



▶ 학습내용

- 01 프로그래밍 언어의 개념을 학습합니다.
- 02 파이썬을 내 컴퓨터에 설치합니다.
- 03 첫 번째 프로그램을 작성해봅니다.
- 04 코드를 파일에 저장하여 실행하는 방법을 알게 됩니다.
- 05 터틀 그래픽으로 여러 가지 그림을 그려봅니다.

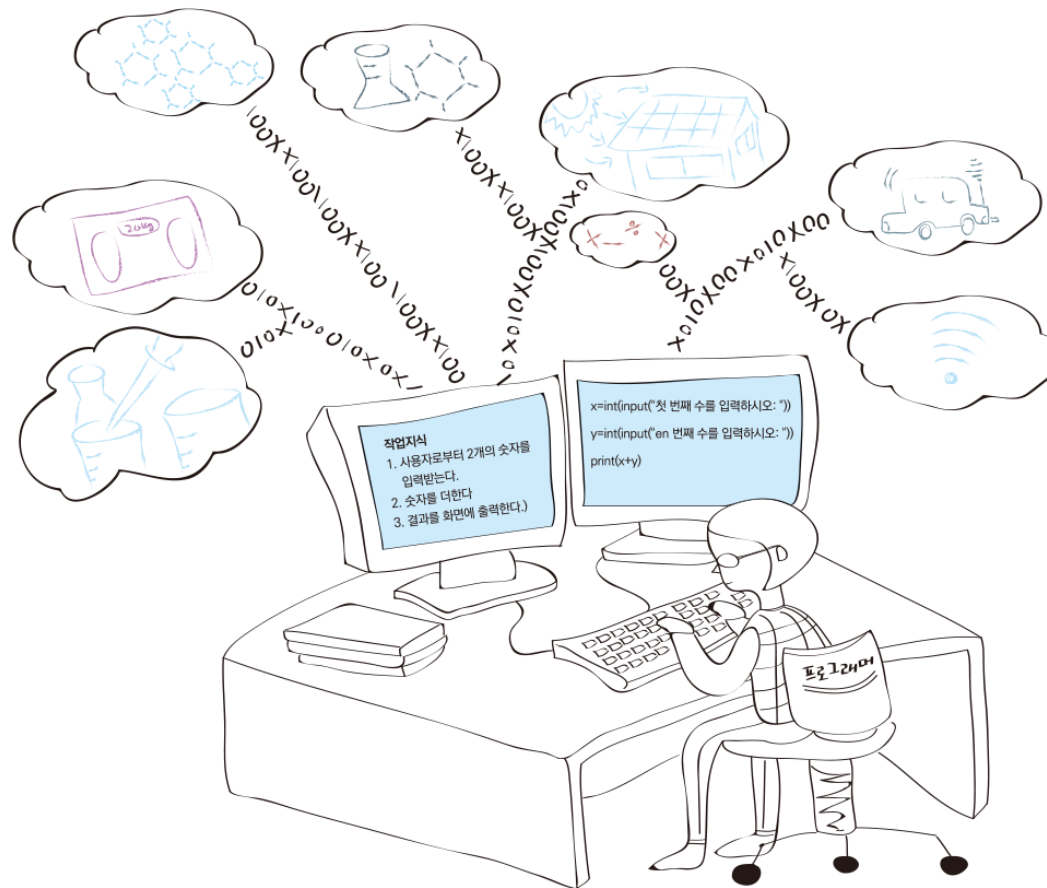
▶ LAB

- 01 print() 실습
- 02 원과 다각형 그리기
- 03 터틀 그래픽 더 살펴보기

01. 프로그래밍이 무엇이죠?



- 우리는 여러 가지 문제를 해결하기 위해 컴퓨터 프로그램을 많이 사용함



01. 프로그래밍이 무엇이죠?



- 프로그램(program)은 컴퓨터에 어떤 일을 어떻게 시킬지를 기록해 놓은 작업 지시서
- 작업의 지시가 컴퓨터에서 작동하려면 프로그래밍 언어로 만들어져야 함.
- 프로그래머 : 프로그래밍 언어를 이용하여 논리적이고 명확하게 프로그램을 작성하여 문제를 해결하는 사람



컴퓨터가 알아듣는 언어는 오직
프로그래밍 언어뿐입니다.

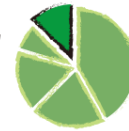
02. 프로그래밍을 배우면 무엇을 할 수 있을까요?

5

컴퓨팅 사고력

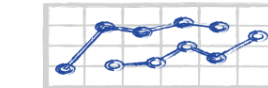
분해

작은 그룹으로 나누어 문제를 해결!



패턴 인식

순서를 찾는다!



데이터를 분석한다!

알고리즘 설계

순서화된 단계를 사용하여 해결책 작성!



02. 프로그래밍을 배우면 무엇을 할 수 있을까요?

6

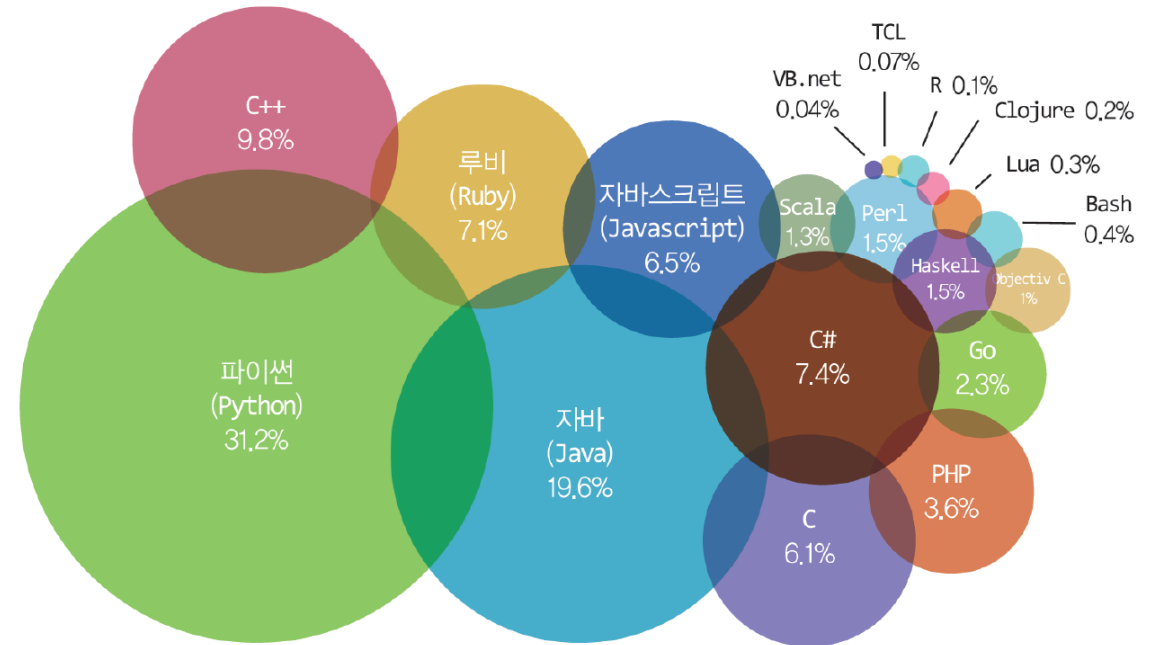
- 컴퓨터를 여러분 마음대로 제어할 수 있습니다.
- 프로그래밍은 생각하는 방법을 가르쳐 줍니다.
- 프로그래밍을 하면 더 창의적인 사람이 됩니다.
- 프로그래밍은 논리적인 사고방식과 문제해결력을 키워줍니다.

