

» 학습목표

- 파이썬 프로그램을 설치하고 실행을 하여 간단한 프로그램을 만들고, 프로그램의 기본적인 연산과 변수, for 명령을 사용하여 똑 같은 작업을 반복하게 실행하는 프로그램을 구현 할 수 있어야 한다.

» Unit 1

파이썬 설치하고 실행하기

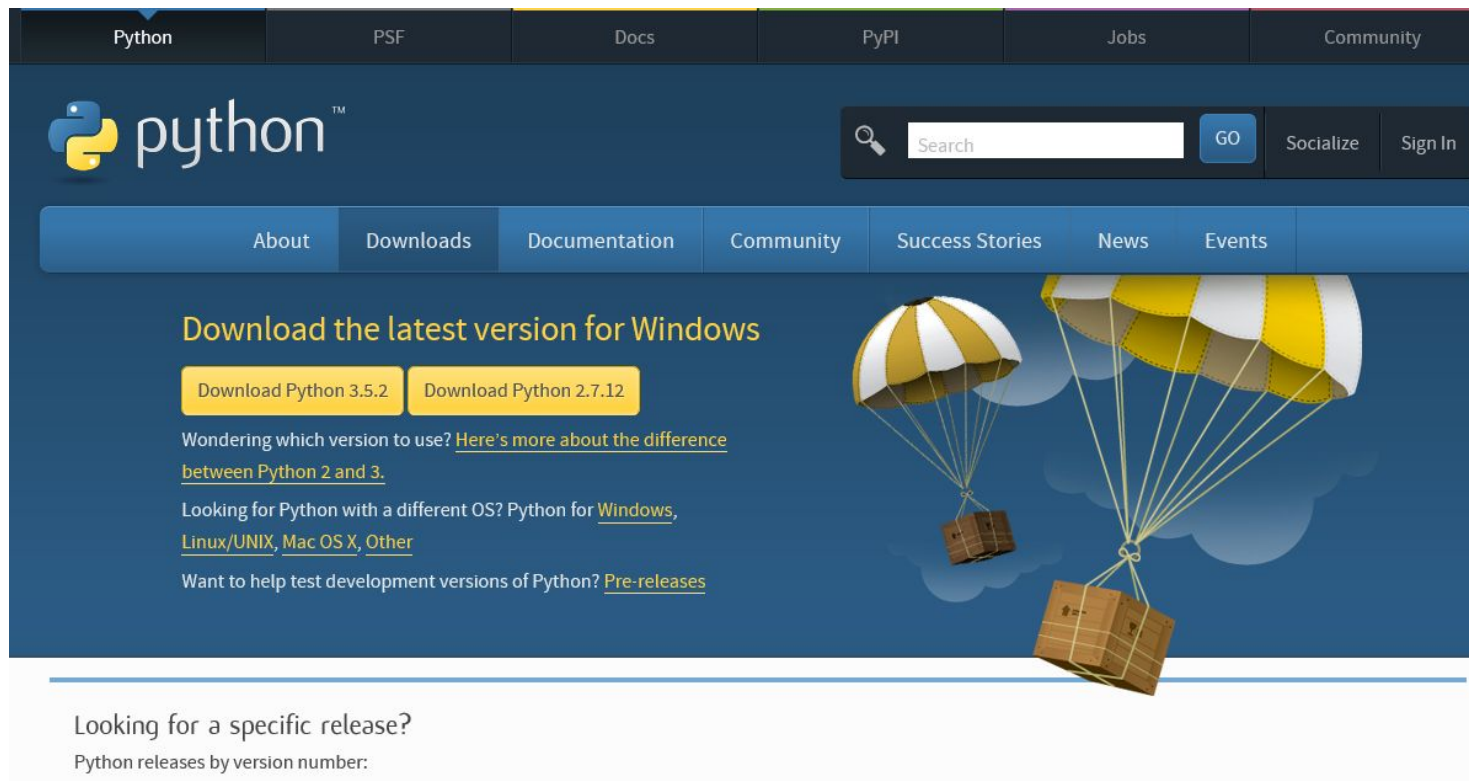
Unit 1. 파이썬 설치하고 실행하기

- 01** 파이썬 설치하기
- 02** 대화형 셸이 뭔가요?
- 03** 거북이 그래픽 모듈 사용하기

슬라이드 1. 파이썬 설치하기

» <http://python.org/dpwnload> 입력

- Download Python 3.5.2 버튼 누름



1. 파이썬 설치하기

» 안정성에 대한 경고와 실행 여부를 묻는 창이 뜨면 실행 버튼을 누름

The screenshot shows the Python.org website's download page for Windows. The navigation bar includes links for About, Downloads, Documentation, Community, Success Stories, News, and Events. The main heading is "Download the latest version for Windows". Below this are two buttons: "Download Python 3.5.2" and "Download Python 2.7.12". There is also a link to "Here's more about the difference between Python 2 and 3." and links for other operating systems like Linux, Mac OS X, and Other. A section titled "Looking for a specific release?" lists Python releases by version number and date. At the bottom, a Windows security warning dialog is displayed, asking if the user wants to run the file "python-3.5.2.exe (27.9MB)".

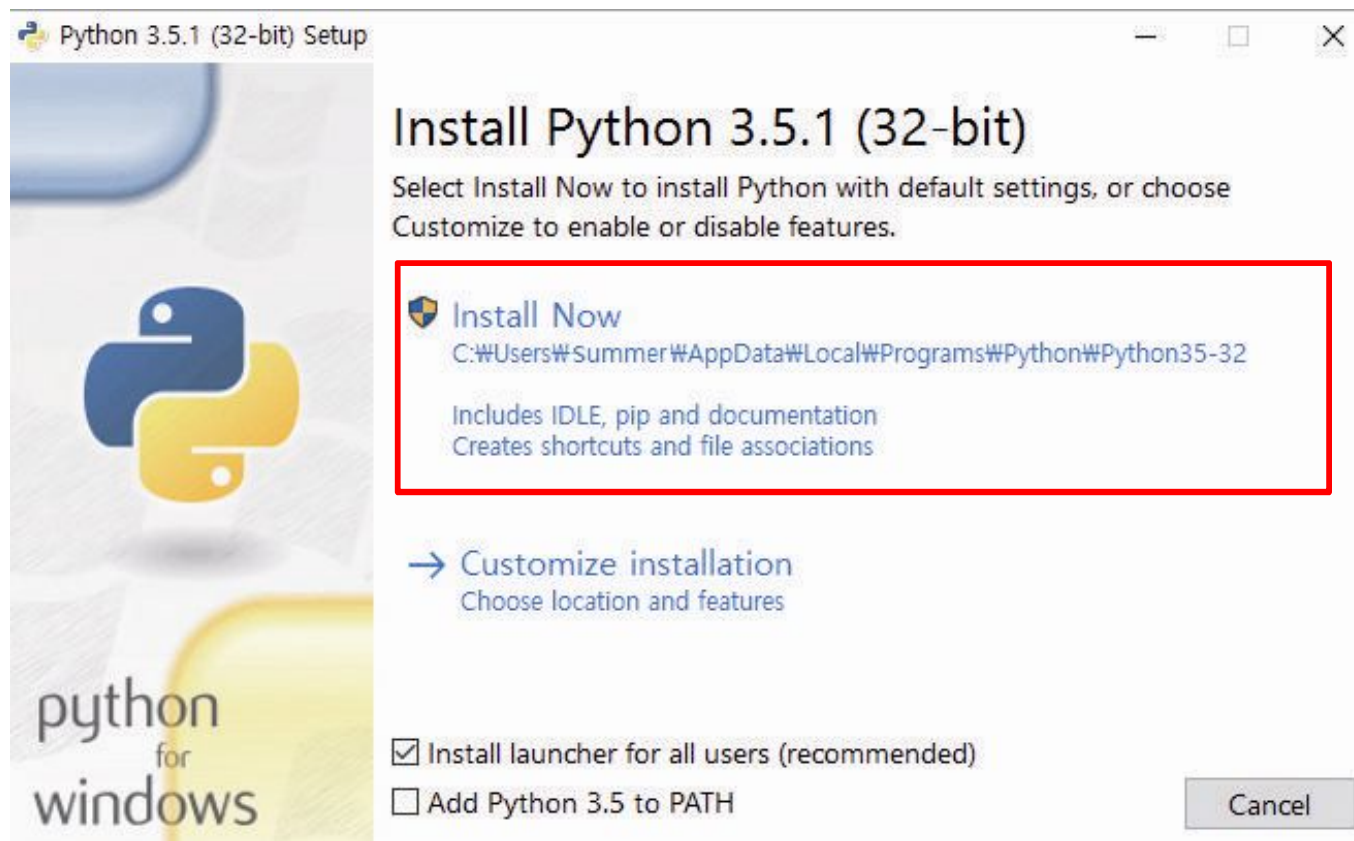
Release version	Release date	Click for more
Python 3.4.5	2016-06-27	Download, Release Notes
Python 3.5.2	2016-06-27	Download, Release Notes
Python 2.7.12	2016-06-25	Download, Release Notes
Python 3.4.4	2015-12-21	Download, Release Notes
Python 3.5.1	2015-12-07	Download, Release Notes
Python 2.7.11	2015-12-05	Download, Release Notes
Python 3.5		

python.org의 python-3.5.2.exe(27.9MB)를(를) 실행하거나 저장하시겠습니까?

