

파이썬 프로그래밍 입문

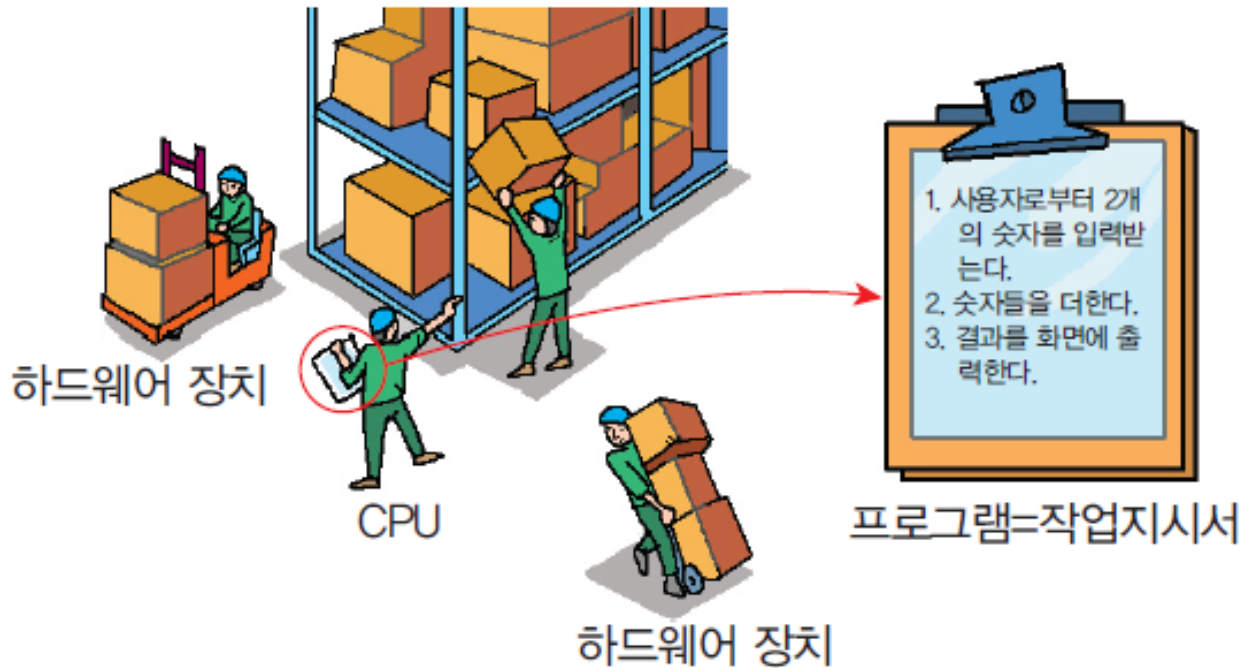
파이썬 소개 / 프로그래밍 과정

최 윤 정

cris.lecture@gmail.com

컴퓨터 프로그램

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



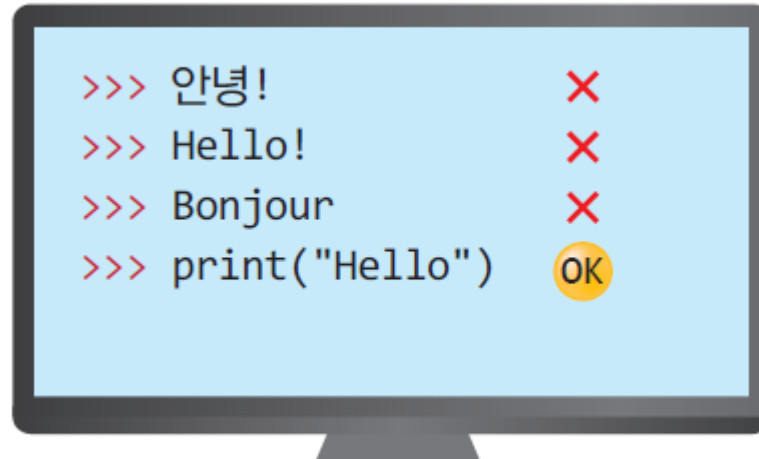
프로그래밍 언어

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



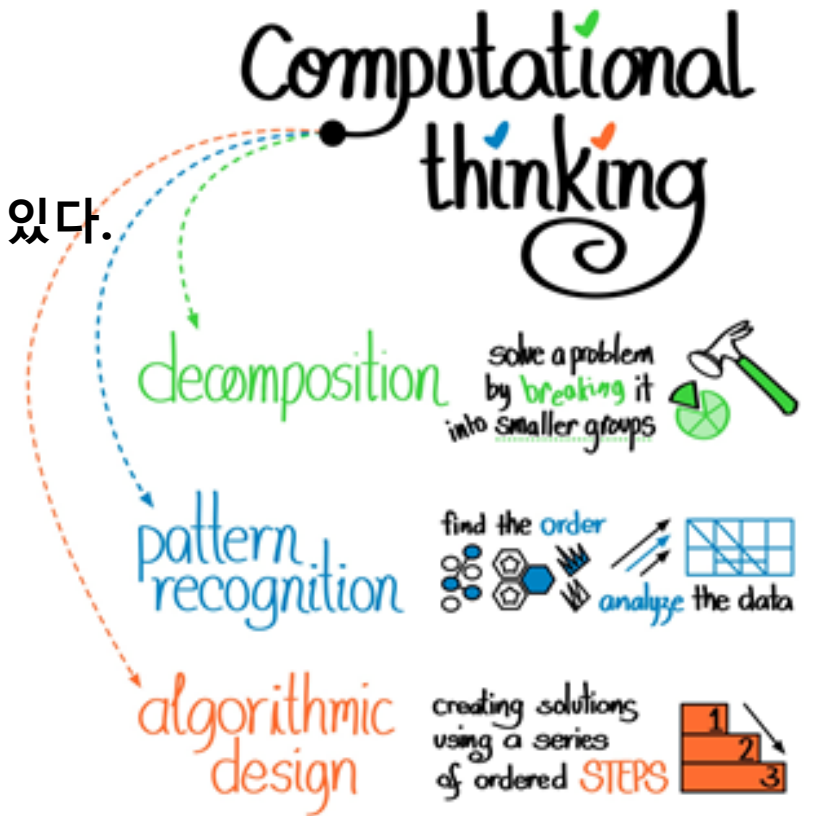
프로그래밍 언어

- 컴퓨터는 사람의 언어를 이해할 수 없다.
 - 내부적으로는 10101111 같은 기계어(이진수)를 사용함
- '프로그래밍 언어'는 컴퓨터가 이해하는 언어이다.
 - `print("Hello")` 로 작성된 문장은 '컴파일러'(해석기) 에 의해
 - 컴퓨터가 이해하는 말로 변환됨.

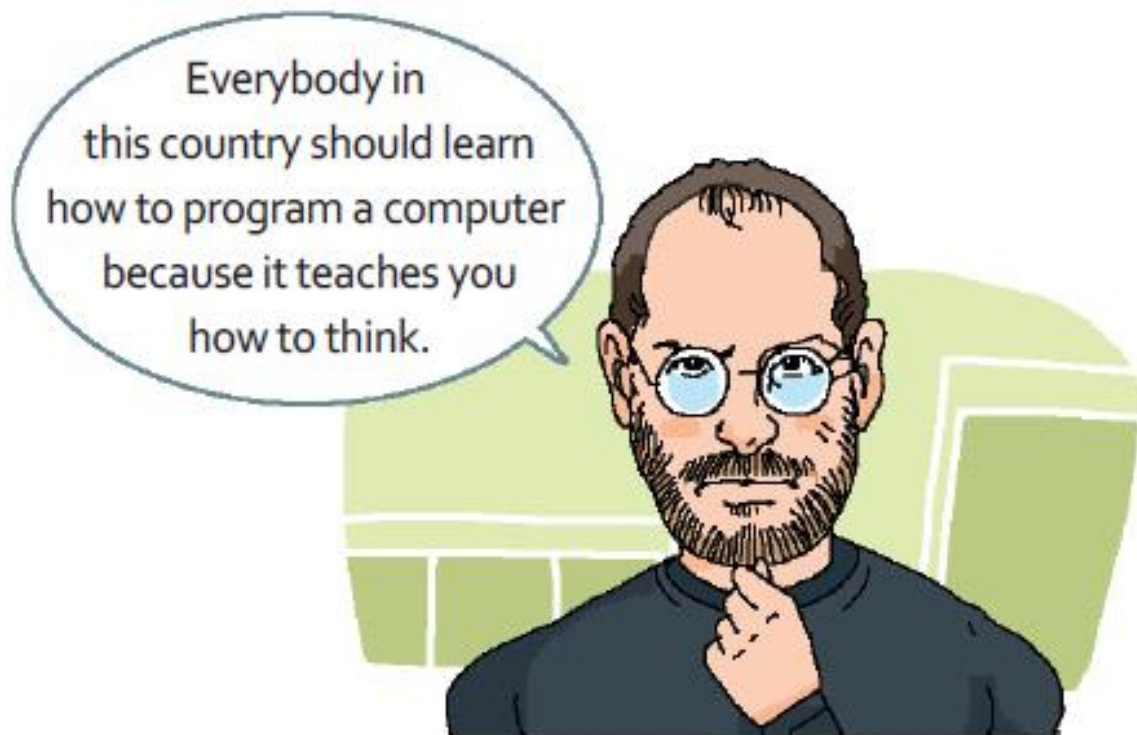


프로그래밍이 어디에 도움이 될까?

- 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.