03. 두근두근 파이썬



- 프로그래밍 언어의 종류는 다양
- 문제를 해결할 때는 문제의 내용, 목적, 주어진 조건을 따져보고 알맞은 프로그래밍 언어를 선택하는 것이 좋습니다.
- 각각의 프로그래밍 언어는 고유한 문법 체계와 장단점을 가지고 있습니다.

[인기있는 프로그래밍 언어]



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

03. 두근두근 파이썬



파이썬의 인기
비결은?



귀도 반 로섬

- 문법이 쉬워 초보자도 배우기에 좋습니다.
- 오픈 소스이면서도 강력한 언어입니다
- 코드가 간결하고 프로그래밍을 즐기게 해줍니다.

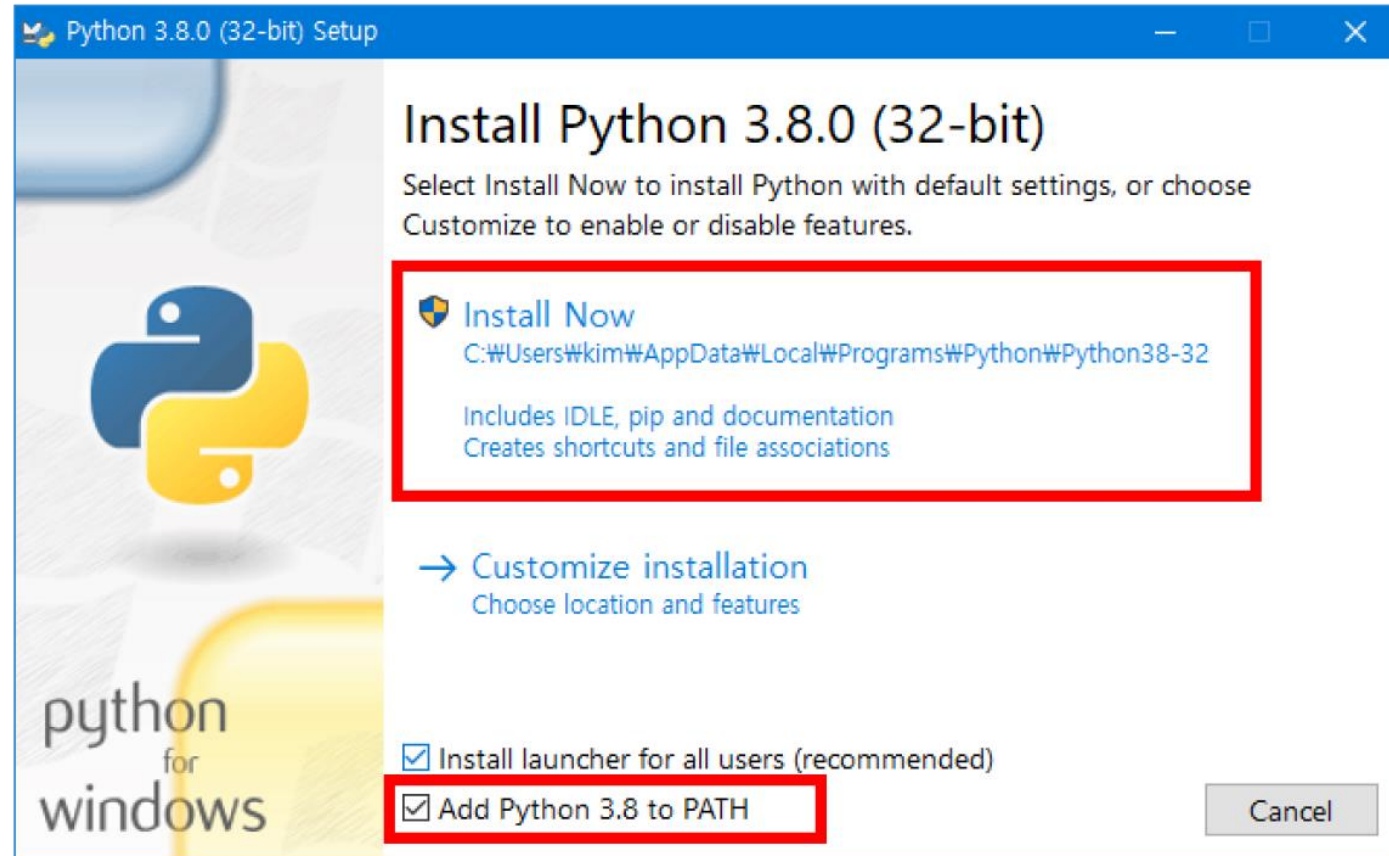
04. 파이썬 설치하기



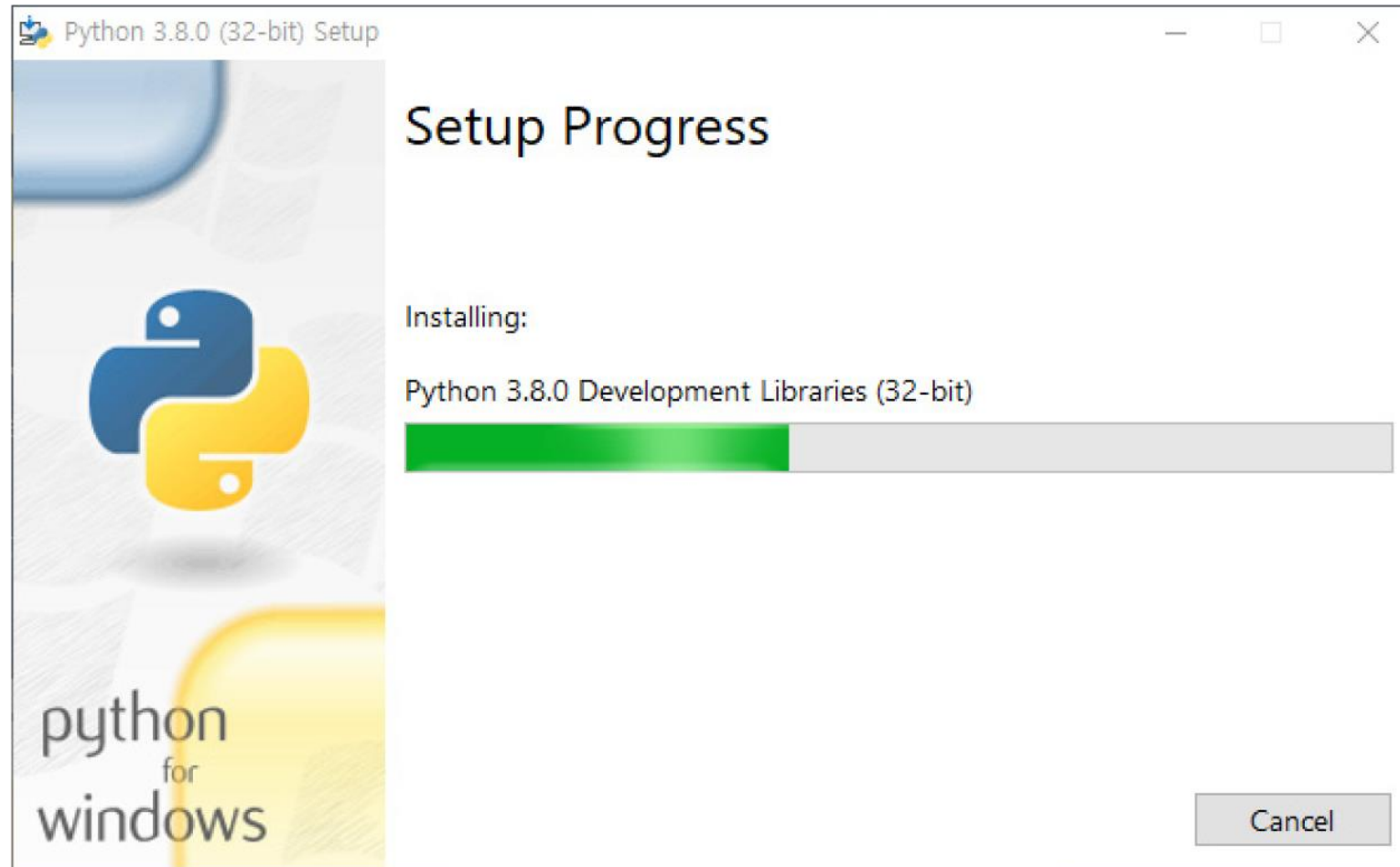
- <http://www.python.org/>에 접속



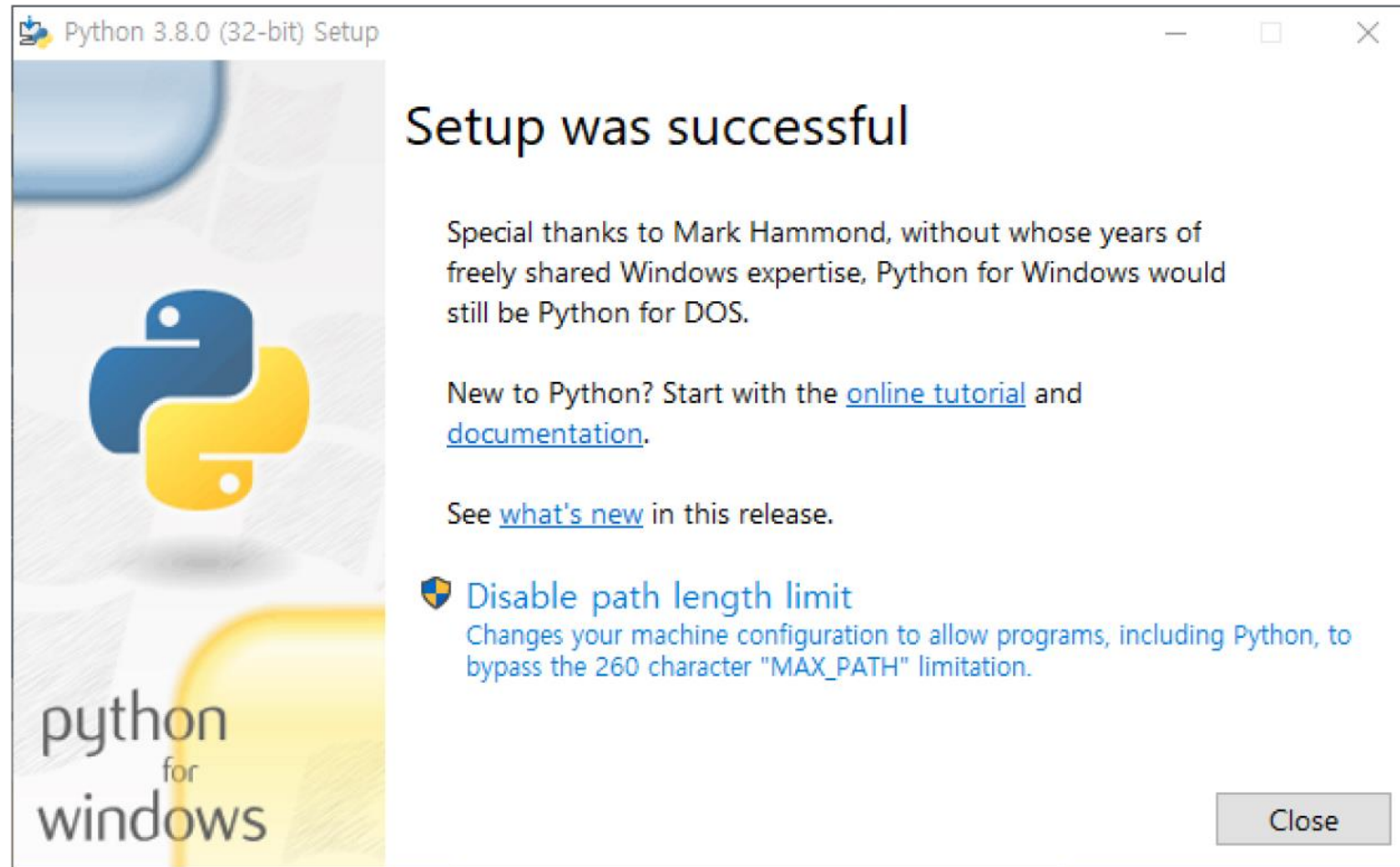
04. 파이썬 설치하기



04. 파이썬 설치하기



04. 파이썬 설치하기



05. 파이썬 IDLE 실행하기



Python 3.8
버전 확인하기!