실행(R) 저장(S) 취소(C) ×

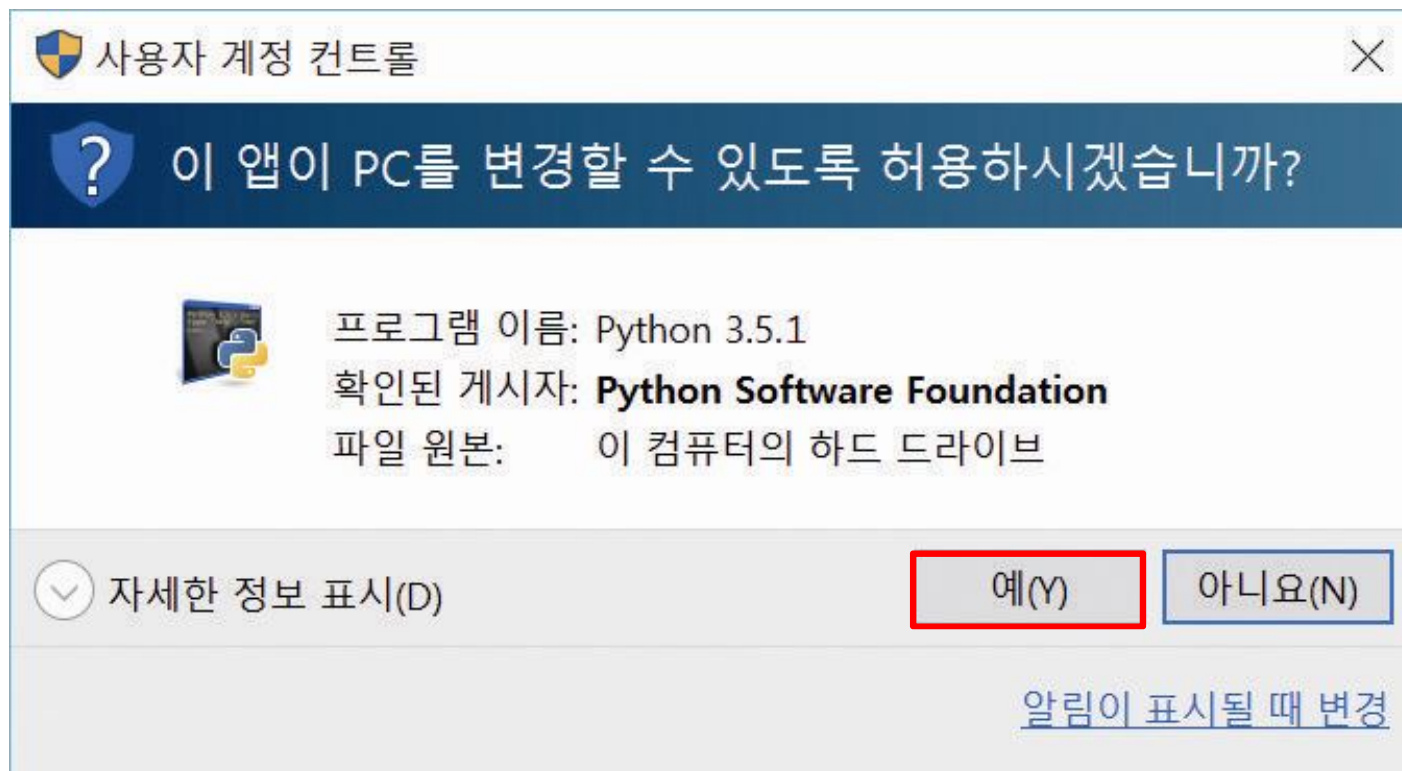
1. 파이썬 설치하기

» Install Now 을 눌러 설치 시작



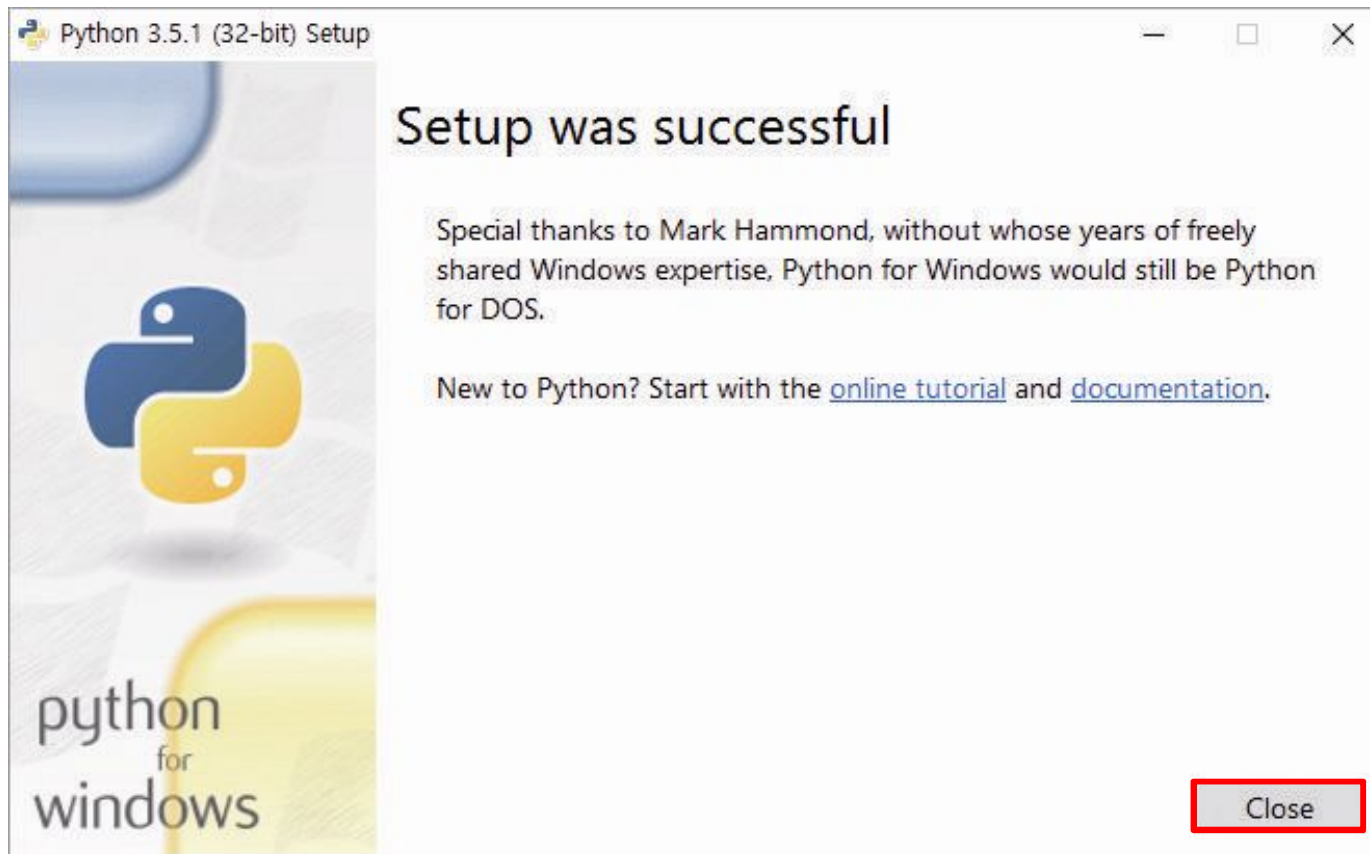
1. 파이썬 설치하기

» 사용자 계정 컨트롤 창에 보안 경고가 뜨면 예(Y) 버튼을 누름



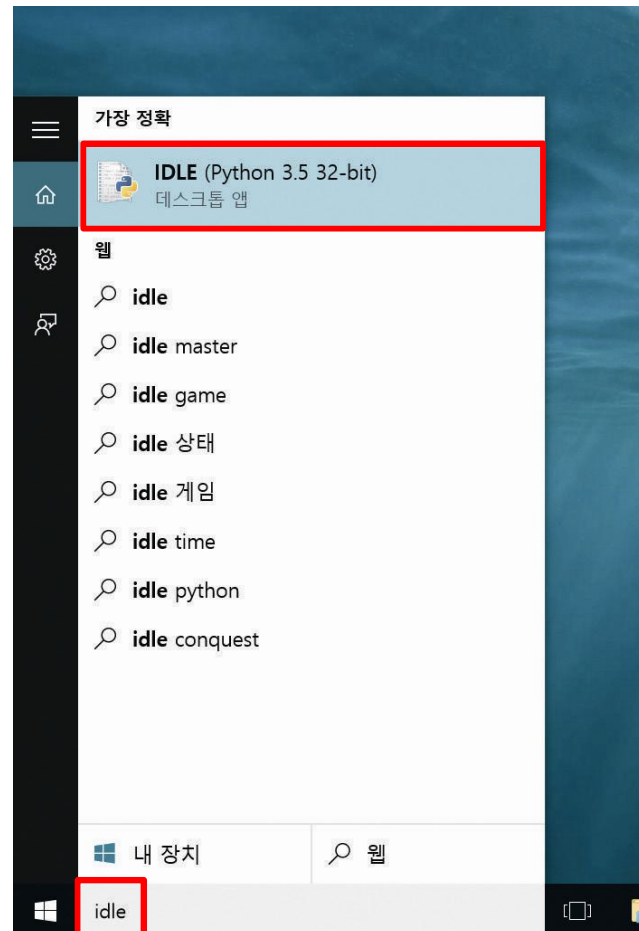
1. 파이썬 설치하기

» 설치 마법사 창에 Setup was successful 이 나타나면 닫음



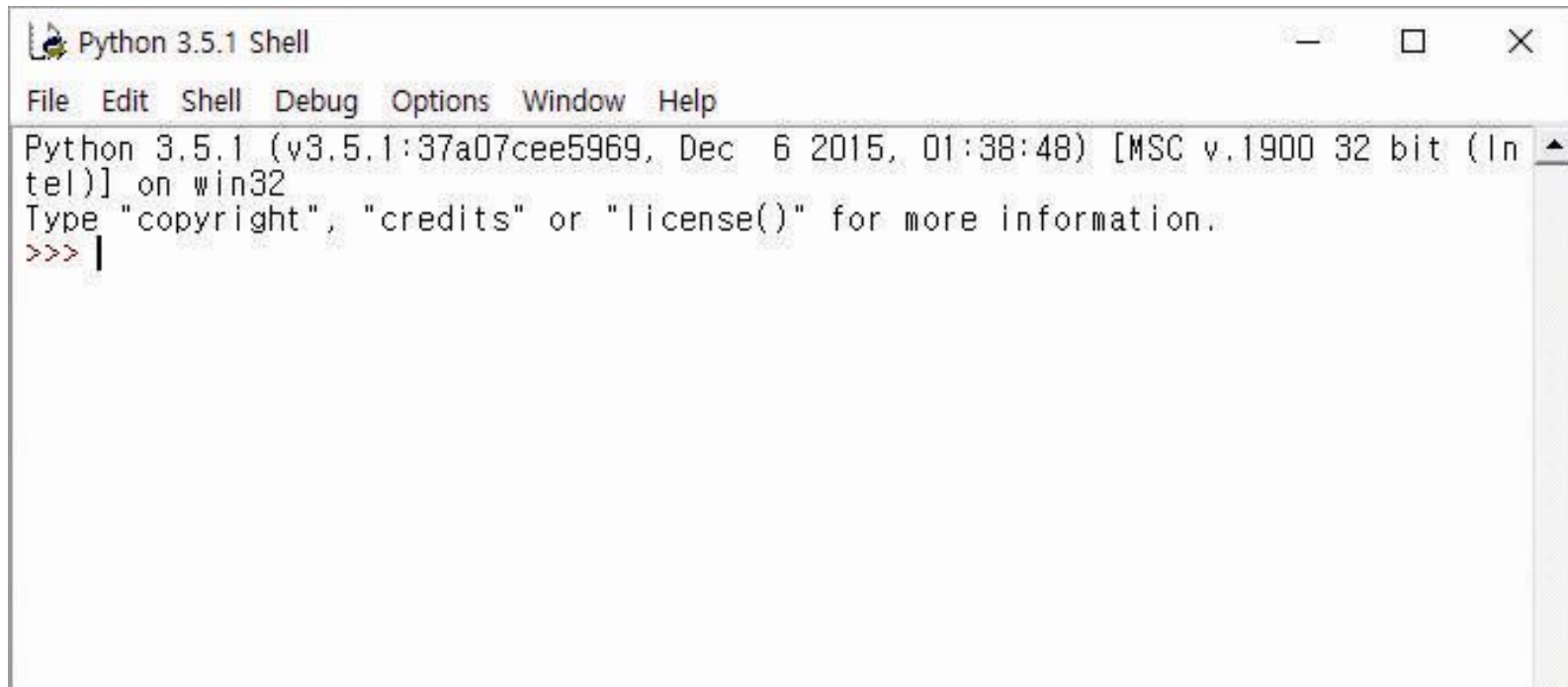
1. 파이썬 설치하기

» 왼쪽 맨 아래의 '웹 및 Windows 검색' 부분에 idle 입력 후 실행



1. 파이썬 설치하기

» IDLE 프로그램이 화면에 나타남

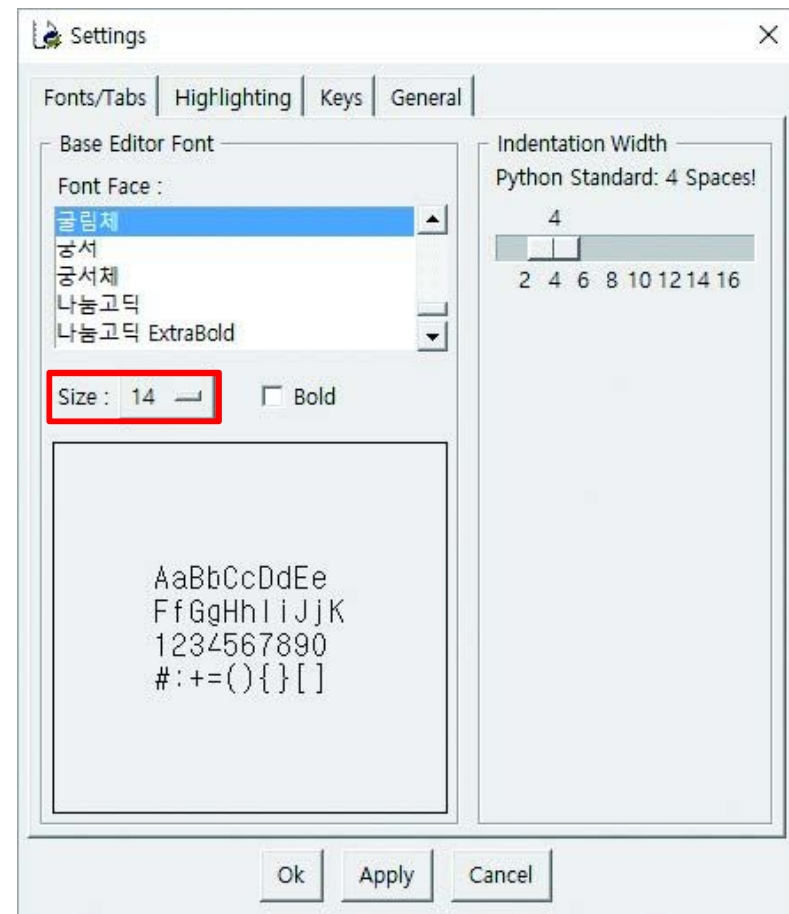
A screenshot of the Python 3.5.1 Shell window. The window has a title bar that says "Python 3.5.1 Shell" and standard Windows window controls (minimize, maximize, close). Below the title bar is a menu bar with the following items: File, Edit, Shell, Debug, Options, Window, and Help. The main area of the window contains 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.
>>> |
```

1. 파이썬 설치하기

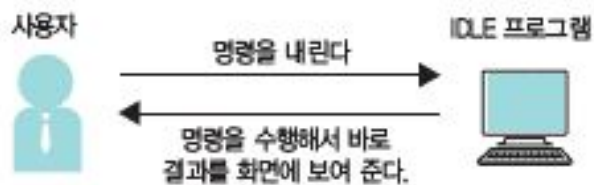
» IDLE 화면에 보이는 글자 크기를 키울 수 있나요?

IDLE 화면에 보이는 글자 크기가 너무 작다고요? 걱정 마세요. IDLE에서는 글자 크기를 줄이거나 늘릴 수 있습니다. 프로그램 위쪽에 나타나는 메뉴에서 Options을 선택하고 Configure IDLE을 선택하면 Settings(설정) 창이 나타납니다. 화면 왼쪽 중간에 있는 Size(크기) 값을 바꿔 글자크기를 조정할 수 있습니다. 숫자를 적당히 키워 보기 좋게 조정합니다(화면 보기와 해상도에 따라 다르지만, 보통 11~14 정도가 적당합니다).



슬라이드 2. 대화형 셸이 뭔가요?

» 대화형 셸 : 사용자가 입력을 하면 바로 결과를 보여줌



- IDLE 프로그램 실행 후 ‘작업 표시줄에 고정’

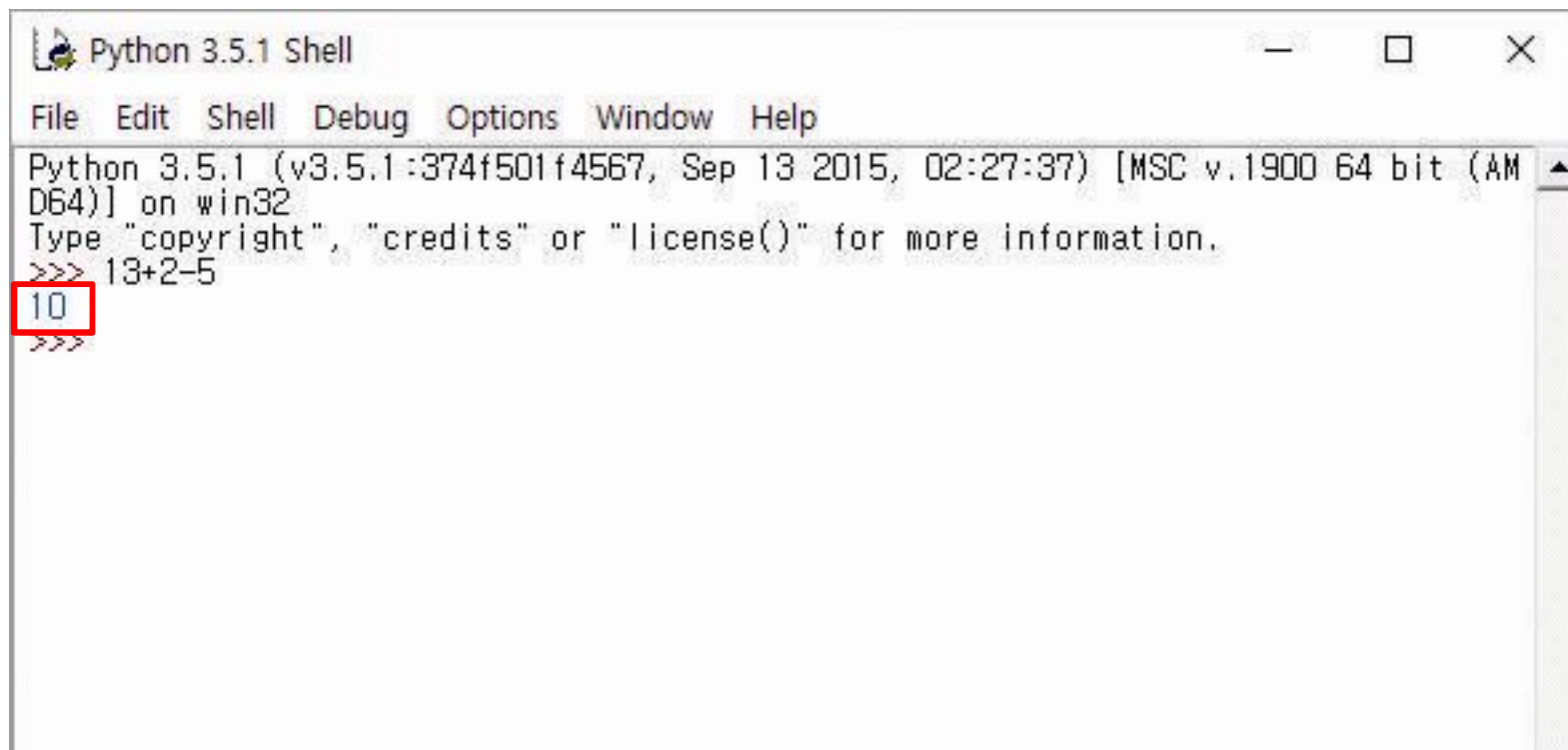


» >>> 기호 뒤에 13+2-5를 입력하고 Enter를 누름

