

Day 2 : 계산과 출력 그리고 거북이

(Lab : 무작정 따라해보세요~)

산술 계산

문자열 Print

거북이(Turtle)

Yun-Jeong Choi(cris2@ewha.ac.kr)

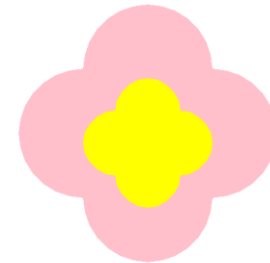
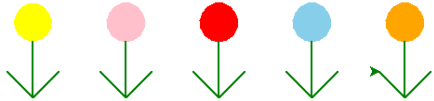
Dept. of Computer Science & Engineering

Ewha Womans Univ.

□ 기본

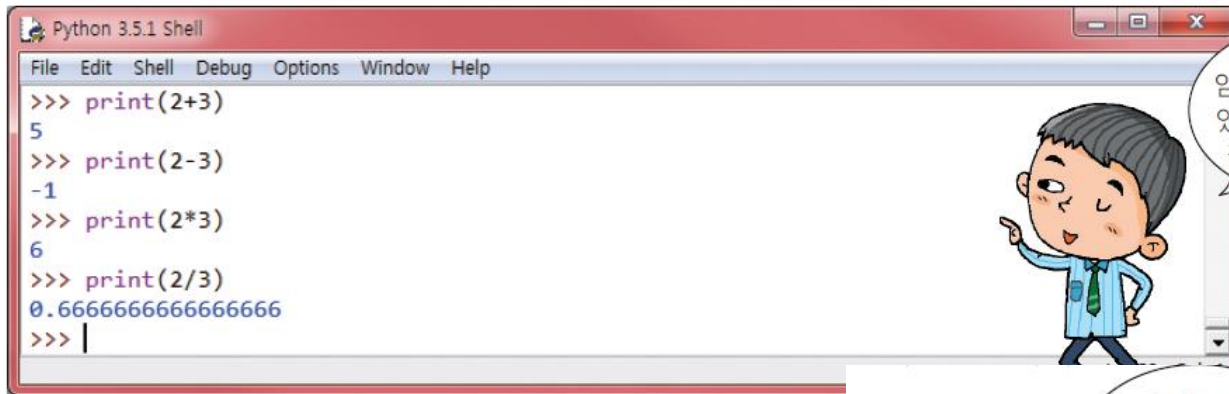
- 산술계산 : $+$, $-$, $*$, $/$ 의 사칙연산
- 문자열 출력하기 : `print`
- 스크립트 모드

□ Lab



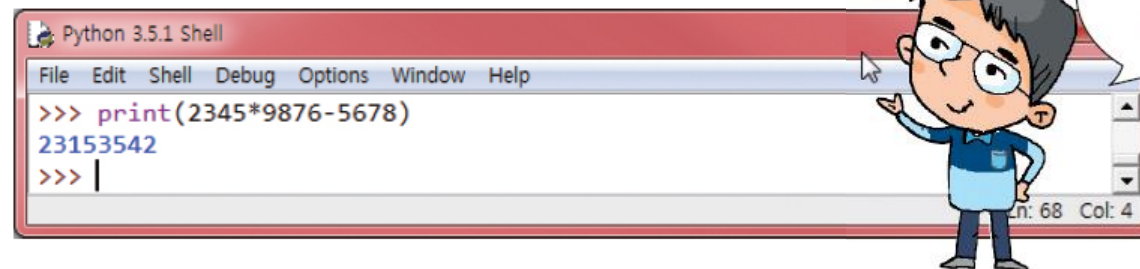
□ 학습활동

산술 계산하기



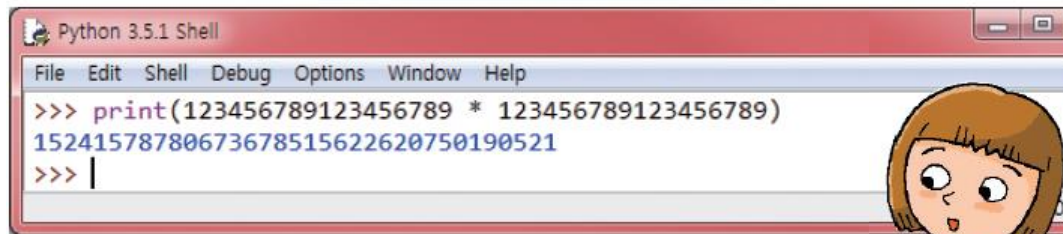
```
Python 3.5.1 Shell
File Edit Shell Debug Options Window Help
>>> print(2+3)
5
>>> print(2-3)
-1
>>> print(2*3)
6
>>> print(2/3)
0.6666666666666666
>>> |
```

이 정도는
암산으로도 할 수
있어요! 파이썬이
필요 없다고요!



```
Python 3.5.1 Shell
File Edit Shell Debug Options Window Help
>>> print(2345*9876-5678)
23153542
>>> |
```

이 정도라면
계산기로
하면 되죠!



```
Python 3.5.1 Shell
File Edit Shell Debug Options Window Help
>>> print(123456789123456789 * 123456789123456789)
15241578780673678515622620750190521
>>> |
```

컴퓨터에서 사용하는 숫자들은 유효범위도 있고,
소수의 계산 결과에는 오차도 있다는 것을 알고 넘
어갑시다~

산술계산하기 퀴즈 : 다음의 결과는 무엇일까요?

$$3.141592 * 10.0 * 10.0 = ?$$

$$(1/100) * 1234 = ?$$

결과는 ?

