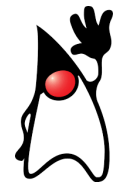


ㄷ ㄱ ㄷ ㄱ
ㅏ ㅓ ㅏ ㅓ
ㄴ ㄴ ㄴ ㄴ

파이썬 수업



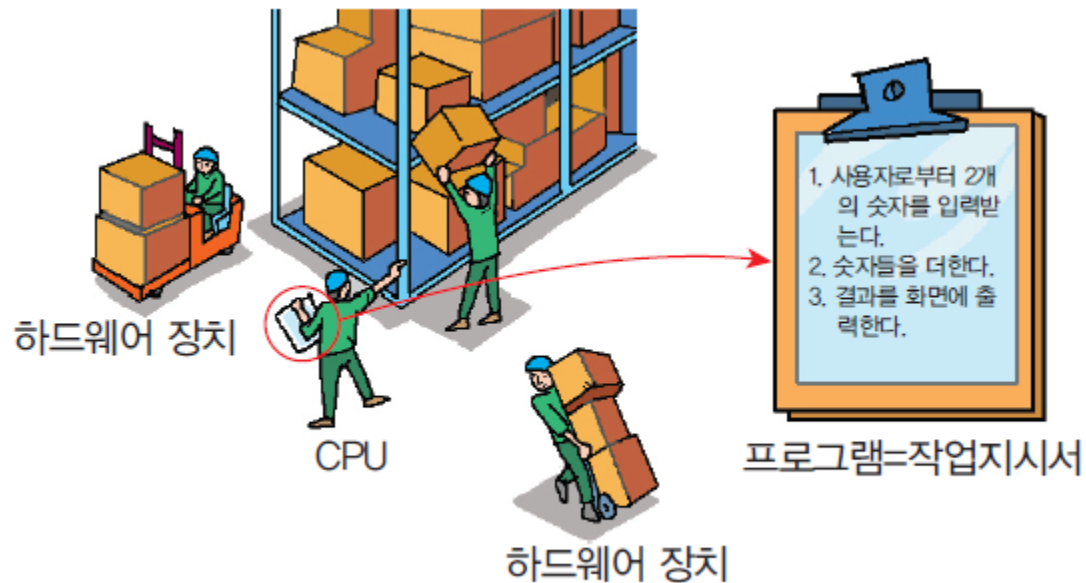
1 장 파이썬 소개

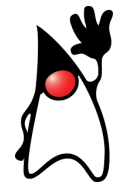


컴퓨터 프로그램



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





프로그래밍 언어

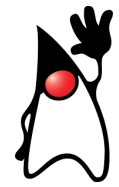


- 프로그램은 ‘프로그래밍 언어’로 작성된다. 프로그램을 만드는 사람을 ‘프로그래머’라고 한다.



→
프로그래밍 언어

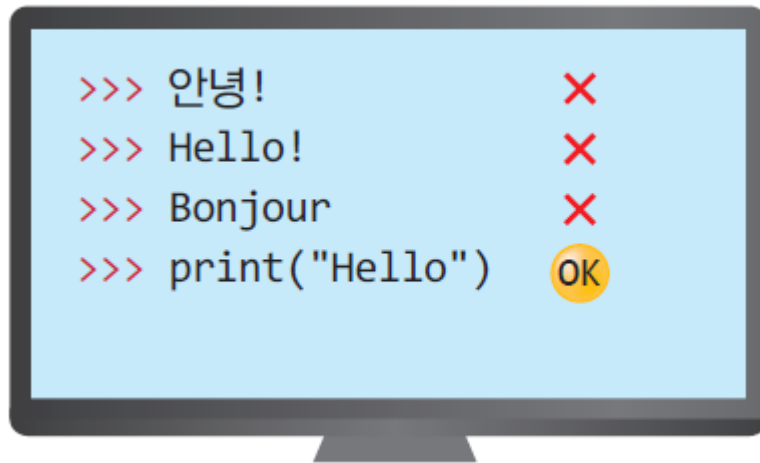




프로그래밍 언어



- 컴퓨터는 사람의 언어를 이해할 수 없다!
- '프로그래밍 언어'는 컴퓨터가 이해하는 언어이다.

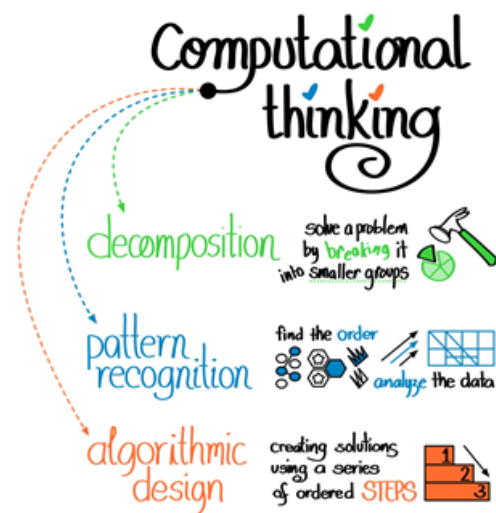


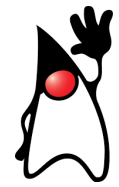


프로그래밍이 어디에 도움이 될까?

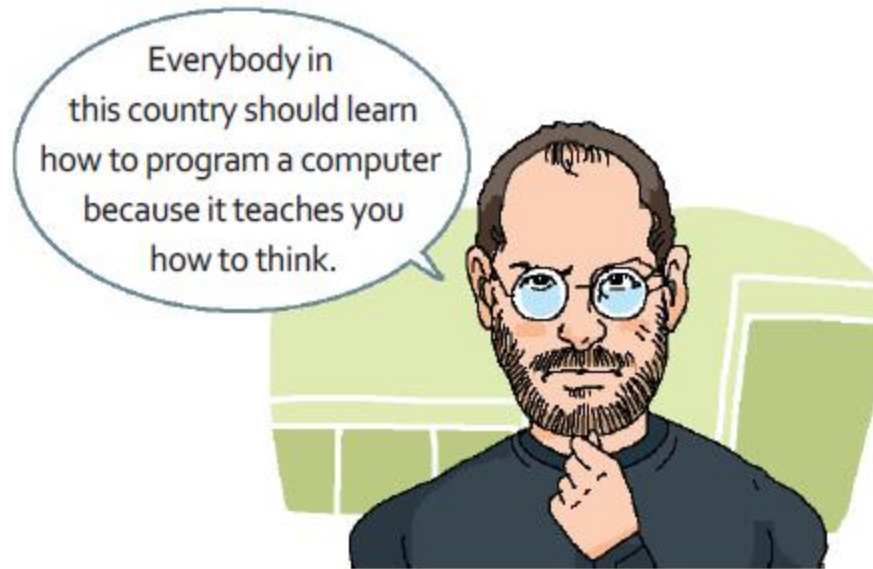


- 컴퓨터를 여러분 마음대로 제어할 수 있다.
- 자신이 해결해야 하는 일에 딱 맞는 프로그램을 작성할 수 있다.
- 프로그래밍을 하면 더 창의적인 사람이 된다(?).
- 프로그래밍을 하면 논리적으로 문제를 해결하는 능력을 배양할 수 있다.





스티브 잡스



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

– 스티브 잡스(Steve Jobs)

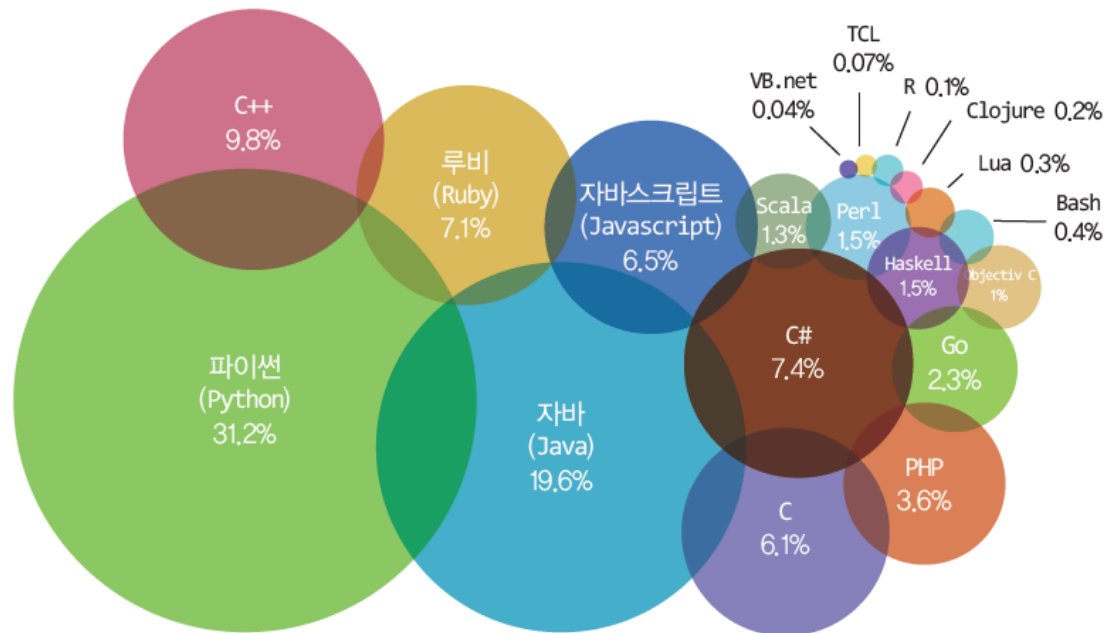


프로그래밍 언어의 종류



- 많이 사용되는 언어들에는 '파이썬', '자바', 'C', 'BASIC' 등이 있다.