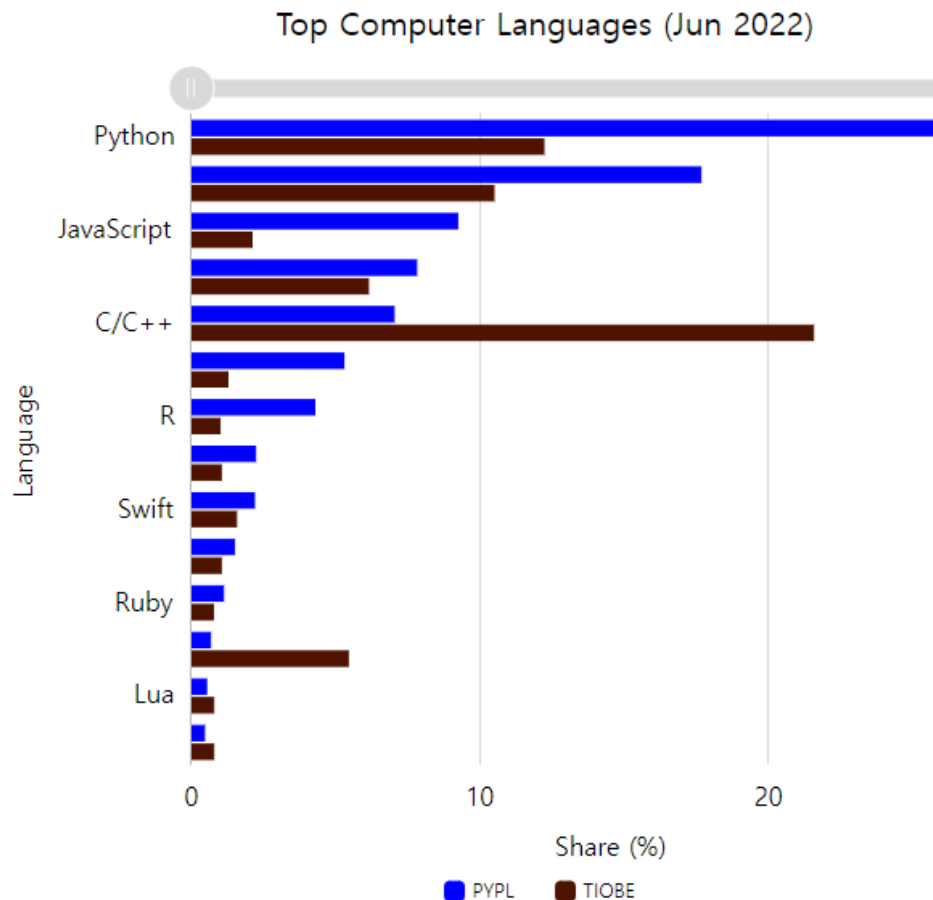
스티브 잡스 왈



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.
- 잠깐 구글링 : 많이 사용되는 컴퓨터언어



티오베 지수 : 구글, Bing, 위키피디아 같은 검색 엔진에서의 검색빈도. 이 수식은 언어와 관련된 숙련된 엔지니어, 학습 과정, 써드파티 업체의 수를 평가한다.

PYPL 지수 : 구글내에서의 검색빈도

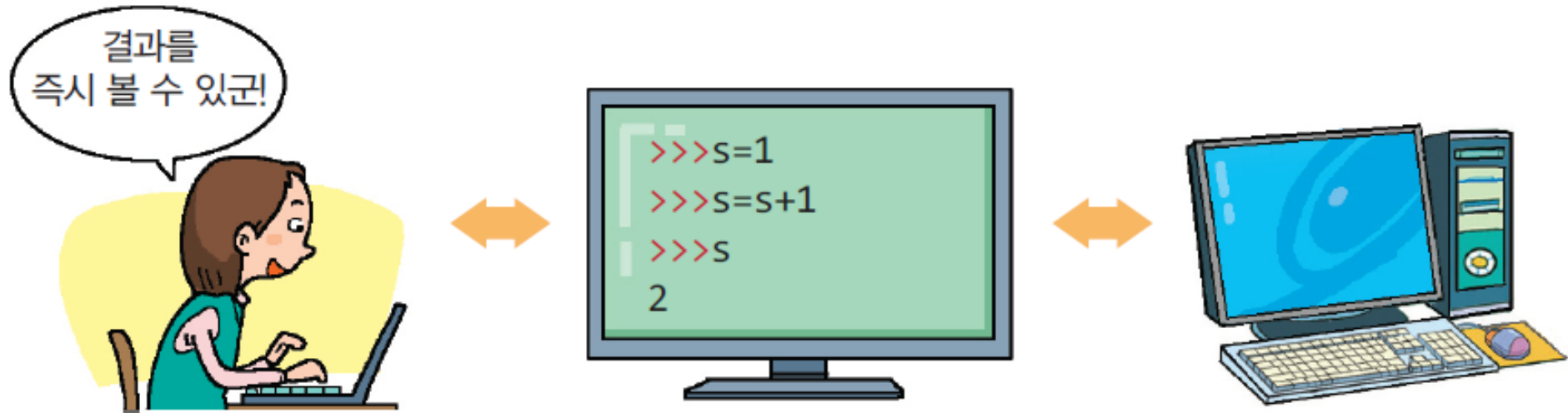
파이썬

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



파이썬의 특징

- 쉽다! 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")
```

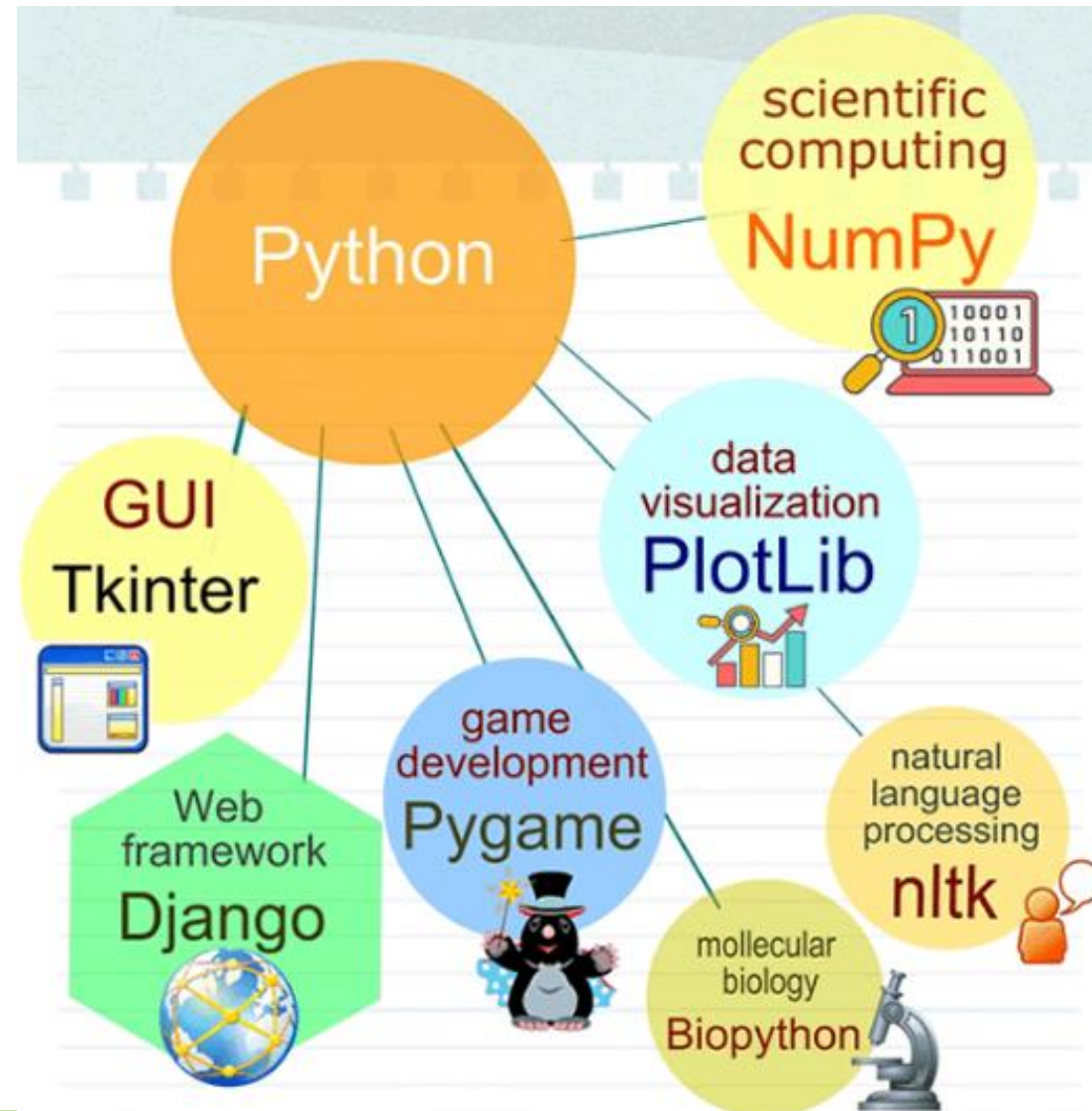
- 파이썬은 다양한 플랫폼에서 사용
 - 라이브러리가 풍부
 - 애니메이션이나 그래픽을 쉽게 사용
-

파이썬을 사용하고 있는 기업들

Top Companies Using Python



파이썬의 라이브러리



개인 PC에 파이썬 설치하기

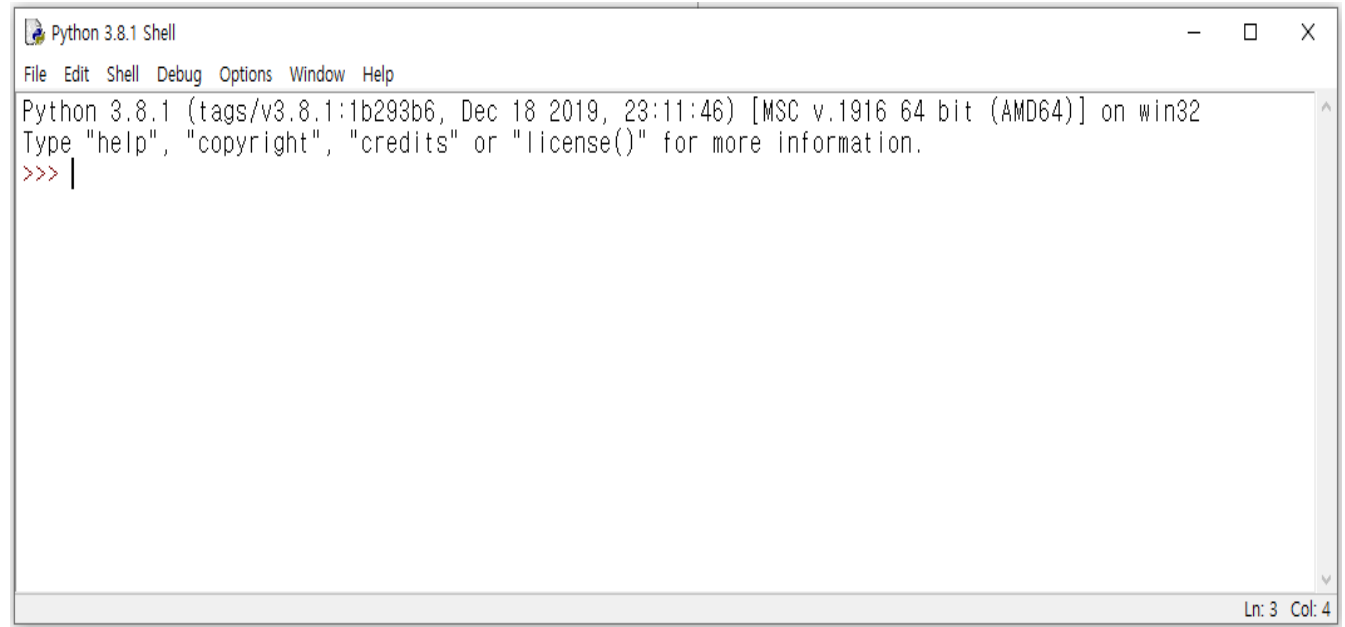
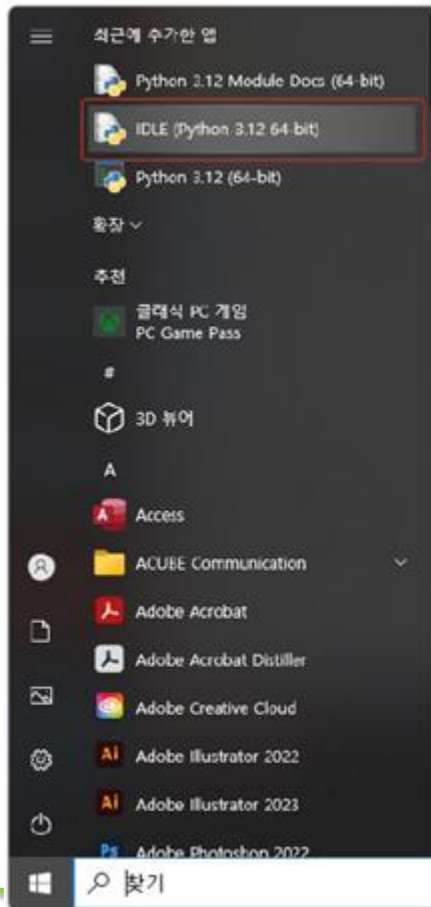
- <http://www.python.org/>에 접속하여 Download 메뉴에서 "Python 3.x"을 선택한다.
- 기존에 설치된 것이 있다면 다시 깔지않아도 됨.



파이썬 시작하기

□ 윈도우의 시작 메뉴에서 "IDLE" 프로그램을 찾아서 실행

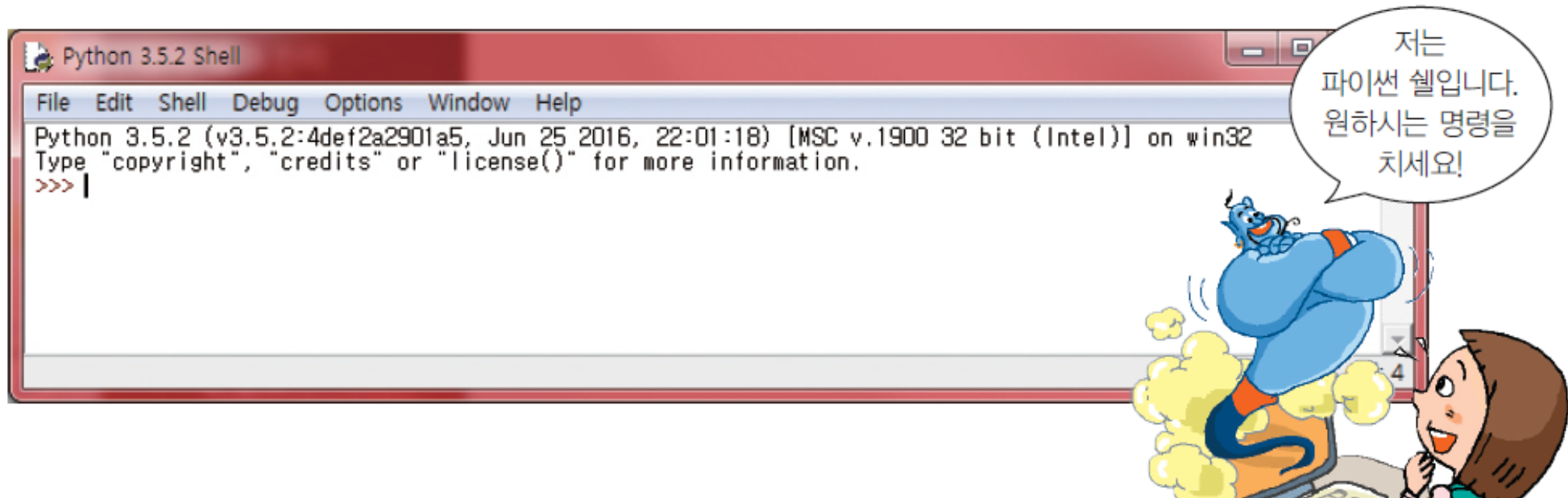
■ 제 노트북환경은 3.8입니다.



파이썬의 프로그래밍 개발 환경을 IDLE(Integrated DeveLopment Environment), 즉 파이썬 통합개발환경이라고 합니다.

파이썬 셸

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



Hello World! 출력하기