□ 나누기의 친구들

$$3 / 4 = 0.75$$

일반적인 나눗셈

$$3 // 4 = 0$$

3을 4로 나누었을 때의 몫을 구합니다.

$$3 \% 4 = 3$$

3을 4로 나누었을 때의 나머지를 구합니다.

문자열 출력하기 : print(), print 함수

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

```
>>> print(Hello Ewha!)
```

SyntaxError: invalid syntax

```
>>> print 'Hello Ewha!'
```

SyntaxError: invalid syntax

```
>>> print "Hello Ewha!"
```

SyntaxError: invalid syntax

```
>>> print('Hello Ewha!')
```

Hello Ewha!

```
>>> print("Hello Ewha!")
```

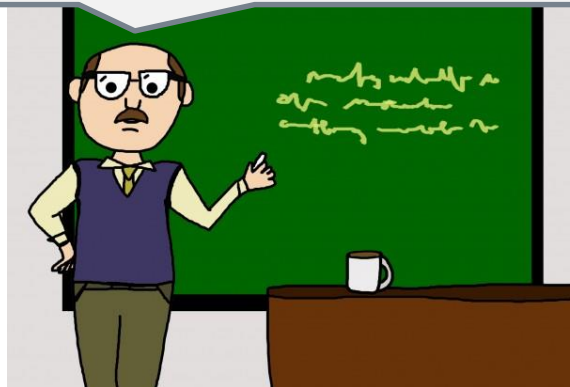
Hello Ewha!

파이썬 예전문법.
지금은 () 필수입니다.

문자열은 어떻게 구별하나요?



따옴표("...")가 붙으면 문자열입니다.



문자열 출력하기 : print() 함수

- , 를 사용하여 여러 개의 값들을 화면에 차례대로 출력할 수 있다.

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

```
결과값은 14 입니다.
```

문자열 출력하기 : +, * 이용하기

□ +

- 숫자들은 산술적으로 더해주지만
- 문자열은 연결한다.
- 사용방법 : 문자열 + 문자열

```
>>> print("강아지" + "고양이")
강아지고양이
>>>
```

□ *

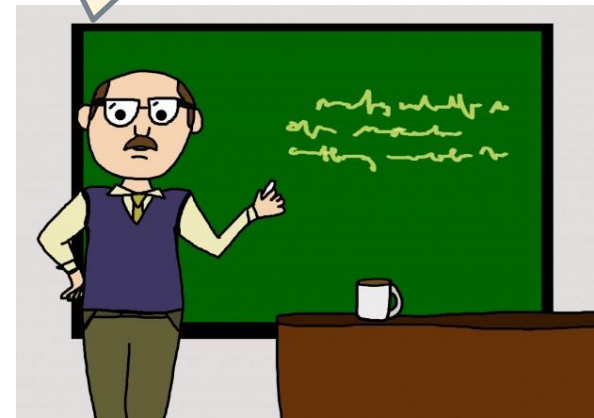
- 숫자들은 산술적으로 곱해주지만
- 문자열들은 숫자만큼 반복한다.
- 사용방법 : 문자열 * 숫자

```
>>> print("반가워요 " * 20)
반가워요 반가워요 반가워요 반가워요 반가워요 반가워요 반가워요 반가워요 반가워요 반가워요
반가워요 반가워요 반가워요 반가워요 반가워요 반가워요 반가워요 반가워요 반가워요 반가워요
>>>
```

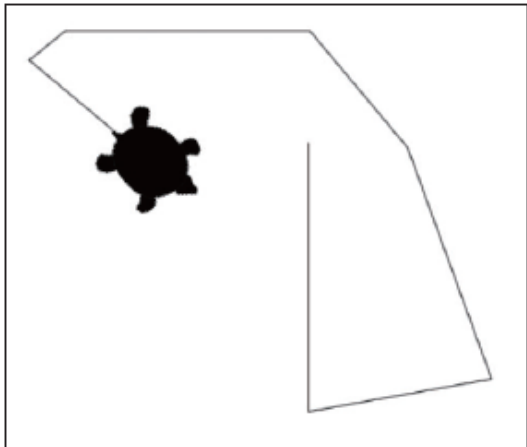
```
>>> print("100" + "200")  
100200  
>>>
```

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

따옴표("...")가
붙으면 문자열
입니다.

