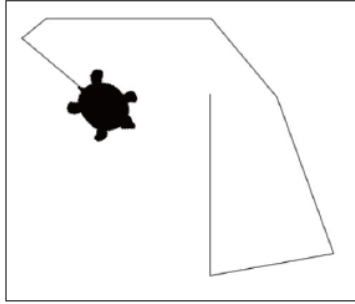


# 터틀 그래픽

- 터틀 그래픽은 화면에서 거북이를 이용하여서 그림을 그리는 기능이다.



터틀 그래픽은 거북이가 화면 위에서 지나가면서 그림을 그리는 것이라고 생각하세요.



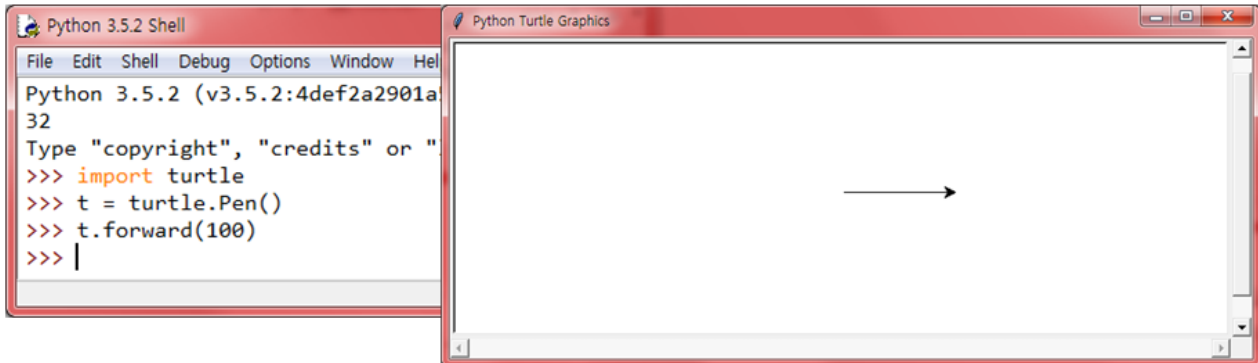
# 터틀 그래픽 시작

- 파이썬 셸에서 다음과 같이 입력하여 보자.

```
Python 3.5.2 Shell
File Edit Shell Debug Options Window Help
Python 3.5.2 (v3.5.2:4def2a2901a
32
Type "copyright", "credits" or "
>>> import turtle
>>> t = turtle.Pen()
>>>
```

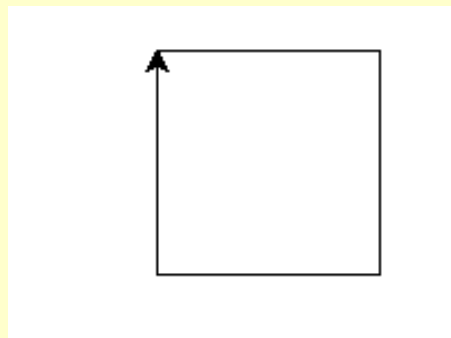
# 직선 그리기

- `t.forward(100)`



# 사각형 그리기

```
>>> import turtle
>>> t = turtle.Pen()
>>> t.forward(100)
>>> t.right(90)
>>> t.forward(100)
>>> t.right(90)
>>> t.forward(100)
>>> t.right(90)
>>> t.forward(100)
```

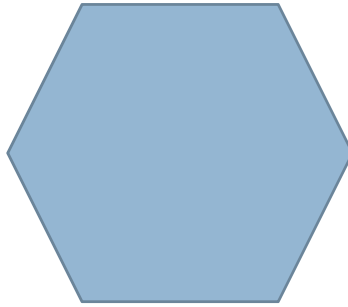


# 도전문제



## 도전문제

화살표를 움직여서 6각형을 그려보자. 회전하는 각도를 몇 도로 하여야 하는가?



# 스크립트 모드

- 코드가 복잡해지면 인터프리터 모드는 번거롭다.

이걸 한 줄씩 입력하라고?

