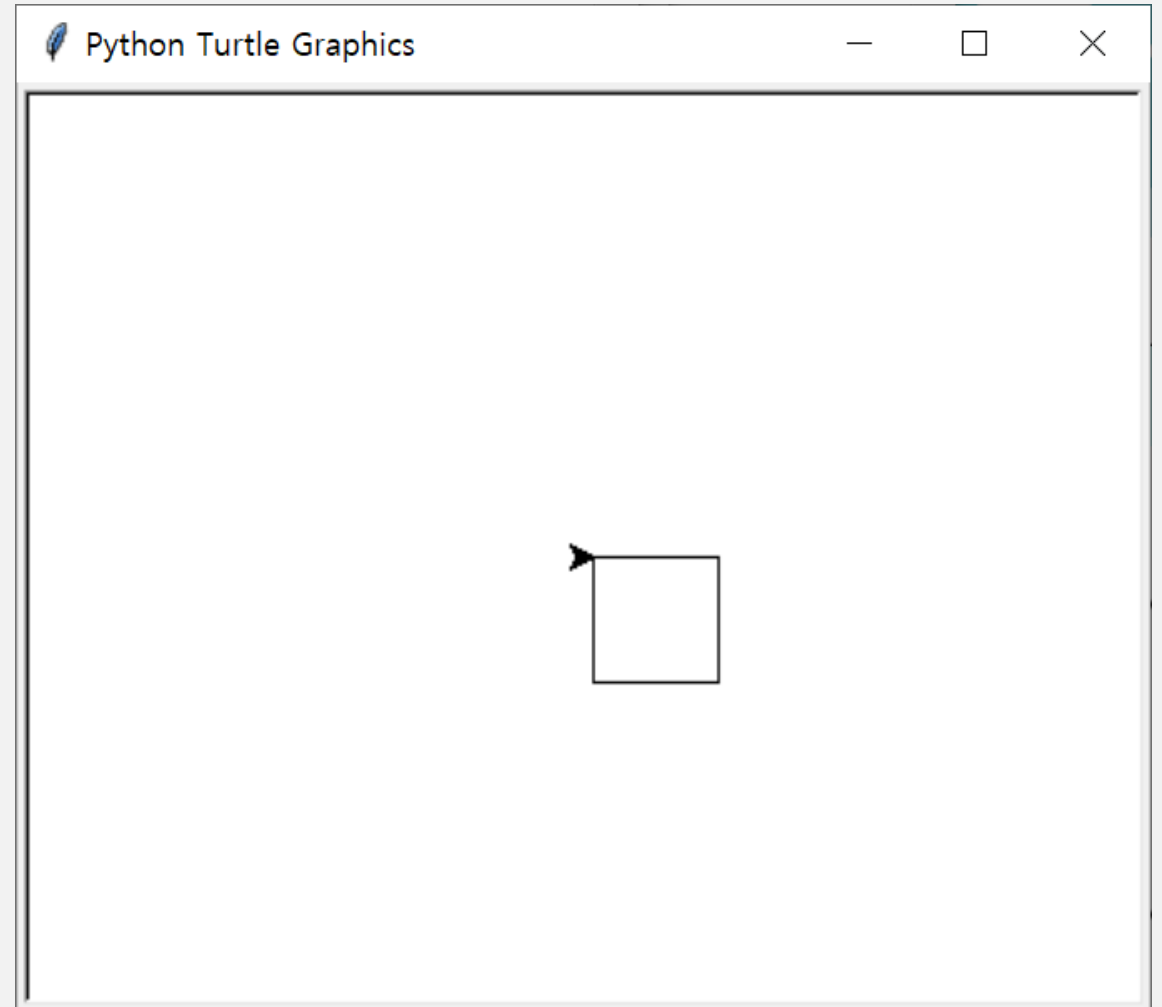
```
1 sign = "stop"
2
3 while sign == "stop":
4     sign = input("현재 신호를 입력하시오: ")
5
6 print("OK! 진행합니다.")
7
```

=====

```
현재 신호를 입력하시오: stop
현재 신호를 입력하시오: stop
현재 신호를 입력하시오: stop
현재 신호를 입력하시오: NO
OK! 진행합니다.
>>>
```