```
>>> 13+2-5
```

2. 대화형 셸이 뭔가요?

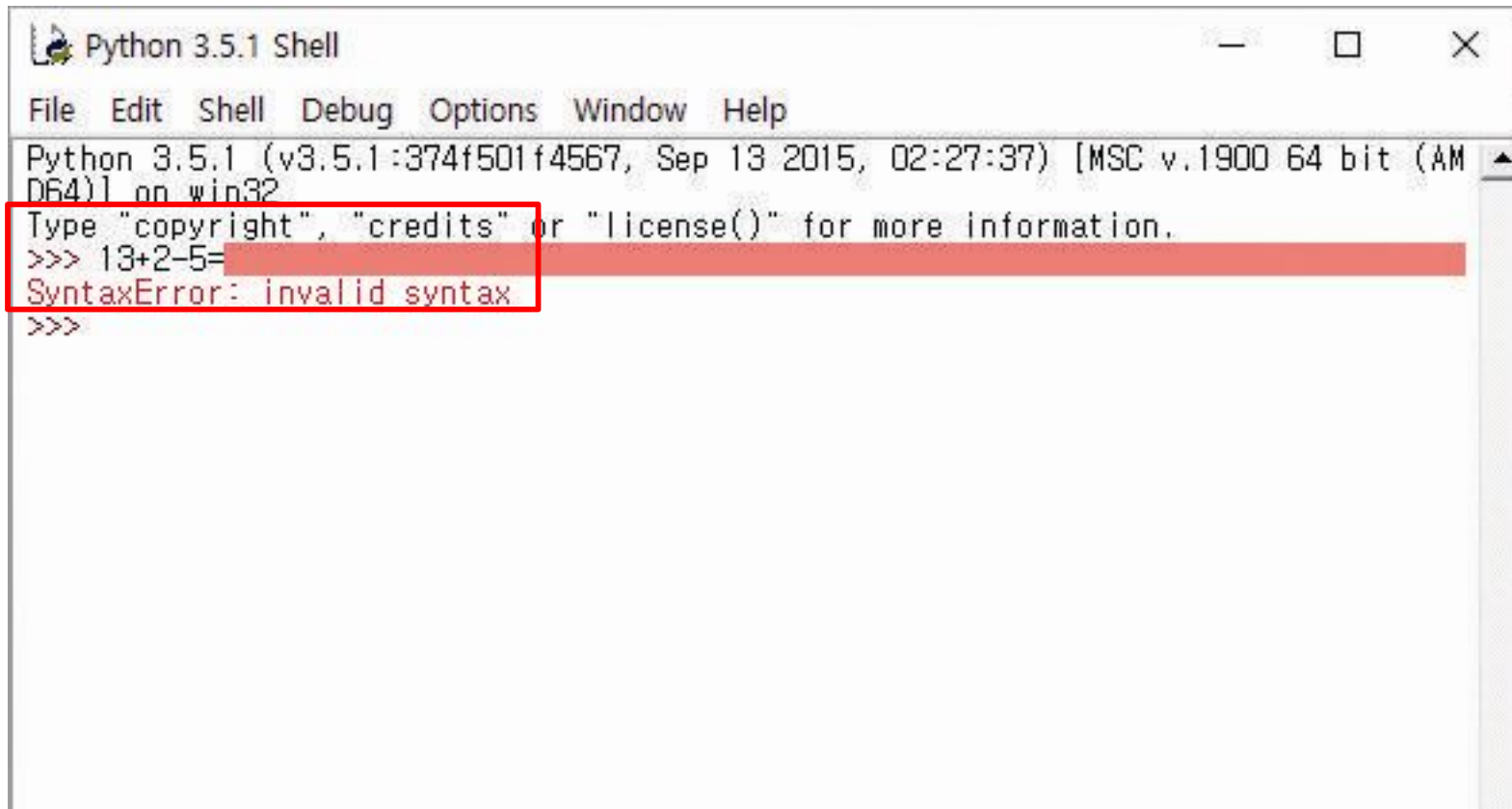
» 파이썬이 $13+2-5$ 를 계산한 후 결과를 보여줌



```
Python 3.5.1 Shell
File Edit Shell Debug Options Window Help
Python 3.5.1 (v3.5.1:374f501f4567, Sep 13 2015, 02:27:37) [MSC v.1900 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> 13+2-5
10
>>>
```

2. 대화형 셸이 뭔가요?

- » 실수로 등호(=)를 포함하여 $13+2-5=$ 으로 입력하면
SyntaxError: invalid syntax라는 에러메시지가 나타남



```
Python 3.5.1 Shell
File Edit Shell Debug Options Window Help
Python 3.5.1 (v3.5.1:374f501f4567, Sep 13 2015, 02:27:37) [MSC v.1900 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> 13+2-5=
SyntaxError: invalid syntax
>>>
```

2. 대화형 셸이 뭔가요?

» 에러가 뭐예요?

컴퓨터 프로그램을 만들다 보면 프로그램이 제대로 작동하지 않을 때가 있습니다. 컴퓨터에서는 이런 문제를 '에러(Error)'라고 부릅니다. 파이썬 프로그램에서 에러가 발생하면 에러가 왜 났는지, 어느 부분에서 났는지 등의 정보를 화면에 표시해 주는데 이것을 '에러 메시지'라고 부릅니다.

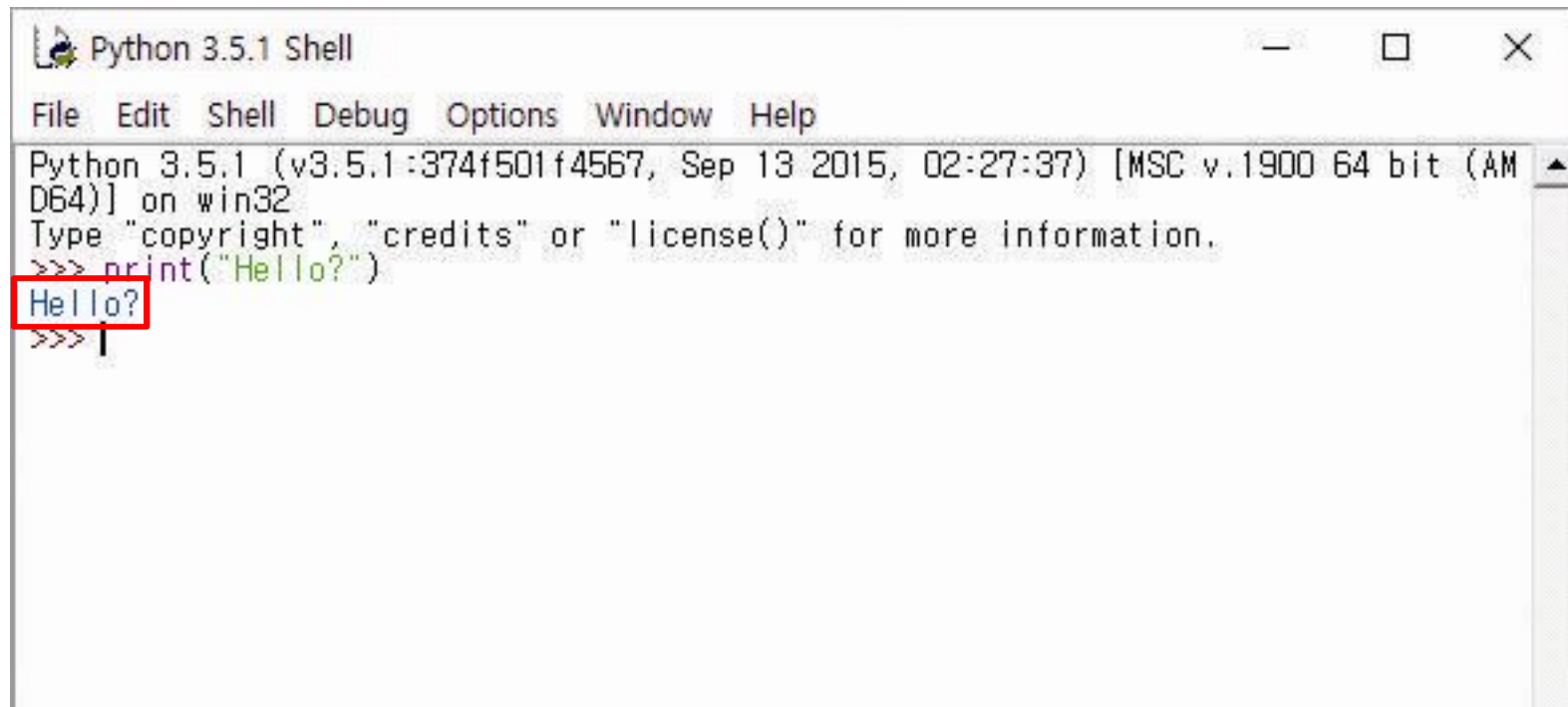
» 에러 해결 방법



2. 대화형 셸이 뭔가요?

» 인사말을 출력하는 프로그램

```
>>> print("Hello?")
```



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

```
Python 3.5.1 (v3.5.1:374f501f4567, Sep 13 2015, 02:27:37) [MSC v.1900 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> print("Hello?")
Hello?
>>> |
```

The output "Hello?" is highlighted with a red rectangular box.

슬라이드 3. 거북이 그래픽 모듈 사용하기

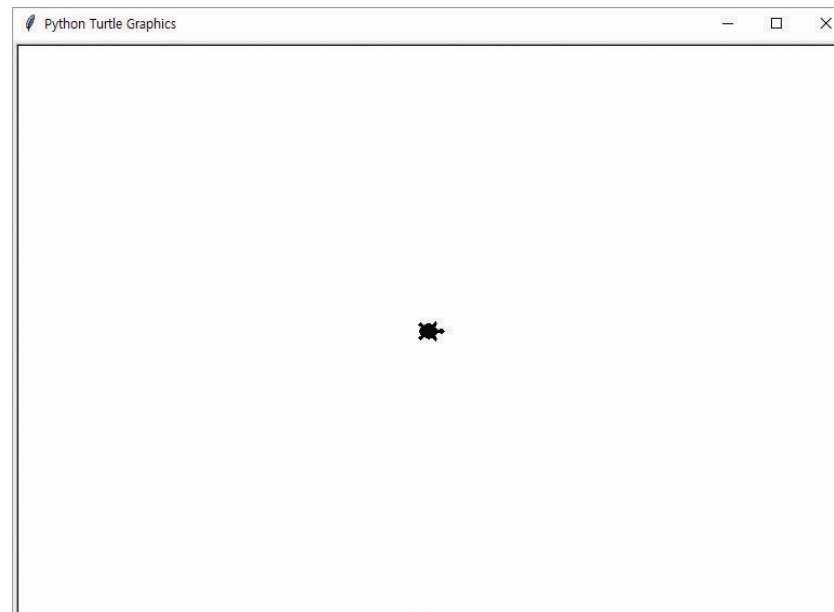
- » 거북이 그래픽을 시작하면 하얀 종이(거북이 그래픽 창) 가운데에 거북이가 등장
- » 거북이에게 명령을 내리면 거북이는 명령에 따라 종이(거북이 그래픽 창) 위를 움직임
- » 거북이가 움직이는 대로 그림이 그려짐

3. 거북이 그래픽 모듈 사용하기

» IDLE 프로그램을 열고 다음과 같이 입력한 후 Enter를 누름

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

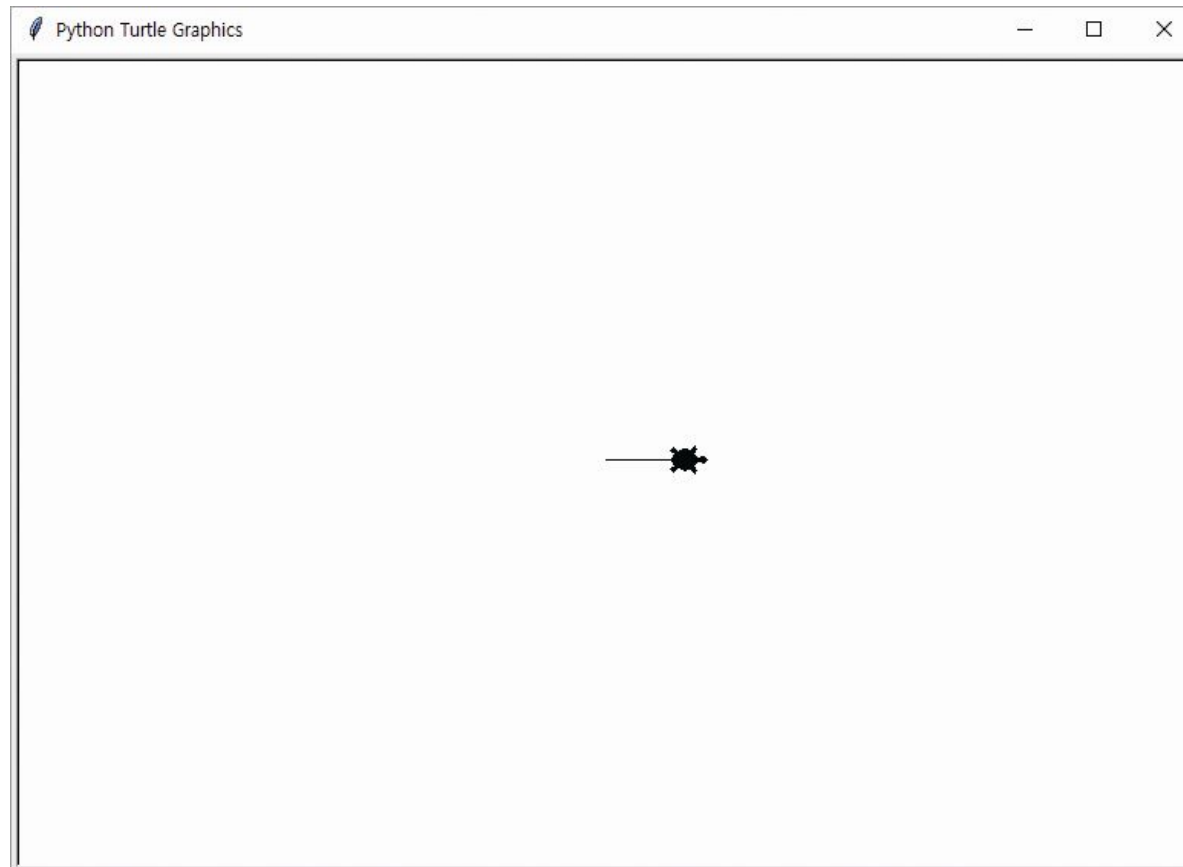
» 거북이 그래픽 창이 뜨고, 창 한가운데에 오른쪽을 바라보는 거북이가 나타남



3. 거북이 그래픽 모듈 사용하기

» t.forward(50)을 입력후 Enter. 거북이가 50픽셀만큼 앞으로 이동

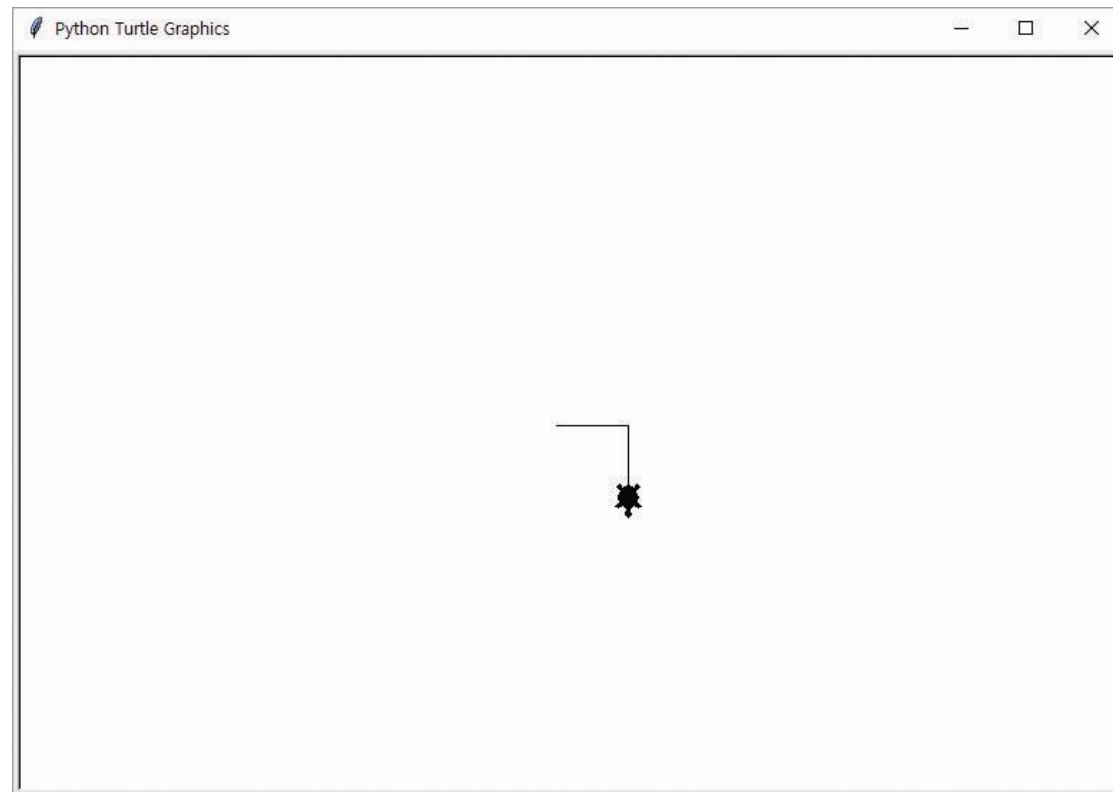
```
>>> t.forward(50)
```



3. 거북이 그래픽 모듈 사용하기

- » `t.right(90)` 입력 시 거북이가 오른쪽으로 90°(직각) 회전
- » 다시 `t.forward(50)` 입력 시 거북이가 앞으로 50픽셀만큼 이동하면서 기억(┐)자를 그림