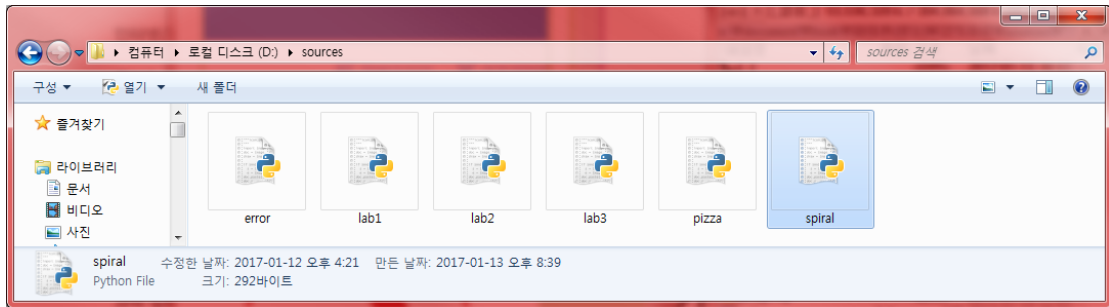


```
def add5(x):
    return x+5

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 in n, childrenumerate(ast[1:]):
            children.append(dotwrite(child))
        print ' %s -> (' % nodename
        for in 'namechildren
            print '%s' % name,
```