인기있는 프로그래밍 언어



자료출처: <http://www.techtechnik.com>(2015년)



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

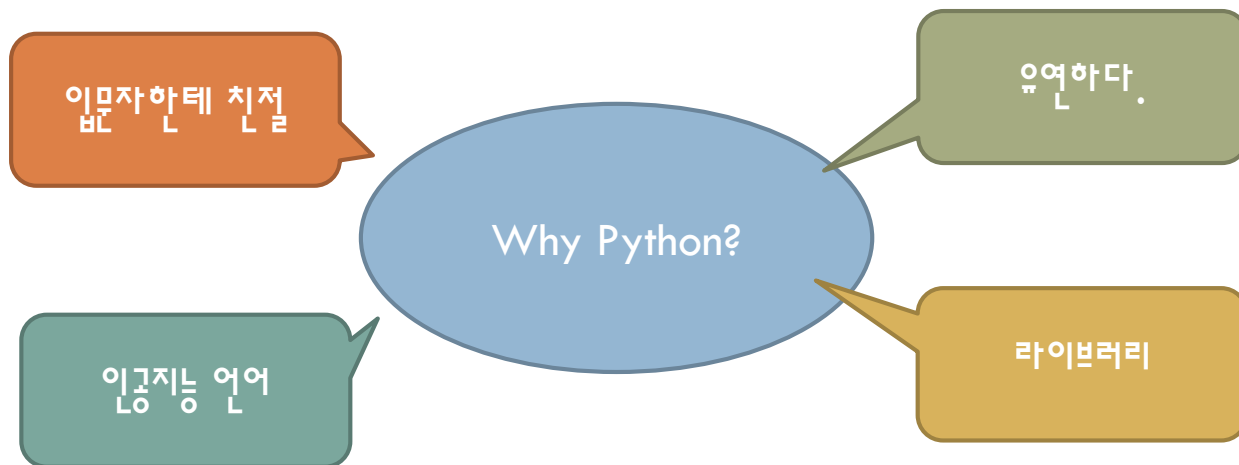


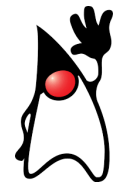


파이썬이 인기있는 이유



- 파이썬은 간결하고 읽기 쉬운 구문, 유용한 라이브러리, 쉬운 유지보수성 등의 이유로 인기가 있다.
- 파이썬은 웹 개발, 시스템 자동화, 데이터 처리, 인공지능 등 다양한 분야에서 사용되기 때문에 인기가 있다.
- 파이썬은 코드를 작성하기 쉽기 때문에 새로운 개발자들이 쉽게 접근할 수 있기 때문에 인기가 있다.

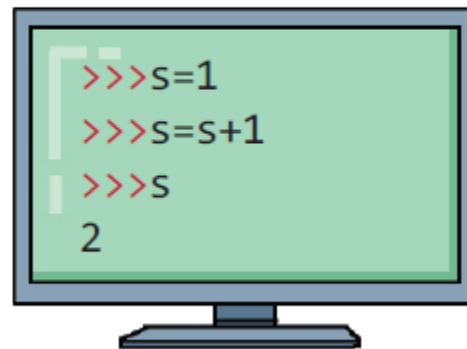




파이썬의 특징



- 생산성이 뛰어나다.
- 초보자한테 좋은 언어 – 인터프리터 언어





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



파이썬을 사용하고 있는 기업들



YAHOO!

Google

You Tube



reddit



BitTorrent™

IBM



Dropbox



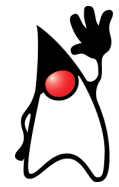
redhat.

CANONICAL

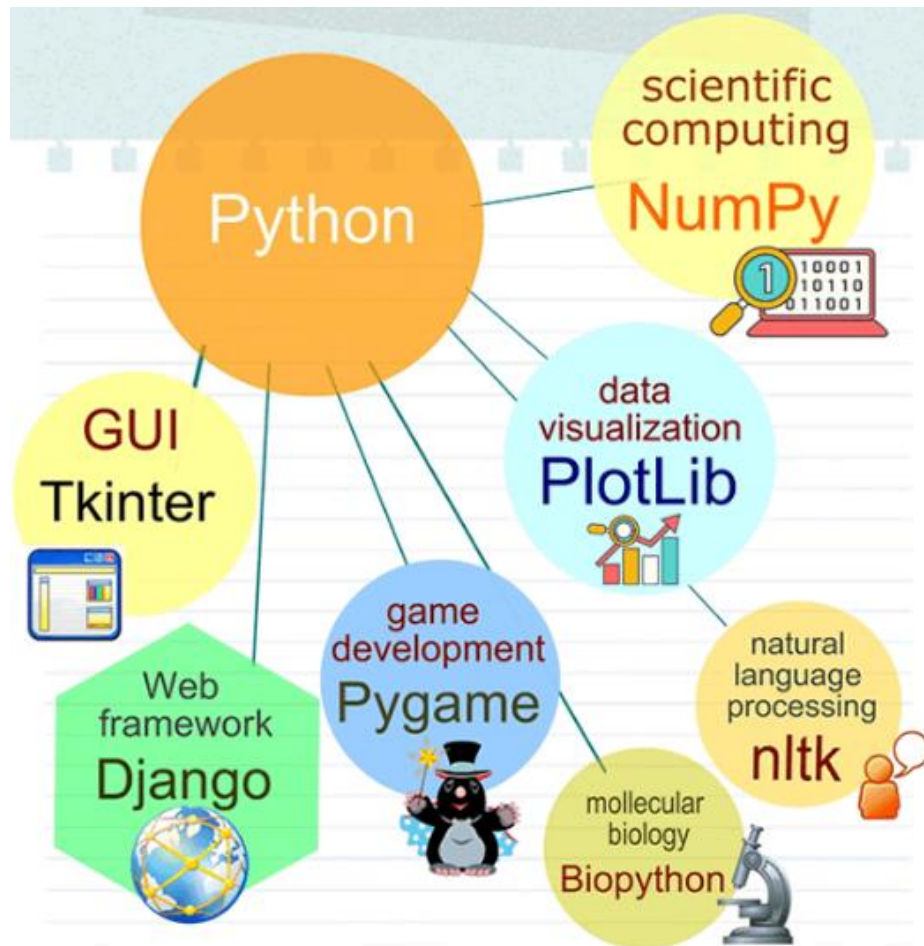
NETFLIX

Quora





파이썬의 라이브러리

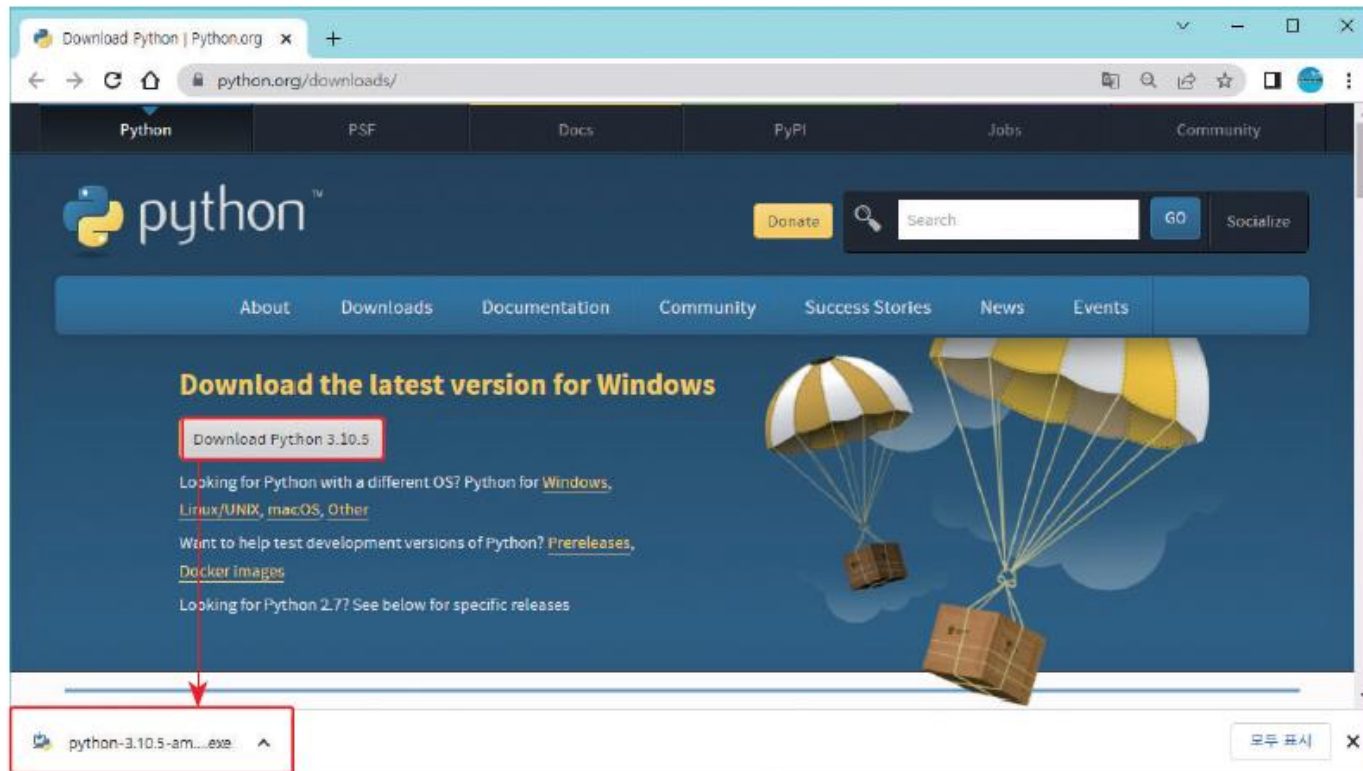


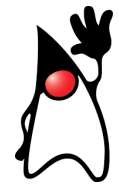


파이썬 설치하기



- 파이썬을 설치하려면 <http://www.python.org/>에 접속하여 Download 메뉴에서 "Python 3.10.5"을 선택한다.

