

[Python]



# Python으로 배우는 소프트웨어 원리

Appendix 00. 파이썬 개발환경 구성 및 경험하기

# 목차

1. Python 설치
2. [IDLE] 명령어 실행
3. [IDLE] Demo 코드
4. PyInstaller 설치
5. Demo 코드 컴파일

# 01. 파이썬 설치

## I. 파이썬 설치

- ① 파이썬 공식 홈페이지(<http://python.org>)에 접속

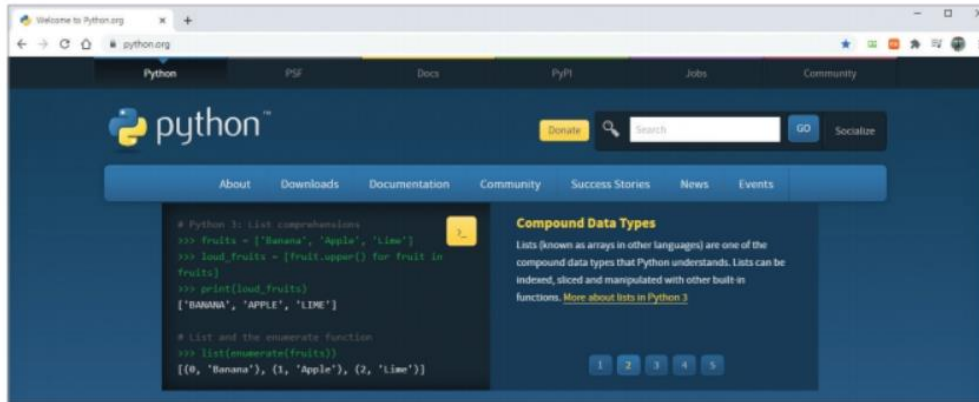


그림 2-6 파이썬 사이트 메인 화면

- ② [Downloads] 메뉴를 클릭하고  
[Windows]를 선택

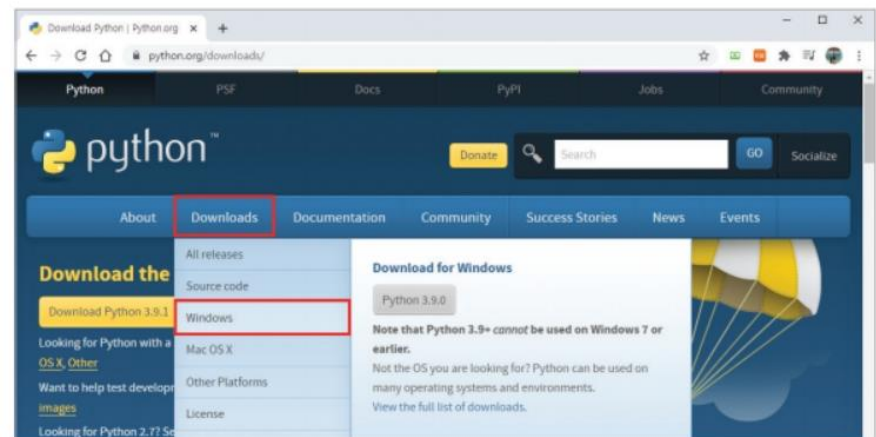


그림 2-7 [Downloads] 메뉴 클릭 후 [Windows] 선택

# 01. 파이썬 설치

## I. 파이썬 설치

- ③ 최근 버전의 [Download Windows Installer (64-bit)] 클릭
- ④ 설치 파일을 다운로드할 폴더를 선택하고 클릭

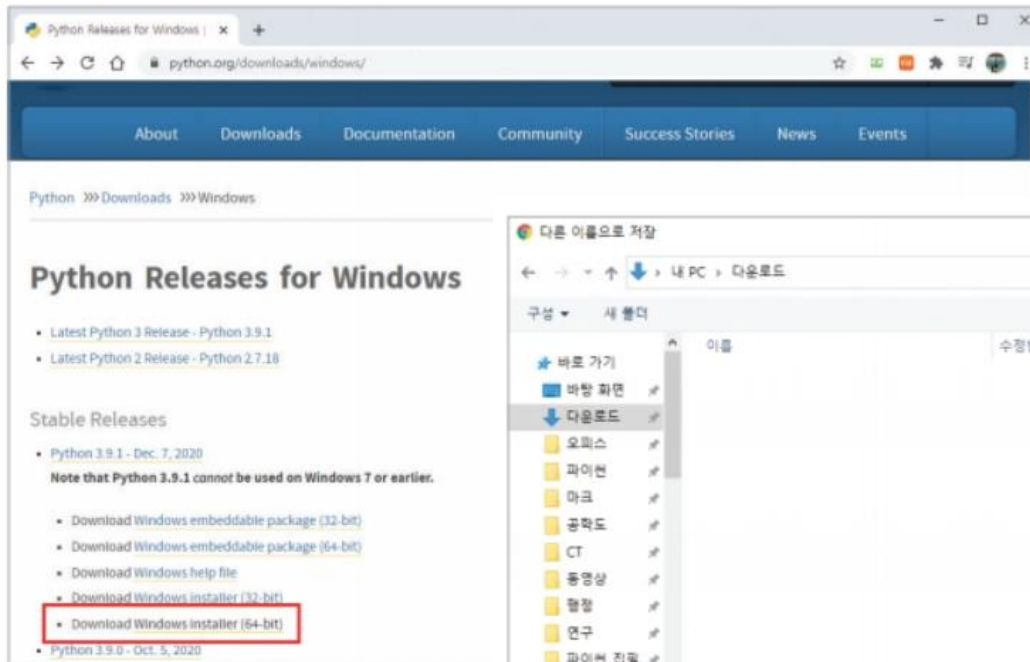


그림 2-8 [Download Windows Installer

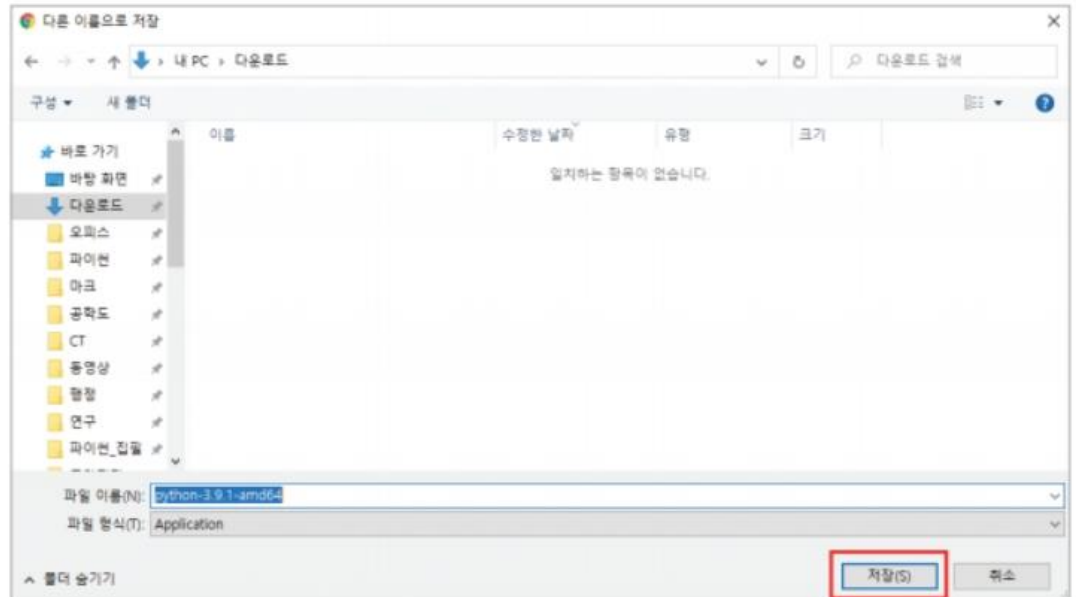


그림 2-9 설치 파일을 다운로드할 폴더 선택

# 01. 파이썬 설치

## I. 파이썬 설치

- ⑤ 다운로드한 폴더에 있는 설치 파일을 실행하고, 설치 시작 화면에서 [Add Python 3.10.1 to PATH] 항목을 반드시 클릭
- ⑥ 설치 과정이 끝난 후 설치 성공을 알리는 화면이 나타나면 [Close] 클릭