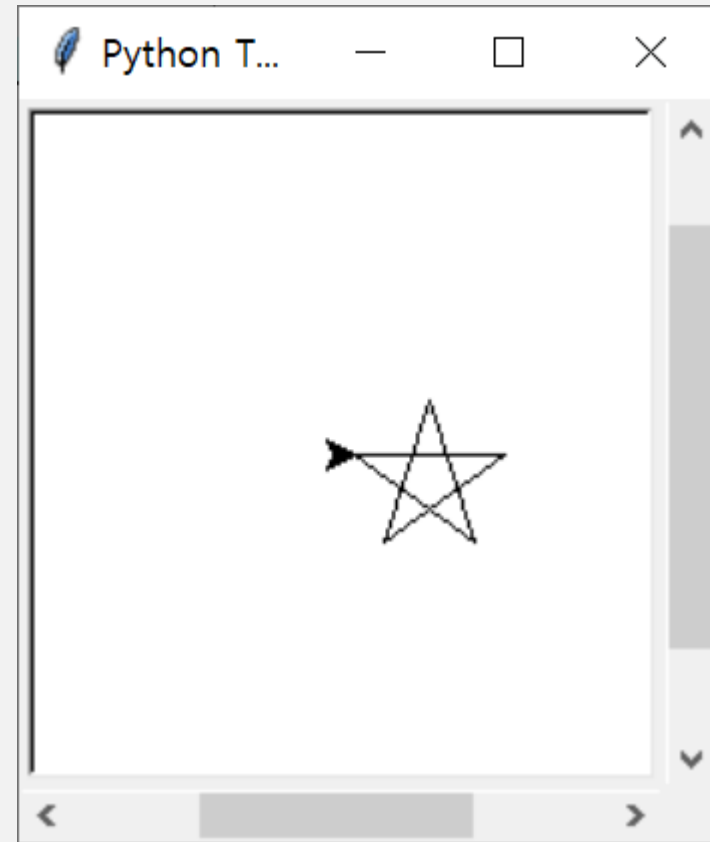
반복문을 사용하는 turtle graphic #1

```
turtle_while1_ch1.py
File Edit Format Run Options Window
1 import turtle
2
3 t = turtle.Turtle()
4
5 for i in [ 0, 1, 2, 3 ]:
6     t.forward(50)
7     t.right(90)
8
```



반복문을 사용하는 turtle graphic #2

```
turtle_while2_ch1.py
File Edit Format Run Options Window Help
1 import turtle
2
3 t = turtle.Turtle()
4
5 for i in [ 0, 1, 2, 3, 4 ]:
6     t.forward(50)
7     t.right(144)
8
```



Lab: 숫자 추측 게임

- 조건문을 사용하여 간단한 숫자 맞추기 게임을 작성해보자



Solution

Choose_number.py

File Edit Format Run Options Window Help

```
#
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.")
```

IDLE Shell 3.11.4

File Edit Shell Debug Options Window Help


Python 3.11.4 (tags/v3.11.4:d2340ef, Jun 7
AMD64) on win32
Type "help", "copyright", "credits" or "lice

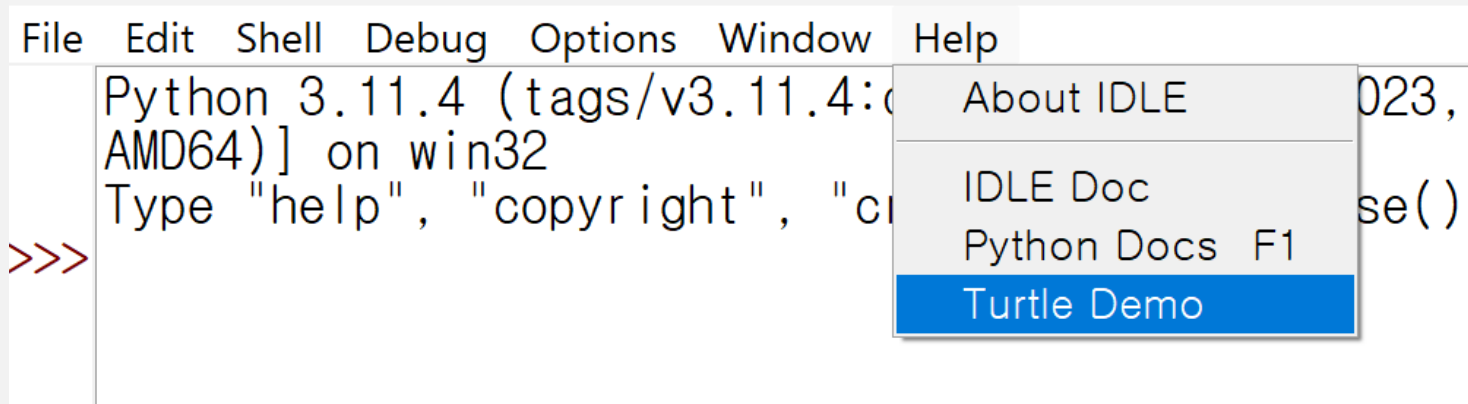
>>>

= RESTART: C:/소사/과소사-강의예제/Choose_nu
>> 숫자게임에 오신 것을 환영합니다.
1부터 100 사이의 숫자를 추측해보세요: 62
>> 맞았습니다.
>> 게임이 종료되었습니다.