```
>>> t.right(90)
>>> t.forward(50)
```



» Unit 2

간단한 프로그램 만들기

간단한 프로그램 만들기

- 01** Hello!를 두 번 출력하는 프로그램
- 02** 거북이 그래픽으로 도형을 그리는 프로그램

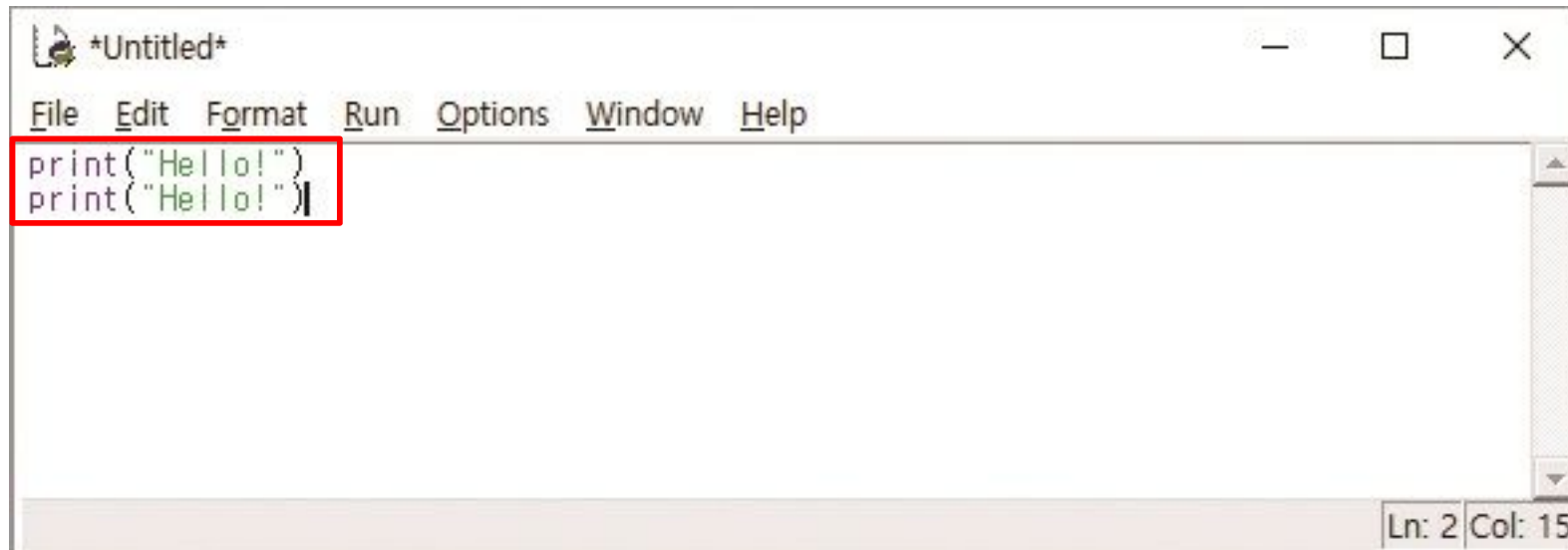
01

슬라이드 1.

Hello!를 두 번 출력하는 프로그램

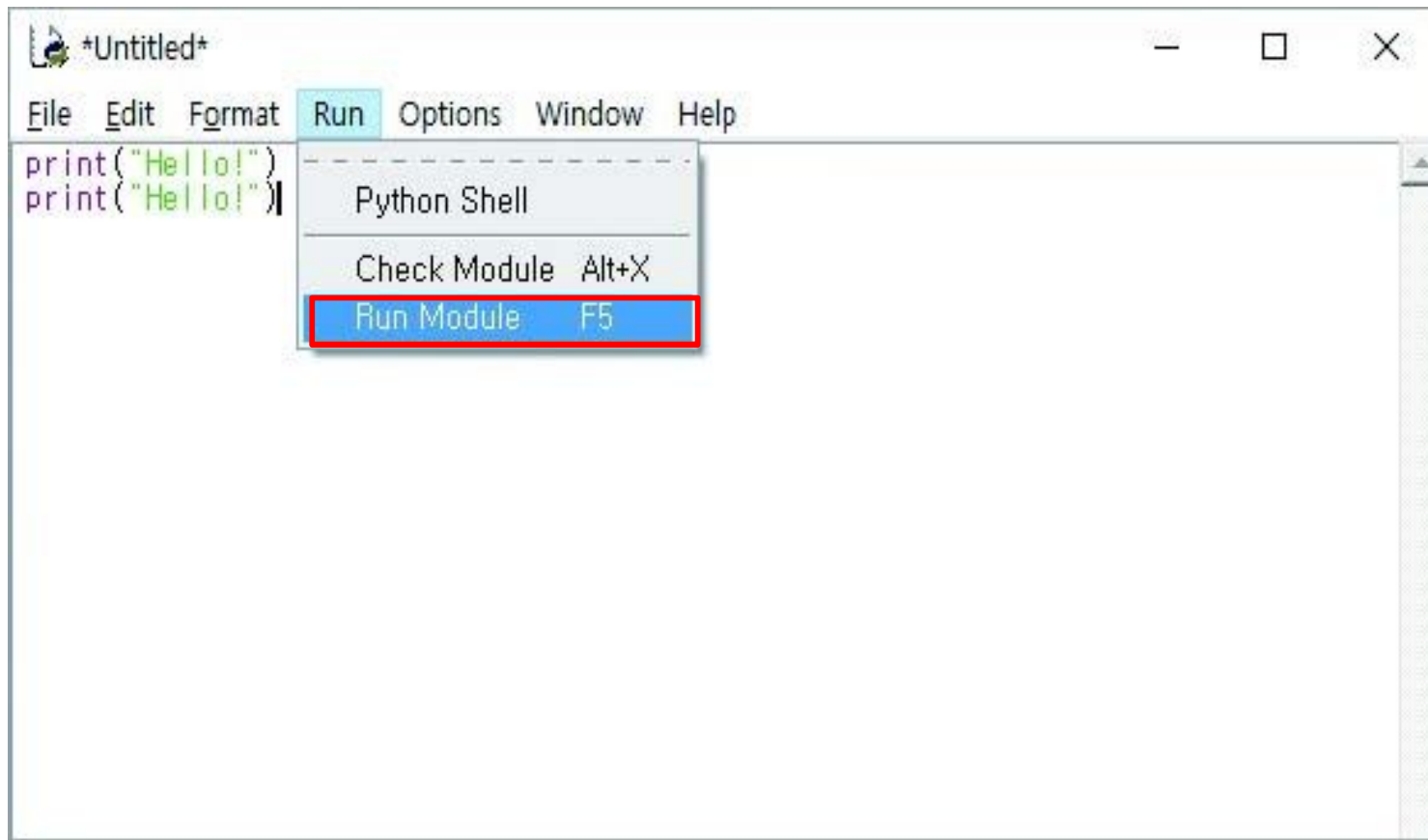
» IDLE을 실행 후 메뉴에서 File → New File을 선택