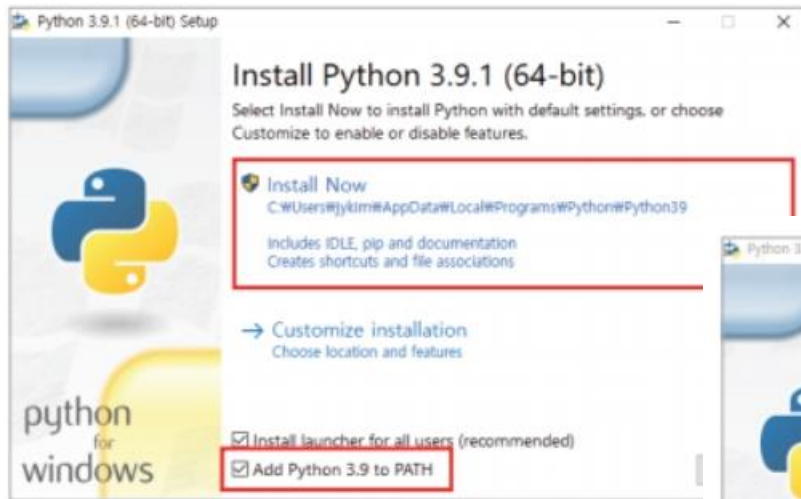


그림 2-10 설치 시작

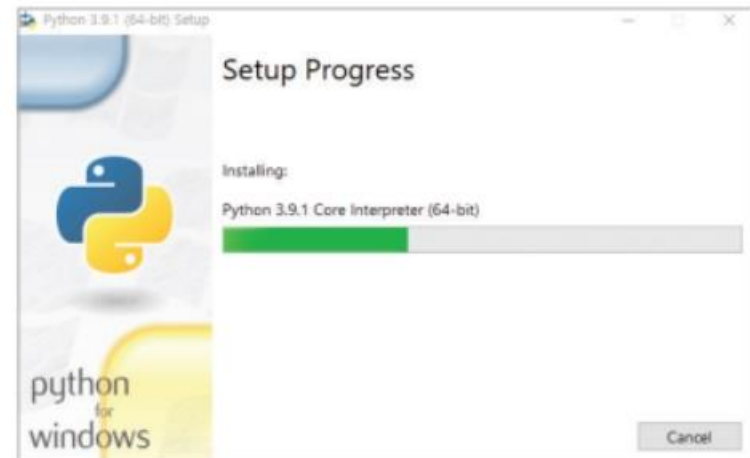


그림 2-11 설치 진행

## 02. IDLE

### I. 명령어 실행

#### ■ IDLE 셸 실행

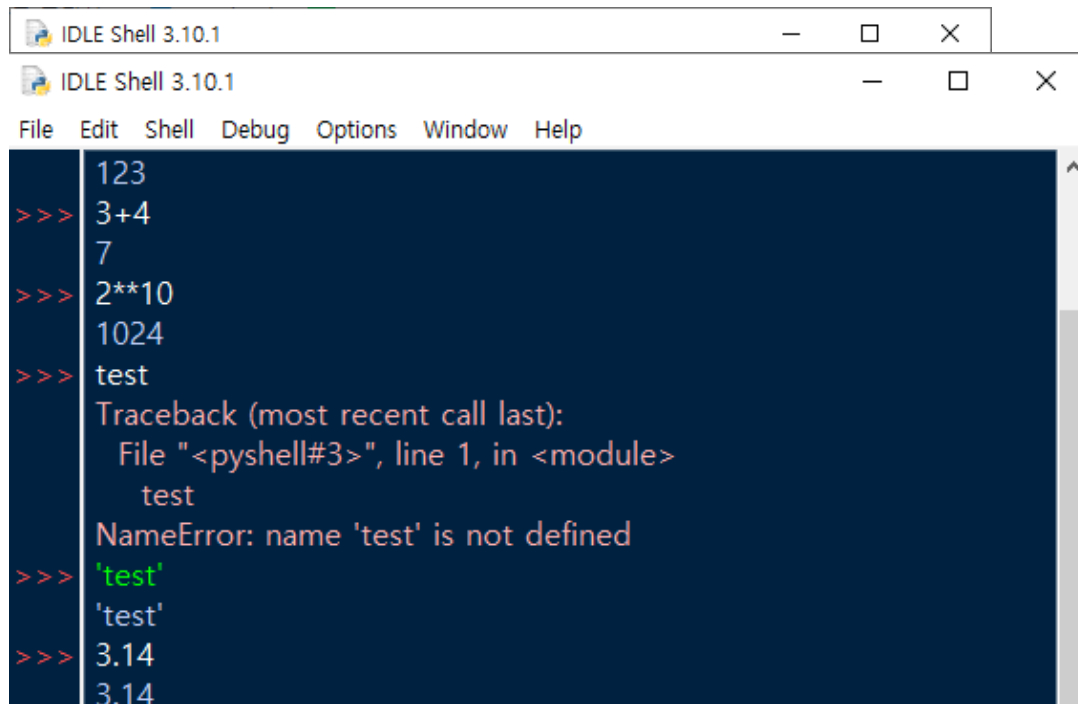
- 윈도우 시작에서 IDLE로 검색

- IDLE (Python 3.10 64-bit) 선택 실행

❖ 셸(shell)은 이용자가 입력한 문장을 읽어  
그 문장이 요구하는 명령을 수행하도록 해주는 명령 해석기

❖ IDLE는 Python 통합개발환경 툴  
통합개발환경(IDE: Integrated Development Environment)