Python과 놀아보자.

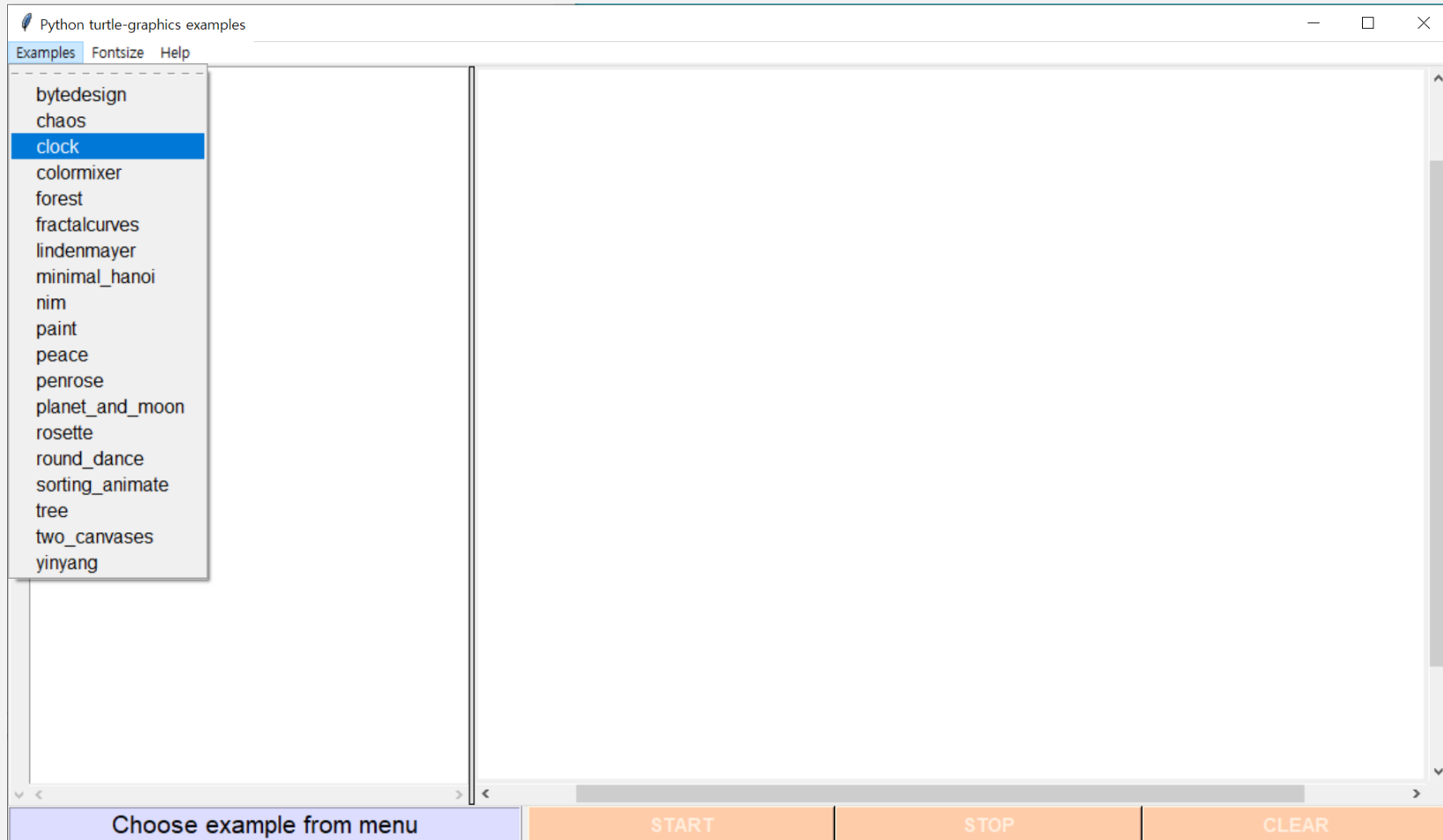
- Python Shell에서 [Help] ->[Turtle Demo]를 선택