```
print("Hello!") # print는 소문자로 입력  
print("Hello!") # Hello!를 두 번 표시
```



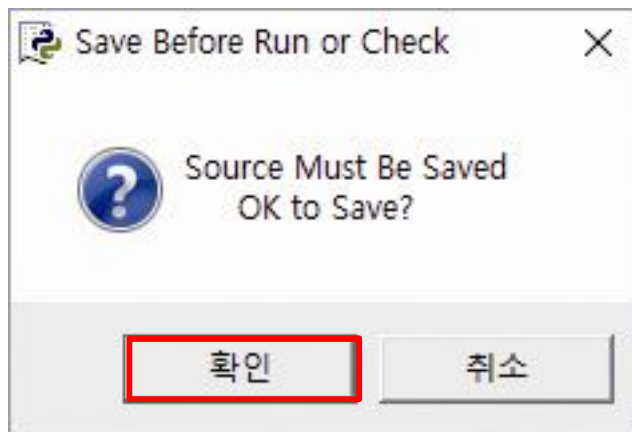
1. Hello!를 두 번 출력하는 프로그램

» Run → Run Module을 선택하거나 F5 를 누름



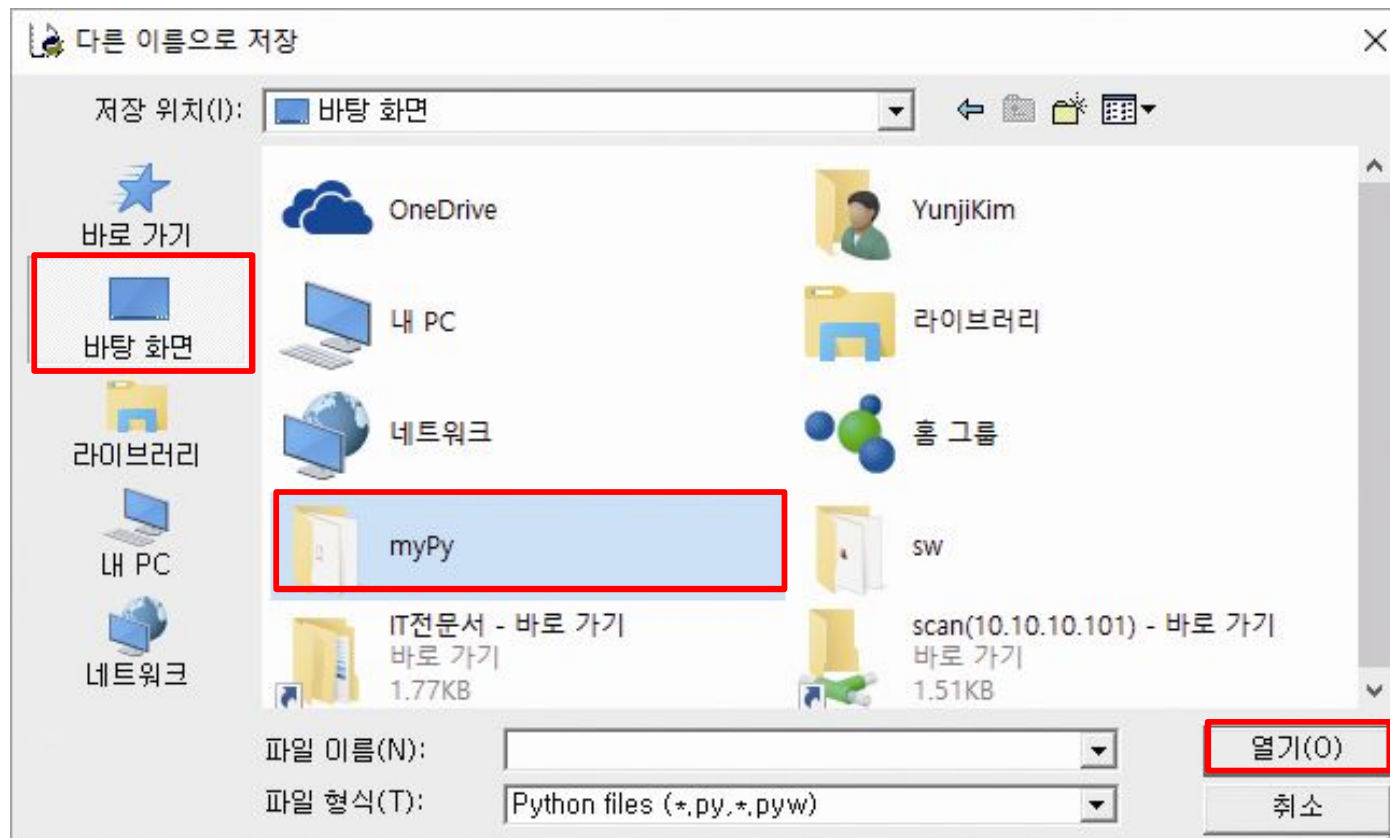
1. Hello!를 두 번 출력하는 프로그램

» 파일을 저장할 것인지 묻는 창이 뜨면 확인 버튼을 누름



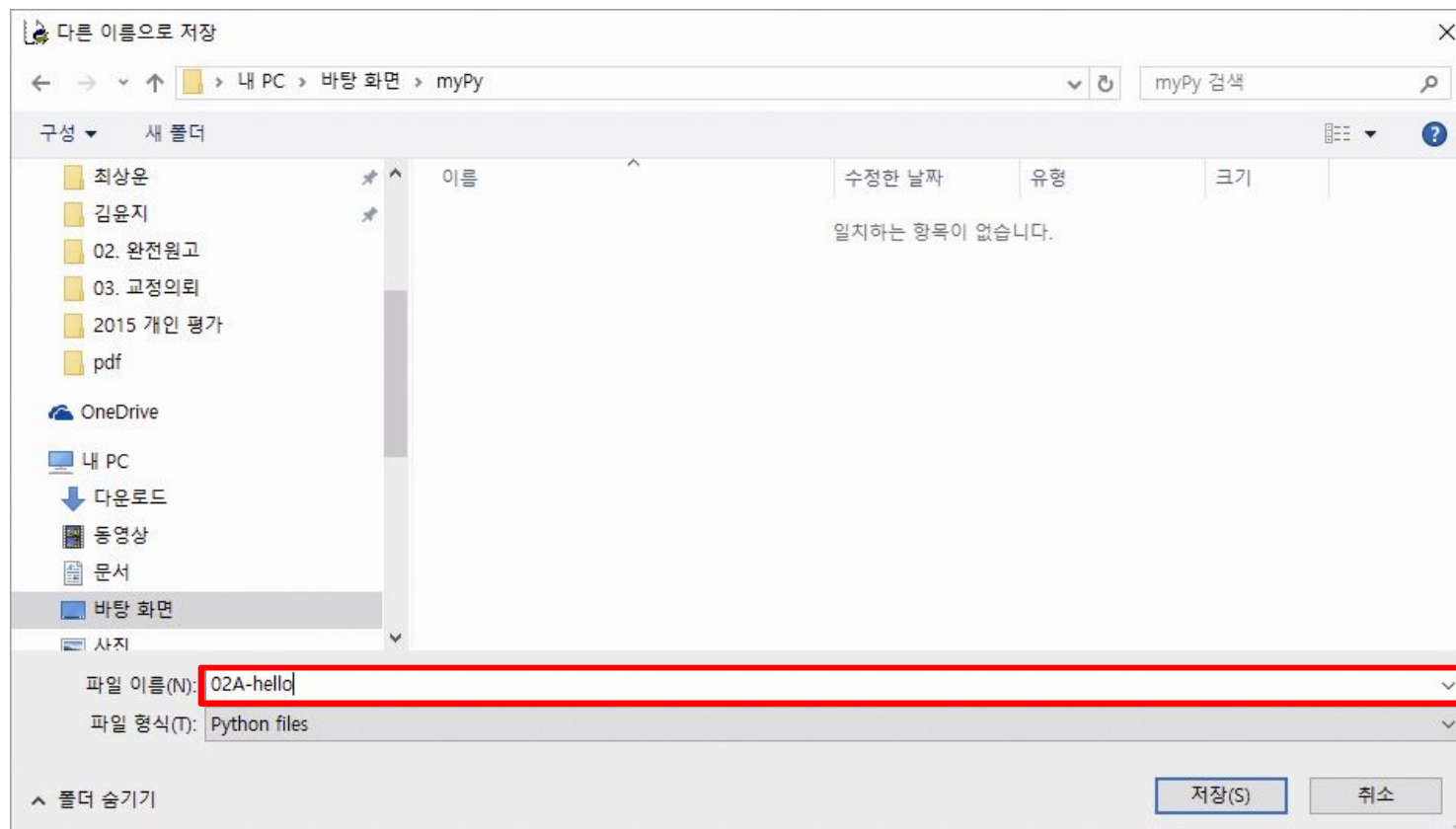
1. Hello!를 두 번 출력하는 프로그램

» 바탕 화면에 myPy 폴더를 만든 후 열기 버튼을 누름.



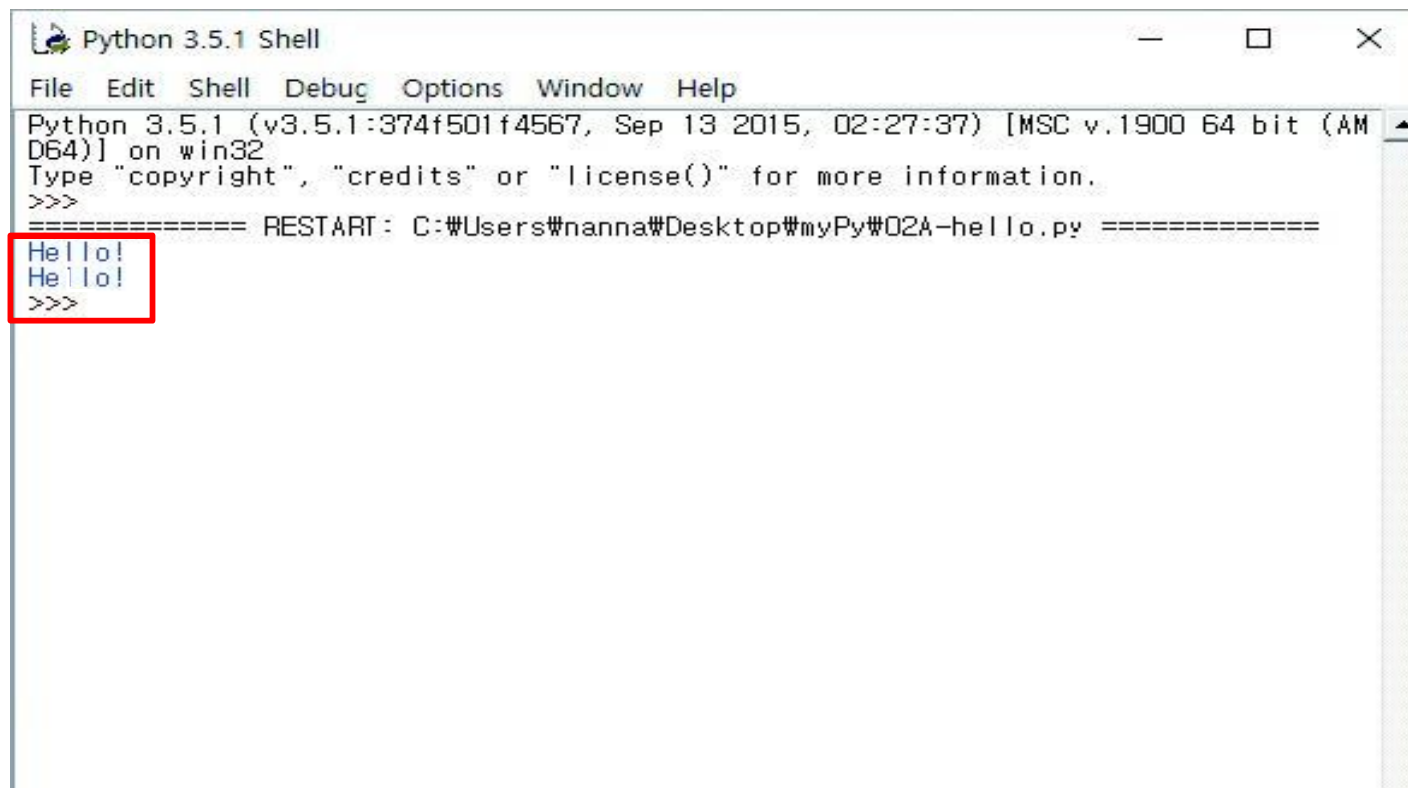
1. Hello!를 두 번 출력하는 프로그램

» myPy 폴더가 열리면 파일 이름을 02A-hello로 입력하고 저장



1. Hello!를 두 번 출력하는 프로그램

» 프로그램이 실행되고 IDLE 대화형 셸에 실행 결과가 출력됨



```
Python 3.5.1 Shell
File Edit Shell Debug Options Window Help
Python 3.5.1 (v3.5.1:374f501f4567, Sep 13 2015, 02:27:37) [MSC v.1900 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\nanna\Desktop\myPy\02A-hello.py =====
Hello!
Hello!
>>>
```

1. Hello!를 두 번 출력하는 프로그램

» 주석이 무엇인가요?

예제 프로그램을 보면 # 기호로 시작하는 문장이 보일 것입니다. 프로그램을 이해하는 데 도움이 되는 내용을 적어 놓은 '설명'으로, 예제를 입력할 때 꼭 읽어 봐야 합니다.

```
print("Hello!") # print는 소문자로 입력합니다.
```

```
print("Hello!") # Hello!를 두 번 표시합니다.
```

파이썬은 # 기호부터 그 줄 끝에 있는 글자까지는 읽지 않고 건너뛰기 때문에 입력하지 않아도 프로그램을 실행하는 데는 아무런 지장이 없습니다. 그렇다면 왜 파이썬이 읽지도 않는 주석을 작성하는 걸까요? 주석은 프로그램을 작성한 사람과 프로그램을 사용할 사람이 함께 이용하는 메모지와 같습니다. 적어 두면 도움이 될 만한 설명이 나 꼭 알아야 할 내용을 메모해 두는 것입니다. 즉, 주석은 프로그램 안에 붙여 놓은 포스트잇과 같은 역할을 한다고 보면 됩니다

1. Hello!를 두 번 출력하는 프로그램

» 거북이 그래픽의 짧은 명령어

forward는 앞으로, left는 왼쪽, right는 오른쪽을 뜻하는 영어 단어인데, 거북이 그래픽에서 가장 많이 사용되는 명령어이기도 합니다. 이렇게 자주 쓰는 명령어는 짧을수록 입력하기가 편합니다. 그래서 파이썬에서는 다음과 같이 짧은 버전 명령어도 제공합니다.

- forward → fd
- left → lt
- right → rt

짧은 버전 명령어를 사용하면 입력하는 시간도 줄고 실수도 줄일 수 있습니다. 예제 코드를 입력할 때 `t.forward(50)`을 간단히 `t.fd(50)`으로 입력해도 좋습니다.

슬라이드 2.

거북이 그래픽으로 도형을 그리는 프로그램

```
import turtle as t
```

```
# 삼각형 그리기
```

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

```
t.left(120)
```

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

```
t.left(120)
```

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

```
t.left(120)
```

```
# 거북이가 100만큼 앞으로 이동
```

```
# 거북이가 왼쪽으로 120도 회전
```

```
# 위 과정을 두 번 반복
```

```
# 사각형 그리기
```

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

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

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

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

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

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

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

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

```
# 거북이가 100만큼 앞으로 이동
```

```
# 거북이가 왼쪽으로 90° 회전
```

```
# 위 과정을 세 번 반복
```

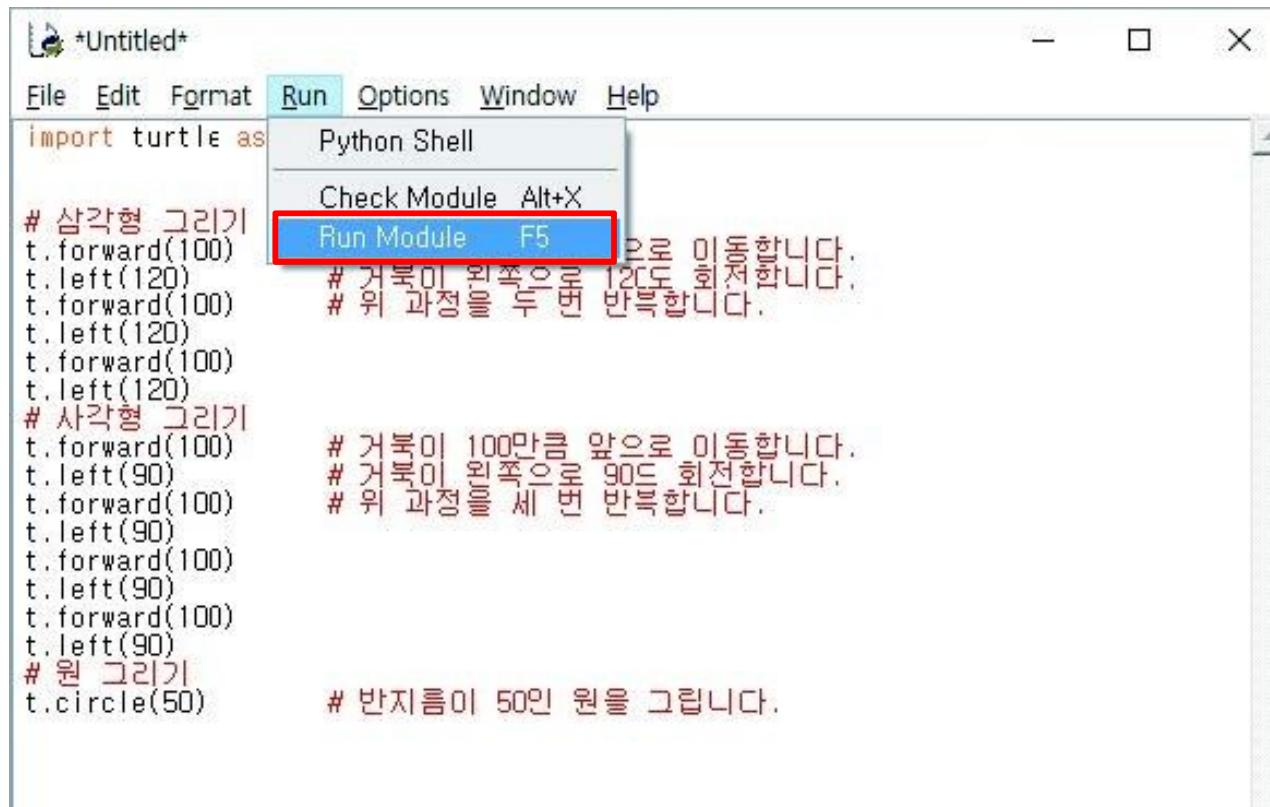
```
# 원 그리기
```

```
t.circle(50)
```

```
# 반지름이 50인 원을 그림
```

2. 거북이 그래픽으로 도형을 그리는 프로그램

- » IDLE을 실행하고 File → New File을 선택해 새 파일을 만든 다음, 위와 같이 코드 입력. forward 대신 fd, left 대신 lt를 입력해도 됨.
- » Run → Run Module을 선택하거나 F5 를 누름.



The screenshot shows the IDLE Python IDE interface. The menu bar includes File, Edit, Format, Run, Options, Window, and Help. The 'Run' menu is open, displaying three options: 'Python Shell', 'Check Module Alt+X', and 'Run Module F5'. The 'Run Module F5' option is highlighted with a red rectangular box. The background code in the editor is as follows:

```
import turtle as t

# 삼각형 그리기
t.forward(100)
t.left(120)
t.forward(100)
t.left(120)
t.forward(100)
t.left(120)

# 사각형 그리기
t.forward(100)
t.left(90)
t.forward(100)
t.left(90)
t.forward(100)
t.left(90)
t.forward(100)
t.left(90)

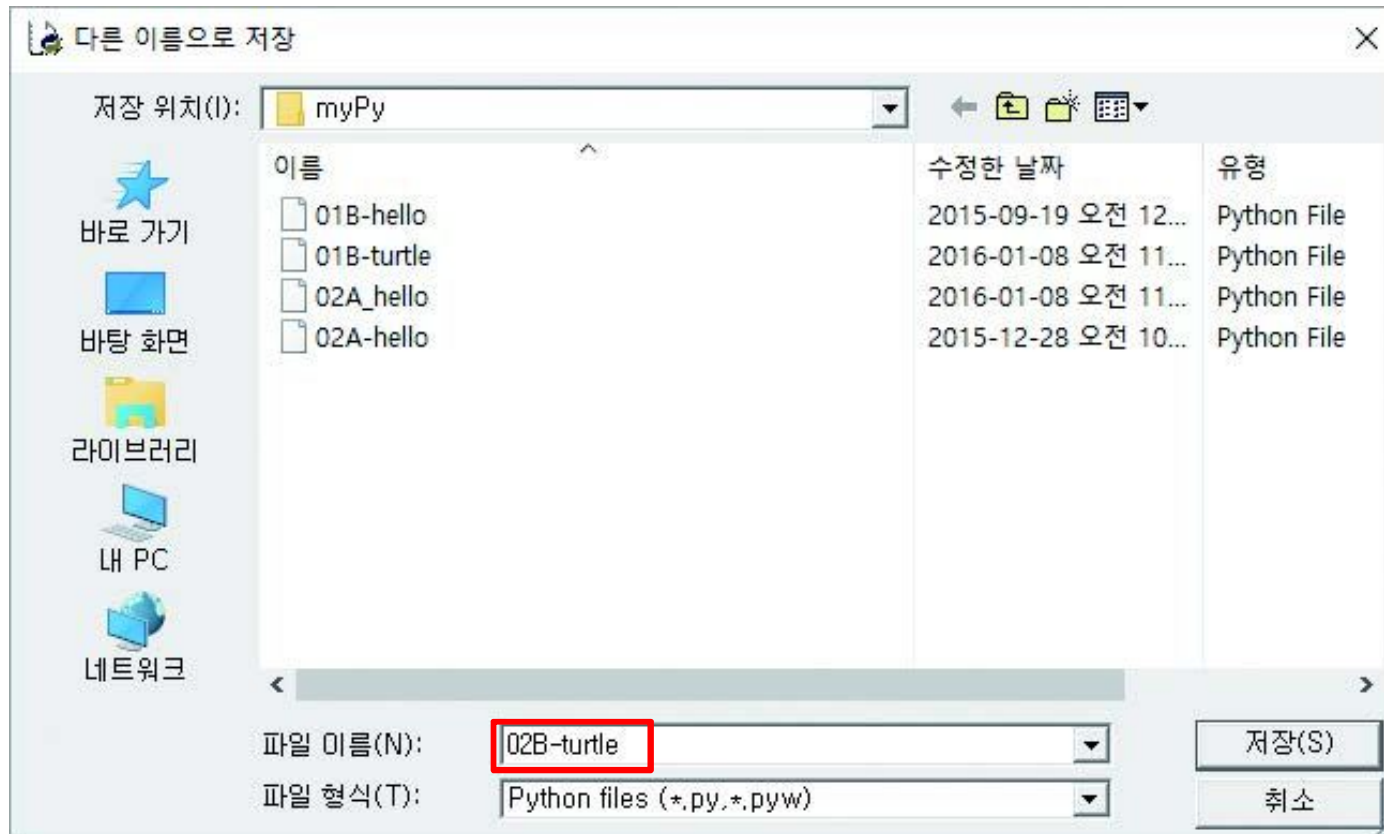
# 원 그리기
t.circle(50)
```

Comments in Korean are present next to the code blocks:

- For the triangle: # 거북이 앞으로 100만큼 이동합니다. # 거북이 왼쪽으로 120도 회전합니다. # 위 과정을 두 번 반복합니다.
- For the square: # 거북이 100만큼 앞으로 이동합니다. # 거북이 왼쪽으로 90도 회전합니다. # 위 과정을 세 번 반복합니다.
- For the circle: # 반지름이 50인 원을 그립니다.

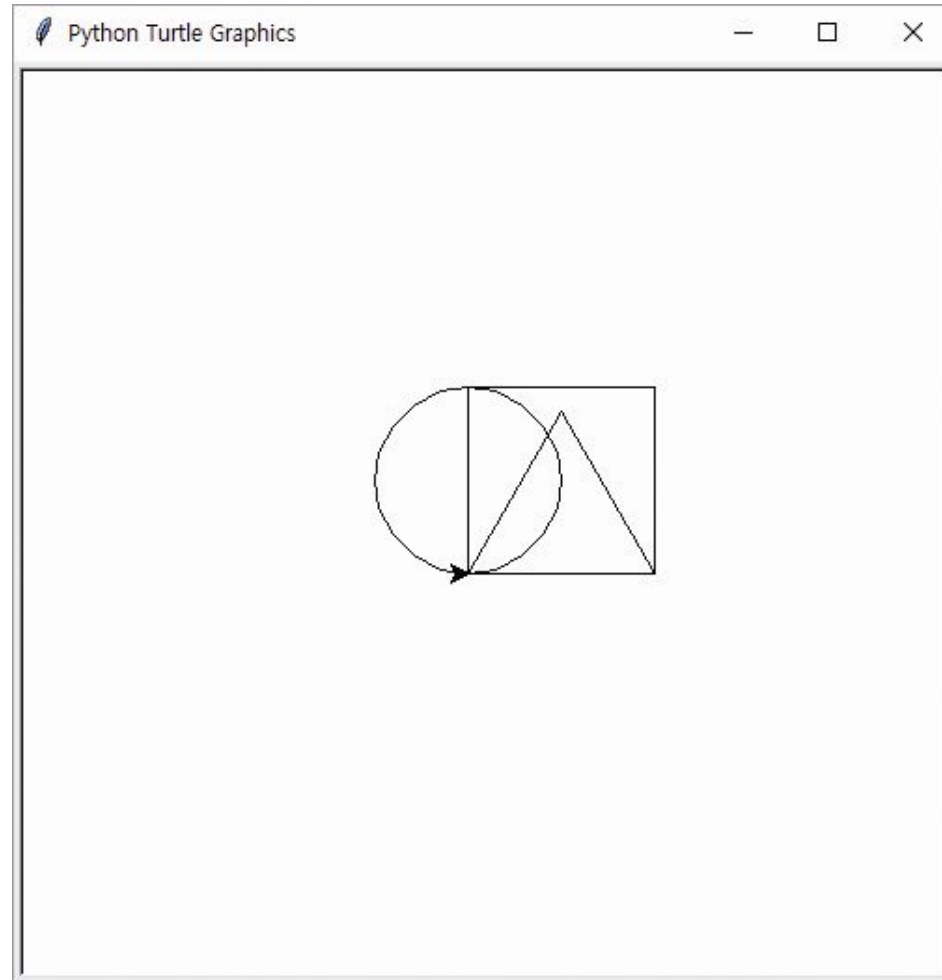
2. 거북이 그래픽으로 도형을 그리는 프로그램

- » 다른 이름으로 저장할 것인지 묻는 창이 뜨면 확인 버튼을 누름.
파일 이름을 02B-turtle로 입력하고 저장 버튼을 누름.



02

2. 거북이 그래픽으로 도형을 그리는 프로그램



2. 거북이 그래픽으로 도형을 그리는 프로그램

» 에러 메시지를 보고 에러 위치 찾기

프로그램을 여러 줄로 만들었을 때 에러가 발생하면 어느 위치에서 에러가 발생했는지 찾기 어렵습니다. 이럴 때는 파이썬이 알려 준 에러 메시지에서 힌트를 얻을 수 있습니다.

Traceback (most recent call last):

```
File "C:\Users\Work\02B-turtle.py", line 14, in <module>  
    t.let(90)
```

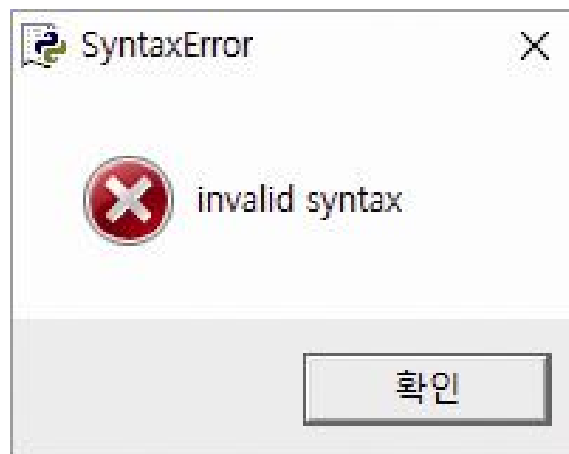
AttributeError: 'module' object has no attribute 'let'

위의 에러는 left 명령어를 let으로 잘못 입력해서 발생한 에러입니다. line 14를 보고 열네 번째 줄에 입력한 t.let(90)에 문제가 있다는 것을 알 수 있습니다. no attribute 'let'은 let이라는 명령어가 없다는 뜻입니다. 곧 left를 잘못 입력했다는 말입니다.

2. 거북이 그래픽으로 도형을 그리는 프로그램

» 에러 메시지 창이 떴을 때 에러 위치 찾기

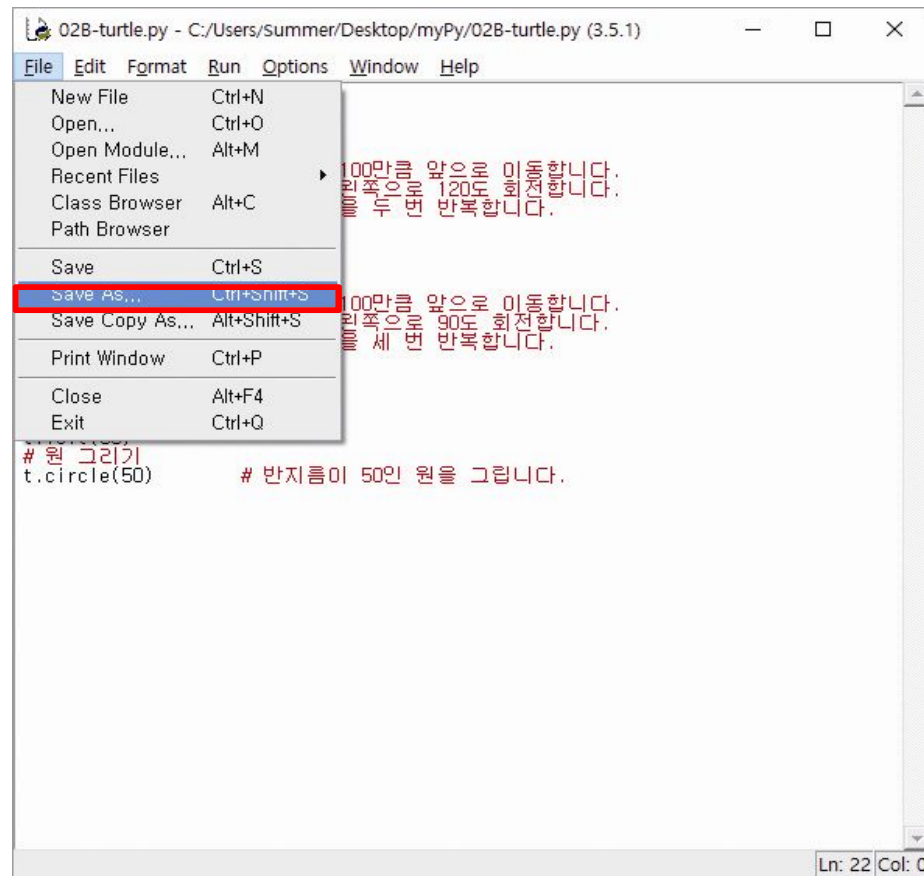
에러 메시지가 대화 창으로 뜨면 어느 위치에서 에러가 발생했는지 정확한 행 번호를 알 수 없습니다. 대신 프로그램의 입력 커서가 에러가 발생한 곳 근처로 옮겨진 것을 알 수 있습니다. 대화 창을 닫고 커서의 앞뒤 부분을 살펴보면 에러를 쉽게 찾을 수 있습니다.



2. 거북이 그래픽으로 도형을 그리는 프로그램

» 응용

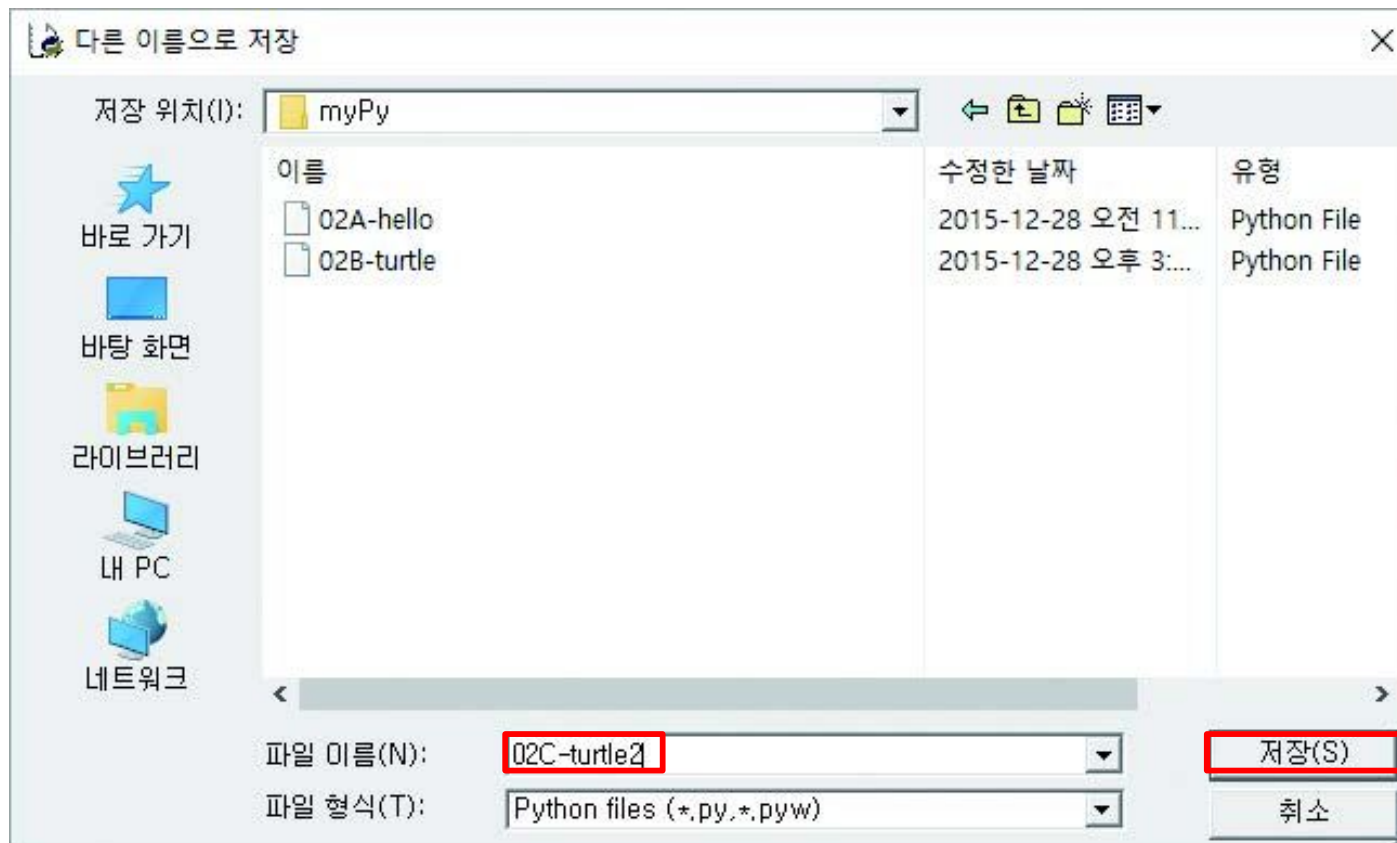
02B-turtle.py 파일이 열린 상태에서 File → Save As...를 선택하거나 Ctrl + Shift + S 를 누름.



02

2. 거북이 그래픽으로 도형을 그리는 프로그램

» 파일 이름을 02C-turtle2로 입력하고 저장 버튼을 누름.
기존 파일이 새로 지정한 파일 이름으로 저장됨.



02

2. 거북이 그래픽으로 도형을 그리는 프로그램