```
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 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> print("Hello World!")
Hello World!
>>> |
```

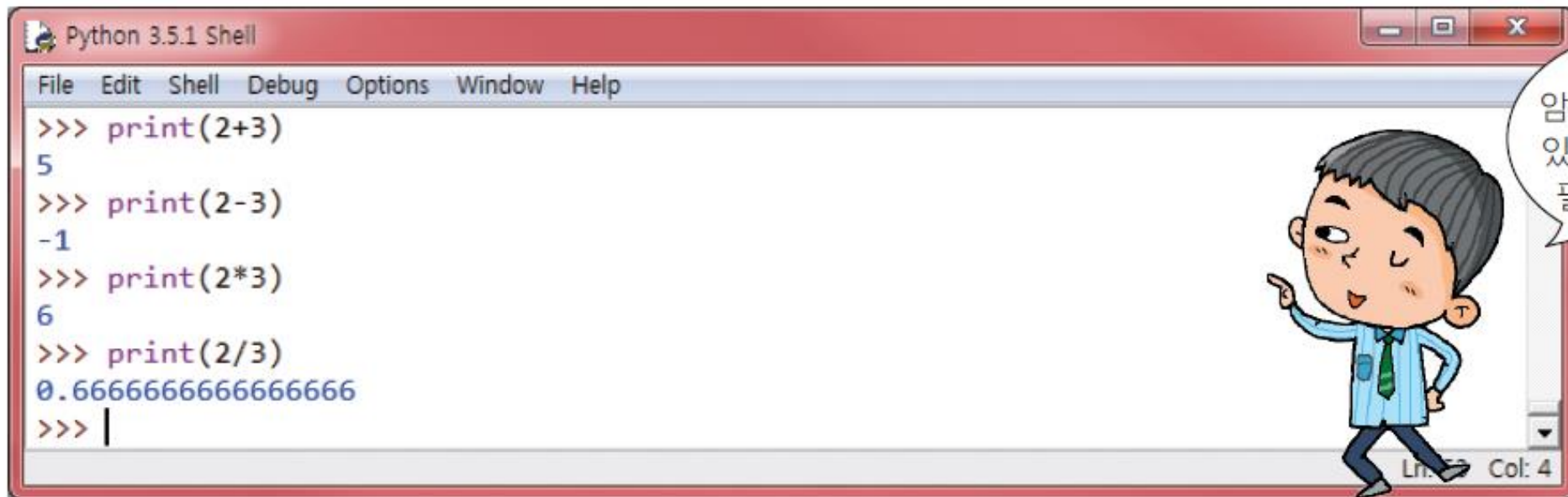
**print()
print("")
print("""**

**print()
print('Hello World')
print("Hello World")**



계산하기 #1

- `print()` 는 출력을 해주는 함수(기능)입니다.
- 출력하고 싶은 내용을 () 에 넣어주세요
- 먼저 계산결과를 출력해봅시다



```
Python 3.5.1 Shell
File Edit Shell Debug Options Window Help
>>> print(2+3)
5
>>> print(2-3)
-1
>>> print(2*3)
6
>>> print(2/3)
0.6666666666666666
>>> |
```

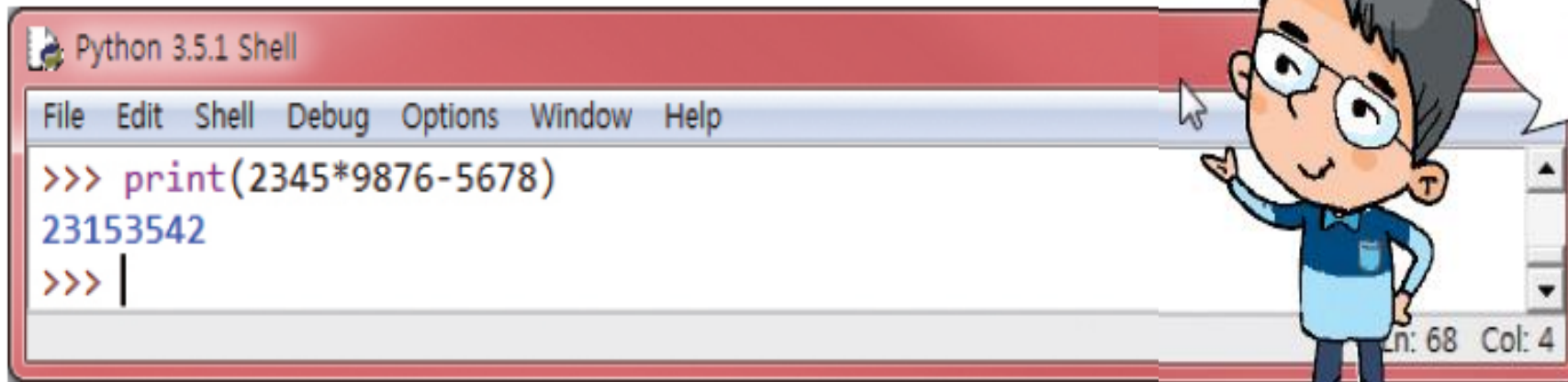
이 정도는
암산으로도 할 수
있어요! 파이썬이
필요 없다고요!

계산하기 #2

- 이번엔 좀 더 어려운 계산~
- $3.141592 * 10.0 * 10.0$
- $(1/100) * 1234$
- 결과를 출력하려면? 그리고 결과는 무엇일까요?

```
print( 3.141592 * 10.0 * 10.0)
```

```
print( 1/100 * 1234)
```



도전! 계산하기 #3

□ 사칙연산 vs. 육칙연산!

■ $+$, $-$, $*$, $/$

□ 나누기는 친구들이 있어요!

■ $//$ 와 $\%$

□ 곱하기도 친구가 있어요!

■ $**$: 거듭제곱

□ 일반적인 수식과 결과

□ $3 / 4 = 0.75$ 일반적인 나눗셈

□ $3 // 4 = 0$ 3을 4로 나누었을 때의 몫을 구합니다.

□ $3 \% 4 = 3$ 3을 4로 나누었을 때의 나머지를 구합니다.

□ $3**4 = 81$ $3*3*3*3$ 의 결과를 구합니다.