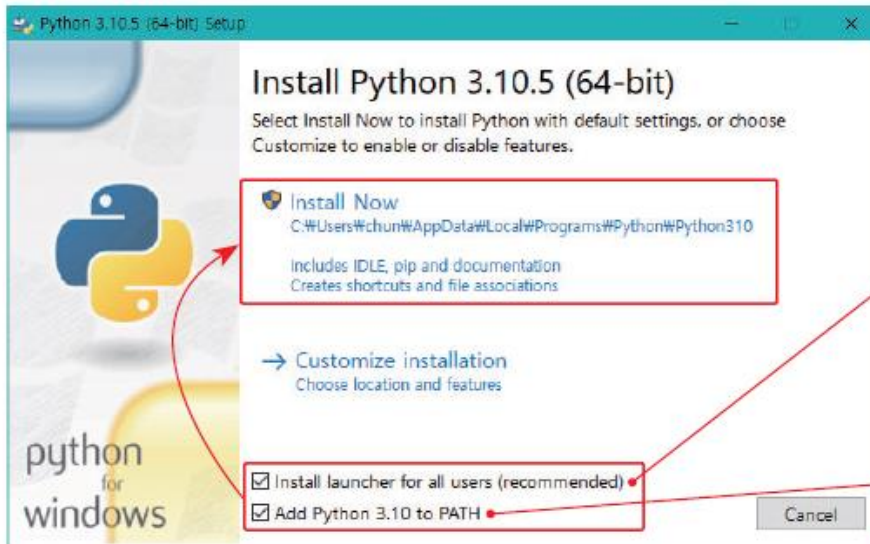




파이썬 설치하기

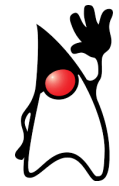


- 반드시 다음을 체크할 것!



위 화면의 아래에 있는 첫 번째 체크 박스는 컴퓨터 사용자 전체가 사용하게 할 것인지, 아니면 현재의 사용자만 사용할 것인지를 묻는 것이다. "Install launcher for all users"를 체크한다.

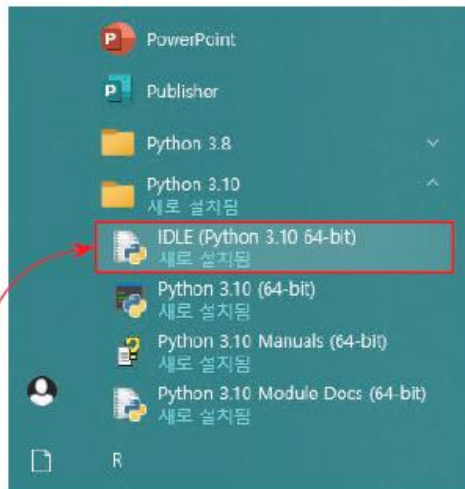
두 번째 체크 박스 "Add Python 3.5 to PATH"는 PATH 환경 변수에 파이썬을 추가할 것인지를 묻는다. 이 부분을 설치 시 체크하지 않으면 나중에 직접 환경 변수 PATH를 변경하여야 한다. PATH에 파이썬이 들어 있어야 편리하다. 반드시 체크하도록 하자.

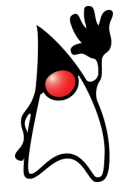


파이썬 시작하기



- 윈도우의 시작 메뉴에서 "IDLE" 프로그램을 찾아서 실행





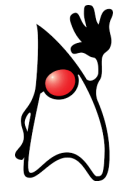
파이썬 셸



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

```
DLE Shell 3.10.7
File Edit Shell Debug Options Window Help
Python 3.10.7 (tags/v3.10.7:6cc6b13, Sep 5 2022, 14:08:36) [MSC v.1933 64 bit (AMD64)] on <os>
Type "help", "copyright", "credits" or "license()" for more information.
>>>
```





Hello World! 출력하기



```
IDLE Shell 3.10.7
File Edit Shell Debug Options Window Help
Python 3.10.7 (tags/v3.10.7:6cc6b13, Sep 5 2022, 14:08:36) [MSC v.1933 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> print("Hello World!")
Hello World!
>>>
```

Ln: 5 Col: 0



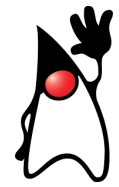


도전문제



도전문제

- (1) 한글도 출력될까? “안녕하세요?”를 화면에 출력하여 보자. 따옴표를 올바르게 입력하여야 한다.
- (2) “programming에 입문하신 것을 축하드립니다.”를 출력하여 보자.



계산하기 #1

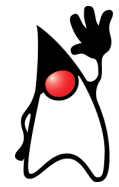


```
IDLE Shell 3.10.5
File Edit Shell Debug Options Window Help
>>> 2+3
5
>>> 2-3
-1
>>> 2*3
6
>>> 2/3
0.6666666666666666
>>>
```



이 정도는
암산으로도 할 수
있어요! 파이썬이
필요 없다고요!

Ln: 7 Col: 3



계산하기 #2

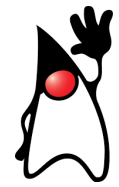


```
IDLE Shell 3.10.5
File Edit Shell Debug Options Window Help
>>> 2345*9876*5678
      23153542
>>> |
```

Ln: 13 Col: 0



이 정도라면
계산기로
하면 되죠!



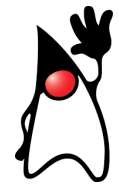
계산하기 #3



```
IDLE Shell 3.10.5
File Edit Shell Debug Options Window Help
>>> 123456789123456789*123456789123456789
      15241578780673678515622620750190521
>>>
```



이것은
정말 계산기로
안 되네요!



도전문제

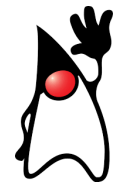


도전문제

파이썬의 IDLE를 이용하여 다음과 같은 계산을 하여 보자.

(1) $3.141592 * 10.0 * 10.0$

(2) $(1/100) * 1234$

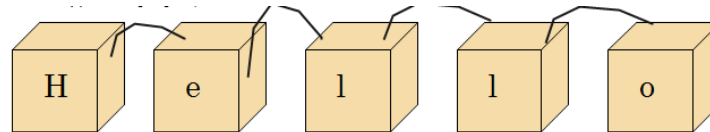


문자열이란?

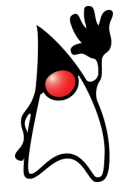


- 문자열(string) : 큰따옴표("...")나 작은따옴표('...') 안에 들어 있는 텍스트 데이터

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



- 반드시 따옴표가 있어야 한다.
 - “100” – 문자열
 - 100 - 정수

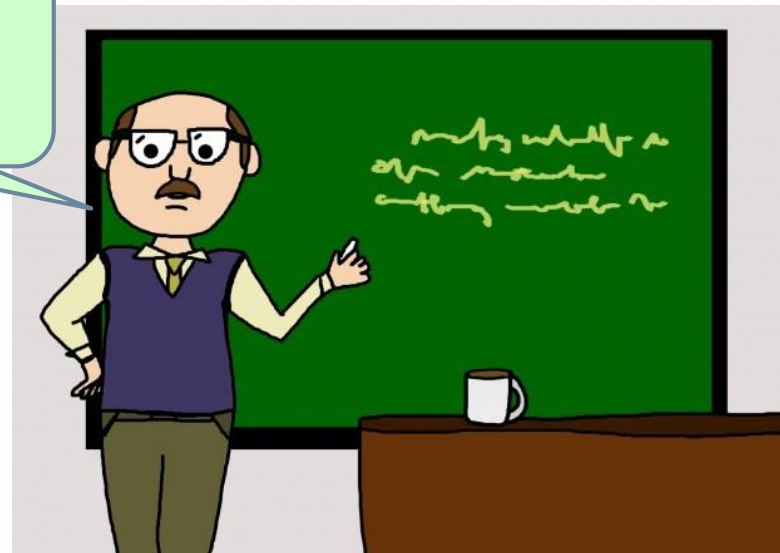


문자열 합치기



```
>>> "강아지" + "고양이"  
'강아지고양이'  
>>>
```

+은
문자열에서는
접합을 의미
합니다.