 IDLE Shell 3.11.4



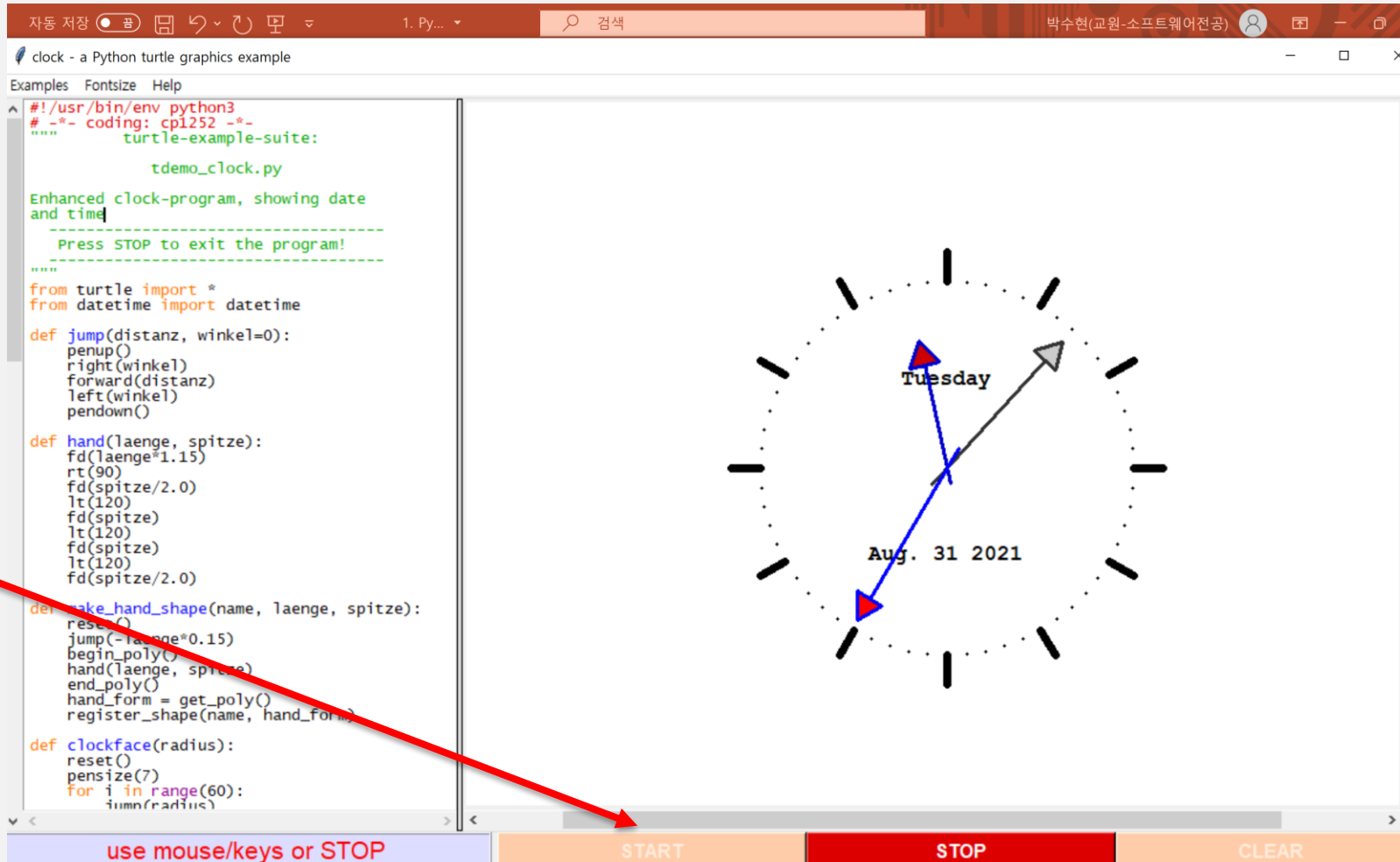
Python과 놀아보자.