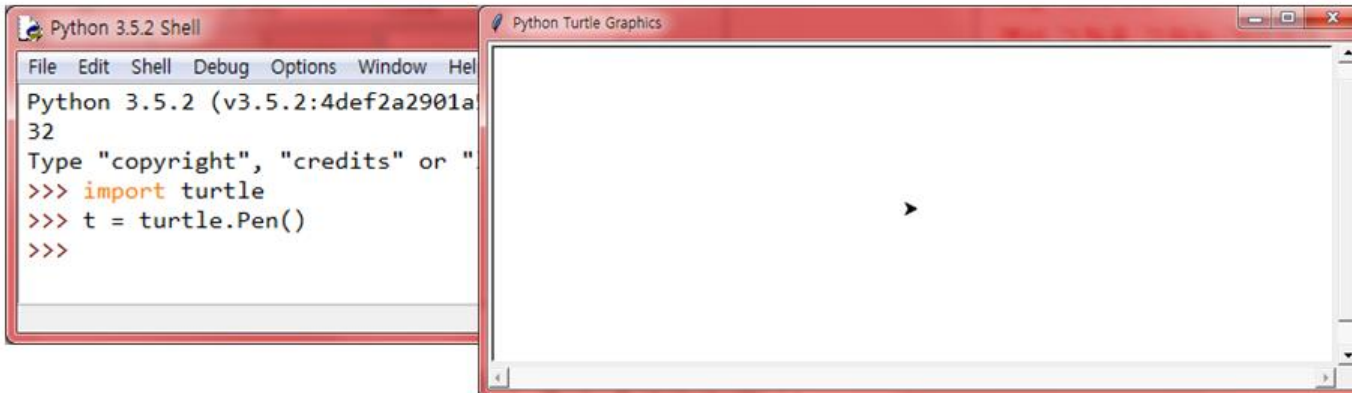


- 거북이로 그림그리기
- 모든 명령은 방향키가 아니라 명령문으로 입력해 줍니다.
 - 예) 거북이 `t.forward(100)` : 거북이 t 야! 앞으로 100칸 가!