문자열 반복하기

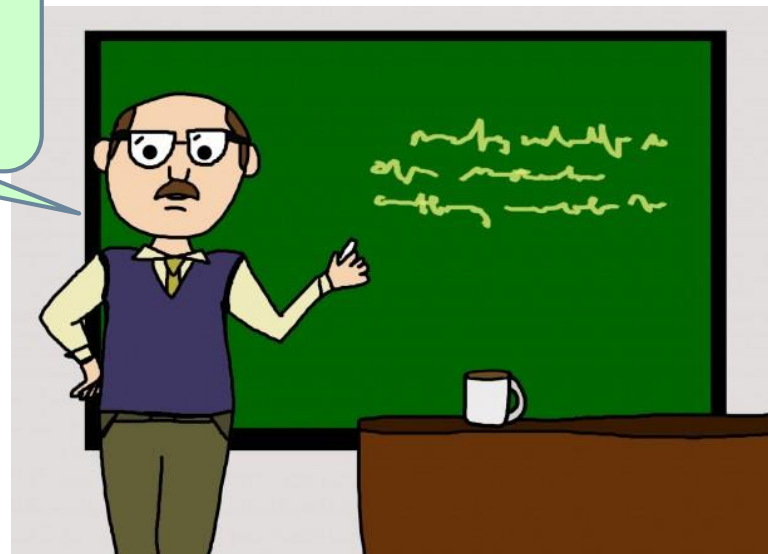


>>> "반가워요 " * 20

반가워요 반가워요 반가워요 반가워요 반가워요 반가워요 반가워요 반가워요
반가워요 반가워요 반가워요 반가워요 반가워요 반가워요 반가워요 반가워요
반가워요 반가워요 반가워요 반가워요

>>>

*은 반복을
의미하기도
합니다.





문자열과 정수의 차이점



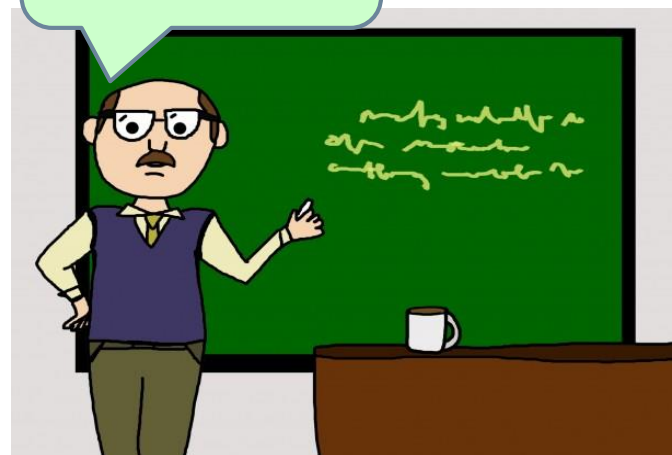
```
>>> "100" + "200"  
'100200'
```

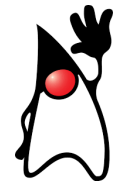
```
>>> 100 + 200  
300
```

문자열은 어떻게
구별하나요?



따옴표("...")가
붙으면
문자열입니다.



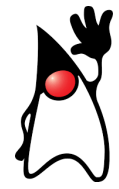


print() 함수



- 여러 개의 값들을 화면에 차례대로 출력할 수 있다.

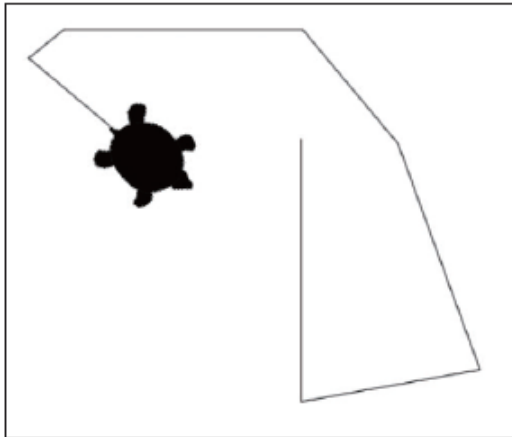
```
>>> print("결과값은", 2*7, "입니다.")  
결과값은 14 입니다.
```



터틀 그래픽

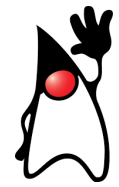


- 터틀 그래픽은 화면에서 거북이를 이용하여서 그림을 그리는 기능이다.



터틀 그래픽은
거북이가 화면 위에서
지나가면서 그림을
그리는 것이라고
생각하세요.

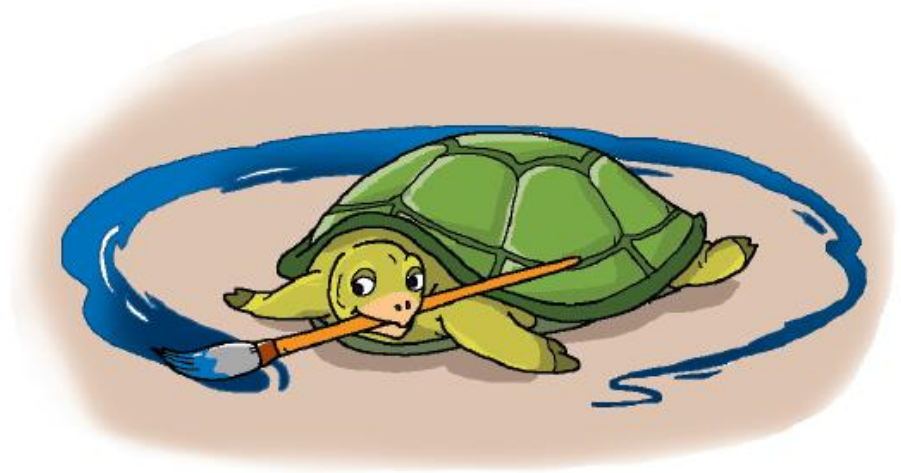


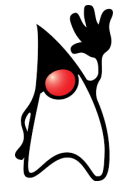


터틀 그래픽



1. 터틀 그래픽을 시작하면 종이의 한 가운데 거북이가 나타난다.
2. 거북이에게 명령을 내리면 거북이가 움직인다. 예를 들어서 “앞으로 전진”, “뒤로 후진”, “왼쪽으로 방향 전환” 등의 명령을 사용할 수 있다.
3. 거북이가 움직이면서 종이 위에 그림이 그려진다. 거북이가 펜을 가지고 움직인다고 생각하면 된다.