- Python 쉘에서 [Examples] ->[Clock]

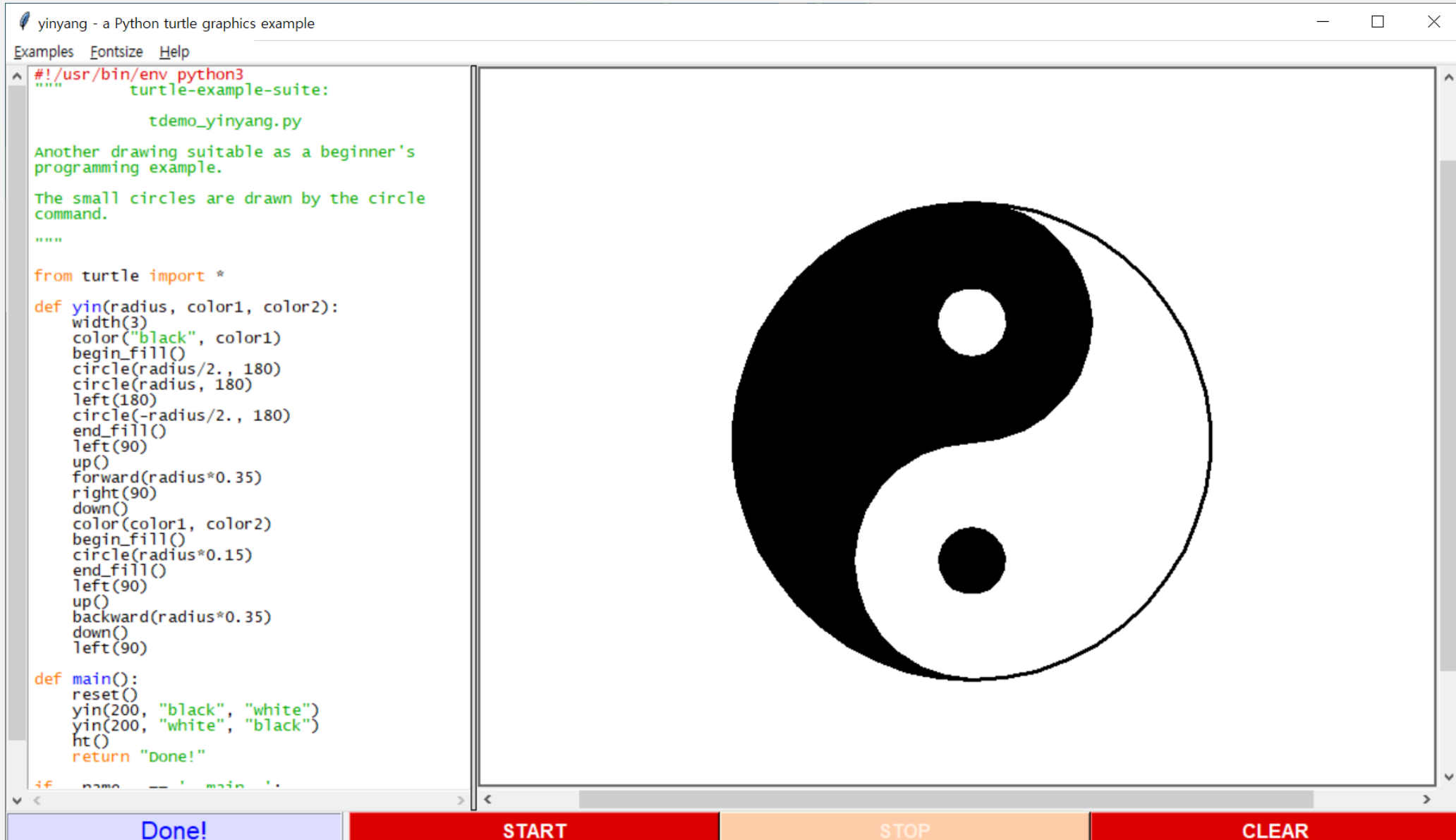


Python과 놀아보자.

- Python 쉘에서 [Examples] ->[Clock]



Python과 놀아보자.



핵심 정리

- program은 명령어들로 이루어진 text file 형태로 작성된다. 이것을 source file이라고 한다.
- Python interpreter는 source file을 해석하여서 computer가 이해할 수 있는 기계어 파일(machine code)로 변환하여 생성한다.
- 문장들은 기본적으로 순차적으로 실행되지만 조건에 따라서 서로 다른 경로로 실행되거나 반복될 수 있다.

Homeworks

- 자신의 실습 컴퓨터에 python 실행환경 install
- 예제code coding 및 실행
 - code
 - zip으로 묶어서 제출
 - File 명 : ch1_이름_학번.zip
 - 예) ch1_김국민_20230123.zip
- 제출방법
 - ecampus 숙제제출 link에 upload
- 제출마감
 - 2023. 9 .19(화) 13:00
 - 제출 마감 일시까지만 제출 가능. 마감일시 이후 ecampus 숙제제출 링크 자동 close