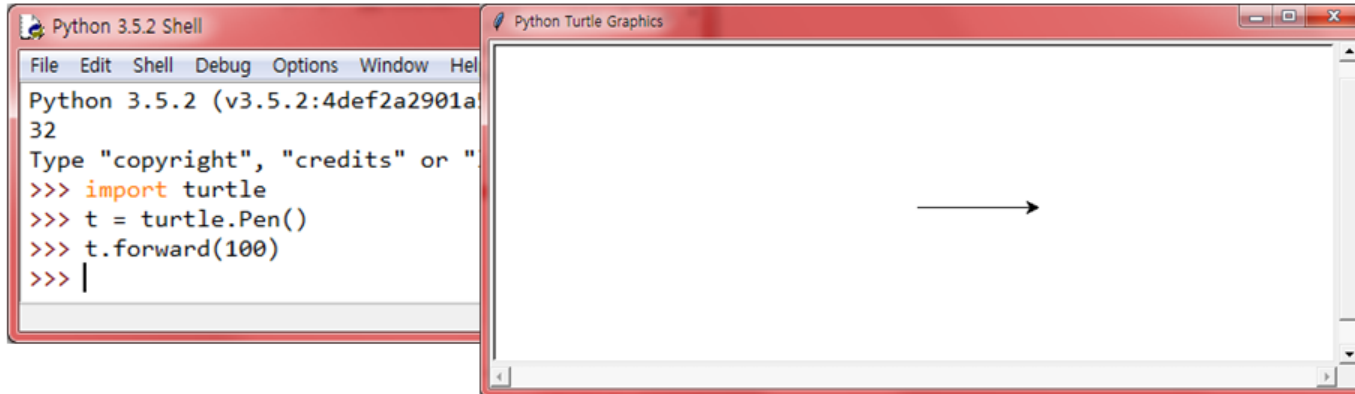


터틀 그래픽 시작

- `import turtle`
- `t=turtle.Pen()`
- 거북이 (그려주는 연필) 이름은 `t`



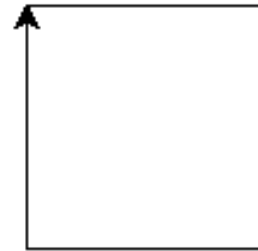
□ `t.forward(100)` 또는 `t.fd(100)`



사각형을 그려보세요

```
>>> 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)
```

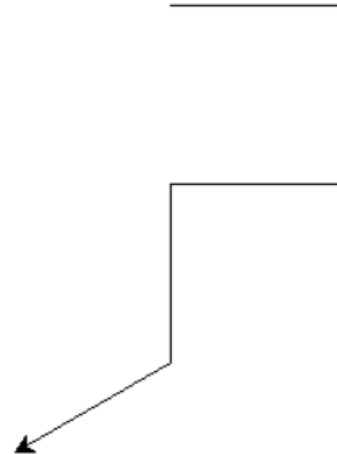
오른쪽으로 90도 방향전환하기



펜 색깔을 바꾸고 싶다면
>>> t.color("red")
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)
>>>
```



☐ 몇 번의 직선을 그려주어야 할까?

☐ 회전각은 몇 도?

☐ .

☐ .

☐ .

☐ 직선그리고 60도 회전하고

☐ 직선그리고 60도 회전하고

☐ 직선그리고 60도 회전하고

☐ 직선그리고 60도 회전하고

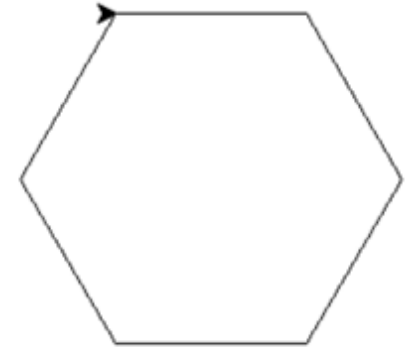
☐ 직선그리고 60도 회전하고

☐ 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.!

```
import turtle

t= turtle.Pen()

for i in range(6) :
    t.forward(100)
    t.right(60)

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

- 한 줄씩 입력하고 바로바로 결과를 확인하는 인터프리트 모드는
- 코드가 복잡해지면 번거롭다! → 에디터를 사용하자

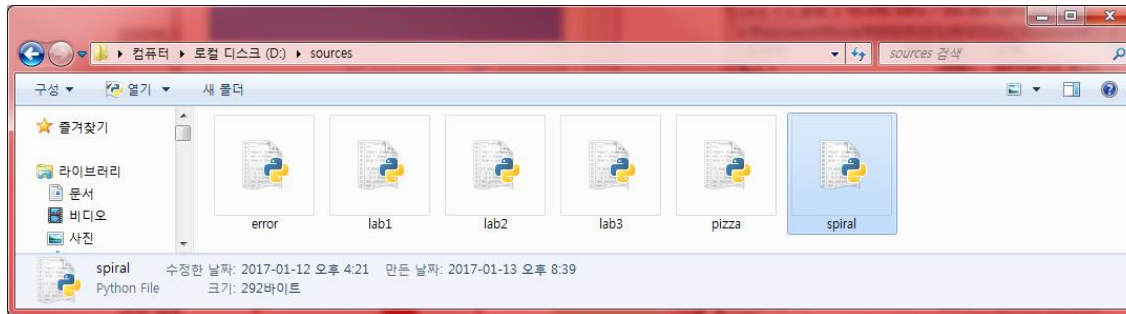
이걸 한 줄씩 입력하라고?