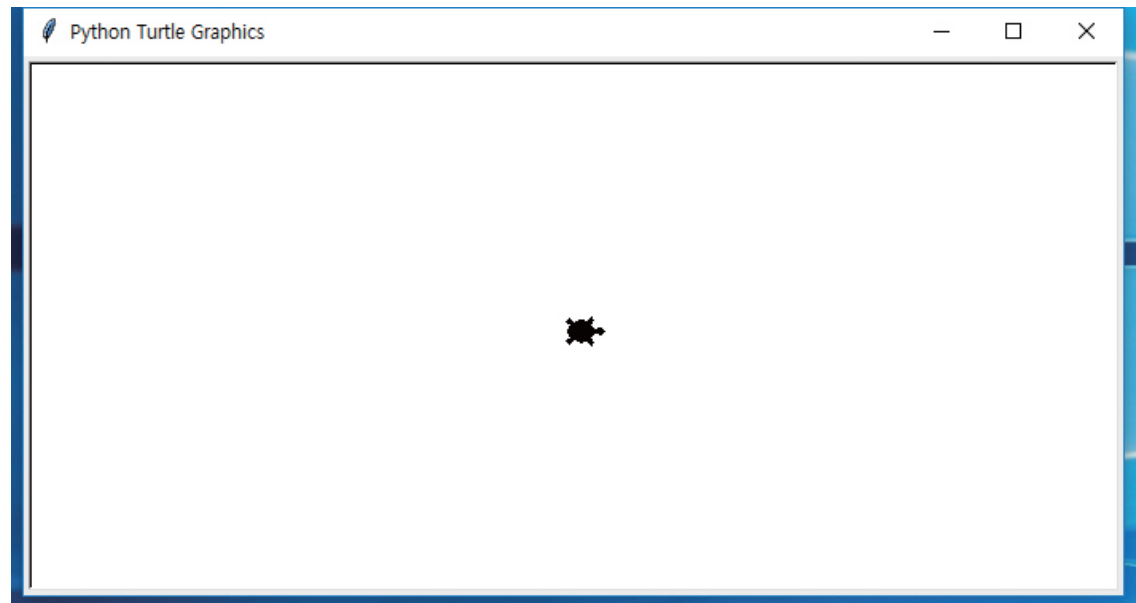


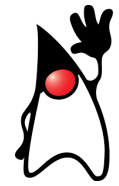


터틀 그래픽 시작



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

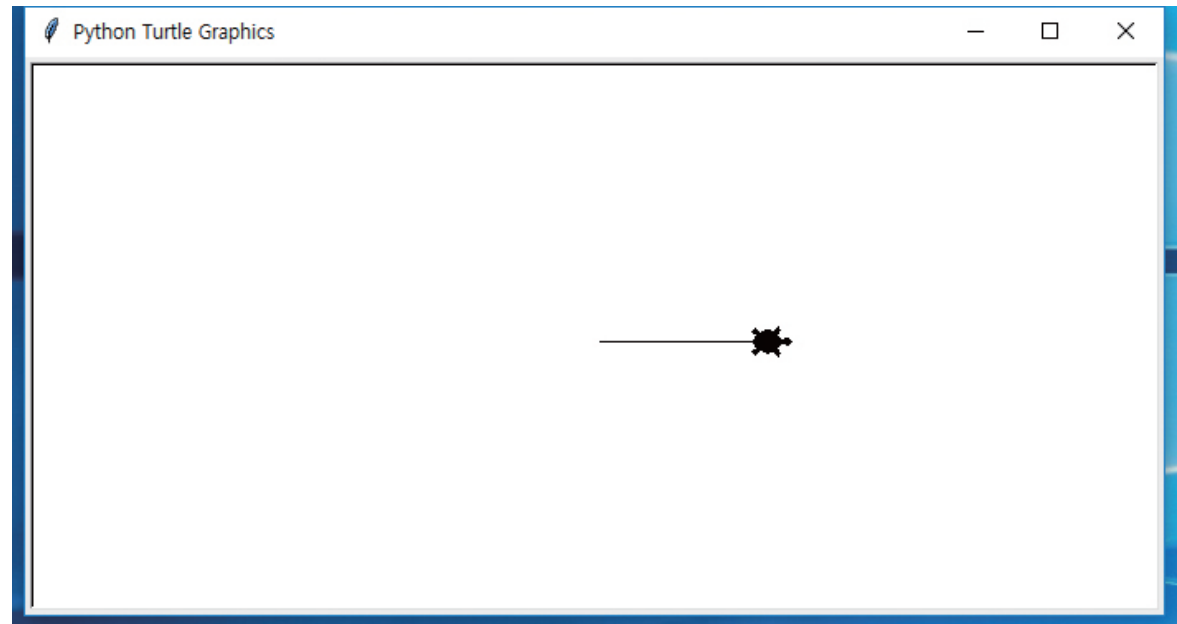


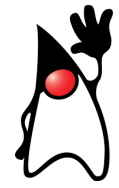


직선 그리기



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



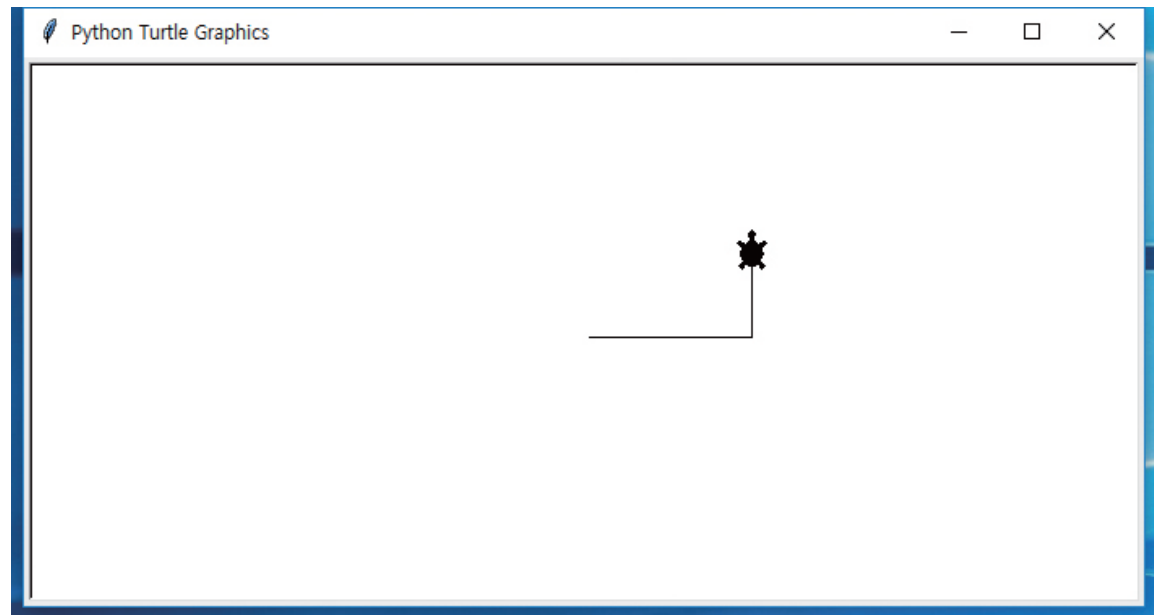


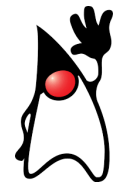
직선 그리기



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

>>> turtle.done()
```

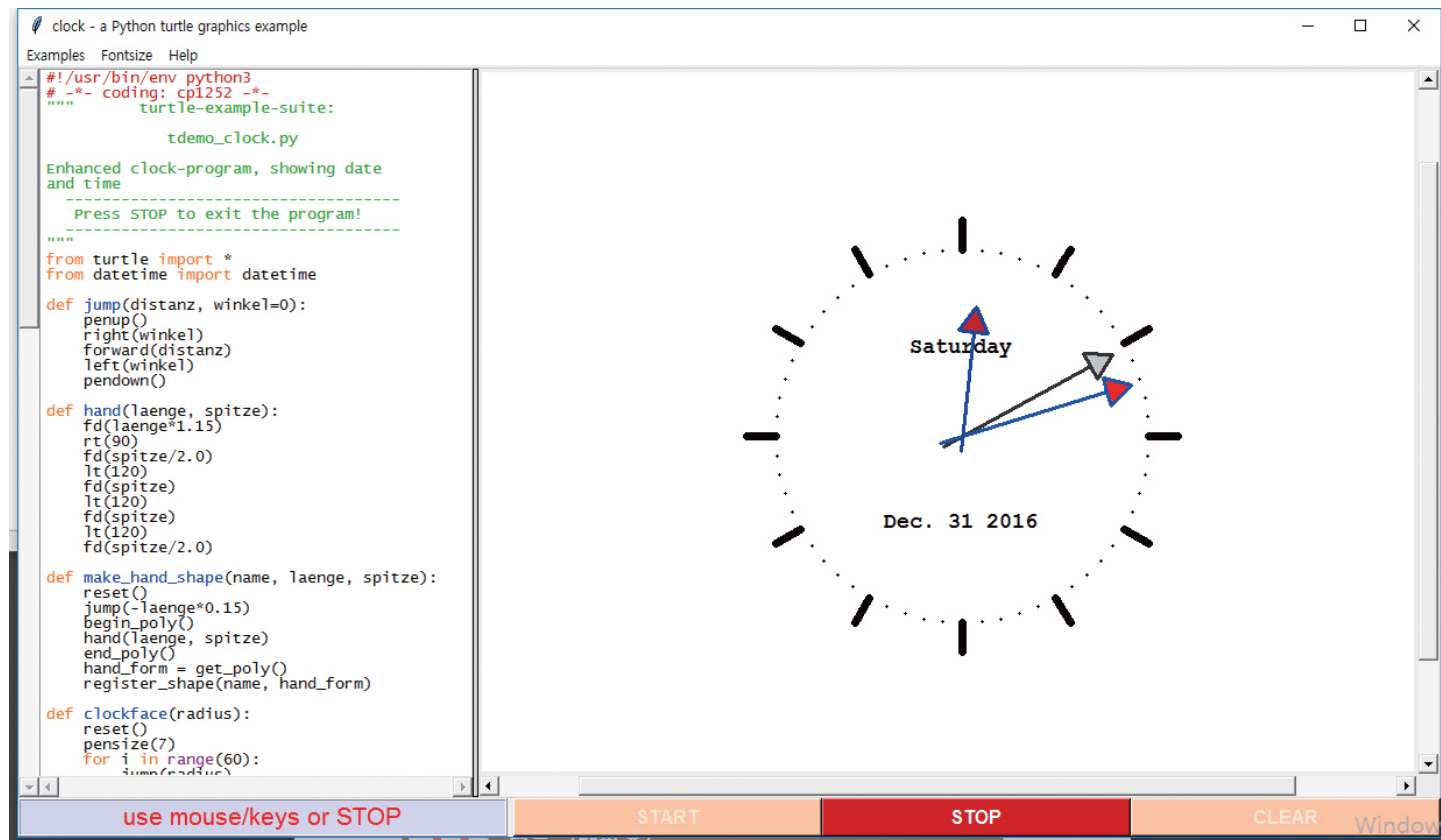


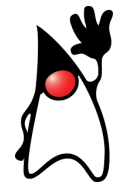


터틀 그래픽 데모 실행해보기



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

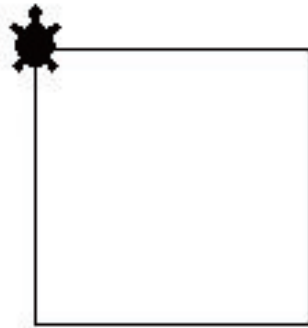




Lab: 터틀 그래픽으로 사각형을 그려보자



- 앞에서 터틀 그래픽에서 거북이를 전진시키고 회전하는 명령어들을 학습하였다. 이들 명령어를 이용하여서 다음과 같이 사각형을 그려보자



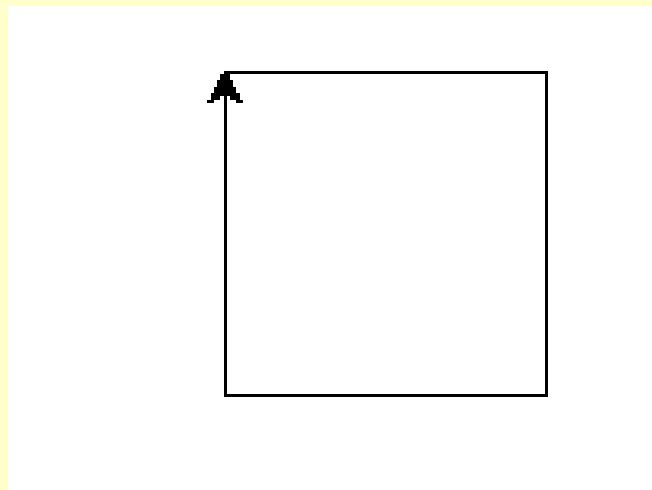


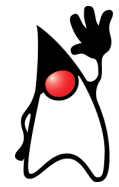
Sol: 터틀 그래픽으로 사각형을 그려보자



rect.py