05. 파이썬 IDLE 실행하기



아래와 같은 화면을 파이썬 셸(python shell)이라고 부릅니다.

```
Python 3.8.0 Shell
File Edit Shell Debug Options Window Help
Python 3.8.0 (tags/v3.8.0:fa919fd, Oct 14 2019, 19:21:23) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> |
Ln: 3 Col: 4
```

05. 파이썬 IDLE 실행하기



```
Python 3.8.0 Shell
File Edit Shell Debug Options Window Help
Python 3.8.0 (tags/v3.8.0:fa919fd, Oct 14 2019, 19:21:23) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> 1+1
2
>>> print("Hello!")
Hello!
>>>
```

Ln: 7 Col: 4

06. 파이썬 첫걸음 - 계산하기



덧셈, 뺄셈, 곱셈, 나눗셈을 파이썬으로 실행해 봅시다. 연습 삼아서 아래의 코드를 한 줄씩 입력하고 실행해 봅시다.

코드 및 실행 결과

```
>>> 2+3
5
>>> 2-3
-1
>>> 2*3
6
>>> 2/3
0.6666666666666666
```

이렇게 명령을 한 줄씩 입력
한 후에 **Enter** 키를 눌러서 실행하는 모드를 대화형 모드 (interpret mode)라고 합니다.

06. 파이썬 첫걸음 - 계산하기



코드 및 실행 결과

```
>>> 2345*9876-5678  
23153542  
>>> 123456789123456789 * 123456789123456789  
15241578780673678515622620750190521
```

이 정도는 정말 계산기로도
안 되네요!

07. 파이썬 첫걸음 - 문자 출력하기



- 파이썬에서 큰따옴표("...")로 둘러싸면 문자열(string)이 됩니다.
- 문자열에 '+' 연산자를 붙이면 문자열과 문자열이 마치 더해진 것처럼 줄줄이 연결됩니다.



다음과 같은 코드를 입력하고 실행해 보세요.

코드 및 실행 결과

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

07. 파이썬 첫걸음 - 문자 출력하기



- 파이썬에서는 텍스트 뒤에 '*'가 붙고 숫자가 있으면 그 숫자만큼 텍스트를 반복하게 됩니다.



이번에는 문자열에서의 '*'연산자를 살펴보겠습니다.

코드 및 실행 결과

```
>>> print("반가워요 " * 20)
```

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

07. 파이썬 첫걸음 - 문자 출력하기



파이썬에서 큰따옴표(" ")의 유무에 따라 문자열(string)과 수(number)를 구별합니다. 예를 들어서 "100"은 문자열이고 100은 숫자입니다.

큰따옴표(" ")로 둘러싸
이면 문자열(string)

☑ "100"+"200"을 실행하면 "100200"이 출력됩니다.
문자열이 서로 연결됩니다.

코드 및 실행 결과

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

☑ 100+200을 실행하면 300이 출력됩니다. 정수에 대
한 덧셈이 수행됩니다.

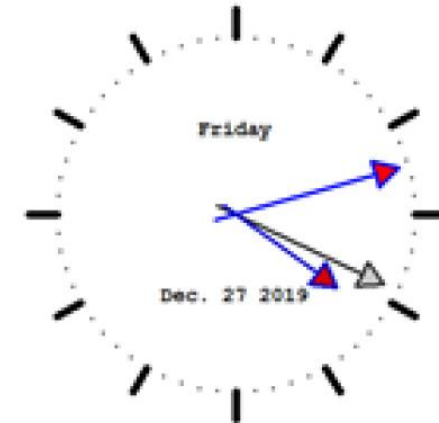
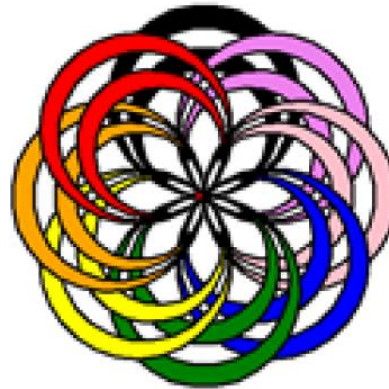
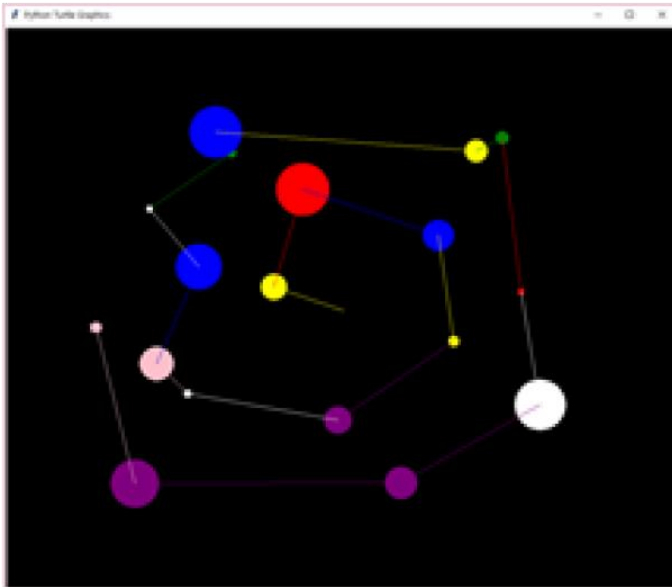
코드 및 실행 결과

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

08. 파이썬 첫걸음 – 그리기1



- 터틀 그래픽(turtle graphic) 모듈



08. 파이썬 첫걸음 – 그리기1



- 터틀 그래픽은 동작
 - ① 프로그램이 시작하면 종이의 한 가운데 거북이가 나타납니다.
 - ② 명령을 내리면 거북이가 움직입니다.
예) “앞으로 전진”, “뒤로 후진”, “좌회전”, “우회전”
 - ③ 거북이가 움직이면서 종이 위에 그림을 그립니다.
(거북이가 펜을 가지고 움직인다고 생각해보세요)



08. 파이썬 첫걸음 – 그리기1



이제부터 그림을 그려볼까요? 파이썬 셸에서 다음과 같이 입력합니다.