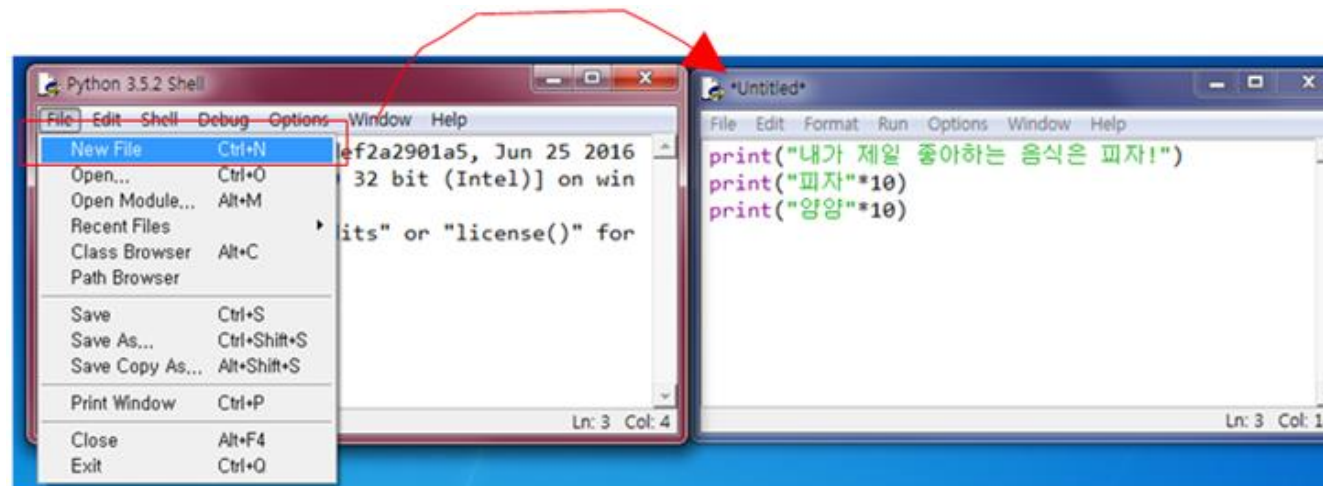
```
def add5(x):  
    return x+5  
  
def dotwrite(ast):  
    nodename = getNodeName()  
    label=symbol.sym_name.get(int(ast[0]), ast[0])  
    print '    %s [label="%s' % (nodename, label)  
    if isinstance(ast[1], str):  
        if ast[1].strip():  
            print '= %s' % ast[1]  
     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


스크립트 모드 : 소스 파일 작성하기

- 텍스트 에디터를 이용하여 명령어들을 파일에 저장한다.
- 명령어들이 저장된 파일을 소스 파일(source file)이라고 한다.