```
>>> 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()
```



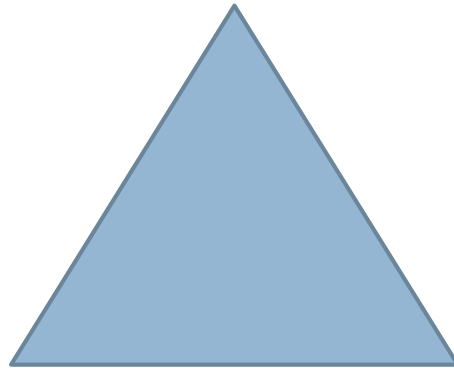


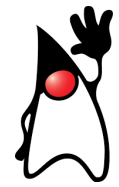
도전문제



도전문제

거북이를 움직여서 삼각형을 그려보자. 회전하는 각도를 몇 도로 하여야 하는가?





스크립트 모드

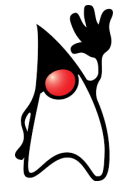


- 코드가 복잡해지면 인터프리트 모드는 번거롭다.

이걸 한 줄씩 입력하라고?



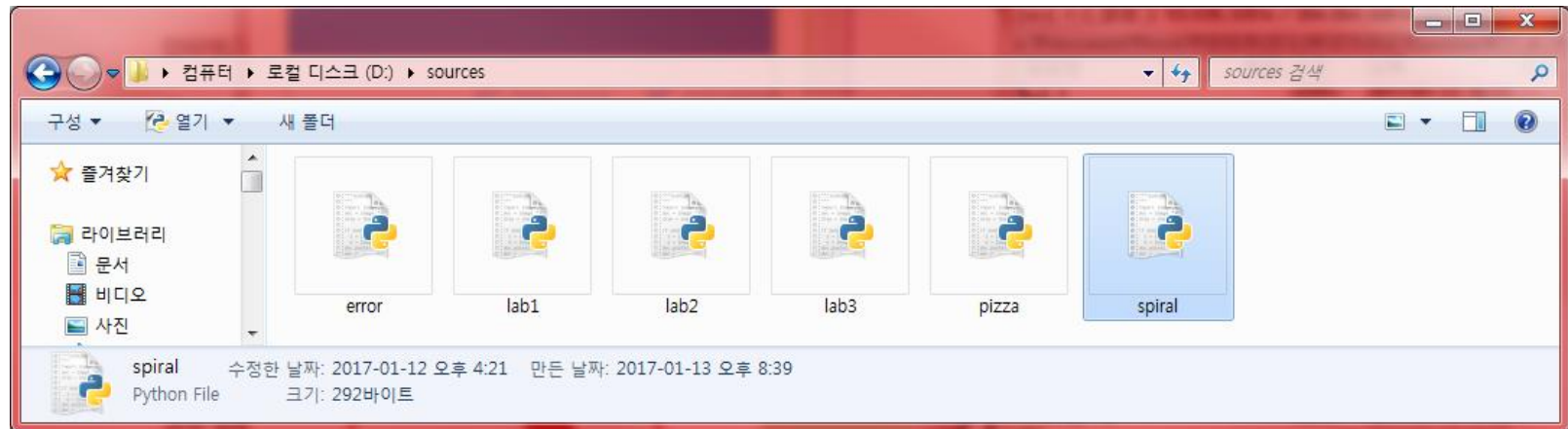
```
def add5(x):  
    return x+5  
  
def dotwrite(ast):  
    nodename = getNodeName()  
    label=symbol.sym_name.get(int(ast[0]),ast[0])  
    print '  %s [label="%s' % (nodename, label)  
    if isinstance(ast[1], str):  
        if ast[1].strip():  
            print '    %s';' % ast[1]  
        else:  
            print ']'  
    else:  
        print '];'  
        children = []  
        for n, child in enumerate(ast[1:]):  
            children.append(dotwrite(child))  
        print '  %s -> (' % nodename  
        for n, child in enumerate(children):  
            print '    %s' % child,
```



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.



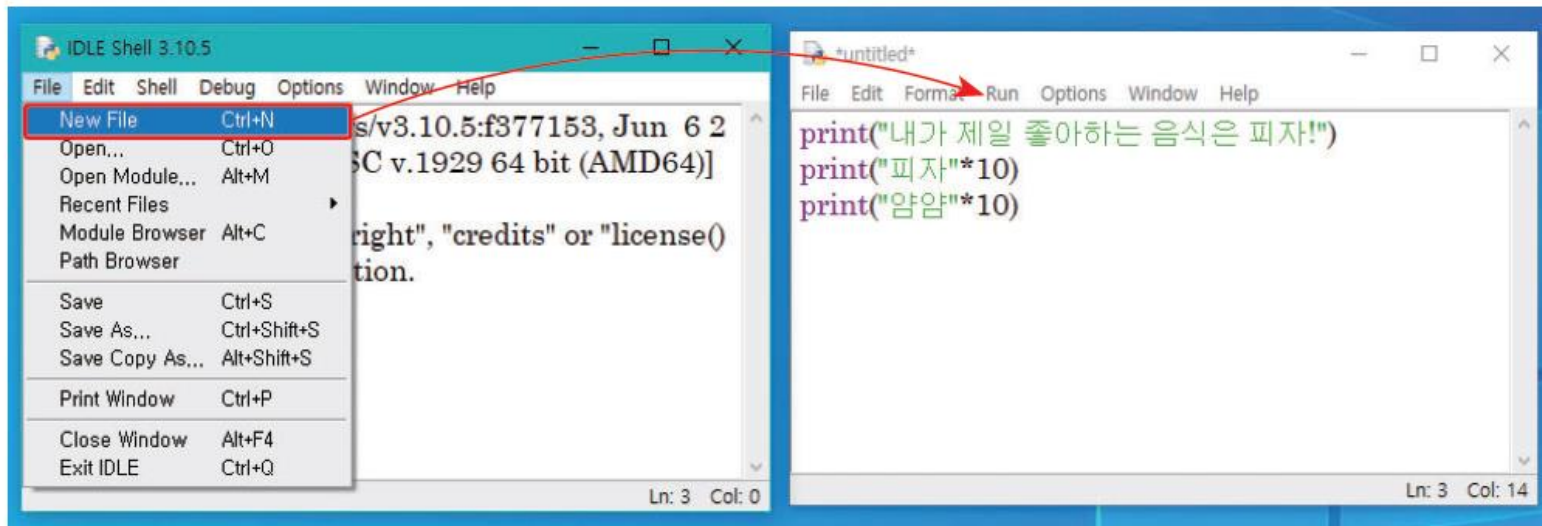


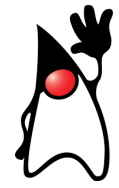
IDLE를 이용한 소스 파일 작성



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

pizza.py

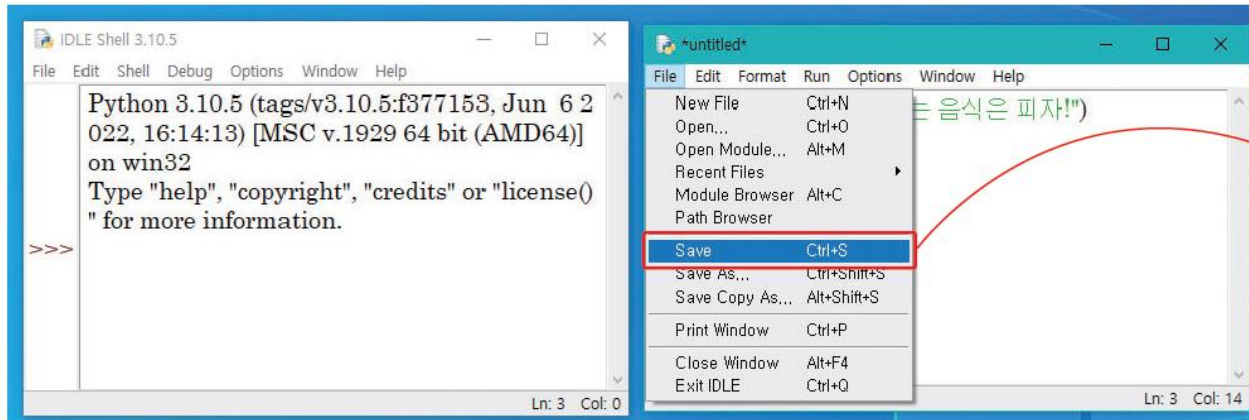


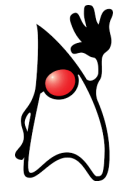


IDLE를 이용한 소스 파일 저장



- 텍스트 에디터의 [File]->[Save] 메뉴를 선택하여 코드를 파일로 저장한다.





IDLE를 이용한 소스 파일 실행



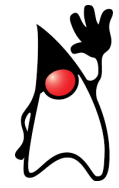
- 파이썬 쉘의 메뉴 중에서 메뉴 [Run]->[Run Module]을 선택한다.

The screenshot shows two windows from the IDLE 3.10.5 environment. The left window is the 'IDLE Shell 3.10.5' with a menu bar (File, Edit, Shell, Debug, Options, Window, Help). It displays the output of a Python script, including a restart message and Korean text. The right window is the 'test.py - D:/test/test.py (3.10.5)' editor with a menu bar (File, Edit, Format, Run, Options, Window, Help). The 'Run' menu is open, showing options: 'Run Module F5', 'Run... Customized Shift+F5', 'Check Module Alt+X', and 'Python Shell'. The 'Run Module' option is highlighted. A red arrow points from the 'Run Module' option to the output in the Shell window.

