



파이썬과 친해지기

while 반복문, 대화 프로그램, turtle 모듈 활용

9주차

복습

apple



사과. 사과나무의 열매.

bear



곰. 갈색의 털북숭이 포유류.

car



자동차. 바퀴가 달린 이동수단.

key

열쇠

value

값

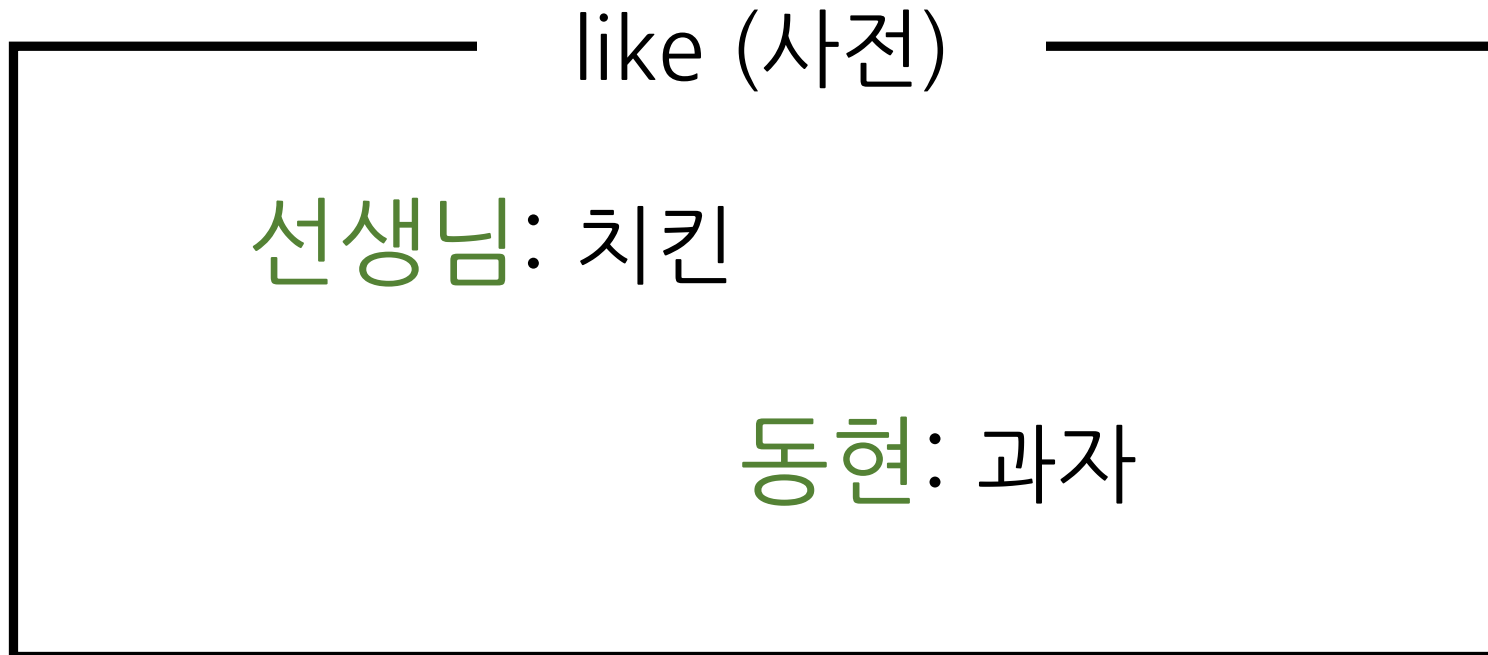
복습

딕셔너리, 즉 사전은 중괄호를 사용한다.

```
word = {"apple": "사과"}  
like = {"선생님": "치킨",  
        "동현": "과자"}  
empty = {}
```

복습

```
like = {"선생님": "치킨",  
        "동현": "과자"}
```



복습

```
like = {"선생님": "치킨",  
        "동현": "과자"}
```



복습

```
>>> 메뉴 = {"치킨": 18000,  
            "공깃밥": 1000,  
            "라면": 3000}
```

```
>>> "치킨" in 메뉴
```

```
True
```

```
>>> "샐스핀" in 메뉴
```

```
False
```

$A \text{ in } B$

A라는 열쇠가 B에 있는가?