코드

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

실행 결과



08. 파이썬 첫걸음 – 그리기1



다음과 같은 명령어를 이용하여 거북이를 앞으로 전진시켜보겠습니다.

코드

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

실행 결과



08. 파이썬 첫걸음 – 그리기1



거북이를 왼쪽으로 90° 회전시킨 후에 전진시켜 보겠습니다.

코드

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

실행 결과



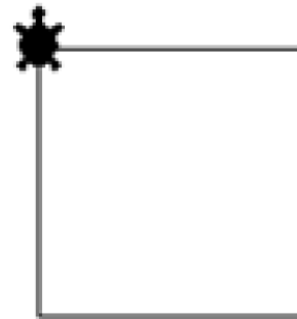
09. 파이썬 첫걸음 – 그리기2



코드

```
>>> 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)
```

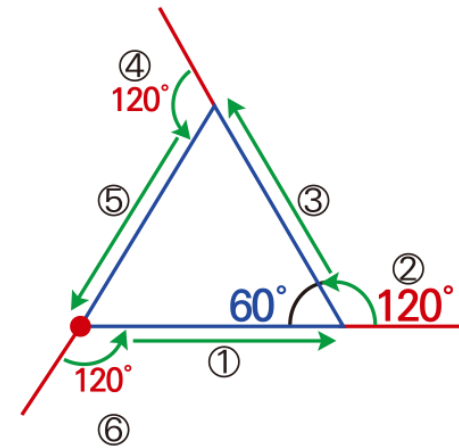
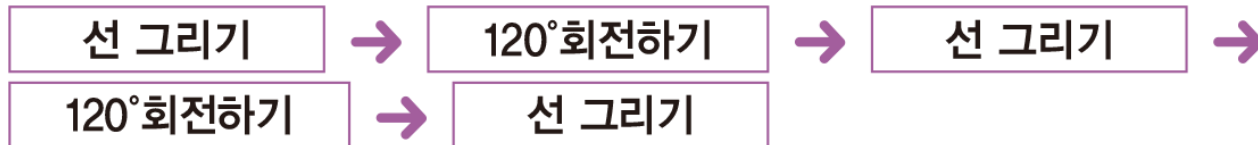
실행 결과



09. 파이썬 첫걸음 – 그리기2



그렇다면 삼각형은 어떻게 해야 할까요? 삼각형의 위치는 신경 쓰지 않고 정삼각형이 되도록 그려보겠습니다.
삼각형은 다음과 같은 과정으로 그려집니다.



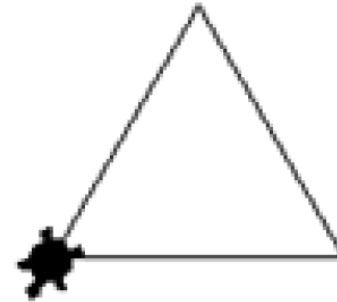
09. 파이썬 첫걸음 – 그리기2



코드

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

실행 결과



10. 스크립트 모드



- 우리의 프로그램 코드는 더욱 복잡해지고 길어지므로 코드를 저장할 필요가 있습니다.
- 파이썬에는 스크립트 모드(script mode)에서 파이썬 프로그램 파일을 만들어 코드를 저장하고 실행할 수 있습니다.



파이썬 파일을 작성하고 저장하기

(1) IDLE의 [File] → [New File] 메뉴를 선택합니다. 윈도우의 ‘메모장’ 프로그램과 같은 텍스트 에디터가 등장합니다. 여기에 아래와 같은 파이썬 코드를 입력합니다.

코드

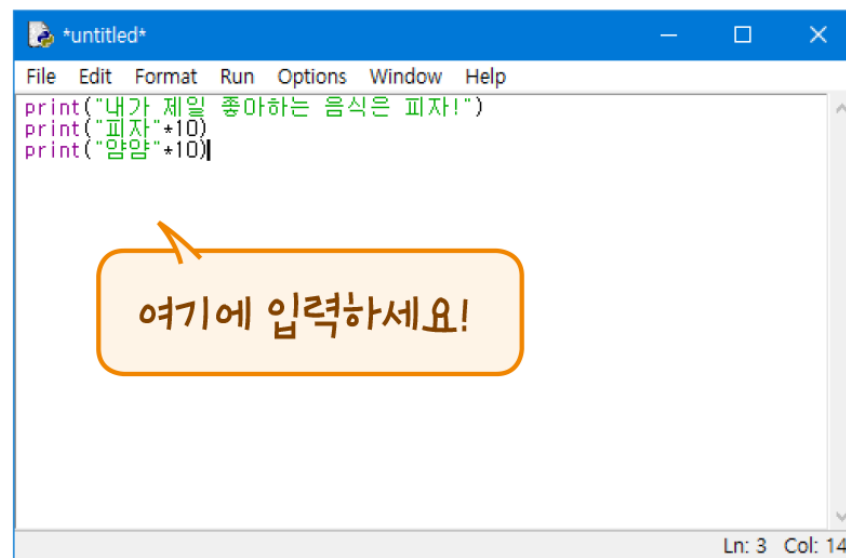
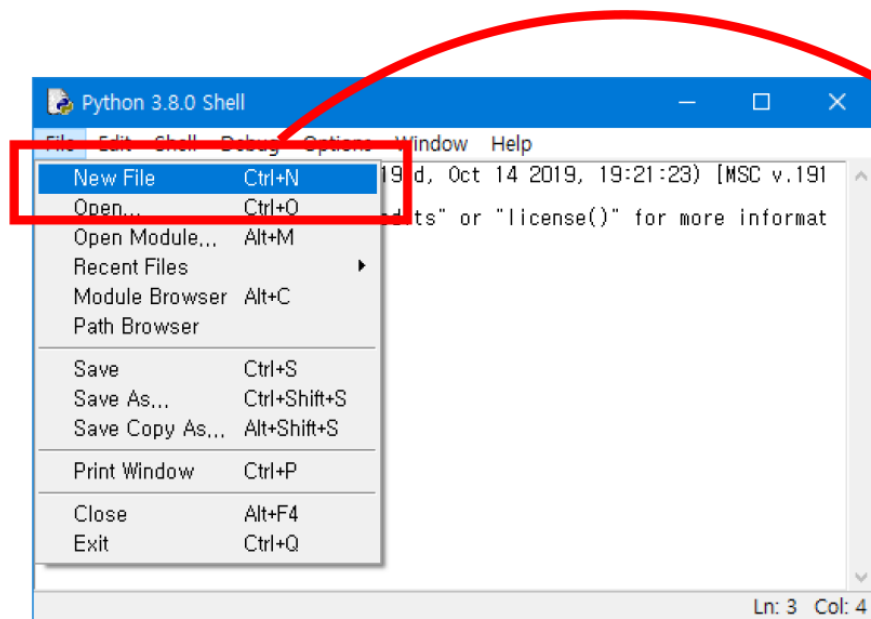
```
print("내가 제일 좋아하는 음식은 피자!")  
print("피자"*10)  
print("암암"*10)
```