- 셸을 종료하려면 'exit( )'를 입력하거나, 단축키 Ctrl + Z 혹은 [X] 단추를 클릭



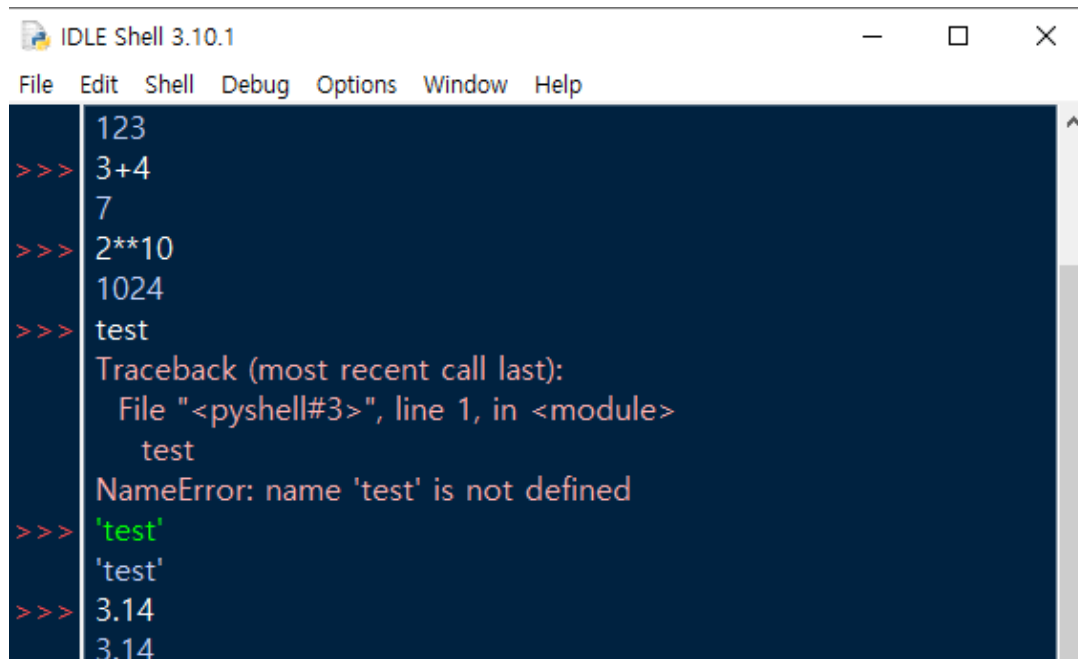
```
123
>>> 3+4
7
>>> 2**10
1024
>>> test
Traceback (most recent call last):
  File "<pyshell#3>", line 1, in <module>
    test
NameError: name 'test' is not defined
>>> 'test'
'test'
>>> 3.14
3.14
```

## 02. IDLE

### I. 명령어 실행

#### ■ 명령어 입력 실행

- 파이썬 프롬프트(>>>)가 표시됨 ❖ 프롬프트(prompt)는 명령어 입력을 기다리는 표시자
- 명령어를 입력하고 엔터 키를 누르면 실행
- 셸은 기본적으로 명령어를 처리해주는 기능을 하므로 값과 명령어만 입력 가능
- 값은 정수, 실수, 문자 형태만 가능 → 숫자는 그대로 입력, 문자는 " 또는 ""로 감싸 입력

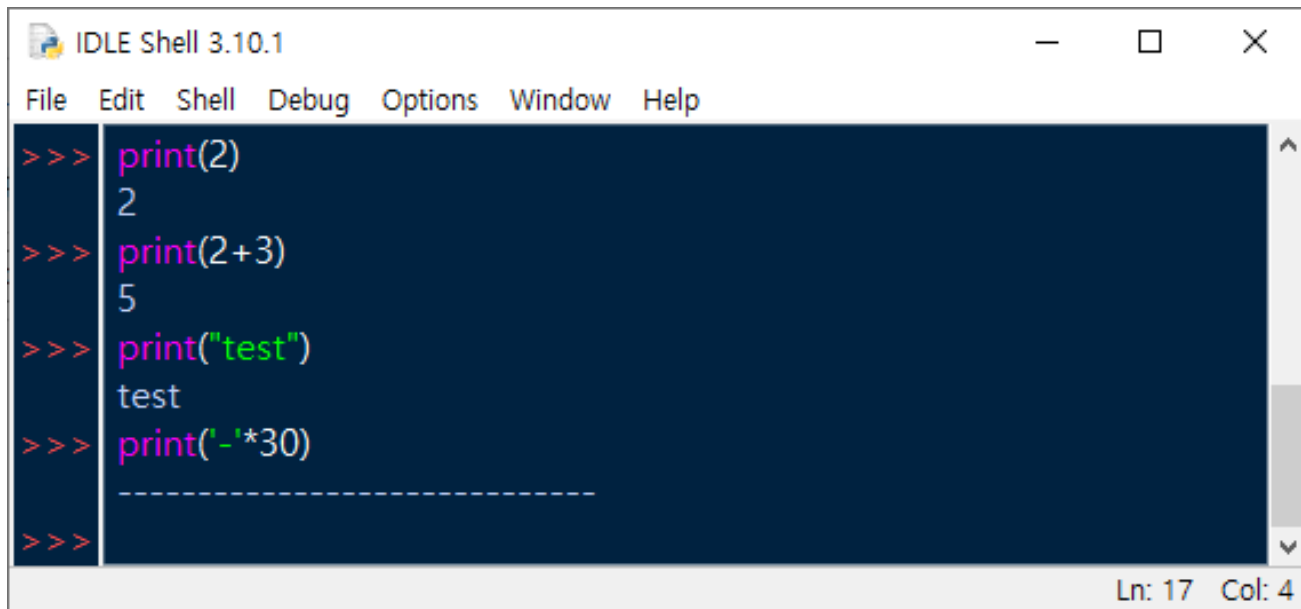


```
IDLE Shell 3.10.1
File Edit Shell Debug Options Window Help
123
>>> 3+4
7
>>> 2**10
1024
>>> test
Traceback (most recent call last):
  File "<pyshell#3>", line 1, in <module>
    test
NameError: name 'test' is not defined
>>> 'test'
'test'
>>> 3.14
3.14
```

## 02. IDLE

### I. 명령어 실행

- 명령어 입력 실행
  - print() 명령어 실행



The screenshot shows the IDLE Shell 3.10.1 window. The title bar reads "IDLE Shell 3.10.1" with standard window controls. The menu bar includes "File", "Edit", "Shell", "Debug", "Options", "Window", and "Help". The main text area has a dark blue background and displays the following commands and their outputs:

```
>>> print(2)
2
>>> print(2+3)
5
>>> print("test")
test
>>> print('-'*30)
-----
>>>
```

The status bar at the bottom right indicates "Ln: 17 Col: 4".