```
>>> 022, 16:14:13) [MSC v.1929 64 bit (AMD64)]
on win32
Type "help", "copyright", "credits" or "license()"
>>>
===== RESTART: D:/test/test.py =====
내가 제일 좋아하는 음식은 피자!
피자피자피자피자피자피자피자피자피자피자
암암암암암암암암암암암암암암암암암암
>>>
```

```
test.py - D:/test/test.py (3.10.5)
File Edit Format Run Options Window Help
Run Module F5
Run... Customized Shift+F5
Check Module Alt+X
Python Shell
```

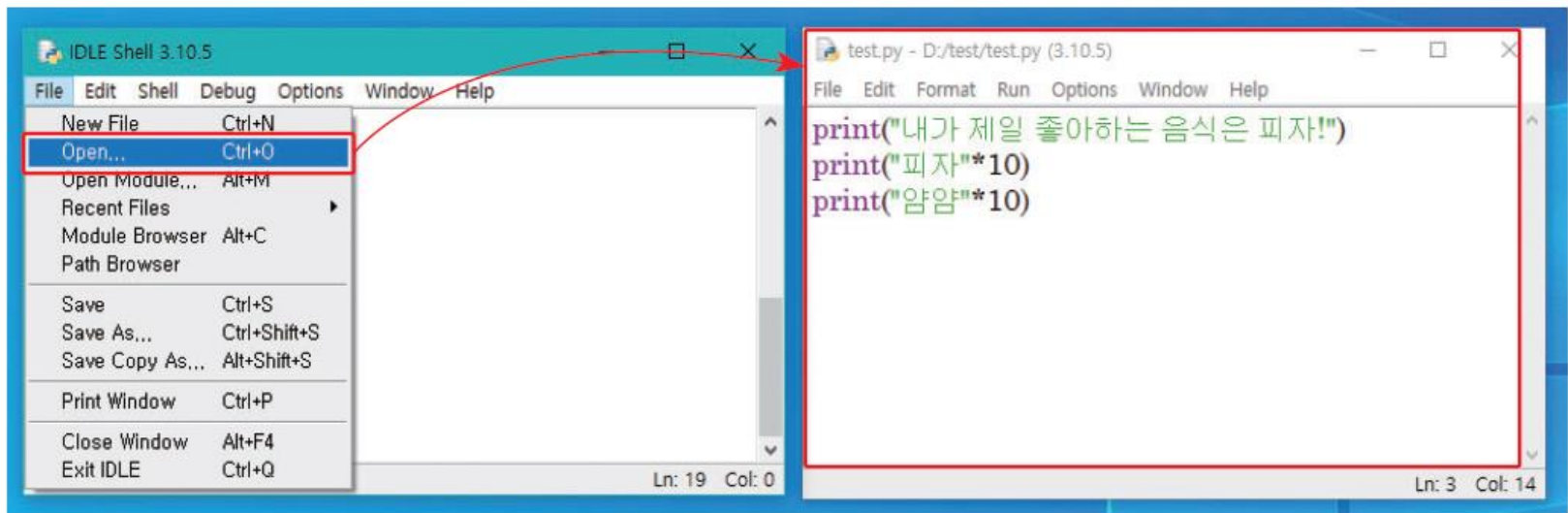
```
print("내가 제일 좋아하는 음식은 피자!")
print("피자"*10)
print("암암"*10)
```

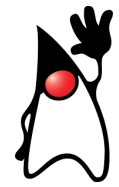


소스 파일 다시 열기



- IDLE의 [File]->[Open] 메뉴를 선택한다. 우리가 저장하였던 폴더로 가서 원하는 파일을 선택한다. 텍스트 에디터가 나오고 우리가 입력하였던 소스가 다시 보일 것이다.

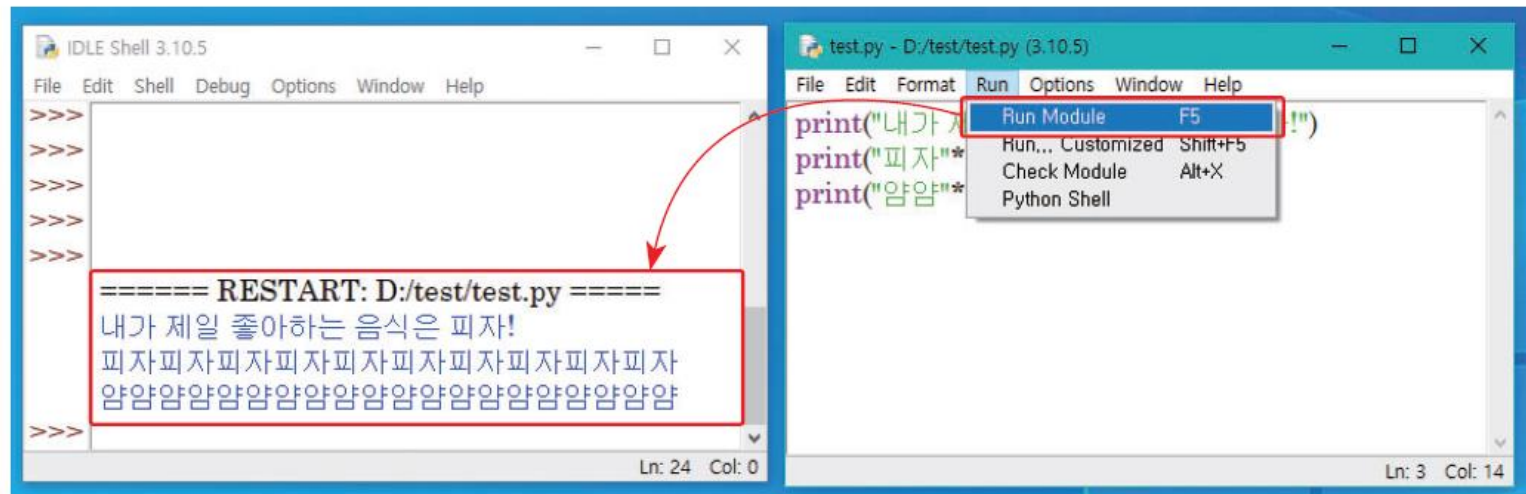


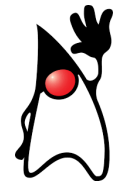


소스 파일 다시 실행하기



- 이 소스 파일을 다시 실행하려면 [Run] → [Run Module] 메뉴를 선택하면 된다.





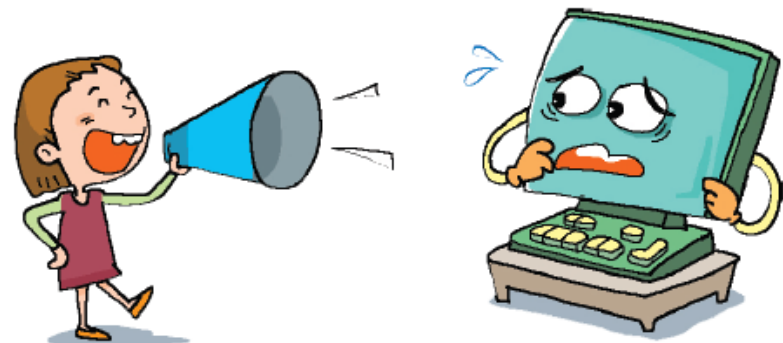
문법적인 오류



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

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

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





실행시 오류



```
print("안녕하세요? 파이썬에 오신 것을 환영합니다!")  
print("프로그래밍 공부를 즐기셨으면 합니다.")  
print("안녕!" + 3)
```

*print("안녕!" * 3)로 수정*

```
===== RESTART: D:\s.py =====  
안녕하세요? 파이썬에 오신 것을 환영합니다!  
프로그래밍 공부를 즐기셨으면 합니다.  
Traceback (most recent call last):  
File "D:\s.py", line 3, in <module>  
print("안녕!" + 3)  
TypeError: Can't convert 'int' object to str implicitly
```

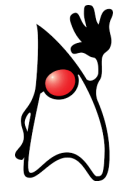


오류가 수정된 프로그램



```
print("안녕하세요? 파이썬에 오신 것을 환영합니다!")  
print("프로그래밍 공부를 즐기셨으면 합니다.")  
print("안녕!" * 3)
```

안녕하세요? 파이썬에 오신 것을 환영합니다!
프로그래밍 공부를 즐기셨으면 합니다.
안녕!안녕!안녕!



Lab: print() 함수 실습



- 빈칸을 채워본다.

```
>>> _____
```

안녕하세요? 여러분

```
>>> _____
```

저는 파이썬을 무척 좋아합니다.

```
>>> _____
```

9*8은 72 입니다.

```
>>> _____
```

안녕히 계세요.



Sol: print() 함수 실습



```
>>> print("안녕하세요? 여러분")
```

안녕하세요? 여러분

```
>>> print("저는 파이썬을 무척 좋아합니다.")
```

저는 파이썬을 무척 좋아합니다.

```
>>> print("9*8은", 9*8, "입니다.")
```

9*8은 72 입니다.