10. 스크립트 모드



파이썬 파일을 작성하고 저장하기

- IDLE의 [File] → [New File]
- 아래의 코드를 입력

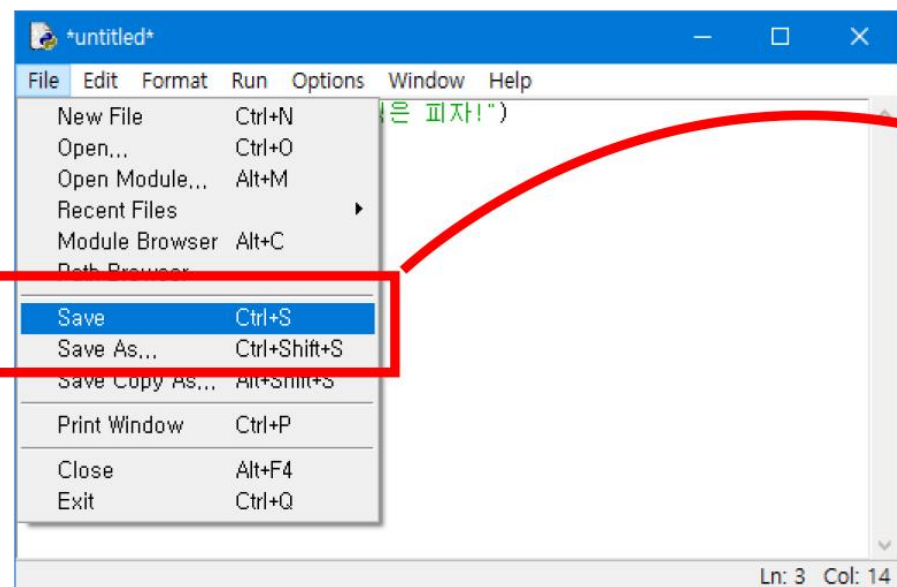


10. 스크립트 모드



파이썬 파일을 작성하고 저장하기

- [File] → [Save]
- 파일 확장자: .py



저장장치

10. 스크립트 모드

32



파이썬 파일을 작성하고 저장하기

- [Run] → [Run Module]
- F5키

```
Python 3.8.0 Shell
File Edit Shell Debug Options Window Help
Python 3.8.0 (tags/v3.8.0:fa919fd, Oct 14 2019, 19:21:23) [MSC v.191
6 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more informat
ion.
>>>
= RESTART: C:/Users/kim/AppData/Local/Programs/Python/Python38-32/te
st.py
내가 제일 좋아하는 음식은 피자!
피자피자피자피자피자피자피자피자피자피자피자피자피자피자피자
암암암암암암암암암암암암암암암암암암암암암암암암암암암암
>>>
```

Ln: 8 Col: 4

```
test.py - C:/Users/kim/AppData/Local/Programs/Python/Pyth...
File Edit Format Run Options Window Help
print("내가 제일 좋아하는 음식은 피자!")
print("피자"*10)
print("암암"*10)
]
```

Run Module F5
Run Customized Shift+F5
Check Module Alt+X
Python Shell

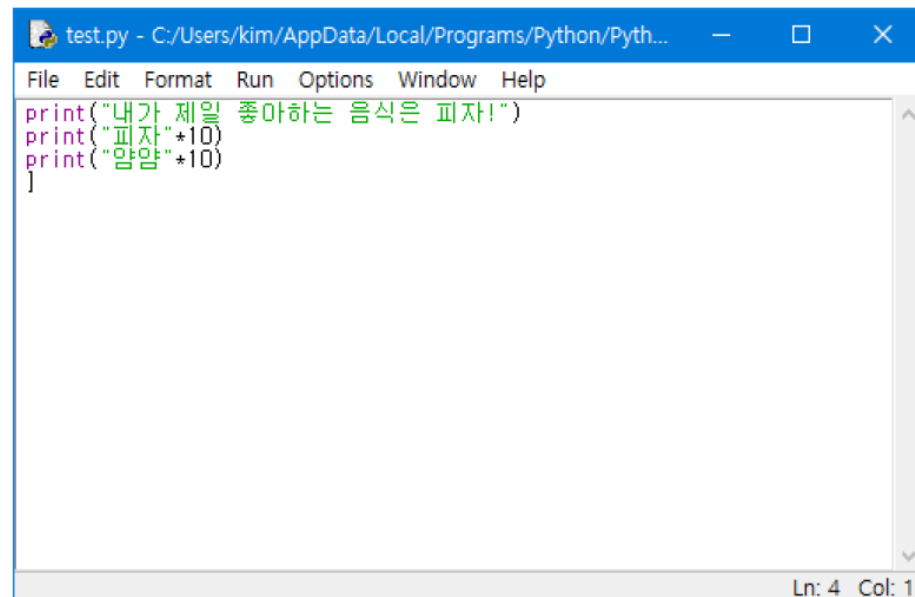
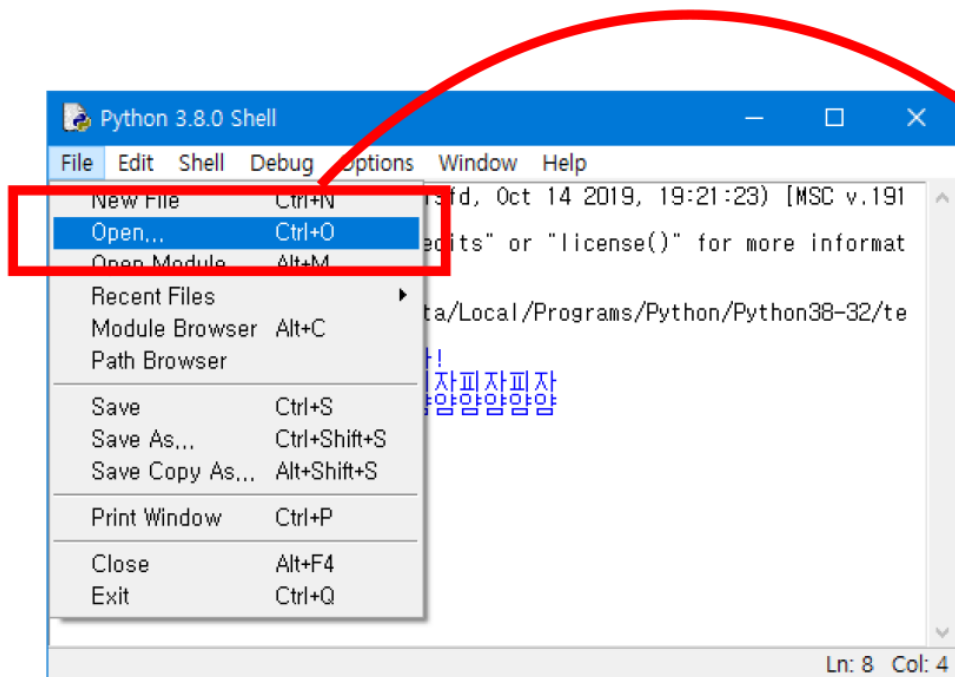
Ln: 4 Col: 1