```
import turtle as t
```

```
# 삼각형 그리기
```

```
t.color("red")
```

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

```
t.left(120)
```

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

```
t.left(120)
```

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

```
t.left(120)
```

```
# 펜 색상을 빨간색으로 바꿈
```

```
# 사각형 그리기
```

```
t.color("green")
```

```
t.pensize(3)
```

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

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

```
t.forward(100
```

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

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

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

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

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

```
# 펜 색상을 녹색으로 바꿈
```

```
# 펜 굵기를 3으로 바꿈
```

```
# 원 그리기
```

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

```
t.pensize(5)
```

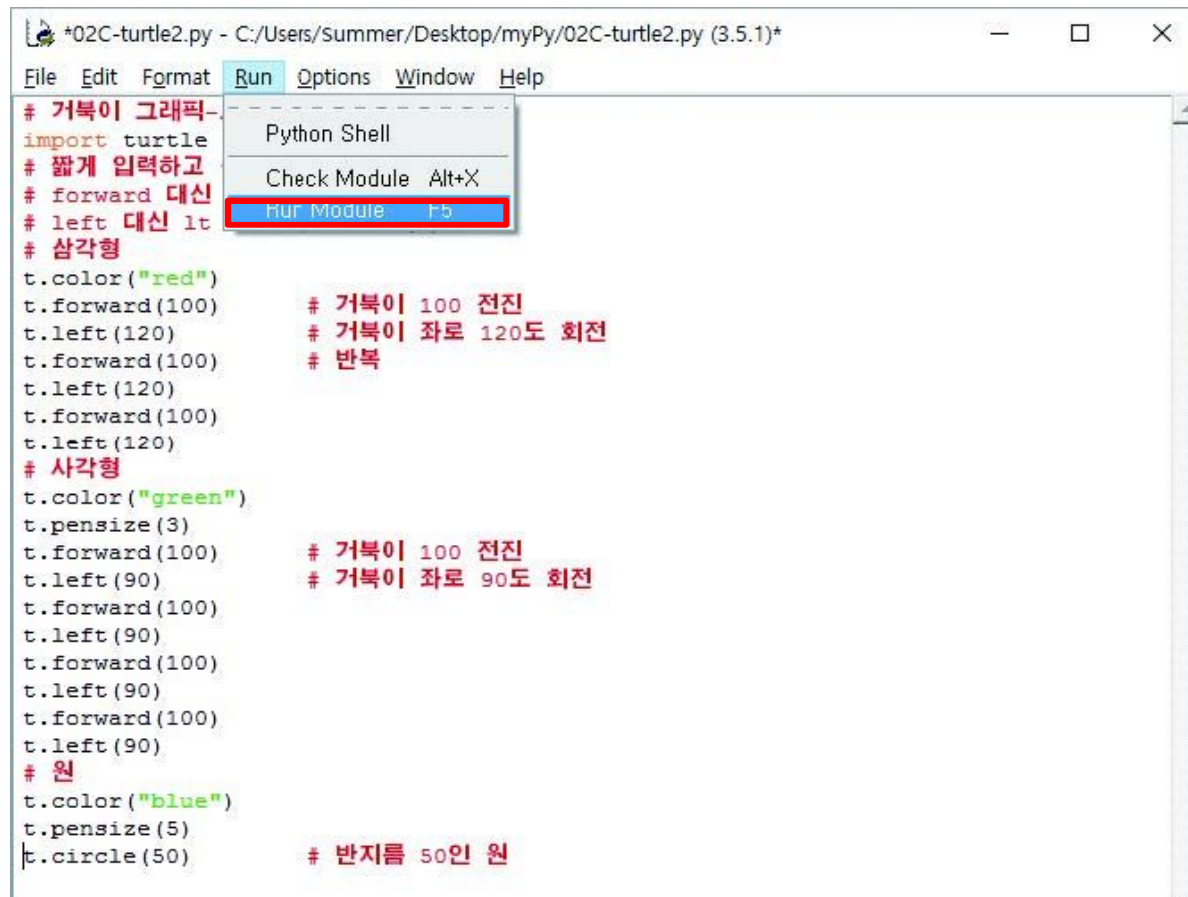
```
t.circle(50)
```

```
# 펜 색상을 파란색으로 바꿈
```

```
# 펜 굵기를 5로 바꿈
```

2. 거북이 그래픽으로 도형을 그리는 프로그램

» 수정을 마쳤다면 Run → Run Module을 선택하거나 F5 를 눌러 프로그램을 실행.



The screenshot shows a Python IDE window titled '*02C-turtle2.py - C:/Users/Summer/Desktop/myPy/02C-turtle2.py (3.5.1)*'. The menu bar includes File, Edit, Format, Run, Options, Window, and Help. The 'Run' menu is open, showing options: Python Shell, Check Module Alt+X, and Run Module F5. The 'Run Module F5' option is highlighted with a red rectangle. The code in the editor is as follows:

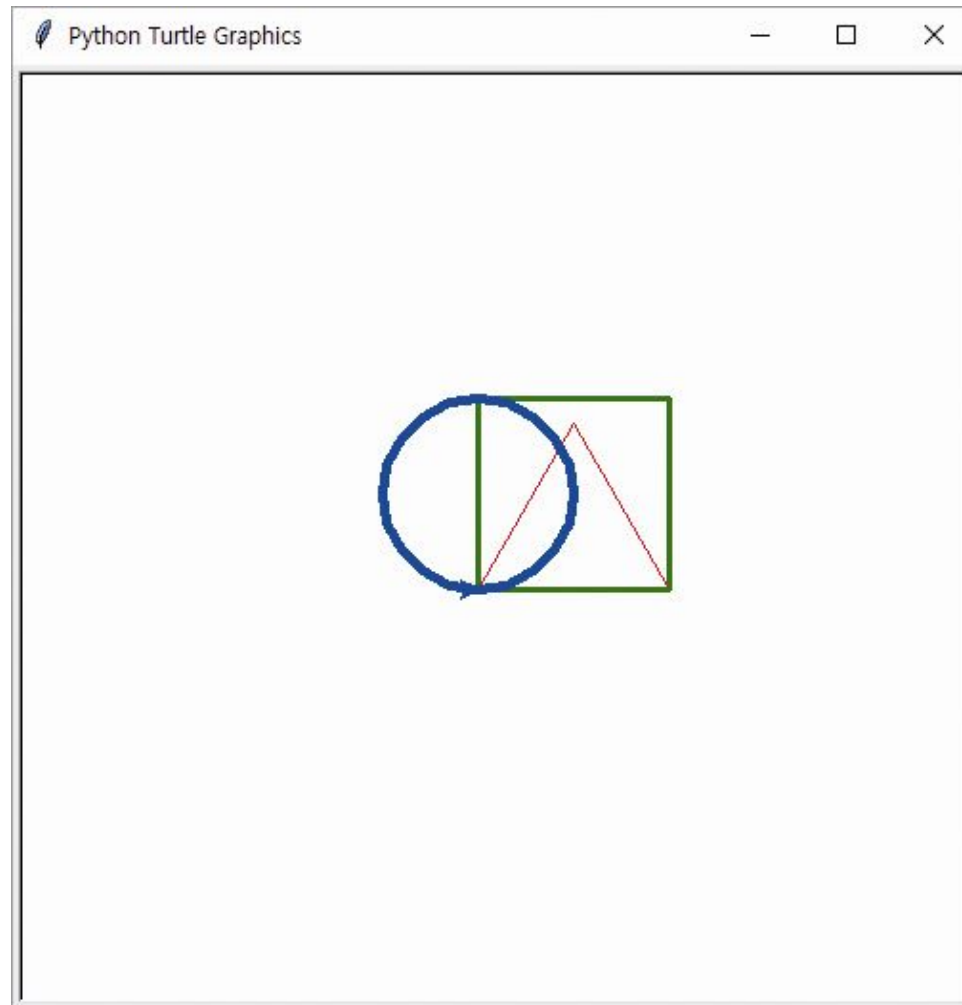
```
# 거북이 그래픽-
import turtle
# 짧게 입력하고
# forward 대신 lt
# left 대신 lt
# 삼각형
t.color("red")
t.forward(100)
t.left(120)
t.forward(100)
t.left(120)
t.forward(100)
t.left(120)
# 사각형
t.color("green")
t.pensize(3)
t.forward(100)
t.left(90)
t.forward(100)
t.left(90)
t.forward(100)
t.left(90)
t.forward(100)
t.left(90)
# 원
t.color("blue")
t.pensize(5)
t.circle(50)
```

Comments in Korean are provided for each shape:

- For the triangle: `# 거북이 100 전진`, `# 거북이 좌로 120도 회전`, `# 반복`
- For the square: `# 거북이 100 전진`, `# 거북이 좌로 90도 회전`
- For the circle: `# 반지름 50인 원`

02

2. 거북이 그래픽으로 도형을 그리는 프로그램



» Unit 3

연산과 변수

연산과 변수

01 파이썬의 연산

02 파이썬의 변수

슬라이드 1. 파이썬의 연산

연산 기호	뜻	예시	결과
+	더하기	7+4	11
-	빼기	7-4	3
*	곱하기	7*4	28
/	나누기	7/4	1.75
**	제곱 (같은 수를 여러 번 곱함)	2**3	8 (2를 세 번 곱함 2*2*2)
//	정수로 나누었을 때의 몫	7//4	1 (나눗셈의 몫)
%	정수로 나누었을 때의 나머지	7%4	3 (나눗셈의 나머지)
()	다른 계산보다 괄호 안을 먼저 계산	2*(3+4)	14

파이썬 문법으로 작성한 $5 + [4 * \{3 + (1 + 2)\}]$

>>> 5+(4*(3+(1+2)))

29

1. 파이썬의 연산

» 수식 계산 프로그램

» 예제 소스

```
print("7+4 = ", 7+4)
print("7*4 = ", 7*4)
print("7/4 = ", 7/4)
print("2**3 = ", 2**3)    # 2를 세 번 곱한 값
print("5%3 = ", 5%3)    # 5를 3으로 나눈 나머지
```

» 실행결과