```
Type "help", "copyright", "credits" or "license()" for more information.
```

```
>>> a=3/4
```

```
>>> b=3//4
```

```
>>> c=3%4
```

```
>>> d=3**4
```

```
>>> a
```

```
0.75
```

```
>>> d
```

```
81
```

```
>>> print(a)
```

```
0.75
```

```
>>> print(a,b,c,d)
```

```
0.75 0 3 81
```

```
>>> |
```

$a=3/4$ 는 $3/4$ 의 결과를 a 에 저장하라는 뜻입니다.

, (쉼표)를 이용하면 여러 개의 값들을 출력할 수 있어요.

수업 중 실습 #1 : 파이썬 설치하고 인증샷

- 앞서 본 과정을 참고하여 자신의 컴퓨터에 파이썬 실습환경을 설치합니다.

- 필수 : 가벼운 테스트 프로그램을 작성한 후 화면 캡취하기
 - 예) `print ("자기소개용 문장출력하기")` 또는 기타 출력하고 싶은 문자나 계산결과
 - 화면저장방법 : `ctl +Print Screen`(보통 키보드 우측 위) 또는 `shirt+윈도우키+s` 는 눌러 화면영역 잡기

- 선택 : 같은 내용을 colab이나 onlinegdb, vscode에서도 실행해봅니다.
- 게시판에 올립니다

Python으로 할 수 있는 일

- 1. 시스템 유틸리티
- 2. GUI
 - Tkinter
 - wxPython etc..
- 여러 다양한 언어 및 도구와의 결합
 - C/C++
 - MatLab / TensorFlow / ...
- 이미지, 수치계산 및 문자열처리
- 웹 프로그래밍
- Database 프로그래밍
- 기계학습/딥러닝/자연어처리...

미리보는 과제#2

- 파이썬으로 할 수 있는 일 한 가지를 찾아 게시판에 올려 공유해봅시다!
- 먼저 올린 내용을 보고 중복되지 않는 문제를 찾아 보세요~

Python으로 할 수 없는 일

- 운영체제 개발처럼 엄청난 횟수의 반복과 연산이 요구되는

프로그래밍에는 적합하지 않습니다.

- 시간적/공간적 성능을 요구하거나
- 하드웨어를 직접 건드려야 하는 작업에는
- 더 적합한 언어가 있습니다.

■ C ^^!!

미리보는 과제#2

- 파이썬으로 할 수 있는 일 한 가지를 찾아 게시판에 올려 공유해봅시다!
- 먼저 올린 내용을 보고 중복되지 않는 문제를 찾아 보세요~

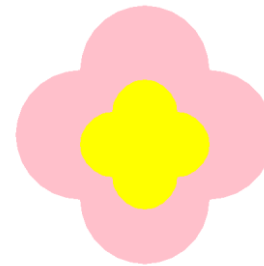
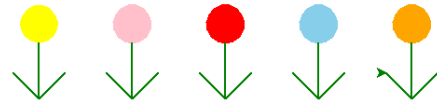
가벼운 실습

가벼운 실습 : 무작정 따라해봅시다~

□ 기본

- 문자열 출력하기 : print
- 스크립트 모드
- 그리고 Turtle

□ Turtle그래픽은 선택입니다



문자열 출력하기 : print(), print 함수

- 문자열(string) : 큰따옴표("...")나 작은따옴표('...') 안에 들어 있는 텍스트 데이터
- 반드시 따옴표가 있어야 한다!!

```
>>> print(Hello 짱구!)
      SyntaxError: invalid syntax
>>> print 'Hello 짱구!'
      SyntaxError: invalid syntax
>>> print "Hello 짱구!"
      SyntaxError: invalid syntax
```

```
>>> print('Hello 짱구!')
Hello 짱구!
>>> print("Hello 짱구!")
Hello 짱구!
```

문자열 출력하기 : 문자열 연결하기

- , 를 사용하면 하나의 print 문장처럼 연결되어 출력된다.
- 여러 개의 값들을 화면에 차례대로 출력할 수 있다.

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

문자열 출력하기 : 문자열 합치기

□ +

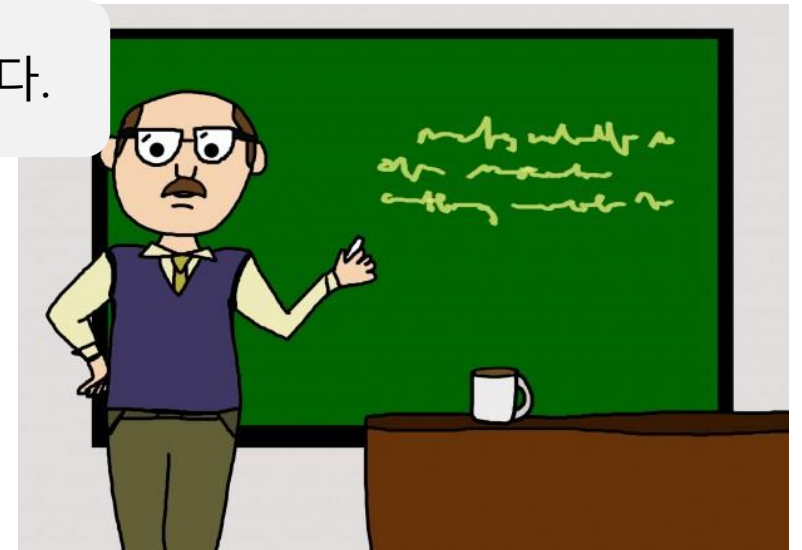
- 문자열 합치기
- 사용방법 : 문자열 + 문자열

```
>>> print("강아지" + "고양이")  
강아지고양이  
>>> print("1" + "2")  
12  
>>print(1 + 2)  
3
```

그럼 문자열과 숫자는 어떻게
구별하나요?



따옴표("...")가 붙으면 문자열입니다.



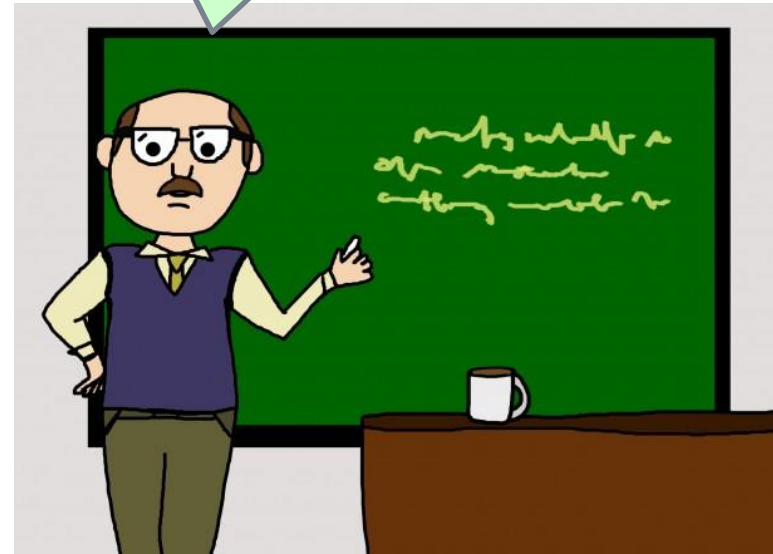
문자열과 숫자

□ 주의해야 해요

```
>>> print("100" + "200")  
100200  
>>>
```

```
>>> print(100 + 200)  
300  
>>>
```

따옴표("...")가
붙으면 문자열
입니다.



문자열 출력하기 : print 문에서 반복하기

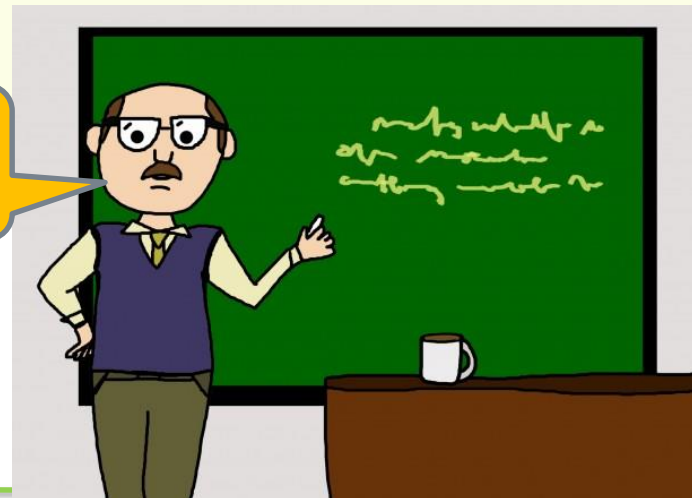
- * 는 산술적으로는 곱하기를, 문자열에서는 반복의 의미가 있어요

```
>>> print("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.



많이 하는 실수 : 문법 오류와 오타

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

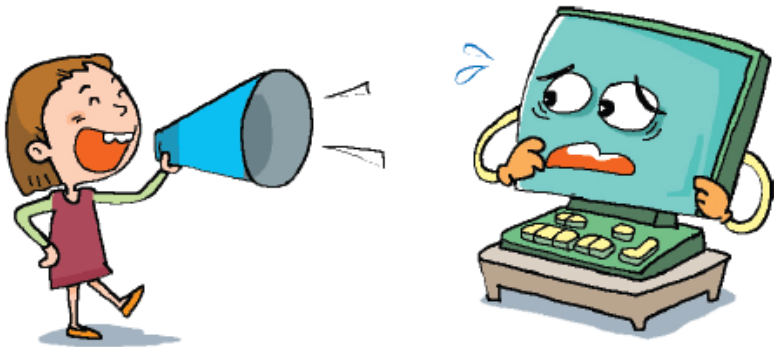
```
>>> 1 +  
SyntaxError: invalid syntax
```

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

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

3번 연속하여 출력하고 싶다면
`print("안녕!" * 3)`로 수정

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



Lab: print() 함수 실습

□ 아래 결과가 출력되도록 빈칸을 채워보세요

```
>>> _____  
안녕하세요? 여러분
```

```
>>> _____  
반갑습니다.^^
```

```
>>> _____  
9*8은 72 입니다.
```

```
>>> _____  
안녕히 계세요.
```

Solution

```
>>> print("안녕하세요? 여러분")
```

안녕하세요? 여러분

```
>>> print("반갑습니다.^^")
```

반갑습니다.^^

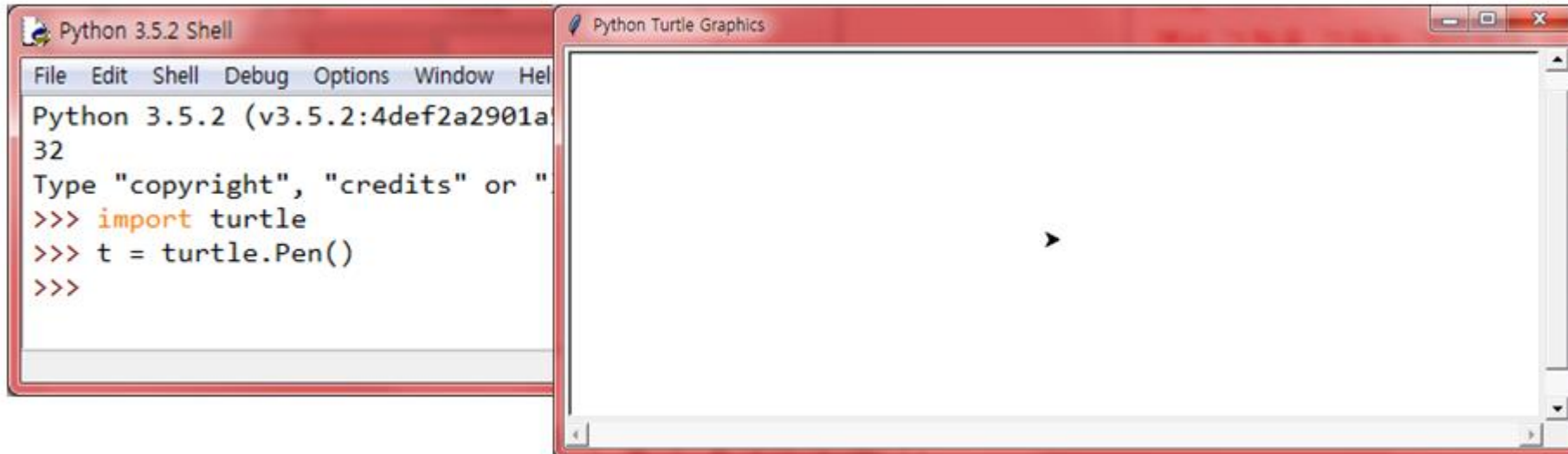
```
>>> print("9*8은", 9*8, "입니다.")
```

9*8은 72 입니다.