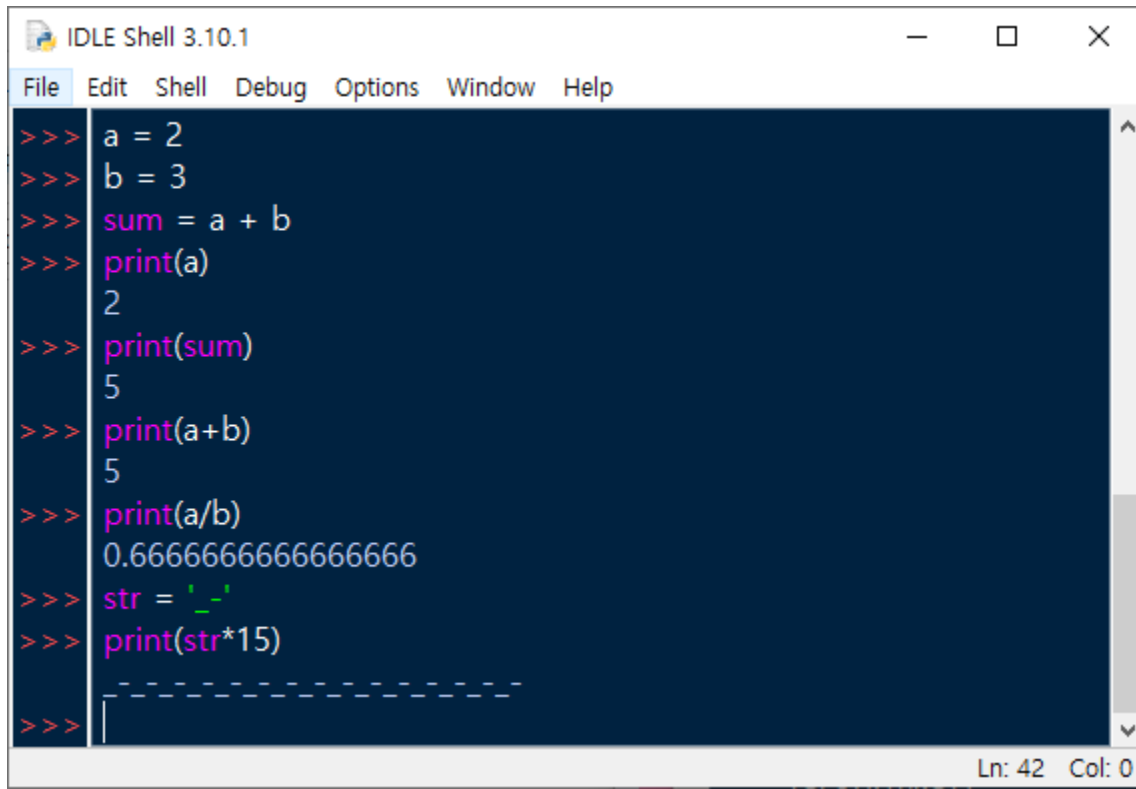


## 02. IDLE

### I. 명령어 실행

#### ■ 변수 사용 실행

- **변수(variable)**는 명령어 실행 과정에서 사용할 값(value)을 일시적으로 저장해 놓는 저장 공간 이름
- 명령어가 해석되는 과정에서 변수명은 해당 값으로 변환되어 실행된다.

A screenshot of the IDLE Shell 3.10.1 window. The window has a menu bar with 'File', 'Edit', 'Shell', 'Debug', 'Options', 'Window', and 'Help'. The main area is a dark blue shell with a light blue prompt '>>>'. The code being executed is: 'a = 2', 'b = 3', 'sum = a + b', 'print(a)' (output: 2), 'print(sum)' (output: 5), 'print(a+b)' (output: 5), 'print(a/b)' (output: 0.6666666666666666), 'str = '-'', and 'print(str\*15)' (output: -----). The status bar at the bottom right shows 'Ln: 42 Col: 0'.

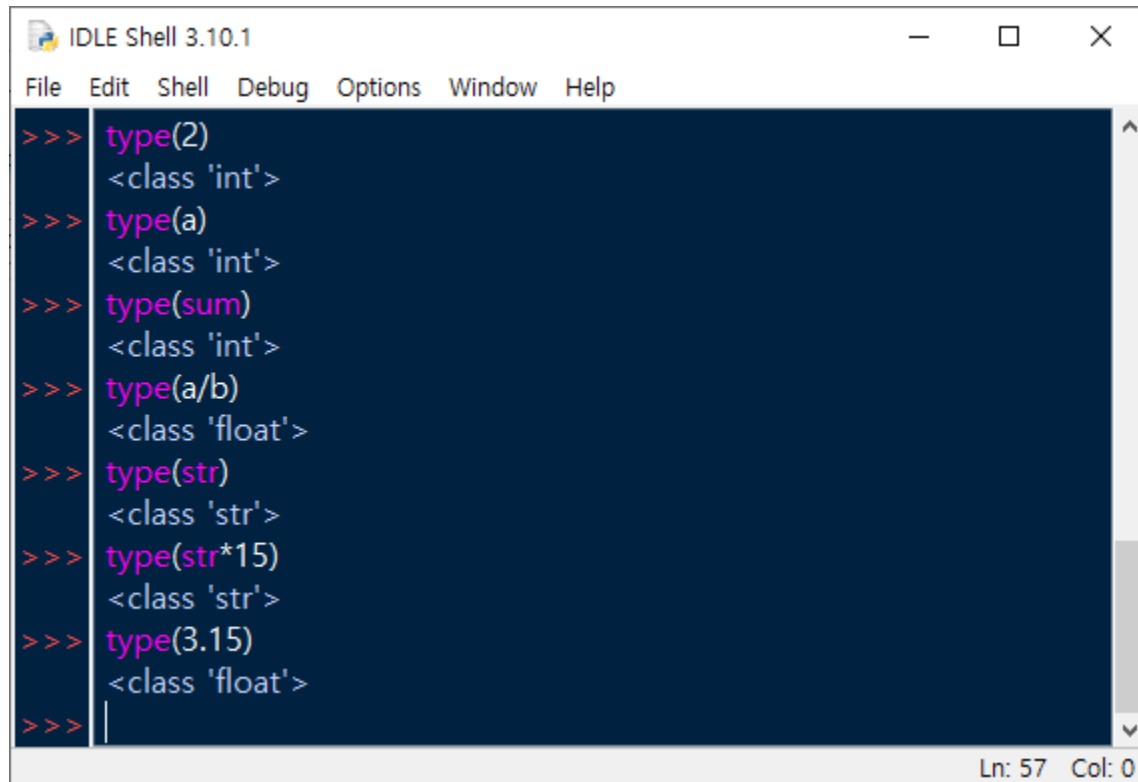
```

IDLE Shell 3.10.1
File Edit Shell Debug Options Window Help
>>> a = 2
>>> b = 3
>>> sum = a + b
>>> print(a)
2
>>> print(sum)
5
>>> print(a+b)
5
>>> print(a/b)
0.6666666666666666
>>> str = '-'
>>> print(str*15)
-----
>>>
Ln: 42 Col: 0
```

## 02. IDLE

### I. 명령어 실행

- 변수에 저장된 값의 형식 확인
  - type() 명령어 사용



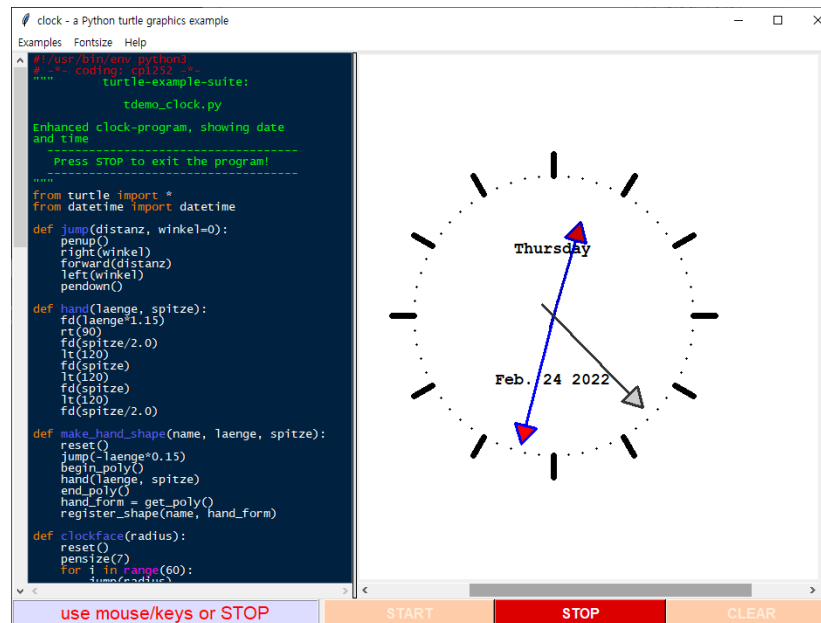
```
IDLE Shell 3.10.1
File Edit Shell Debug Options Window Help
>>> type(2)
<class 'int'>
>>> type(a)
<class 'int'>
>>> type(sum)
<class 'int'>
>>> type(a/b)
<class 'float'>
>>> type(str)
<class 'str'>
>>> type(str*15)
<class 'str'>
>>> type(3.15)
<class 'float'>
>>>
Ln: 57 Col: 0
```