```
>>> print("안녕히 계세요.")
```

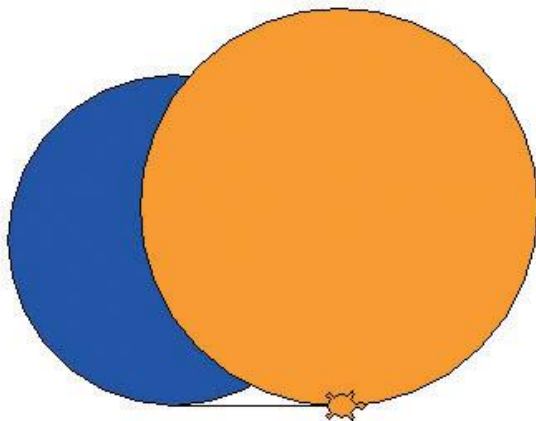
안녕히 계세요.



Lab: 컬러 원을 그려보자



- 터틀 그래픽을 사용하여 색상으로 채워진 원을 그려보자. 원을 그리는 명령어는 다음과 같다.



```
t.fillcolor("blue") # 채우는 색상 지정  
t.begin_fill() # 채우기 시작  
t.circle(100) # 반지름이 100인 원이 그려진다.  
t.end_fill() # 채우기 종료
```



Sol: 컬러 원을 그려보자

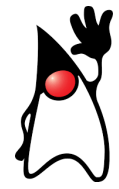


circle.py

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

t.shape("turtle")           # 거북이 모양으로 설정
t.fillcolor("blue")         # 채우는 색상 지정
t.begin_fill()              # 채우기 시작
t.circle(100)               # 반지름이 100인 원이 그려진다.
t.end_fill()                # 채우기 종료
t.forward(100)              # 100 만큼 전진
t.fillcolor("orange")
t.begin_fill()
t.circle(120)               # 반지름이 120인 원이 그려진다.
t.end_fill()

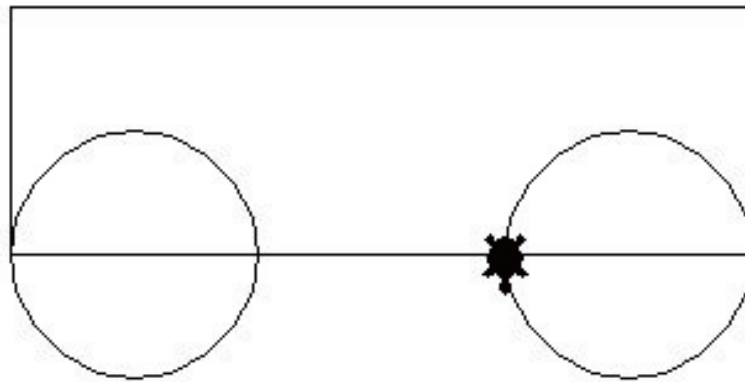
turtle.done()
```

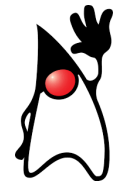


Lab: 자동차를 그려보자



- 터틀 그래픽을 이용하여 자동차 모양을 그려보자.





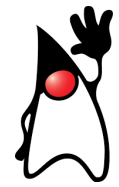
Sol:



car.py

```
import turtle

t = turtle.Turtle()
t.shape("turtle")
t.forward(300) # 자동차 몸체를 그린다.
t.left(90)
t.forward(100)
t.left(90)
t.forward(300)
t.left(90)
t.forward(100)
```



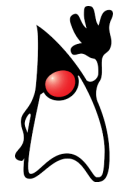
Sol:



car.py

```
# 타이어를 그린다.  
t.up() # 펜을 들어서 거북이가 움직이더라도 그림이 그려지지 않게 한다.  
t.goto(0, 0) # 좌표 (0, 0)으로 간다  
t.down() # 펜을 내린다.  
t.circle(50) # 원을 그린다.
```

```
t.up()  
t.goto(200, 0)  
t.down()  
t.circle(50)  
turtle.done()
```



파이썬으로 무엇을 만들 수 있을까?



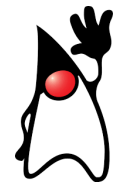
- 아래의 소스를 입력하고 실행해보자.

spiral.py

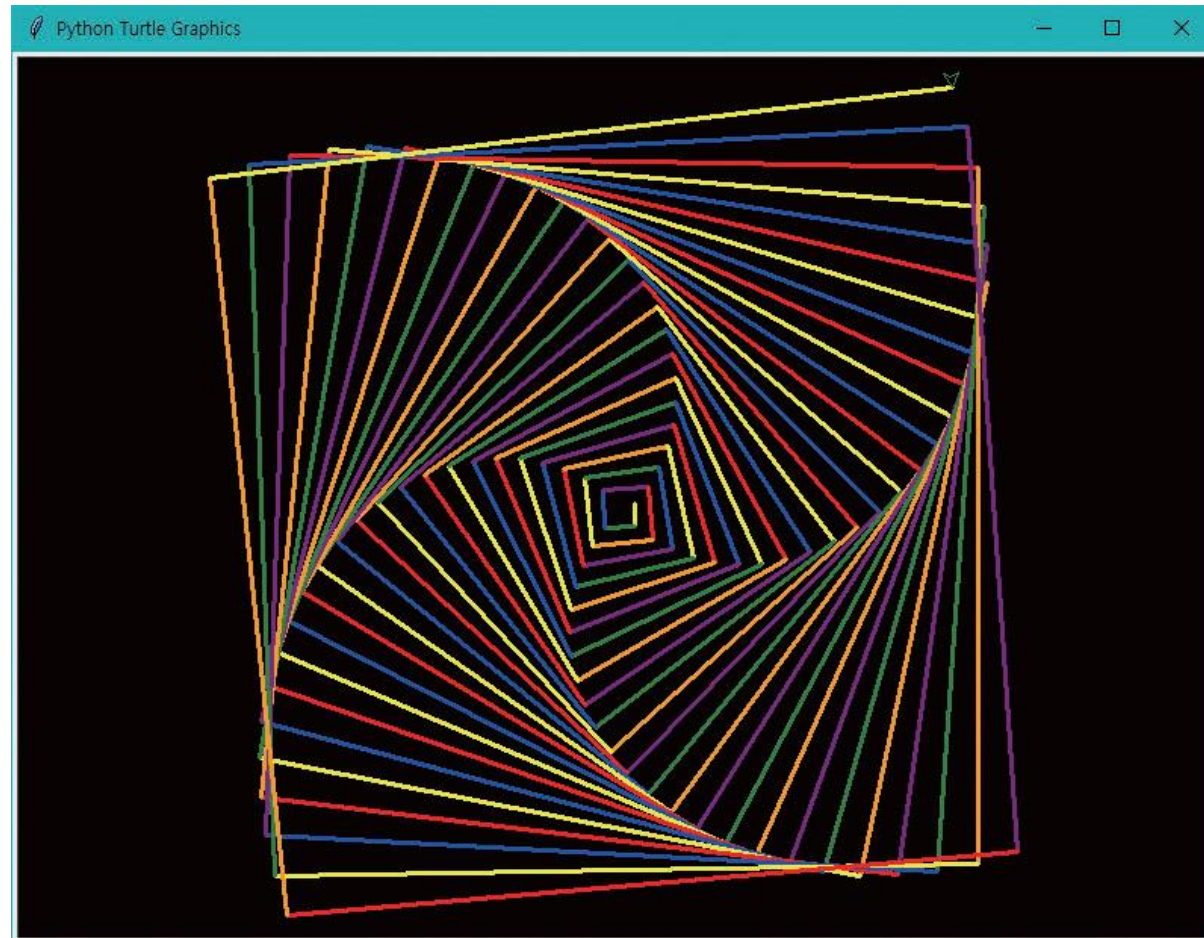
```
import turtle
colors = ["red", "purple", "blue", "green", "yellow", "orange"]
t = turtle.Turtle()

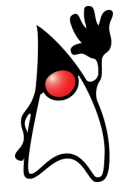
turtle.bgcolor("black")
t.speed(0)
t.width(3)
length = 10

while length < 500:
    t.forward(length)
    t.pencolor(colors[length%6])
    t.right(89)
    length += 5
turtle.done()
```



실행 결과는? 실행 결과는?



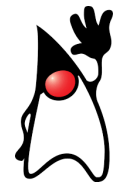


이번장에서 배운 것



- **프로그래밍 언어의 역할을 설명할 수 있나요?**
 - 프로그래밍 언어는 컴퓨터가 이해할 수 있는 언어입니다. 개발자의 의도를 컴퓨터에 전달하는 일종의 번역가의 역할을 합니다.
- **파이썬을 컴퓨터에 설치할 수 있나요?**
 - 파이썬은 <http://www.python.org> 웹사이트에서 다운로드받아서 설치할 수 있습니다.
- **간단한 파이썬 프로그램을 작성하여 실행할 수 있나요?**
 - IDLE은 파이썬으로 프로그램을 작성하기 위한 개발 환경입니다. 파이썬 셸에서는 >>> 프롬프트 다음에 코드를 입력하고 **Enter** 를 누르면 코드가 실행됩니다.



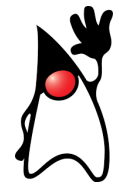


이번 장에서 배운 것



- 파이썬 코드를 파일에 저장하여 실행할 수 있나요?
 - IDLE을 이용하여 스크립트 모드나 대화형 모드로 프로그램을 개발할 수 있습니다. 스크립트 모드를 사용하면 코드를 파일에 저장하여 실행할 수 있습니다.
- 터틀 그래픽으로 여러 가지 그림을 그릴 수 있나요?
 - 터틀 그래픽은 거북이에 펜을 매달아서 그림을 그리는 개념입니다. 거북이를 이동하면 그림이 그려 집니다.
- `print()` 함수의 사용법을 설명할 수 있나요?
 - `print()`는 화면에 문자열이나 계산 결과를 출력할 수 있습니다.





Q & A