거북이(선택활동)
따라합니다

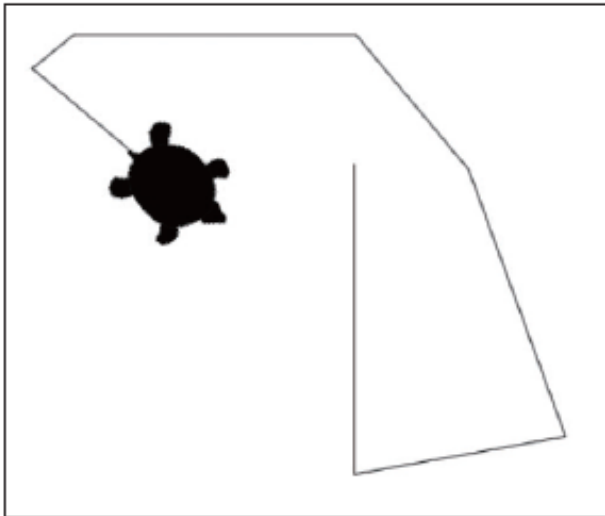
터틀 그래픽 시작

- `import turtle`
- `t=turtle.Pen()`
- 거북이 (그려주는 연필) 이름은 **t**
- 진짜 거북이 모양을 나타나게 하려면 `t.shape("turtle")` 을 추가합니다.



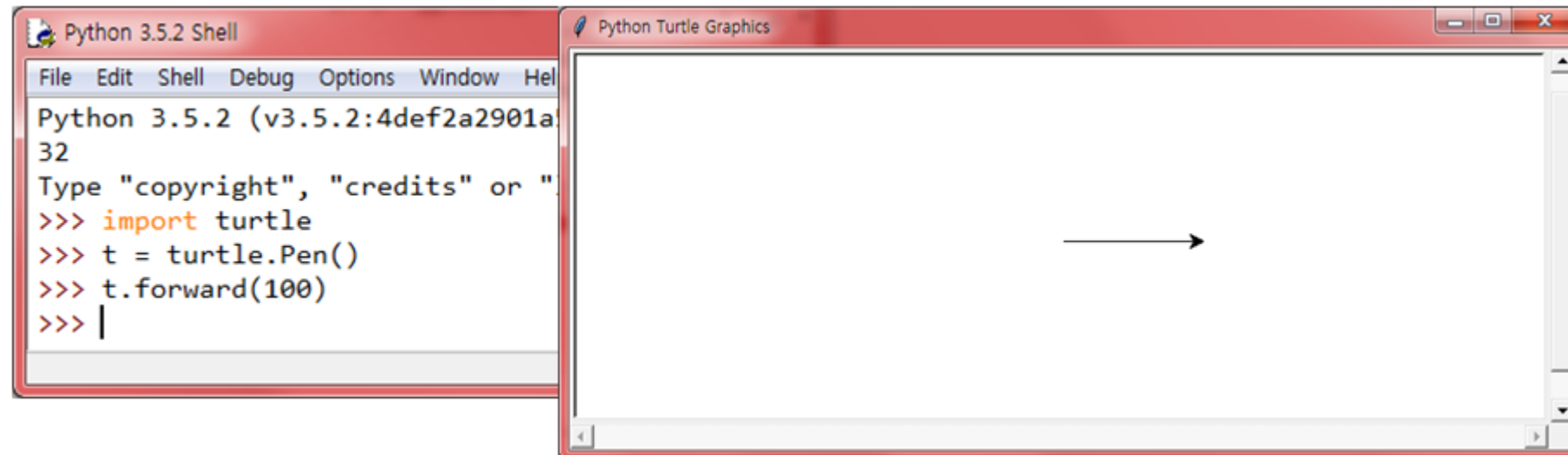
터틀 그래픽

- 거북이로 그림그리기
- 모든 명령은 방향키가 아니라 명령문으로 입력해 줍니다.
 - `t.forward(100)` : 거북이 t 야! 앞으로 100칸 가~
 - 또는 `t.fd((100))`



직선 그리기

□ `t.forward(100)`

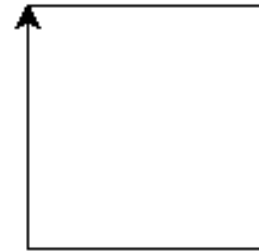


사각형을 그려보세요

- `t.right(90)` : 오른쪽으로 90도 회전하기
- `t.left(90)` : 왼쪽으로 90도 회전하기

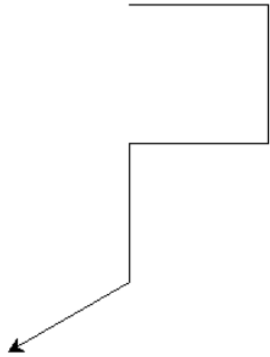
```
>>> import turtle
>>> t = turtle.Pen()
>>> t.forward(100)
>>> t.right(90)
>>> t.forward(100)
>>> t.right(90)
>>> t.forward(100)
>>> t.right(90)
>>> t.forward(100)
```

오른쪽으로 90도 방향전환하기



하나더? 펜 색깔을 바꾸고 싶다면
`t.color("red")` 를
코드 중간에 넣어보세요~

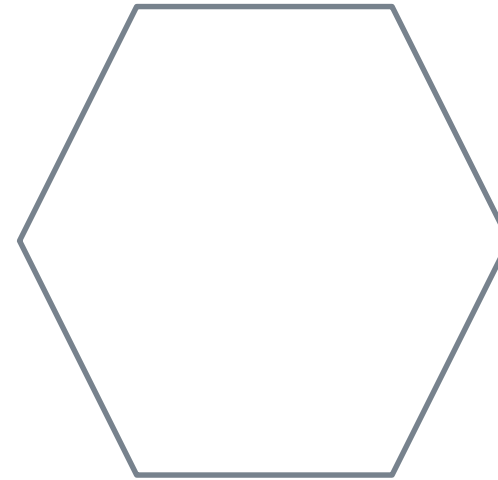
방향을 조정하며 뱀돌뱀돌



```
C:\Windows\py.exe
Python 3.6.1 (v3.6.1:69c0db5, Mar 21 2017, 17:54:52) [MSC v.1900 32 bit (Intel) on win32]
Type "help", "copyright", "credits" or "license()" for more information.
>>> import turtle
>>> t=turtle.Pen()
>>> t.forward(100)
>>> t.right(90)
>>> t.forward(100)
>>> t.right(90)
>>> t.forward(100)
>>> t.left(90)
>>> t.forward(100)
>>> t.right(60)
>>> t.forward(100)
>>>
```

도전문제 : 6각형 그리기

- ☐ 몇 번의 직선을 그려주어야 할까?
- ☐ 회전각은 몇 도?
- ☐ .
- ☐ 직선그리고 60도 회전하고
- ☐ 직선그리고 60도 회전하고
- ☐ 직선그리고 60도 회전하고
- ☐ 직선그리고 60도 회전하고
- ☐ 직선그리고 60도 회전하고
- ☐ 직선그리고 60도 회전하고



Solution

```
C:\Windows\py.exe
Python 3.6.1 (v3.6.1:69c0db5, Mar 21 2017, 17:54:52) [MSC v.190
on win32
Type "help", "copyright", "credits" or "license" for more infor
>>> import turtle
>>> t=turtle.Pen()
>>> t.forward(100) t.right(60)
File "<stdin>", line 1
    t.forward(100) t.right(60)
      ^
SyntaxError: invalid syntax
>>> t.forward(100); t.right(60)
>>> t.forward(100); t.right(60)
>>> t.forward(100); t.right(60)
>>> t.forward(100); t.right(60)
>>> t.forward(100); t.right(60)
>>> t.forward(100); t.right(60)
>>> _
```



명령어들을 한 줄에 쓸 때는 ;를 이용합니다.
라인이 길어져서 붙여봤어요~
6각형이라 6줄을 썼지만 60각형을 그린다면?
이럴 때 반복문을 씁니다! 나중에 같이 배워요

```
(미리보기)
import turtle

t= turtle.Pen()
i=0
for i in range(6) :
    t.forward(100)
    t.right(60)

print("wow!!");
```


스크립트 모드

에디터기능 + 소스화일 저장하기

스크립트 모드

- 한 줄씩 입력하고 바로바로 결과를 확인하는 인터프리트 모드는
- 틀려도 커서를 올려 수정할 수가 없으며
- 코드가 복잡해지면 번거롭다! ➔ 에디터를 사용하자

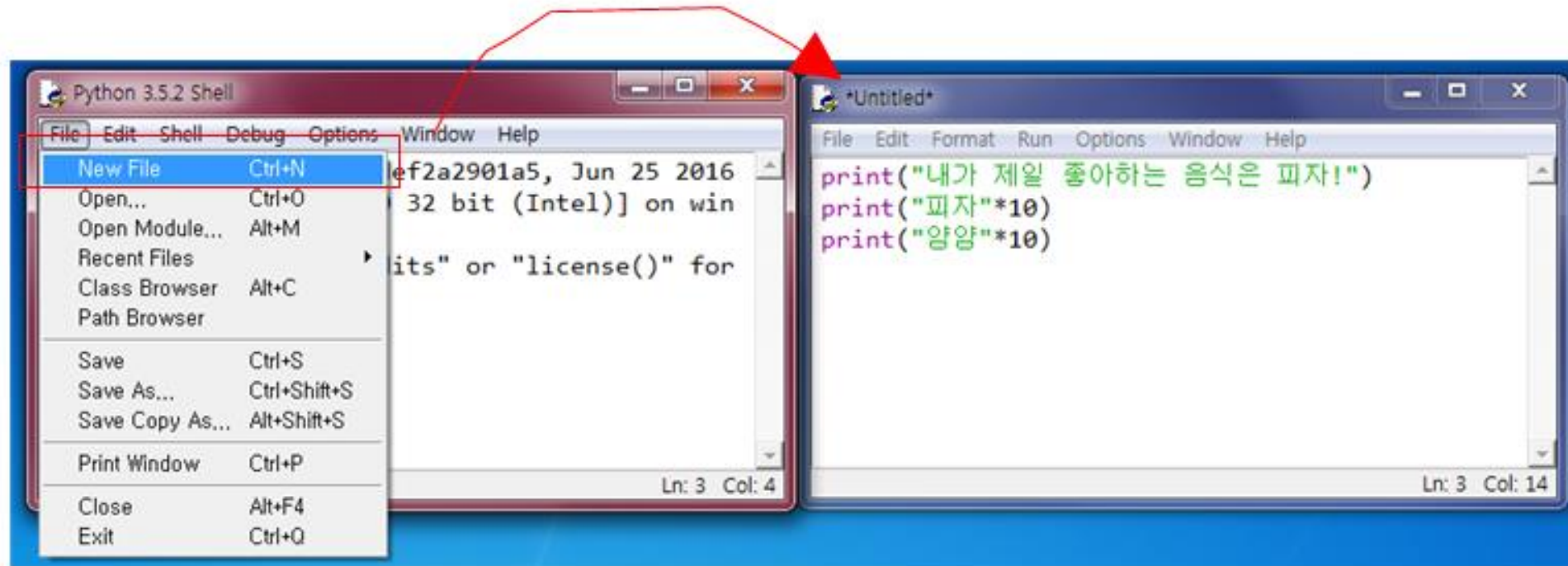
이걸 한 줄씩 입력하라고?



```
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 ', ' % ast[1] -> (' % nodename  
        for n, child in enumerate(children):  
            print '%s' % child,
```