## 02. IDLE

### II. Demo 예제

#### ■ Turtle Demo 예제 실행

- 웰 [File] 메뉴에서 [Help] → [Turtle Demo] 선택
- 새로 생성된 [Turtle Demo] 창에서 [Examples]를 눌러 원하는 예제 선택, 실행

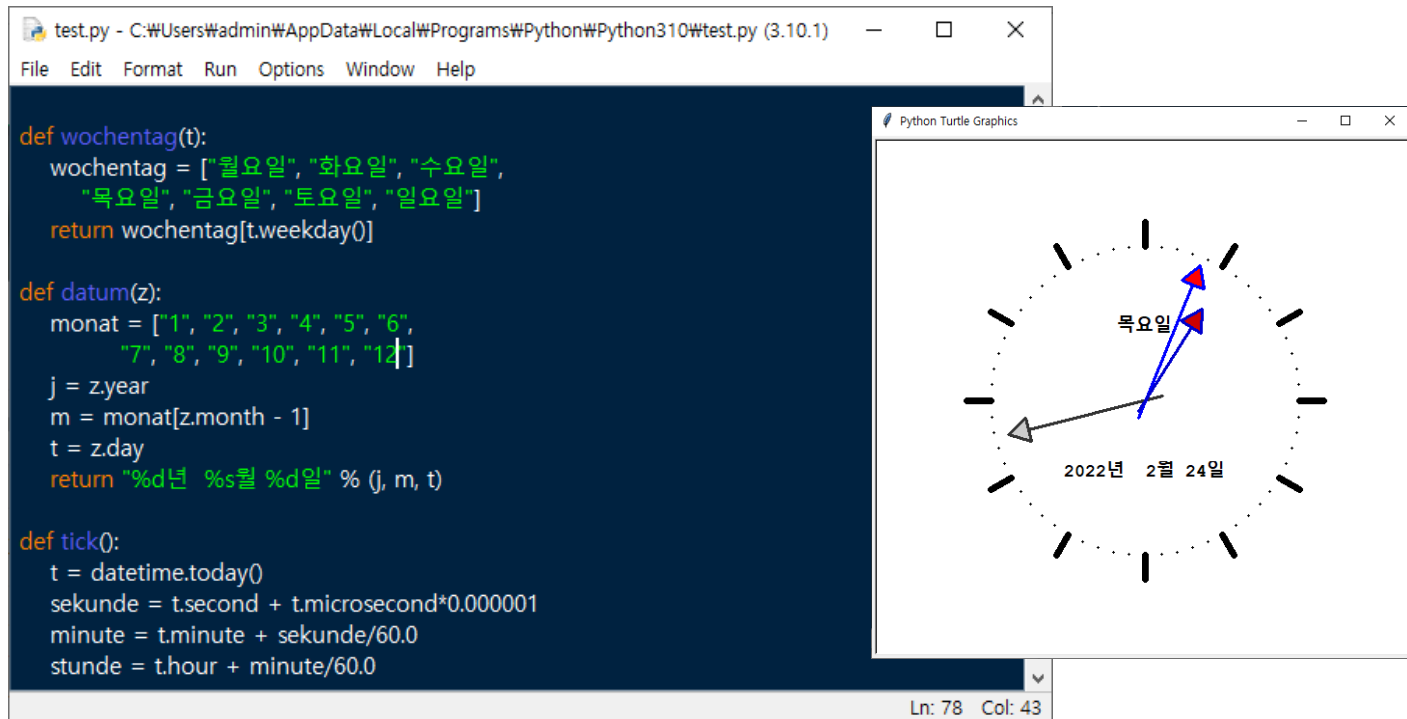


## 02. IDLE

### II. Demo 예제

#### ■ Clock 예제 변경 사용

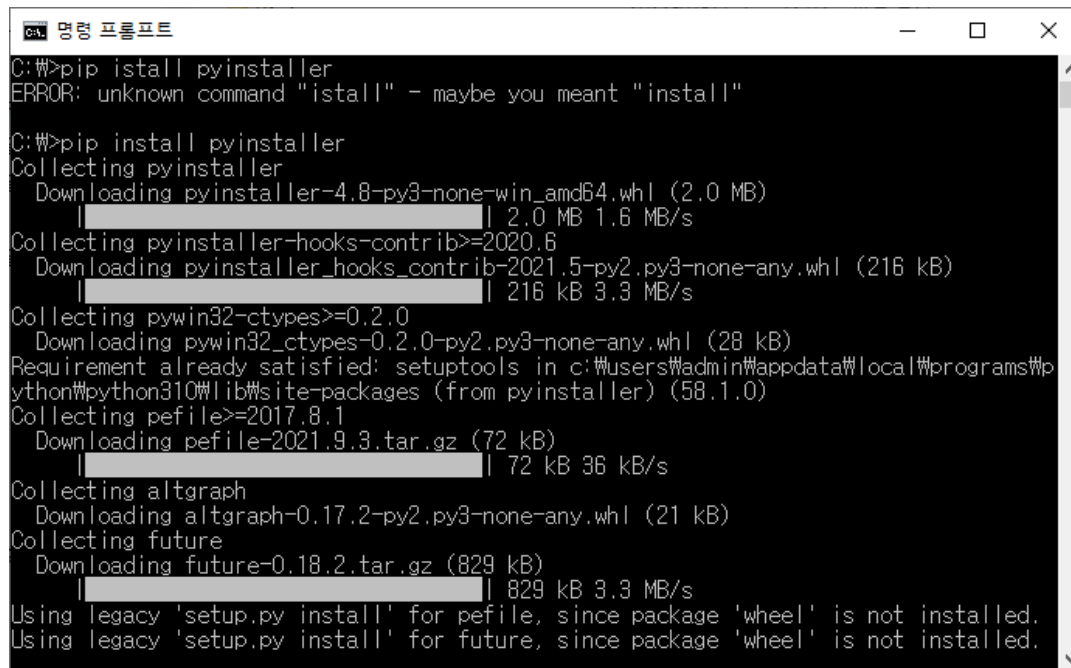
- 쉘 [File] 메뉴에서 [New File] 선택
- clock 예제 코드를 복사하여 new file에 붙여넣기 → 코드 상단의 붉은 2라인은 삭제
- 아래와 같이 초록색 코드 부분으로 변경하여 실행 → [Run] → [Run Module]