10. 스크립트 모드



파이썬 파일을 불러서 열어보기

- IDLE의 [File] → [Open]



11. 읽으면 도움이 돼요 - 주석



- 주석(comment) : 프로그램을 설명하는 글
- '#'로 시작하는 부분부터 줄의 끝까지 주석으로 취급

코드

```
# 파이썬 프로그램  
print("안녕")    # '안녕' 출력
```

실행 결과

안녕

11. 읽으면 도움이 돼요 - 주석



그렇다면 주석이 파이썬 프로그램과 무슨 상관관계가 있는 것일까요?

- 주석은 코드의 해석을 쉽게 하도록 도와주는 설명글

코드

```
print("파이"*10) #파이 10번 출력
```

실행 결과

파이파이파이파이파이파이파이파이파이파이파이

- 소스코드의 특정 부분을 잠시 사용하지 않게 만들고 싶은 경우에 사용

코드

```
print("프로그래밍 공부를 즐기셨으면 합니다.")  
#print("안녕!" * 3)  
print("화이팅!" * 5)
```

실행 결과

프로그래밍 공부를 즐기셨으면 합니다.
화이팅!화이팅!화이팅!화이팅!화이팅!

12. 문제가 생겼어요 - 오류



문법 오류

- SyntaxError: invalid syntax

코드 및 실행 결과

```
>>> pront("Hello World")
Traceback (most recent call last):
  File "<pyshell#0>", line 1, in <module>
    pront("Hello World")
NameError: name 'pront' is not defined
```

무엇이 잘못 되었을까요?

코드 및 실행 결과

```
>>> 1 +
SyntaxError: invalid syntax
>>> 3 +* 2
SyntaxError: invalid syntax
```

← 문법을 지키라는 의미입니다.

12. 문제가 생겼어요 - 오류



실행 중간에 발생하는 오류

코드

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

실행 결과

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

Traceback (most recent call last):

File "DWs.py", line 3, in <module>

`print("안녕!" + 3)`

TypeError: can only concatenate str
(not "int") to str

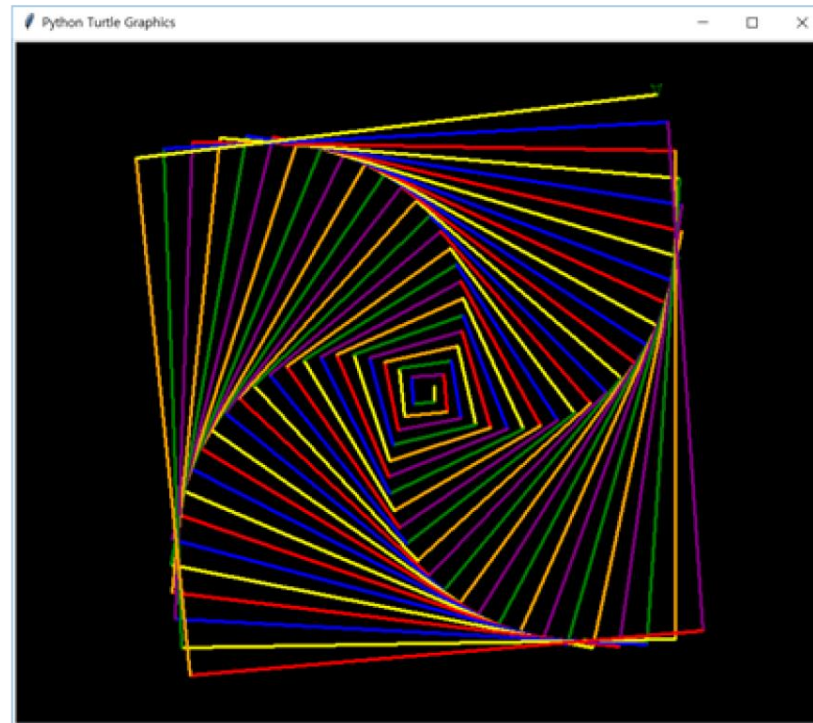
`print("안녕!" + 3)`



수정

`print("안녕!" * 3)`

13. 파이썬으로 무엇을 할 수 있을까?



13. 파이썬으로 무엇을 할 수 있을까?



코드

```
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
```

Lab. print() 실습

41



(1)

코드

```
>>> print("시간순삭 파이썬")
```

실행 결과

시간순삭 파이썬

(2)

코드

```
>>> -----
```

실행 결과

9*8은 72입니다.