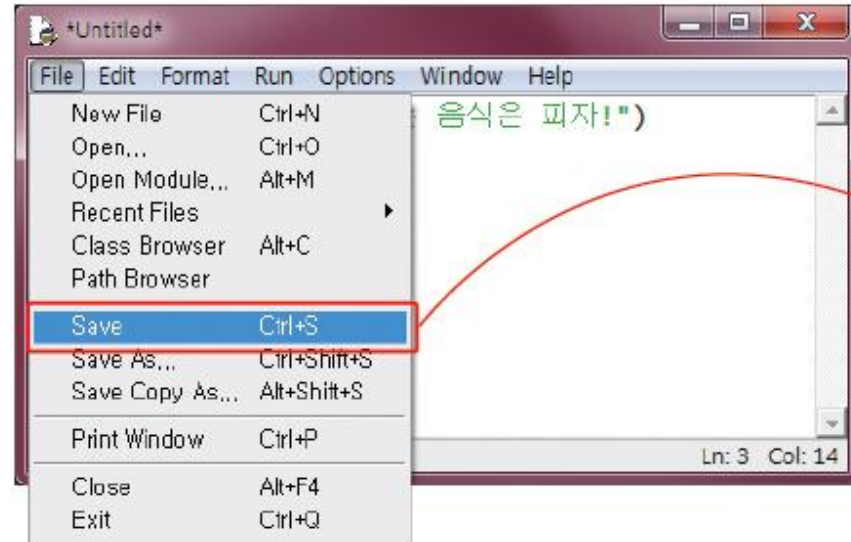
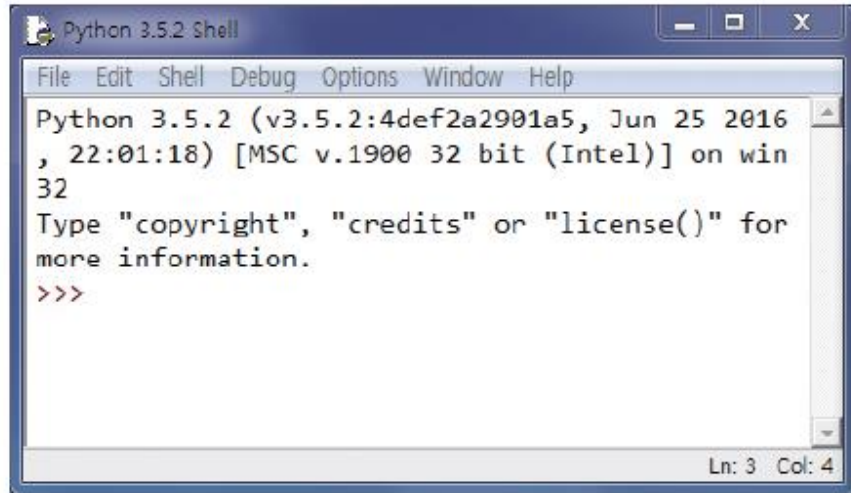
IDLE를 이용한 소스 파일 작성

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



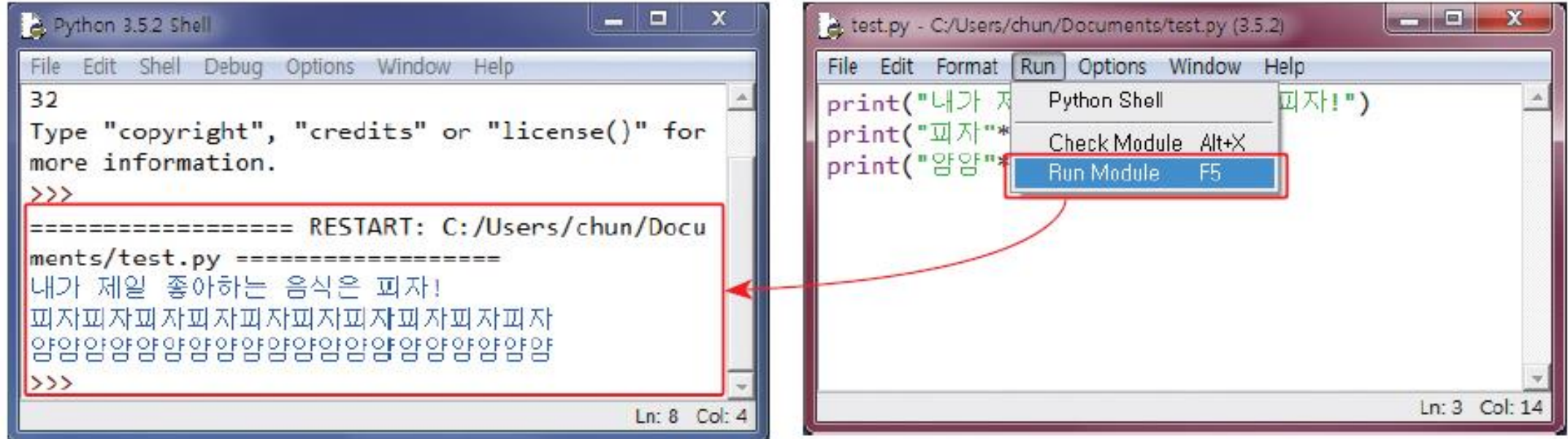
IDLE를 이용한 소스 파일 저장

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



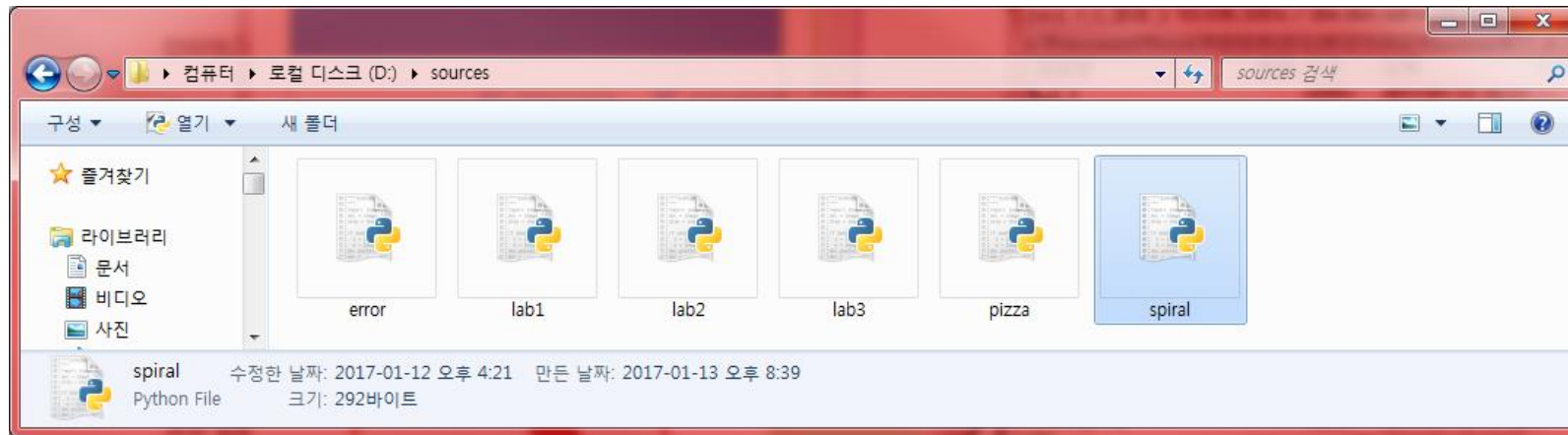
IDLE를 이용한 소스 파일 실행

- 파이썬 셸의 메뉴 중에서 메뉴 [Run]->[Run Module]을 선택한다.
- 단축기 F5 !



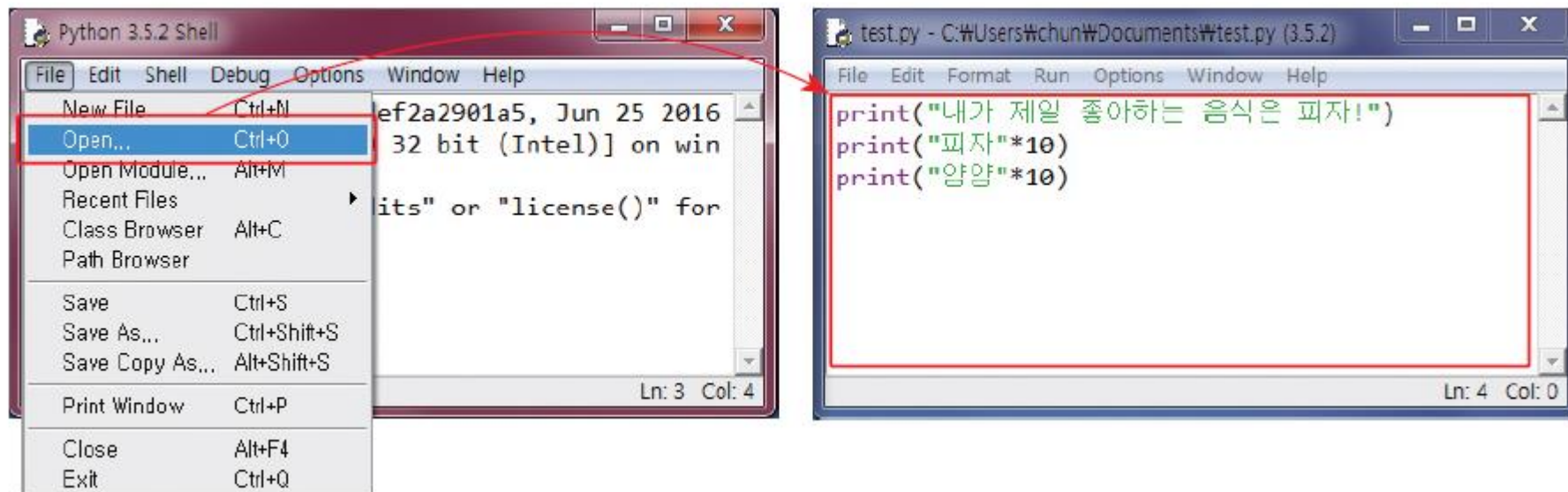
소스 파일 작성하기

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



소스 파일 다시 열기

- IDLE의 [File]->[Open] 메뉴를 선택한다.
- 원하는 폴더에서 원하는 파일을 선택한다.