## 03. PyInstaller

### I. PyInstaller 설치

- 명령프롬프트다음 명령어를 실행하여 PyInstaller 패키지를 설치  
➤ pip install pyinstaller



```
명령 프롬프트
C:\>pip install pyinstaller
ERROR: unknown command "install" - maybe you meant "install"

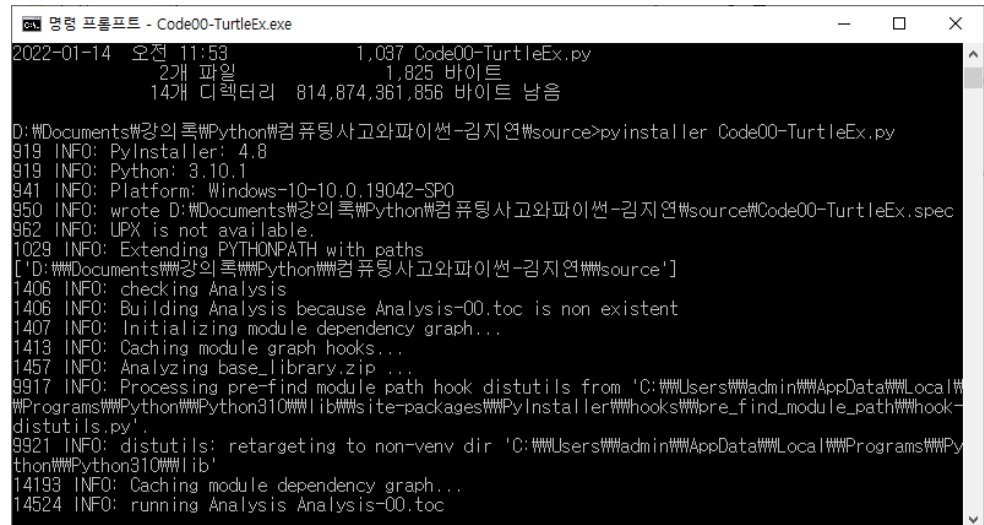
C:\>pip install pyinstaller
Collecting pyinstaller
  Downloading pyinstaller-4.8-py3-none-win_amd64.whl (2.0 MB)
    | 2.0 MB 1.6 MB/s
Collecting pyinstaller-hooks-contrib>=2020.6
  Downloading pyinstaller_hooks_contrib-2021.5-py2.py3-none-any.whl (216 kB)
    | 216 kB 3.3 MB/s
Collecting pywin32-ctypes>=0.2.0
  Downloading pywin32-ctypes-0.2.0-py2.py3-none-any.whl (28 kB)
Requirement already satisfied: setuptools in c:\users\admin\appdata\local\programs\python\python310\lib\site-packages (from pyinstaller) (58.1.0)
Collecting pefile>=2017.8.1
  Downloading pefile-2021.9.3.tar.gz (72 kB)
    | 72 kB 36 kB/s
Collecting altgraph
  Downloading altgraph-0.17.2-py2.py3-none-any.whl (21 kB)
Collecting future
  Downloading future-0.18.2.tar.gz (829 kB)
    | 829 kB 3.3 MB/s
Using legacy 'setup.py install' for pefile, since package 'wheel' is not installed.
Using legacy 'setup.py install' for future, since package 'wheel' is not installed.
```

## 03. PyInstaller

### II. 실행파일 생성(Compile)

❖ `compile`은 소스코드로부터 실행파일을 만드는 과정

- 명령프롬프트에서 \*.py 코드가 있는 디렉터리로 이동
  - `cd D:\Documents\강의록\Python\source` (예)
- 실행파일 생성
  - `pyinstaller Code00-TurtleEx.py` (예)
  - `pyinstaller -w -F Code00-TurtleEx.py` (예)
    - ✓ -w 옵션은 실행 시 명령프롬프트 창 안 뜨게 생성
    - ✓ -F 옵션은 실행파일 한 개만 생성



```
명령 프롬프트 - Code00-TurtleEx.exe
2022-01-14 오전 11:53      1,037 Code00-TurtleEx.py
      2개 파일      1,825 바이트
      14개 디렉터리 814,874,361,856 바이트 남음

D:\Documents\강의록\Python\컴퓨팅사고와파이썬-김지연\source>pyinstaller Code00-TurtleEx.py
919 INFO: PyInstaller: 4.8
919 INFO: Python: 3.10.1
941 INFO: Platform: Windows-10-10.0.19042-SP0
950 INFO: wrote D:\Documents\강의록\Python\컴퓨팅사고와파이썬-김지연\source\Code00-TurtleEx.spec
962 INFO: UPX is not available.
1029 INFO: Extending PYTHONPATH with paths
['D:\Documents\강의록\Python\컴퓨팅사고와파이썬-김지연\source']
1406 INFO: checking Analysis
1406 INFO: Building Analysis because Analysis-00.toc is non-existent
1407 INFO: Initializing module dependency graph...
1413 INFO: Caching module graph hooks...
1457 INFO: Analyzing base_library.zip ...
9917 INFO: Processing pre-find module path hook distutils from 'C:\Users\admin\AppData\Local\Programs\Python\Python310\lib\site-packages\PyInstaller\hooks\pre_find_module_path_hook-distutils.py'.
9921 INFO: distutils: retargeting to non-venv dir 'C:\Users\admin\AppData\Local\Programs\Python\Python310\lib'
14193 INFO: Caching module dependency graph...
14524 INFO: running Analysis Analysis-00.toc
```

## 03. PyInstaller

### III. 실행파일 확인 및 실행

- 실행파일이 생성된 디렉터리인 dist로 이동
  - cd dist
- 실행파일 실행
  - Code00-TurtleEx.exe (예)

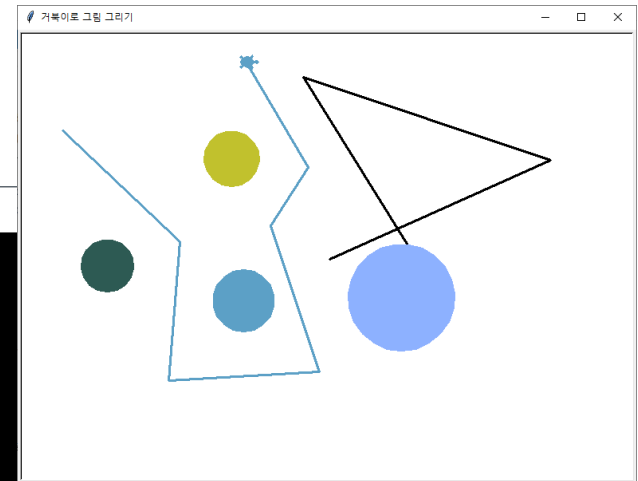
```
명령 프롬프트 - Code00-TurtleEx.exe
D:\Documents\강의록\Python\컴퓨팅사고와파이썬-김지연\source>cd dist
D:\Documents\강의록\Python\컴퓨팅사고와파이썬-김지연\source\dist>dir
D 드라이브의 볼륨: Samsung_T5
볼륨 일련 번호: EE5D-2765

D:\Documents\강의록\Python\컴퓨팅사고와파이썬-김지연\source\dist 디렉터리

2022-01-15 오후 01:32 <DIR>          .
2022-01-15 오후 01:32 <DIR>          ..
2022-01-15 오후 01:32 <DIR>          Code00-TurtleEx
               0개 파일              0 바이트
               3개 디렉터리  814,718,910,464 바이트 남음

D:\Documents\강의록\Python\컴퓨팅사고와파이썬-김지연\source\dist>Code00-TurtleEx.exe
'Code00-TurtleEx.exe'은(는) 내부 또는 외부 명령, 실행할 수 있는 프로그램, 또는
배치 파일이 아닙니다.

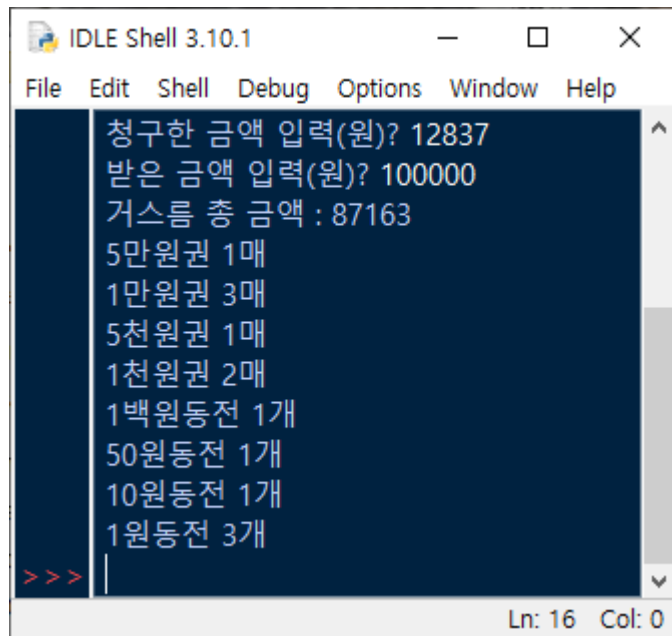
D:\Documents\강의록\Python\컴퓨팅사고와파이썬-김지연\source\dist>cd Code00-TurtleEx
D:\Documents\강의록\Python\컴퓨팅사고와파이썬-김지연\source\dist\Code00-TurtleEx>Code00-TurtleEx.exe
```