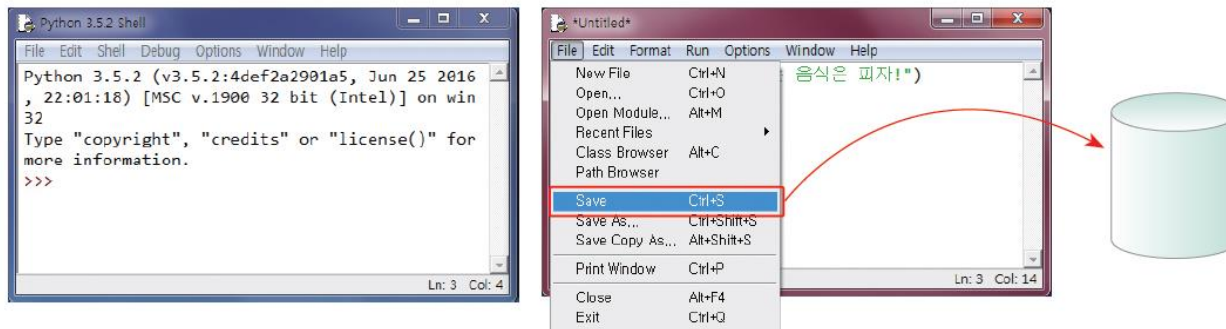


- 파이썬 셸의 메뉴 중에서 [File] -> [New File]을 선택한다.

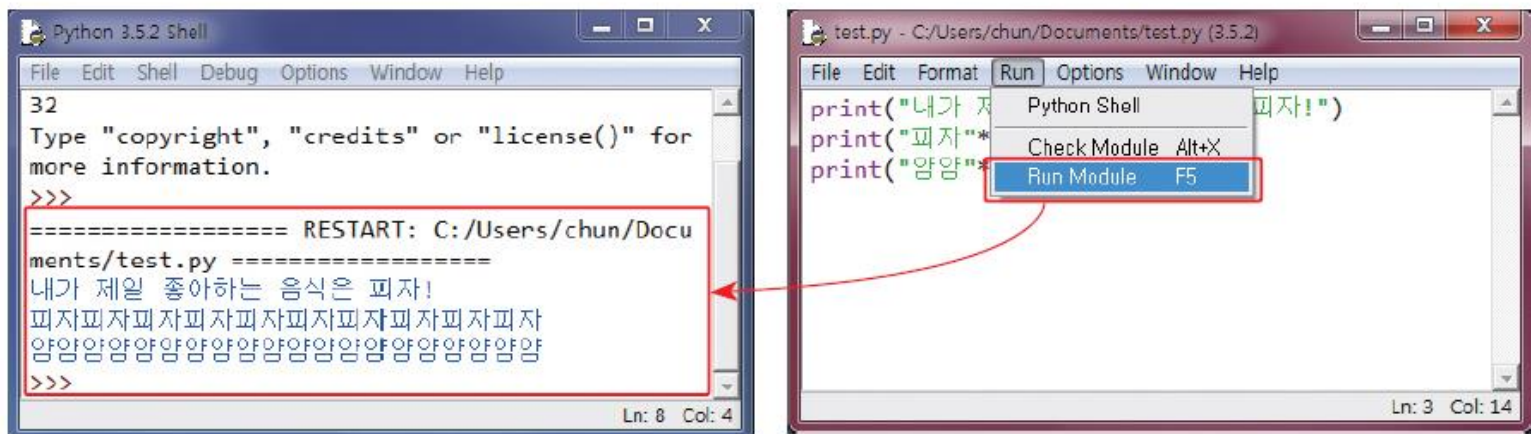


스크립트 모드 : IDLE를 이용한 소스 파일 저장과 실행

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

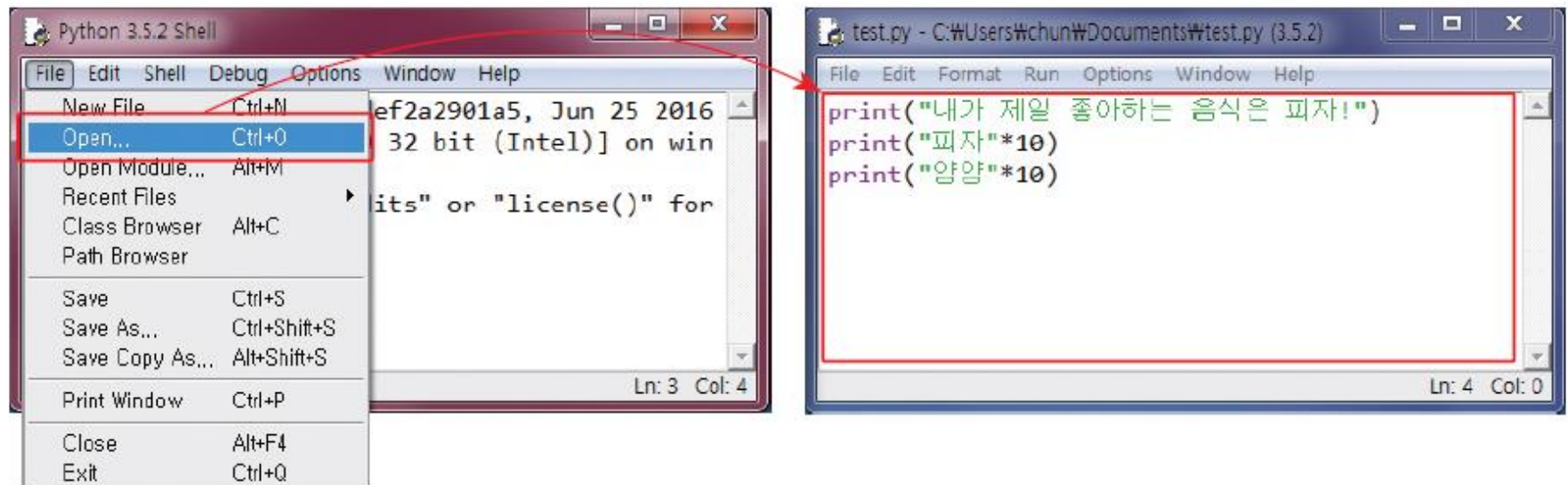


- 파이썬 셸의 메뉴 중에서 메뉴 [Run]->[Run Module]을 선택한다. 단축기 F5 !



스크립트 모드 : 소스 파일 다시 열기

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



도전! : 스크립트 모드에서 소스화일 작성

□ 이 파일을 작성하여 저장한 후 실행해보세요

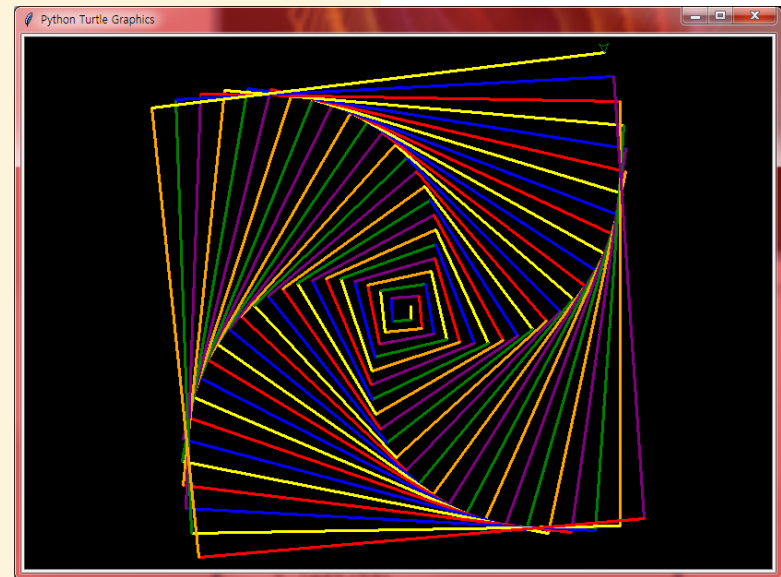
```
import turtle