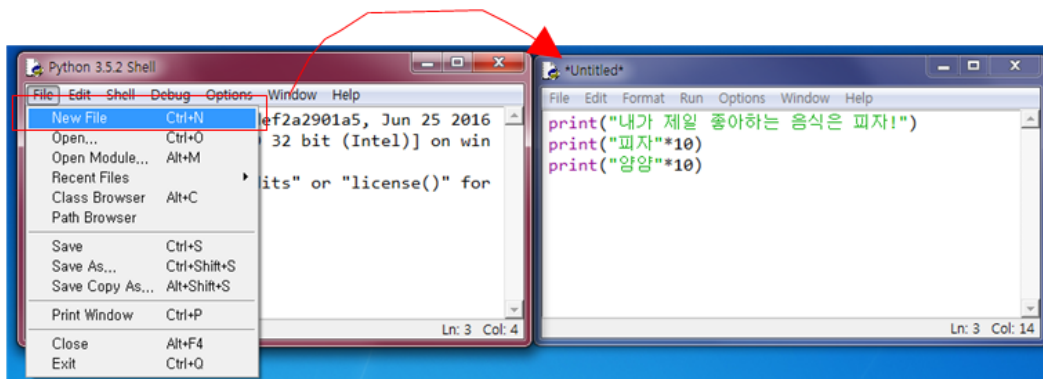
# 소스 파일 작성하기

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



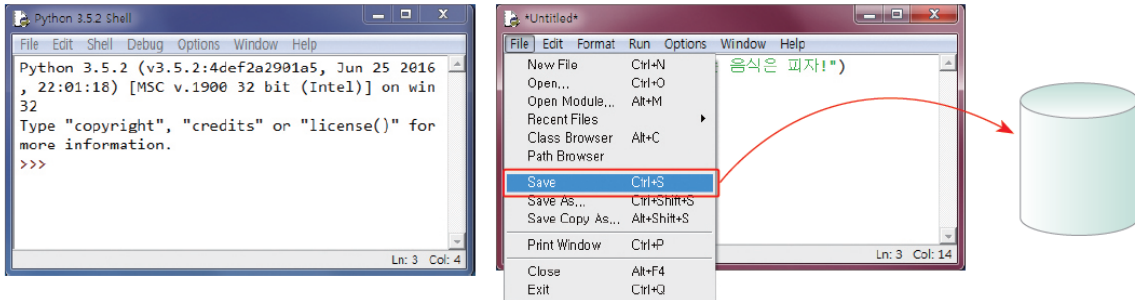
# IDLE를 이용한 소스 파일 작성

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



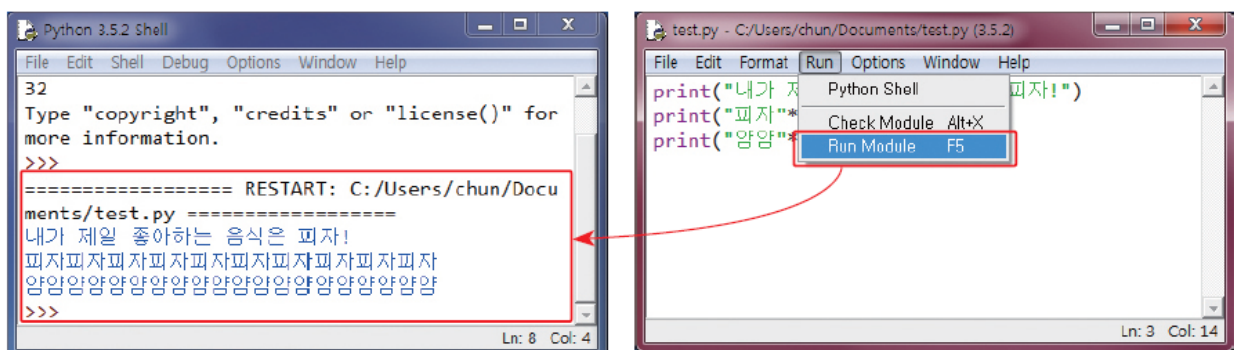
# IDLE를 이용한 소스 파일 저장

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



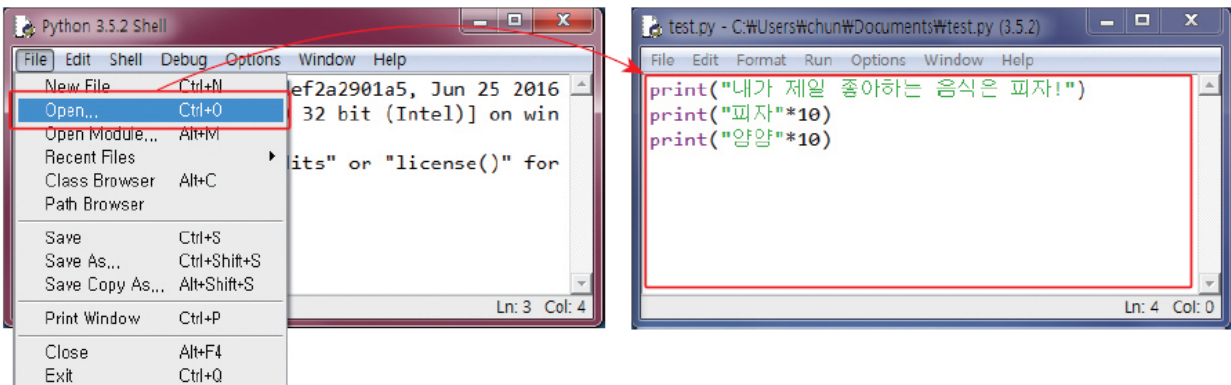
# IDLE를 이용한 소스 파일 실행

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



## 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.

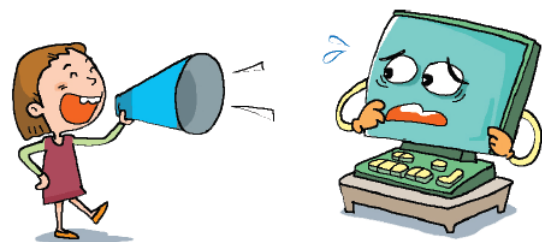


## 문법적인 오류

```
>>> 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
```

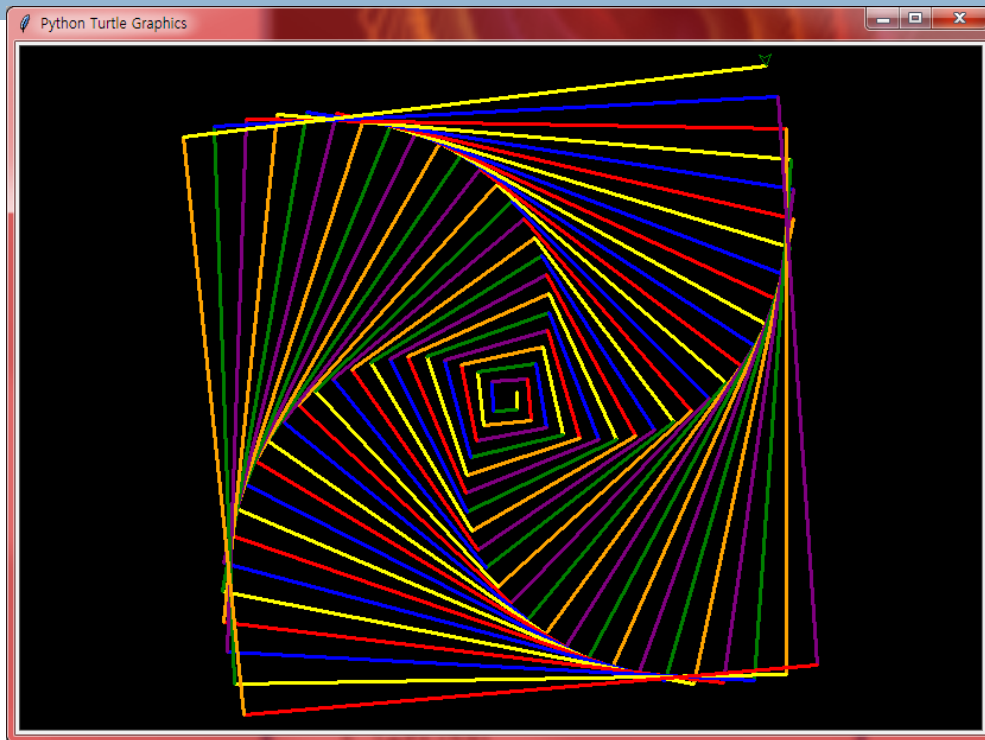
## 파이썬으로 무엇을 만들 수 있을까?