## 04. 어디까지 해보나

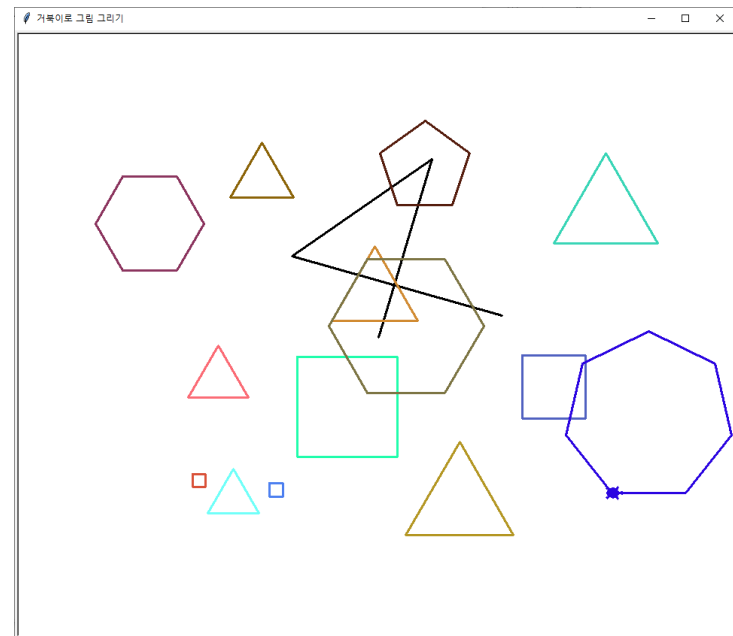
### I. 각 단원별 코딩

- Ch04 연산
  - 거스름 돈 최소 개수로 반환
  - 다각형 그리기 [Turtle Module 활용]



```
IDLE Shell 3.10.1
File Edit Shell Debug Options Window Help
청구한 금액 입력(원)? 12837
받은 금액 입력(원)? 100000
거스름 총 금액 : 87163
5만원권 1매
1만원권 3매
5천원권 1매
1천원권 2매
1백원동전 1개
50원동전 1개
10원동전 1개
1원동전 3개
>>> |
```

Ln: 16 Col: 0



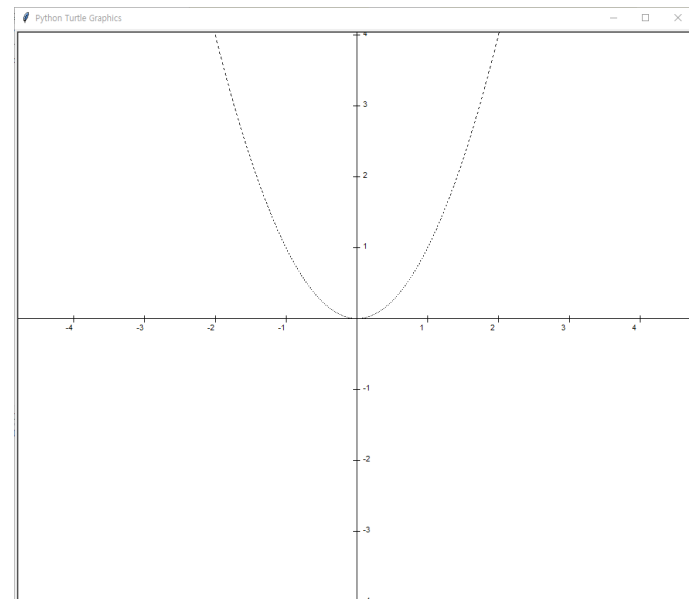


## 04. 어디까지 해보나

### I. 각 단원별 코딩

- Ch05 조건
  - 점수에 대한 등급 판단
  - 2차함수 그래프 그리기 [Turtle Module 활용]

```
D:\Documents\강의록\Pyt...
>점수 입력 : 100
>>100의 등급은 A
>점수 입력 : 97
>>97의 등급은 A+
>점수 입력 : 90
>>90의 등급은 A
>점수 입력 : 88
>>88의 등급은 B+
>점수 입력 : 59
>>59의 등급은 F+
>점수 입력 : 0
>>0의 등급은 F
>점수 입력 : 101
>>101는 범위 외를 초과한 값입니다.
>점수 입력 : -10
>>-10는 범위 외를 초과한 값입니다.
>점수 입력 :
```