도전 !: 좀 긴 소스코드인데요, 따라서 쳐봅시다!

- 정확히 보고 정확히 타이핑하기 연습! 이 파일을 작성하여 저장한 후 실행해보세요

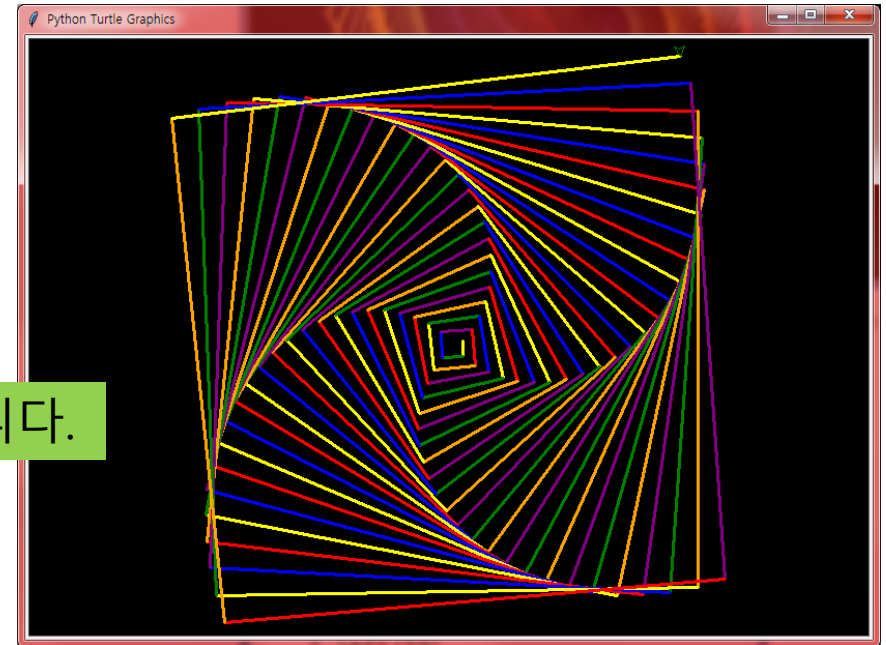
```
import turtle
colors = ["red", "purple", "blue", "green", "yellow", "orange"]
t = turtle.Turtle()
```

```
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
```

빈칸 여러 개가 아니라, 탭키를 한번 넣어줍니다.



대략, 회전각은 89도! 색도 바꾸고 길이는 5씩 증가시키면서 반복하라는 내용

오늘 배운 내용 정리

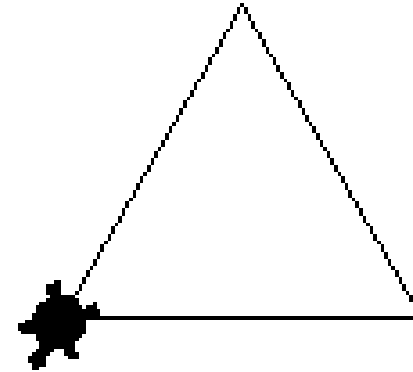
- 출력함수 : `print()`는 화면에 문자열이나 계산 결과를 출력할 수 있다.
- 산술연산자 중 사칙연산: `+` , `-` , `*` , `/`
 - `/` 에서, 몫과 나머지를 구분할 때는 `//` , `%` 를 사용한다
- `+` , `*` , 십표는 문자열에도 사용될 수 있다.
- `" "` 가 있으면 문자열
- 거북이는 파이썬에서 사용할 수 있는 간단한 그래픽라이브러리!
- 코드감각을 익힐 수 있도록 여러 번 실습하자!



Supp. 거북이로 그림그리기

Lab : 터틀 그래픽으로 삼각형을 그려보자

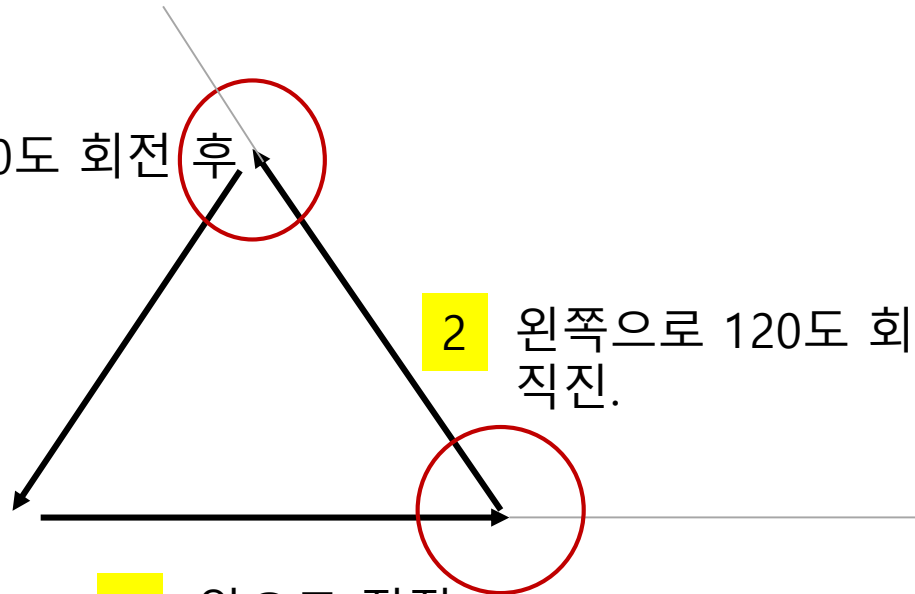
- `t.forward(100)` : 100칸 앞으로 전진
- `t.left(120)`
- 그리는 방법에 따라 `t.right()`을 사용해도 됩니다.
- 각은 진행방향을 기준으로 계산합니다.



3 왼쪽으로 120도 회전 후
직진.

2 왼쪽으로 120도 회전 후
직진.

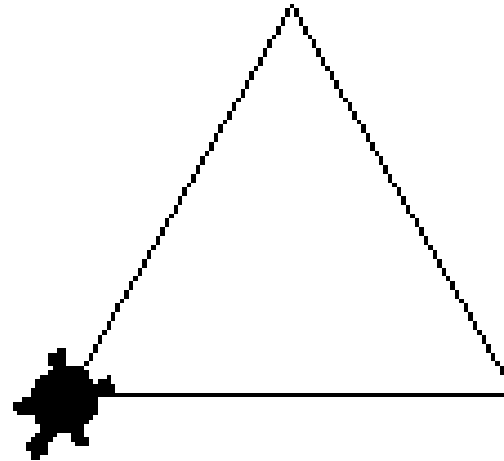
1 앞으로 직진



Solution

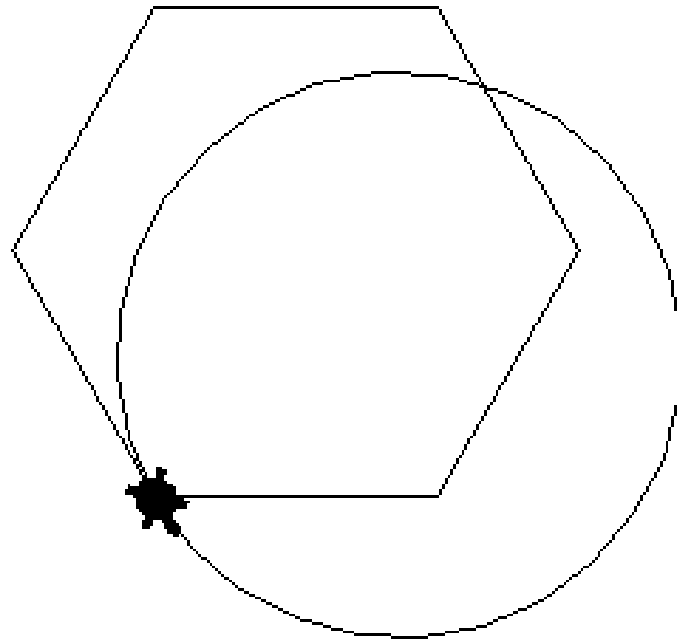
```
>>> import turtle
>>> t=turtle.Pen()
>>> t.shape("turtle")

>>> t.forward(100)
>>> t.left(120)
>>> t.forward(100)
>>> t.left(120)
>>> t.forward(100)
```



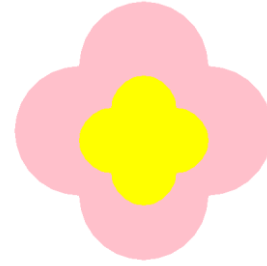
Lab: 터틀 그래픽으로 원, 다각형을 그려보자

- 6각형을 그린 후 원을 그립니다.
- 원 그리기
- `t.circle(100)` : 반지름이 100인 원을 그린다.