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
```



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

Lab : print – 삼각형 – 다각형&원 그리기

Lab#1: print() 함수 실습

□ 아래 결과가 출력되도록 빈칸을 채워보세요

```
>>> _____  
안녕하세요? Ewha 여러분
```

```
>>> _____  
반갑습니다.
```

```
>>> _____  
9*8은 72 입니다.
```

```
>>> print("안녕하세요? Ewha 여러분")
```

안녕하세요? Ewha 여러분

```
>>> print("반갑습니다^^")
```

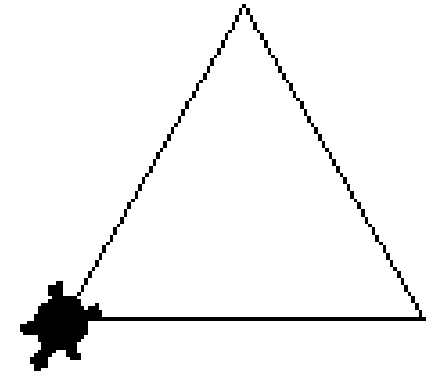
반갑습니다^^

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

9*8은 72 입니다.

Lab#2: 터틀 그래픽으로 삼각형을 그려보자

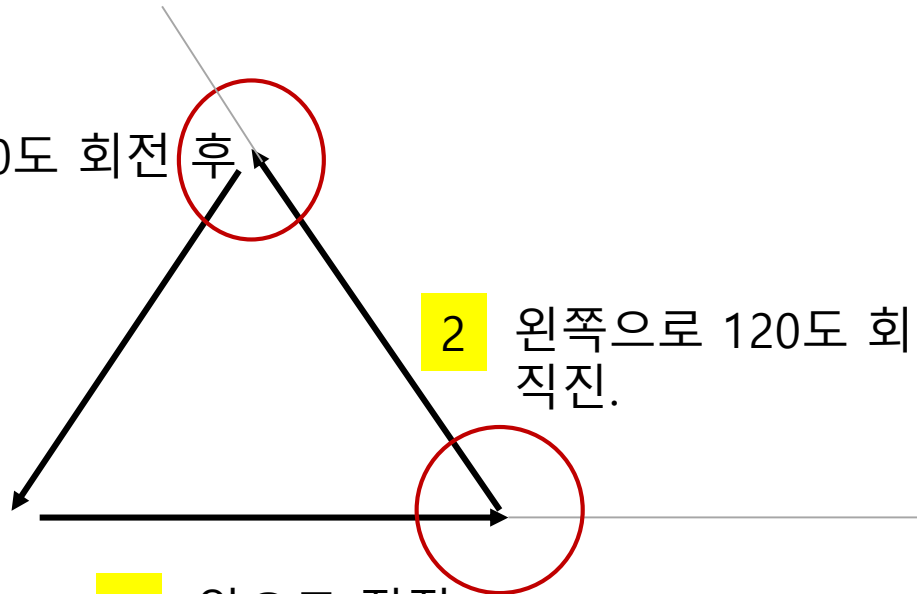
- `t.forward(100)` : 100칸 앞으로 전진
- `t.left(120)`
- 그리는 방법에 따라 `t.right()`을 사용해도 됩니다.



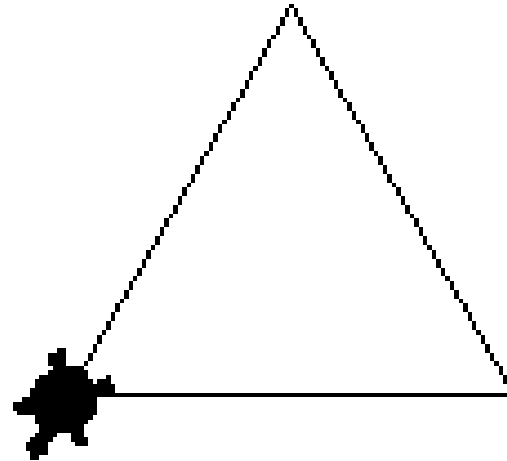
3 왼쪽으로 120도 회전 후
직진.

2 왼쪽으로 120도 회전 후
직진.

1 앞으로 직진

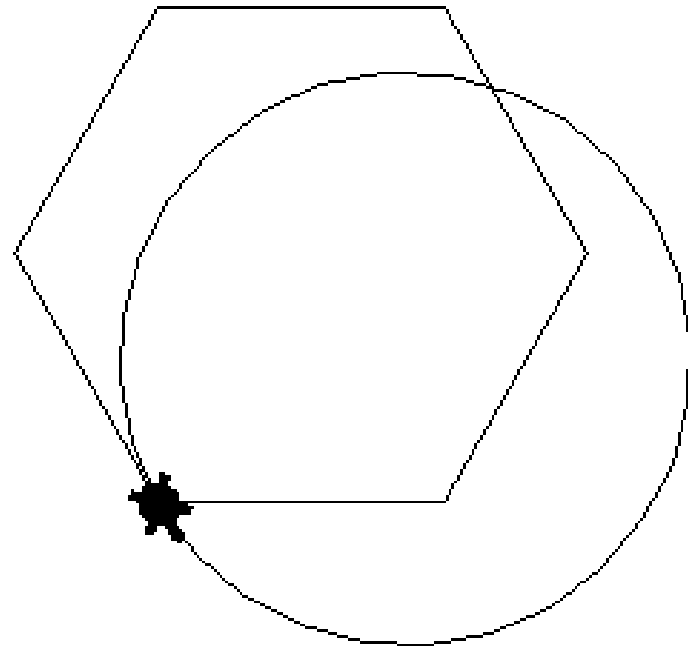



```
>>> import turtle  
>>> t=turtle.Pen()  
>>> t.shape("turtle")  
  
>>> t.forward(100)  
>>> t.left(120)  
>>> t.forward(100)  
>>> t.left(120)  
>>> t.forward(100)
```



Lab#3: 터틀 그래픽으로 원, 다각형을 그려보자

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