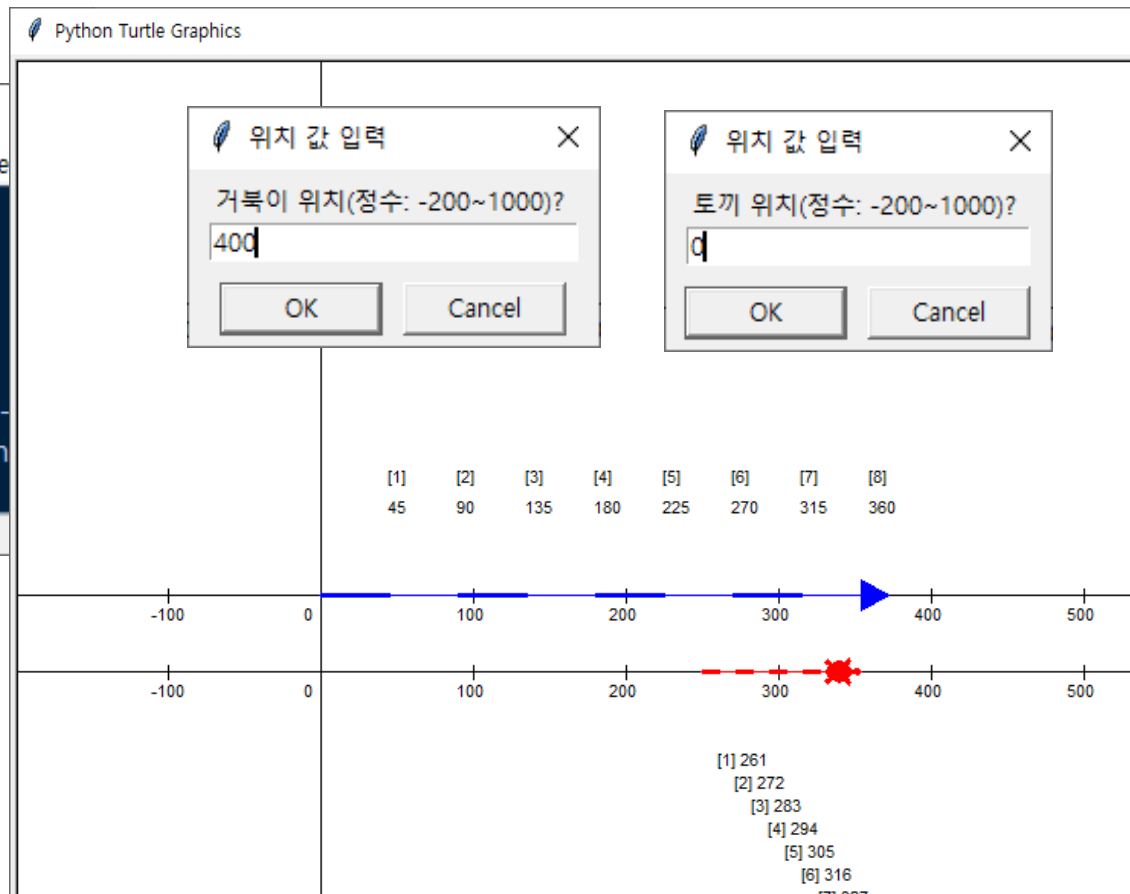


## 04. 어디까지 해보나

### I. 각 단원별 코딩

- Ch06 반복
  - 토끼와 거북이 경주
  - 토끼와 거북이 경주 [Turtle Module 활용]

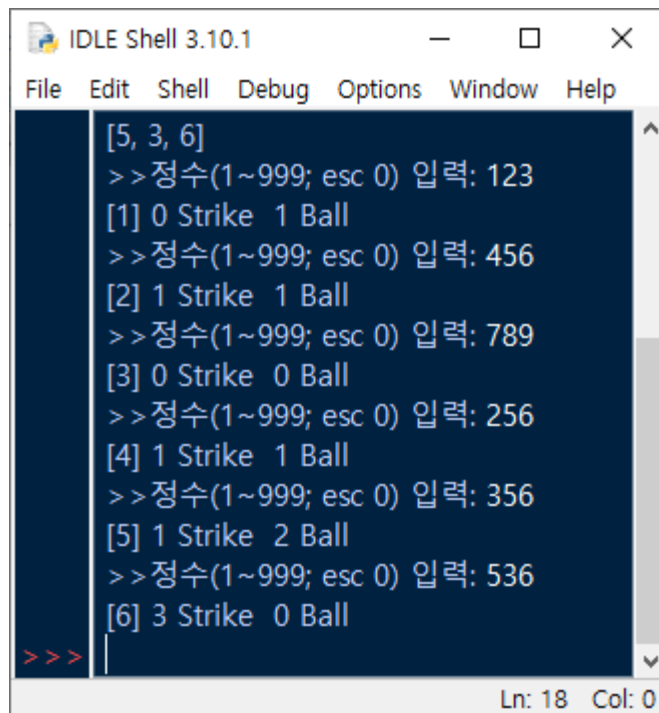
```
IDLE Shell 3.10.1
File Edit Shell Debug Options Window Help
>26분 r: 1170m, t: 1286m
>27분 r: 1215m, t: 1297m
>28분 r: 1260m, t: 1308m
>29분 r: 1305m, t: 1319m
>30분 r: 1350m, t: 1330m
-----
>30분 후에 1330m에 거북이, 1350m
>>>
```



## 04. 어디까지 해보나

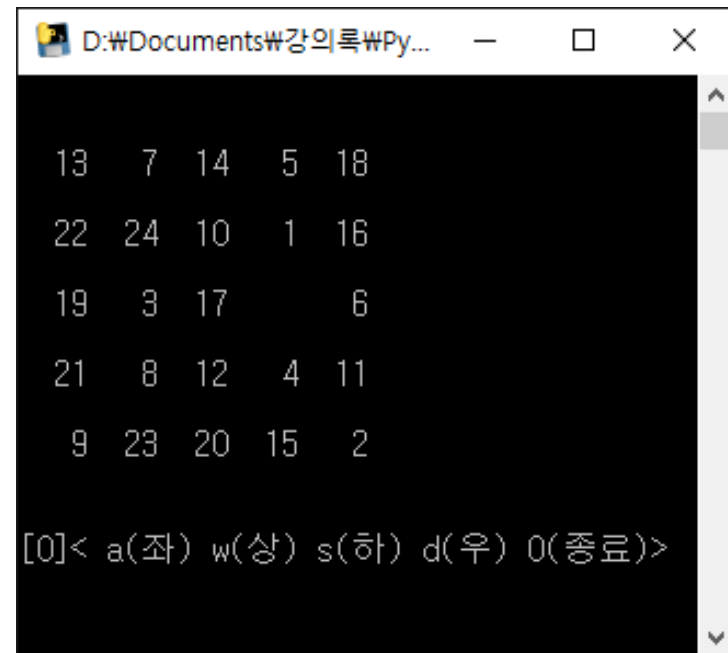
### I. 각 단원별 코딩

- Ch07~Ch08 리스트, 튜플, 딕셔너리
  - 숫자 야구 게임



```
IDLE Shell 3.10.1
File Edit Shell Debug Options Window Help
[5, 3, 6]
>>정수(1~999; esc 0) 입력: 123
[1] 0 Strike 1 Ball
>>정수(1~999; esc 0) 입력: 456
[2] 1 Strike 1 Ball
>>정수(1~999; esc 0) 입력: 789
[3] 0 Strike 0 Ball
>>정수(1~999; esc 0) 입력: 256
[4] 1 Strike 1 Ball
>>정수(1~999; esc 0) 입력: 356
[5] 1 Strike 2 Ball
>>정수(1~999; esc 0) 입력: 536
[6] 3 Strike 0 Ball
>>>
```

Ln: 18 Col: 0



```
D:\Documents\강의록\Py...
13  7  14  5  18
22 24 10  1  16
19  3 17   6
21  8 12  4  11
 9 23 20 15  2

[0]< a(좌) w(상) s(하) d(우) 0(종료)>
```

## 04. 어디까지 해보나

### I. 각 단원별 코딩

- Ch10 문자열과 파일
  - 성적 관리

성적처리 프로그램

코드열람 현황 파일로 저장

메뉴

검색

학생검색

학번: 2230001 검색

✓ 검색 결과 유지

학번	이름	학과
2230001	이선우	컴퓨터과(30)

끝내기 뒤로가기

## 04. 어디까지 해보나

### I. 각 단원별 코딩

- Ch11 그래픽 프로그래밍
  - Café 주문 시스템

부천대학 컴퓨터소프트웨어과 Dreammer Cafe

검색 주문내역 매출\_통계 매출\_차트

COFFEE	에스프레소 3.0	아메리카노 3.0	카페라떼 4.0	카프치노 4.0
LATTE	말차라떼 4.0	초코라떼 4.0	카페라떼 4.0	
TEA	청귤차 4.0	자몽차 4.0	레몬차 4.0	카모마일 4.5
ADE	자몽에이드 4.5	레몬에이드 4.5	청포도에이드 4.5	
JUICE	망고 4.5	바나나 4.5	딸기 4.5	키위 4.5
SMOOTHIE	청귤스무디 4.5	요거트스무디 4.5		
MILK_TEA	흑당밀크티 4.5	달고나밀크티 4.5		

주문 수정

[ 1 ]	에스프레소	3.0	1	3000.0
[ 2 ]	말차라떼	4.0	1	4000.0
[ 3 ]	초코라떼	4.0	2	8000.0
[ 4 ]	아메리카노	3.0	3	9000.0

주문 합계 : 7잔 24000.0원

[취소]재시작 주문 확정

# Thank You !

[Python]