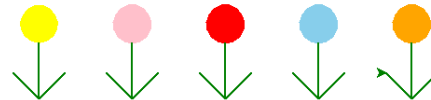


Solution

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



거북이 + 선과 원으로 할 수 있는 일



라이언 그리기

- `t.forward()` → `t.fd()`
- `t.right()` → `t.rt()`
- 움직일 땐 펜 들기 : `t.penup()`;
- 펜 다시 내려놓기 : `t.pendown()`;
- 색깔 칠하기 :
 - `t.color("orange")` : 색깔 정하고
 - `t.begin_fill()` : 칠하기 시작
 - `t.end_fill()` : 칠하기 끝



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

t.color("orange")
t.begin_fill()

r=100
t.circle(r)

t.left(90)
t.fd(200)
t.left(90)
t.end_fill()

t.penup()
t.fd(60)
t.begin_fill()
t.pendown()
t.circle(40)
t.end_fill()

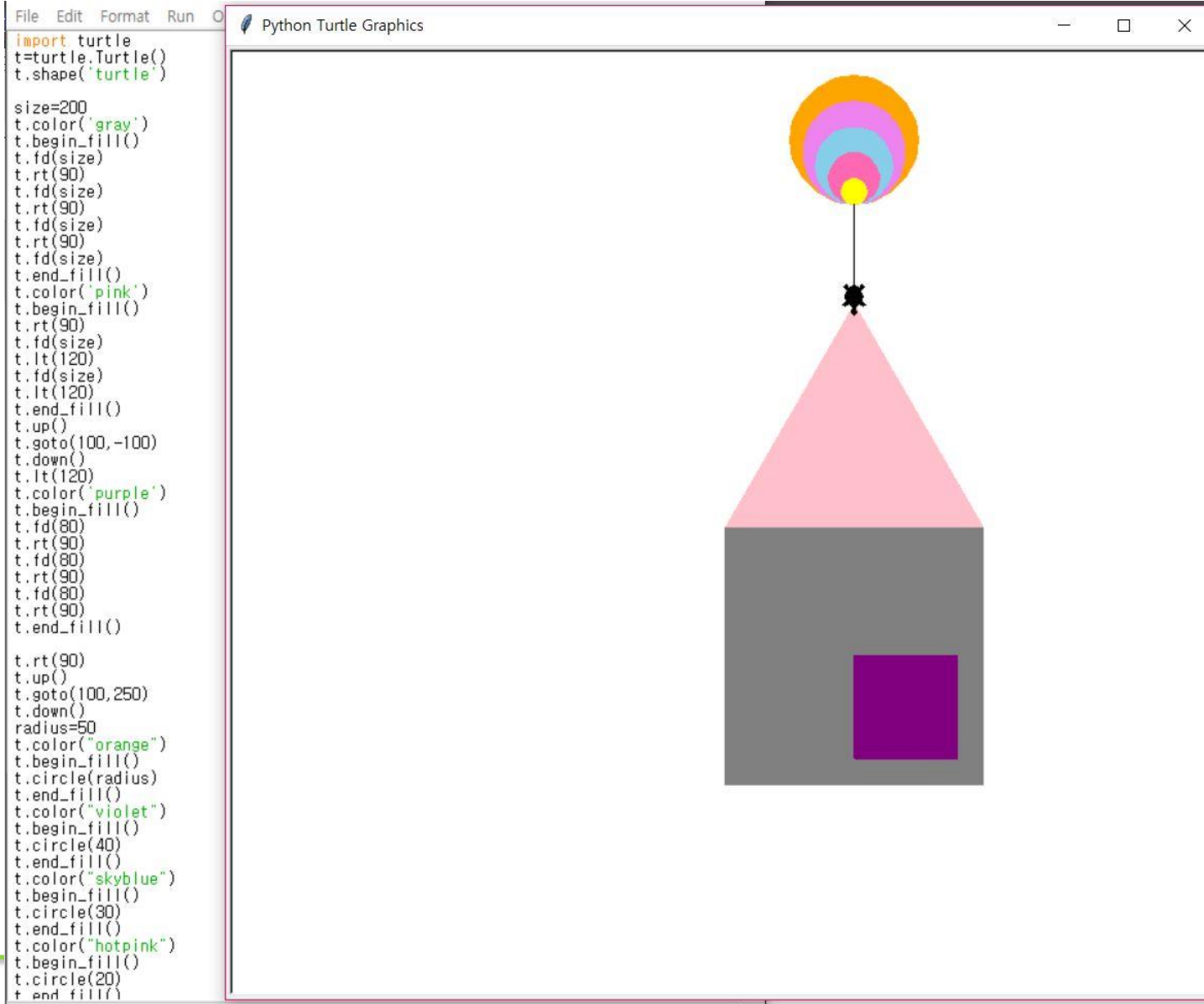
t.penup()
t.rt(180)
t.fd(120)
t.rt(180)
t.begin_fill()
t.pendown()
t.circle(40)
t.end_fill()

t.penup()
t.rt(90)
t.fd(60)
t.lt(90)
t.fd(80)
t.color("black")
t.rt(90)
t.begin_fill()
t.fd(40)
t.pendown()
t.pendown()
t.circle(10)
t.end_fill()

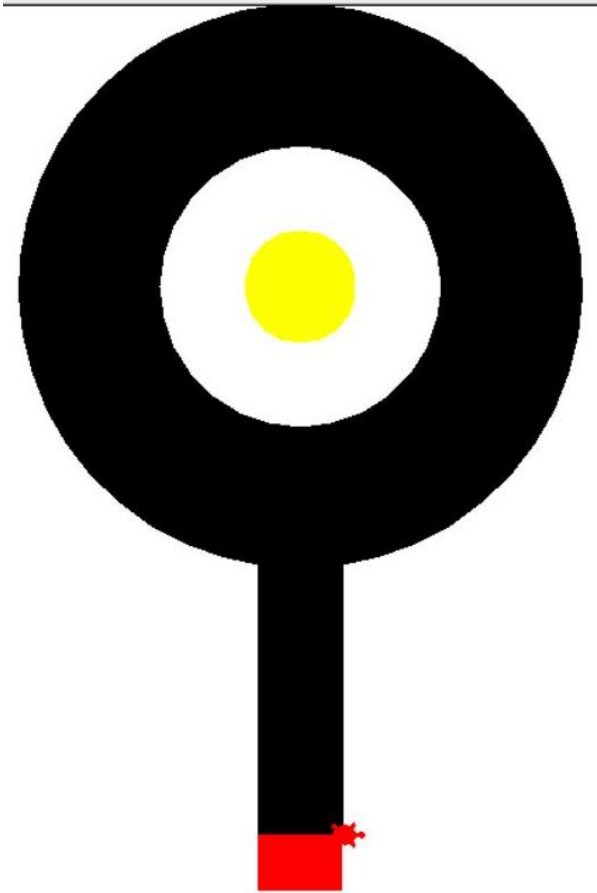
t.penup()
t.rt(180)
t.fd(80)
t.rt(180)
t.begin_fill()
t.pendown()
t.pendown()
t.circle(10)
t.end_fill()

t.penup()
t.rt(90)
t.fd(20)
t.rt(90)
```


UP 처럼 풍선타고 날아가는 집을 그리려다가..^^



후라이팬과 계란후라이



후라이.py - C:/Users/User/Desktop/이화여대/파이썬/후라이.py (3.6.4)

```
File Edit Format Run Options Window Help
import turtle
t = turtle.Turtle()
t.shape("turtle")

t.penup()
t.goto(0,0)
t.down()

t.color("black")
t.begin_fill()
t.circle(200)
t.end_fill()

t.penup()
t.goto(0,100)
t.down

t.color("white")
t.begin_fill()
t.circle(100)
t.end_fill()

t.penup()
t.goto(0,160)
t.down

t.color("yellow")
t.begin_fill()
t.circle(40)
t.end_fill()

t.penup()
t.goto(30,10)
t.pendown()

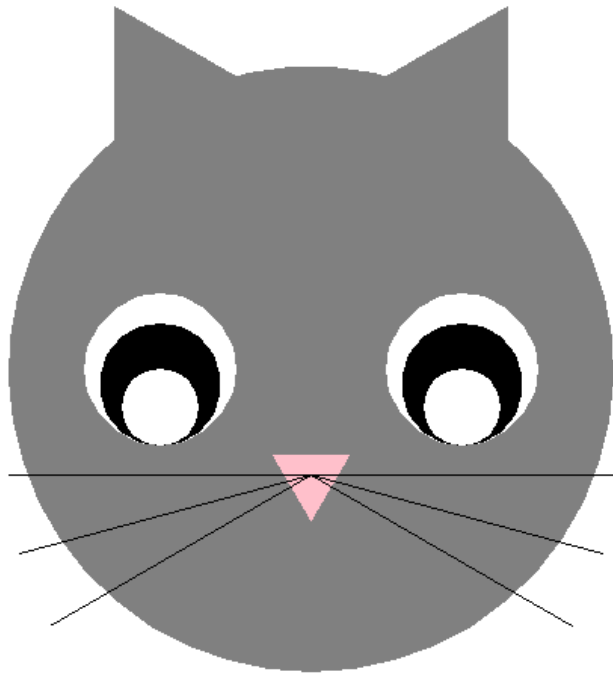
t.color("black")
t.begin_fill()
t.rt(90)
t.fd(200)
t.rt(90)
t.fd(60)
t.rt(90)
t.fd(200)
t.rt(90)
t.fd(60)
t.end_fill()

t.penup()
t.goto(30,-190)
t.pendown

t.color("red")
t.begin_fill()
t.rt(90)
t.fd(40)
t.rt(90)
t.fd(60)
t.rt(90)
t.fd(40)
t.end_fill()
```

//코드잘림
t.color("red")
t.begin_fill()
t.rt(90)
t.fd(40)
t.rt(90)
t.fd(60)
t.rt(90)
t.fd(40)
t.end_fill()

```
Python 3.5.0 Shell
File Edit Shell Debug Options Window Help
Python 3.5.0 (v3.5.0:374f501f4567, Sep 13 2015, 02:16:59) [MSC v.1900
32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/user/Desktop/2.py =====
>>> 반지름의 크기는 얼마입니까?200
>>>
```



```
*고양이.py - C:/Users/user/Desktop/고양이.py
File Edit Format Run Options Window Help
import turtle
t=turtle.Turtle()

r=int(input("반지름의 크기는 얼마입니까?"))
t.goto(0,-100)
t.begin_fill()
t.color("grey")
t.circle(r)
t.end_fill()
t.goto(100,50)
t.begin_fill()
t.color("white")
t.circle(r/4)
t.end_fill()
t.begin_fill()
t.color("black")
t.circle(r/5)
t.end_fill()
t.begin_fill()
t.color("white")
t.circle(r/8)
t.end_fill()
t.penup()
t.goto(-100,50)
t.pendown()
t.begin_fill()
t.color("white")
t.circle(r/4)
t.end_fill()
t.begin_fill()
t.color("black")
t.circle(r/5)
t.end_fill()
t.begin_fill()
t.color("white")
t.circle(r/8)
t.end_fill()
t.penup()
t.goto(0,0)
t.pendown()
t.begin_fill()
t.color("pink")
t.left(60)
t.fd(r/4)
t.left(120)
t.fd(r/4)
t.left(120)
t.fd(r/4)
t.end_fill()
t.goto(0,30)
t.color("black")
```

```
*고양이.py - C:/Users/user/Desktop/고양이.py
File Edit Format Run Options Window Help
t.left(30)
t.fd(r)
t.goto(0,30)
t.left(15)
t.fd(r)
t.goto(0,30)
t.left(15)
t.fd(r)
t.goto(0,30)
t.left(15)
t.fd(r)
t.goto(0,30)
t.left(15)
t.fd(r)
t.goto(0,30)
t.left(15)
t.fd(r)
t.penup()
t.goto(130,340)
t.pendown()
t.begin_fill()
t.color("grey")
t.fd(r/2)
t.left(120)
t.fd(r/2)
t.left(120)
t.fd(r/2)
t.end_fill()
t.penup()
t.goto(-130,340)
t.pendown()
t.begin_fill()
t.color("grey")
t.right(180)
t.fd(r/2)
t.left(120)
t.fd(r/2)
t.left(120)
t.fd(r/2)
t.end_fill()
t.goto(0,150)
```

수고하셨습니다.