- 아래의 소스를 입력하고 실행해보자.

```
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
```



# 실행 결과는? 리ㅇ 리ㅇ 리ㅇ



## Lab: print() 함수 실행

- 빈칸을 채워본다.

```
>>> _____
```

안녕하세요? 여러분

```
>>> _____
```

저는 파이썬을 무척 좋아합니다.

```
>>> _____
```

9\*8은 72 입니다.

```
>>> _____
```

안녕히 계세요.



# Solution

```
>>> print("안녕하세요? 여러분")
안녕하세요? 여러분
```

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

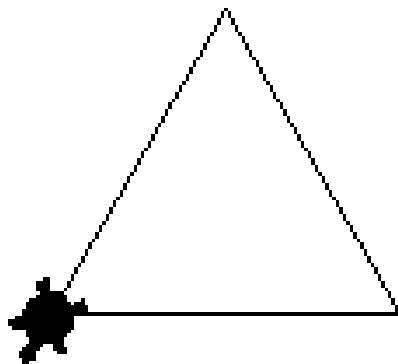
```
>>> print("9*8은", 9*8, "입니다.")
9*8은 72 입니다.
```

```
>>> print("안녕히 계세요.")
안녕히 계세요.
```

## Lab: 터틀 그래픽으로 삼각형을 그려보자

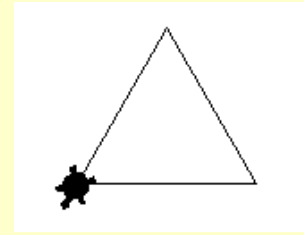


- 터틀 그래픽을 이용하여 삼각형을 그려보자.



# Solution

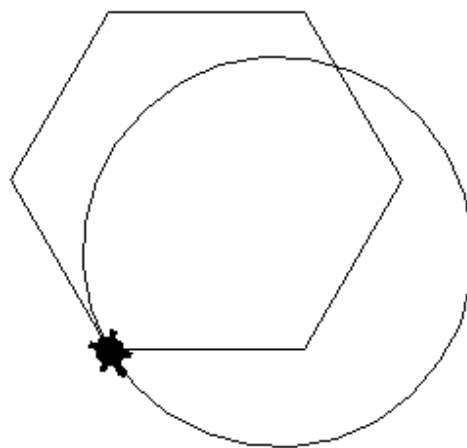
```
>>> import turtle
>>> t=turtle.Pen()
>>> t.shape("turtle")
>>> t.forward(100)
>>> t.left(120)
>>> t.forward(100)
>>> t.left(120)
>>> t.forward(100)
```



## Lab: 터틀 그래픽으로 원, 다각형을 그려보자



- 터틀 그래픽을 이용하여 원, 다각형을 그려보자.



Run Python

# Solution

```
import turtle
t = turtle.Turtle()
t.shape("turtle")
t.forward(100)
t.left(60)
t.forward(100)
t.left(60)
t.forward(100)
t.left(60)
t.forward(100)
t.left(60)
t.forward(100)
t.left(60)
t.forward(100)
t.left(60)
t.forward(100)

t.circle(100)      # 반지름이 100인 원이 그려진다.
```

## 이번 장에서 배운 것

- 프로그램은 컴퓨터에 내리는 명령으로 이루어지는 작업지시서이다.
- 다양한 종류의 프로그래밍 언어가 있고 파이썬도 프로그래밍 언어의 일종이다.
- 파이썬은 <http://www.python.org> 웹사이트에서 다운로드 받아서 설치할 수 있다.
- IDLE은 파이썬으로 프로그램을 작성하기 위한 개발 환경이다.
- 파이썬에서 산술 계산을 하는 연산자에는 +, -, \*, /가 있다.
- print()는 화면에 문자열이나 계산 결과를 출력할 수 있다.
- 스크립트 모드를 사용하면 코드를 파일에 저장하였다가 한 꺼번에 실행할 수 있다.



# Q & A