```
7+4 = 11
7*4 = 28
7/4 = 1.75
2**3 = 8
5%3 = 2
```

슬라이드 2. 파이썬의 변수

» 변수 : 글자 그대로 '변할 수 있는 수' . 정보 보관소 역할

변할 수 있는 수이므로 필요에 따라 몇 번이라도 저장된 값을 바꿀 수 있음.
보통 프로그램 하나는 변수를 여러 개 사용.

따라서 여러 개의 변수를 구분하려면 변수마다 이름을 따로 붙여야 함.

2. 파이썬의 변수

» 변수를 사용해서 삼각형을 그리는 프로그램

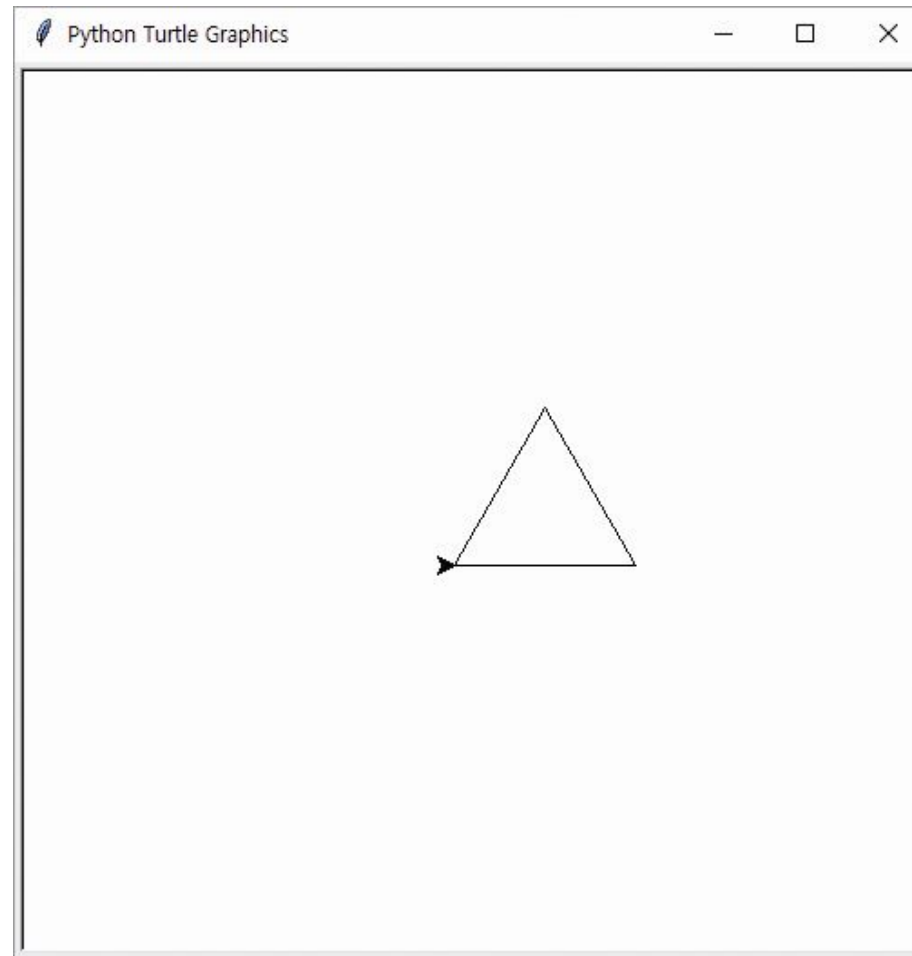
```
import turtle as t
```

```
d = 100      # 변수 d에 값 100을 저장  
             (수치를 바꾸면 삼각형 크기가 변함)
```

```
# 삼각형 그리기
```

```
t.forward(d)  # 거북이가 d만큼 앞으로 이동  
t.left(120)   # 거북이가 왼쪽으로 120도 회전함  
t.forward(d)  
t.left(120)  
t.forward(d)  
t.left(120)
```

2. 파이썬의 변수



2. 파이썬의 변수

» 대화형 셸에서 변수를 사용한 예제

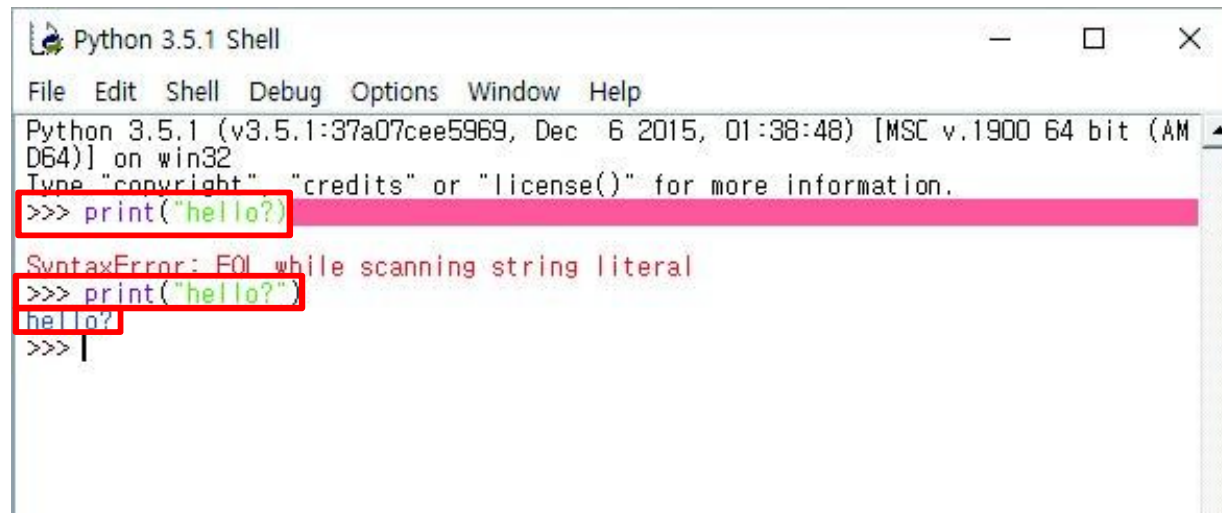
```
>>> a =                # 변수 a에 3을 저장
>>>                    # a 값을 확인
3
>>> b = 1.1+2           # 변수 b에 1.1+2의 결과인 3.1을 저장
>>> b                  # b 값을 확인
3.1
>>> c = a+b             # a와 b를 합한 값을 변수 c에 저장
>>> c                  # c 값을 확인
6.1
>>> d = 2               # 변수 d에 2를 저장
>>> d = d+1             # d에 1을 더한 값을 다시 d에 저장
>>> d                  # d 값을 확인하면 3임
3
```

2. 파이썬의 변수

» 대화형 셸에 명령어를 입력하다 에러가 나면?

대화형 셸에서 명령어를 입력하다 에러가 나면, 에러가 생긴 원인을 찾아내 문장을 올바르게 입력한 다음 다시 한 번 실행하면 됩니다. 매번 처음부터 다시 입력하기가 번거롭다면 다음과 같은 방법으로 수정해 보세요.

- ① 잘못 입력해서 에러가 난 줄을 클릭합니다.
- ② Enter 를 한 번 누르면 >>> 뒤에 잘못 입력한 줄이 자동으로 입력됩니다.
- ③ 자동으로 입력된 줄에서 잘못된 부분을 고치고 Enter 를 누릅니다.
- ④ 수정된 줄에 대한 실행 결과를 얻을 수 있습니다.



```
Python 3.5.1 Shell
File Edit Shell Debug Options Window Help
Python 3.5.1 (v3.5.1:37a07cee5969, Dec 6 2015, 01:38:48) [MSC v.1900 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> print("hello?)
SyntaxError: EOL while scanning string literal
>>> print("hello?")
hello?
>>> |
```

» Unit 4

for 명령을 사용하여
똑같은 작업 반복하기

for 명령을 사용하여 똑같은 작업 반복하기

- 01** Hello!를 반복하여 출력하는 프로그램
- 02** 반복 블록을 알아보는 프로그램
- 03** 반복 기능으로 도형을 그리는 프로그램

슬라이드 1. Hello!를 반복하여 출력하는 프로그램

» 들여쓰기는 몇 칸 해야 하나요?

콜론(:)을 입력한 후 Enter 를 누르면 다음 줄부터는 IDLE이 자동으로 들여쓰기를 해 줍니다. 따라서 다음 줄에 편하게 반복할 내용을 입력할 수 있습니다. 프로그램을 편집하다가 들여쓰기를 직접 해야 할 때는 SpaceBar 를 네 번 눌러 '네 칸 띄어쓰기'를 하는 것이 파이썬이 추천하는 표준 방법입니다.

» 반복을 멈추려면?

range(10) 안의 값을 바꿔 보라고 하면 굉장히 큰 수를 입력해서 테스트하는 사람이 많습니다. 10을 100000으로 바꾸면 어떻게 될까요? Hello!가 십만 번 출력됩니다. 당연히 출력을 마칠 때까지 시간이 오래 걸립니다. 언제 끝날지 모른다고 봐야 합니다. 이럴 때는 파이썬 IDLE을 강제로 닫을 수도 있지만, 다음과 같은 방법으로 프로그램을 멈출 수도 있습니다.

- ① 대화형 셸(Python 3.x.x Shell 창)을 클릭하여 선택합니다.
- ② Ctrl + C 를 누릅니다

슬라이드 2. 반복 블록을 알아보는 프로그램

```
>> for x in range(3):  
    print(100)      # 들여쓰기(빈칸 네 개 뒤에 입력)  
    print(200)      # 들여쓰기(빈칸 네 개 뒤에 입력)  
print(300)         # 내어 쓰기(빈칸 없이 바로 입력)
```

>> 실행 결과

```
100  
200  
100  
200  
100  
200  
300
```

2. 반복 블록을 알아보는 프로그램

» 블록이 무엇인가요?

파이썬에서는 들여쓰기 해서 입력한 문장의 묶음을 블록이라고 부릅니다. 콜론(:) 다음에 들여쓰기로 입력한 두 문장이 바로 '블록'입니다.

```
print(100)
print(200)
```

들여쓰기 되지 않은 마지막 문장은 블록에 속하지 않습니다.

```
print(300)
```

블록은 오늘 배운 반복 외에도 뒤에서 배울 판단이나 함수 등 여러 곳에서 쓰입니다. 블록은 프로그램의 실행에 사용되는 '프로그램의 한 묶음 또는 단위'라는 점을 꼭 기억하세요

슬라이드 3. 반복 기능으로 도형을 그리는 프로그램

```
>> import turtle as t
```

```
# 삼각형 그리기
```

```
for x in range(3):  
    t.forward(100)  
    t.left(120)
```

```
# 세 번 반복
```

```
# 거북이가 100만큼 앞으로 이동  
# 거북이가 왼쪽으로 120도 회전
```

```
# 사각형 그리기
```

```
for x in range(4):  
    t.forward(100)  
    t.left(90)
```

```
# 네 번 반복
```

```
# 거북이가 100만큼 앞으로 이동  
# 거북이가 왼쪽으로 90도 회전
```

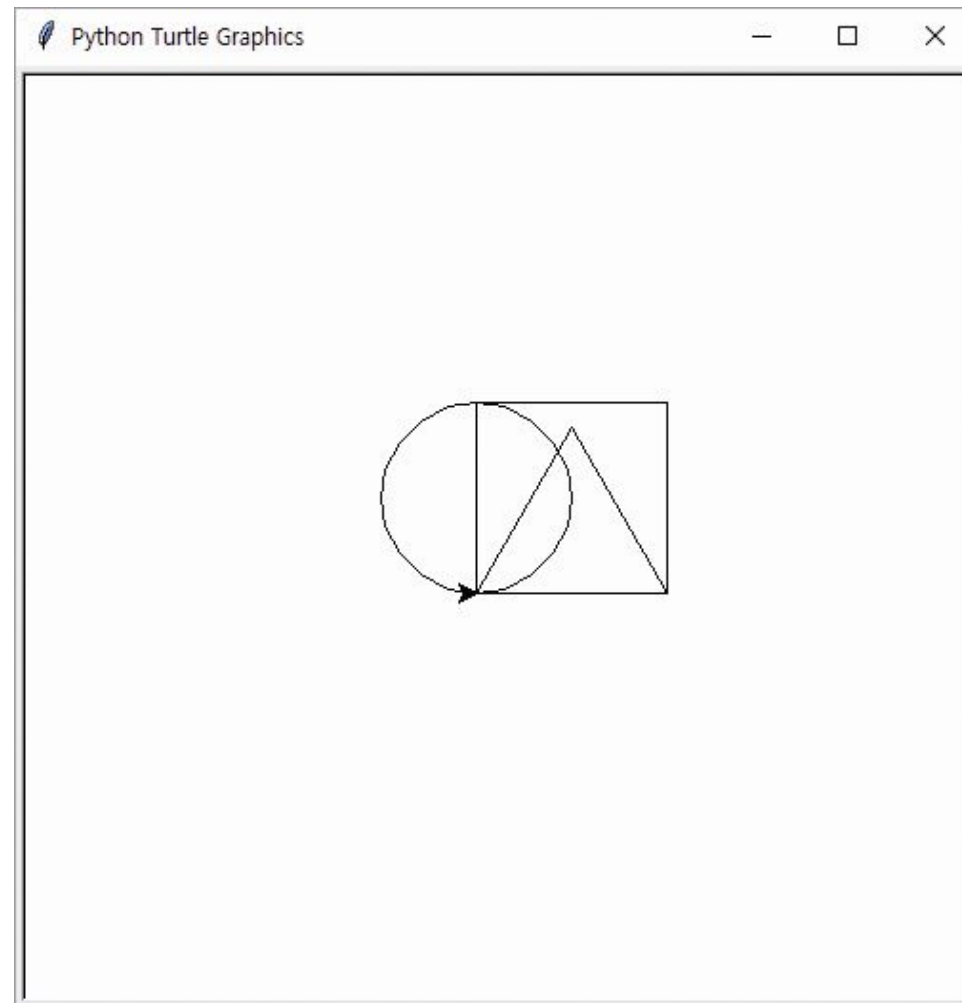
```
# 원 그리기
```

```
t.circle(50)
```

```
# 반지름이 50인 원을 그림
```

3. 반복 기능으로 도형을 그리는 프로그램

» 실행결과



» 학습정리

- 파이썬 프로그램을 다운받을 수 있는 사이트
<http://python.org/dpwnload>
- 대화형 셸(Interactive shell) 혹은 셸(shell)은 사용자가 입력을 하면 바로 결과를 보여주는 형태 프로그램 임.
- 명령어 에러 종류
 - 이름 에러 : `NameError: name '...' is not defined`
 - 구문 에러 : `SyntaxError: EOL while scanning string literal`
 - 임포트 에러 : `ImportError: Not module named`
 - 거북이 모양 에러 : `TurtleGraphicError: There is no shape named...`
 - 속성에러 : `AttributeError: 'module' object has no attribute...`
 - 거북이 색상 에러 : `TurtleGraphicError: bad color string...`

» 학습정리

- 주석 : #기호로 시작하는 문장은 프로그램을 이해하는 데 도움이 되는 내용을 적어 놓은 '설명'임.
- 그래픽의 짧은 명령어
 - forward -> fd
 - left -> lt
 - right -> rt

» 학습평가

1. 다음 파이썬 프로그램의 명령어 에러 종류가 아닌 것은? 4

1. 이름 에러 : `NameError: name '...' is not defined`
2. 구문 에러 : `SyntaxError: EOL while scanning string literal`
3. 임포트 에러 : `ImportError: Not module named`
4. 거북이 속성 에러 : `TurtleAttributeGraphicError: There is no shape named...`

» 학습평가

2. 다음 파이썬 프로그램의 실행 결과로 올바른 것은? 1

```
for x in range(3);
```

```
    print(100)
```

```
    print(200)
```

```
Print(300)
```

1. 100
200
100
200
100
200
300

2. 100
200
300

3. 100
200
300
100
200
300
100
200
300

4. 100
200
300
300
300