Lab. print() 실습



(3) 운율

- '운율'이란 흔히 '리듬(rhythm)'이라고 부르는 것
- print() 함수를 이용하여 출력. 단, 효과적인 프로그램이 될 수 있도록 고민해 보세요

너무 반짝반짝 눈이 부셔 No No No No No
너무 깜짝깜짝 놀란 나는 Oh Oh Oh Oh Oh
너무 짜릿짜릿 몸이 떨려 Gee Gee Gee Gee Gee

Lab. 원과 다각형 그리기

43



1 원 그리기

코드

```
import turtle  
t = turtle.Turtle()  
t.shape("turtle")  
t.circle(100) #반지름이 100인 원을 그림
```

실행 결과



Lab. 원과 다각형 그리기

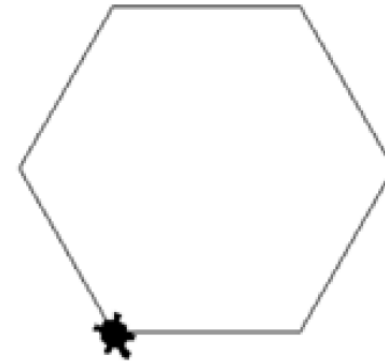


2 정육각형 그리기

코드

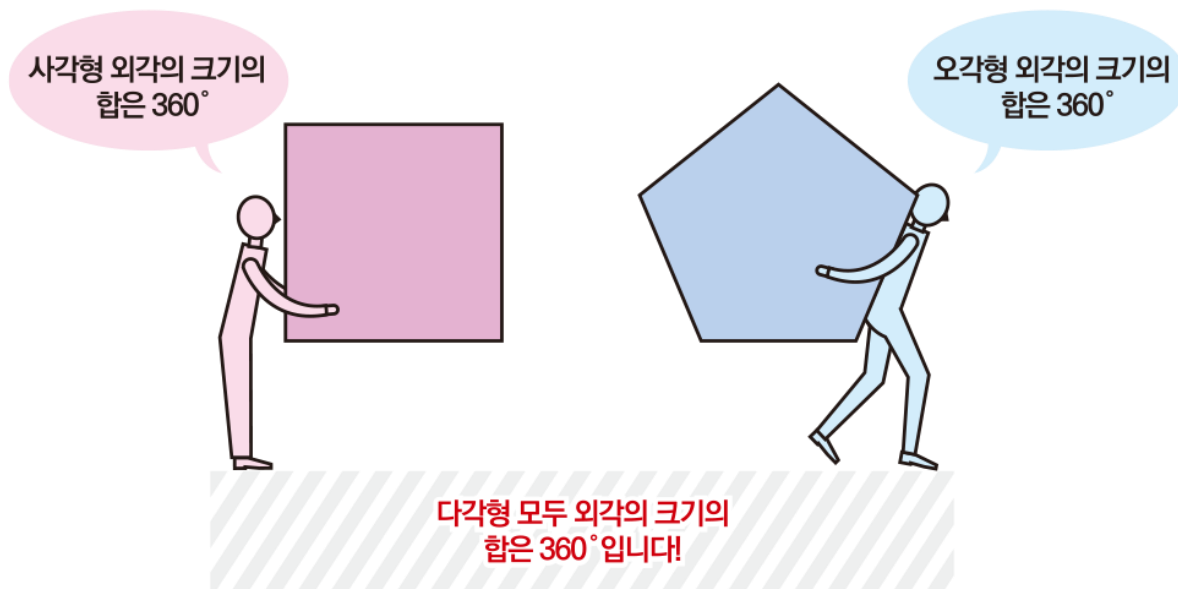
```
import turtle
t = turtle.Turtle()
t.shape("turtle")
t.forward(100)
t.left(60)
t.forward(100)
t.left(60)
t.forward(100)
t.left(60)
t.forward(100)
t.left(60)
t.forward(100)
t.left(60)
t.forward(100)
```

실행 결과





3 정다각형 그리기



$$\begin{aligned} & (\text{정}n\text{각형 한 외각의 크기}) \\ &= (n\text{각형의 외각의 크기의 합}) \div n \\ &= 360^\circ \div n \\ &= \frac{360^\circ}{n} \end{aligned}$$

Lab. 터틀 그래픽 더 살펴보기

46



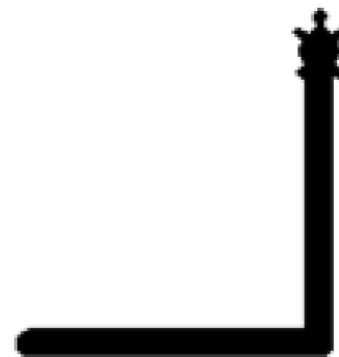
생각 1 : t.width(10)을 호출하면 선의 두께가 10이 됩니다.

코드

```
import turtle
t = turtle.Turtle()
t.shape("turtle")
t.width(10)

t.forward(100)
t.left(90)
t.forward(100)
```

실행 결과



Lab. 터틀 그래픽 더 살펴보기

47



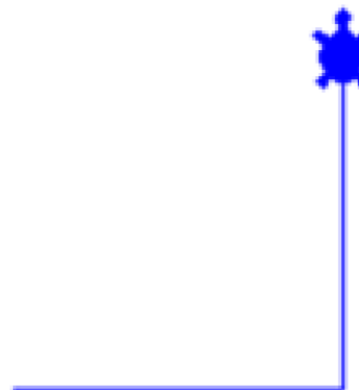
생각 2 : `t.color("blue")`을 호출하면 선의 색상이 파란색이 됩니다. 색상은 영어 단어로 표시합니다.

코드

```
import turtle
t = turtle.Turtle()
t.shape("turtle")
t.color("blue")

t.forward(100)
t.left(90)
t.forward(100)
```

실행 결과



Lab. 터틀 그래픽 더 살펴보기

48



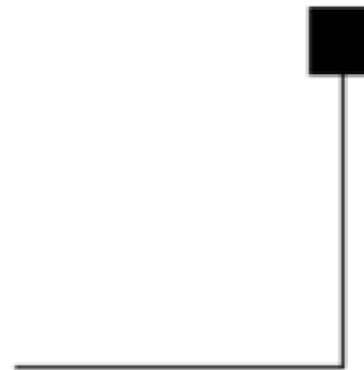
생각 3 : `t.shape("square")`를 호출하면 터틀의 모양을 사각형으로 바꿀 수 있습니다. "arrow", "turtle", "circle", "square", "triangle", "classic" 등의 모양이 가능합니다.

코드

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

t.forward(100)
t.left(90)
t.forward(100)
```

실행 결과



Lab. 터틀 그래픽 더 살펴보기



생각 4 : 터틀 그래픽의 좌표계는 수학의 평면 좌표와 동일하게 (x, y)로 생각하면 됩니다. 화면 중앙이 원점=(0,0)이 됩니다.

코드

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

t.down()          # 펜을 내림
t.goto(100, 0)    #(100, 0)까지 선을 그리며 이동
t.up()           # 펜을 올림
t.goto(0, 200)    #(0, 200)까지 선을 그리지 않고 이동
t.down()         # 펜을 내림
t.goto(100,200)   #(100,200)까지 선을 그리며 이동
```

실행 결과