```
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.
```

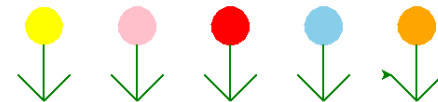
- 프로그램은 컴퓨터에 내리는 명령으로 이루어지는 작업지시서이다.
- 다양한 종류의 프로그래밍 언어가 있고 파이썬도 프로그래밍 언어의 일종이다.
- 파이썬은 <http://www.python.org> 웹사이트에서 다운로드받아서 설치할 수 있다.
- IDLE은 파이썬으로 프로그램을 작성하기 위한 개발 환경이다.

- 산술연산자 : + , - , * , /
- 출력함수 : print()는 화면에 문자열이나 계산 결과를 출력할 수 있다.
- 스크립트 모드와 인터프리트 모드가 있다.
- 거북이를 움직이면서 코딩의 재미를 느껴보자.





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.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()

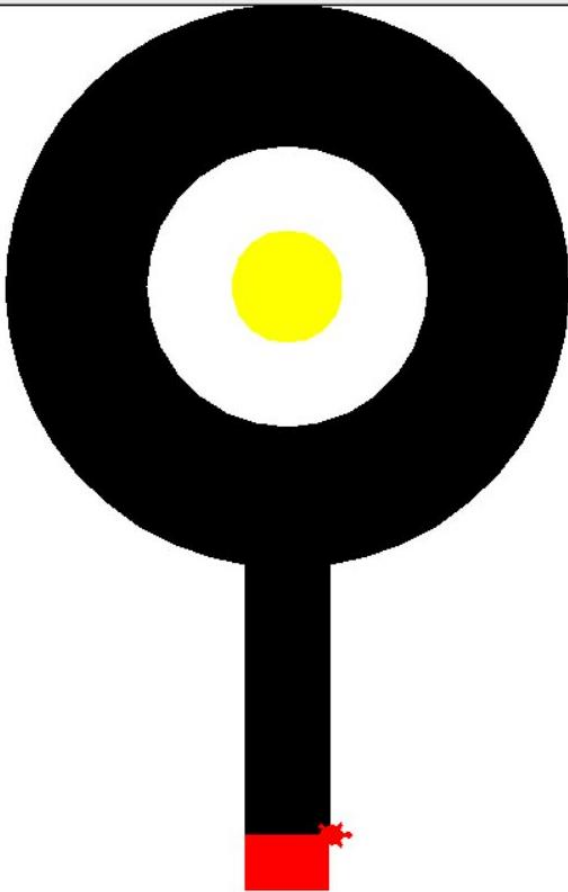
t.penup()
t.rt(180)
t.fd(80)
t.rt(180)
t.begin_fill()
t.pendown()
t.pendown()
t.circle(10)
t.end_fill()

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

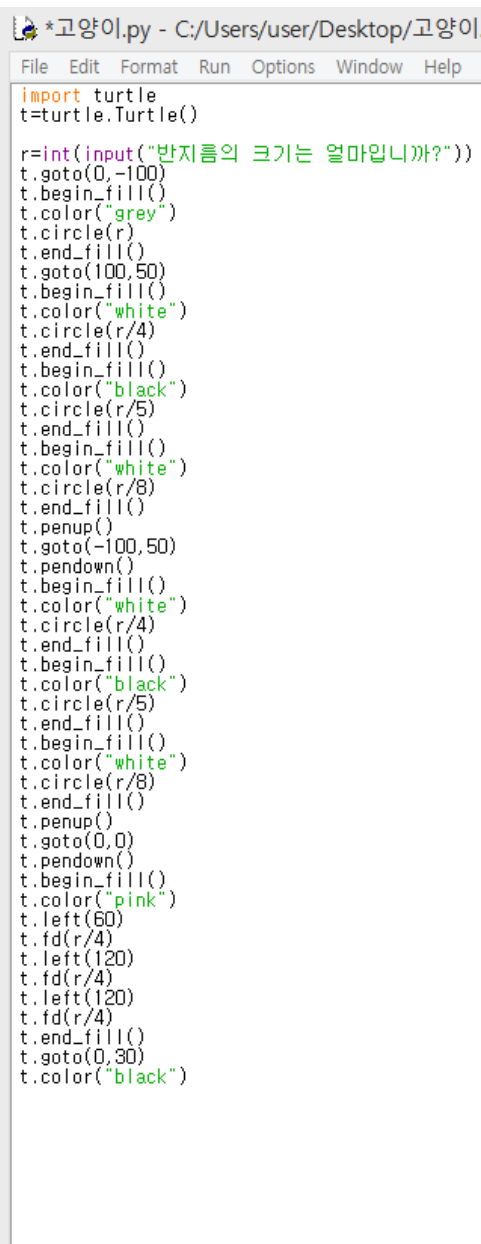
t.color("yellow")
t.begin_fill()
t.circle(40)
t.end_fill()

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.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(180)
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)
```

Day 2 학습활동 : 거북이를 이용하는 코딩 아무거나 ^^

앞서 본 과정을 참고하여 거북이를 이용한 프로그램을 작성 한 후
코드와 함께 화면을 캡취합니다.

재미있는 그림을 만들어보세요 너무 많은 라인이 반복되고 있죠?

Day3에서는 변수와 비교/반복을 이용한 효율적인 방법을 알아보겠습니다

기한 : 2월 21일 금요일까지.

수고하셨습니다