복습

```
>>> 메뉴 = {"치킨": 18000,  
            "공깃밥": 1000,  
            "라면": 3000}
```

```
>>> for a in 메뉴:  
    print(a)
```

치킨
공깃밥
라면

for a in 메뉴
“메뉴” 사전 속 모든 열쇠를
a 에 한번 씩 담아 반복

생각해보기



방청소 하는 법

눈에 보이는 쓰레기를 치운다.
눈에 보이는 쓰레기를 치운다.
눈에 보이는 쓰레기를 치운다.
눈에 보이는 쓰레기를 치운다.
눈에 보이는 쓰레기를 치운다.

생각해보기



방청소 하는 법

눈에 보이는 쓰레기를 치운다.
눈에 보이는 쓰레기를 치운다.
눈에 보이는 쓰레기를 치운다.
눈에 보이는 쓰레기를 치운다.
눈에 보이는 쓰레기를 치운다.



어떤 조건이 참인 동안:
어떤 행동을 한다.

생각해보기



방청소 하는 법

눈에 보이는 쓰레기를 치운다.
눈에 보이는 쓰레기를 치운다.
눈에 보이는 쓰레기를 치운다.
눈에 보이는 쓰레기를 치운다.
눈에 보이는 쓰레기를 치운다.



방에 쓰레기가 있는 동안:
눈에 보이는 쓰레기를 치운다.

생각해보기



치킨을 다 먹는 법

치킨을 한조각 먹는다.
치킨을 한조각 먹는다.
치킨을 한조각 먹는다.
치킨을 한조각 먹는다.

생각해보기



치킨을 다 먹는 법

치킨을 한조각 먹는다.
치킨을 한조각 먹는다.
치킨을 한조각 먹는다.
치킨을 한조각 먹는다.



어떤 조건이 참인 동안:
어떤 행동을 한다.

생각해보기



치킨을 다 먹는 법

치킨을 한조각 먹는다.
치킨을 한조각 먹는다.
치킨을 한조각 먹는다.
치킨을 한조각 먹는다.



치킨 조각이 남아있는 동안:
치킨을 한조각 먹는다.

while 반복문

while A:

A(조건)가 참인 동안 블록을 반복한다.

```
>>> while True:  
    print("안녕!")
```

안녕?
안녕?
안녕?
(무한반복)

while 반복문 예제

```
a = 0
while a < 3:
    print("꼬꼬댁!")
    a += 1
```



```
>>> %Run chicken.py
꼬꼬댁!
꼬꼬댁!
꼬꼬댁!
```


while 반복문 예제

```
치킨 = 50
while 치킨 > 0:
    치킨 -= 1
print("다 먹었다.", 치킨, "마리 남음")
```



```
>>> %Run chicken.py
다 먹었다. 0 마리 남음
```


while 반복문 예제

```
while True:  
    print("꼬꼬댁!")
```



```
>>> %Run chicken.py
```

```
꼬꼬댁!  
꼬꼬댁!  
꼬꼬댁! (무한반복)  
꼬꼬댁!  
꼬꼬댁!
```

break

break

반복문을 탈출한다.

```
>>> while True:  
    print("안녕?")  
    break  
    print("좋은 아침!")
```

안녕?

break 예제

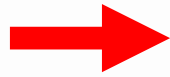
```
a = 1
while True:
    a = a * 2
    if a > 1000:
        break
print(a)
```



```
>>> %Run chicken.py
1024
```

break 예제

```
합 = 0
while True:
    a = input("몇을 합칠까요?")
    if a == "그만":
        break
    합 += int(a)
print(합)
```



```
>>> %Run chicken.py
몇을 합칠까요?5
몇을 합칠까요?8
몇을 합칠까요?3
몇을 합칠까요?그만
16
```

직접 해보기

1. **a = 10** 으로 두자. a가 0보다 큰 동안 a를 출력하고 1씩 빼자. 이를 통해 로켓 카운트다운을 만들어보자.

힌트: while 반복문을 이용하자.

```
>>> %Run chicken.py
```

```
10 ...
```

```
9 ...
```

(생략)

```
2 ...
```

```
1 ...
```

```
발사!
```

2. 823의 배수 중에서 12000을 처음으로 넘는 수를 구해서 출력해보자.

힌트: while 반복문을 이용하자. “합” 변수를 0으로 두고 12000보다 작은 동안 계속 823을 더해보자.

```
합 = 0
while ????:
    ???
print(합)
```

3. 1, 2, 3, 4, ... 와 같이 수를 끝없이 출력하자.

힌트: while True를 이용해 무한반복을 만들 수 있다.

```
>>> %Run hey.py
```

```
1
```

```
2
```

```
3
```

```
4
```

```
5
```

```
6
```

```
7
```

```
...
```

대화 프로그램

```
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 10.0.18362.239]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\lumer>ipconfig

Windows IP Configuration

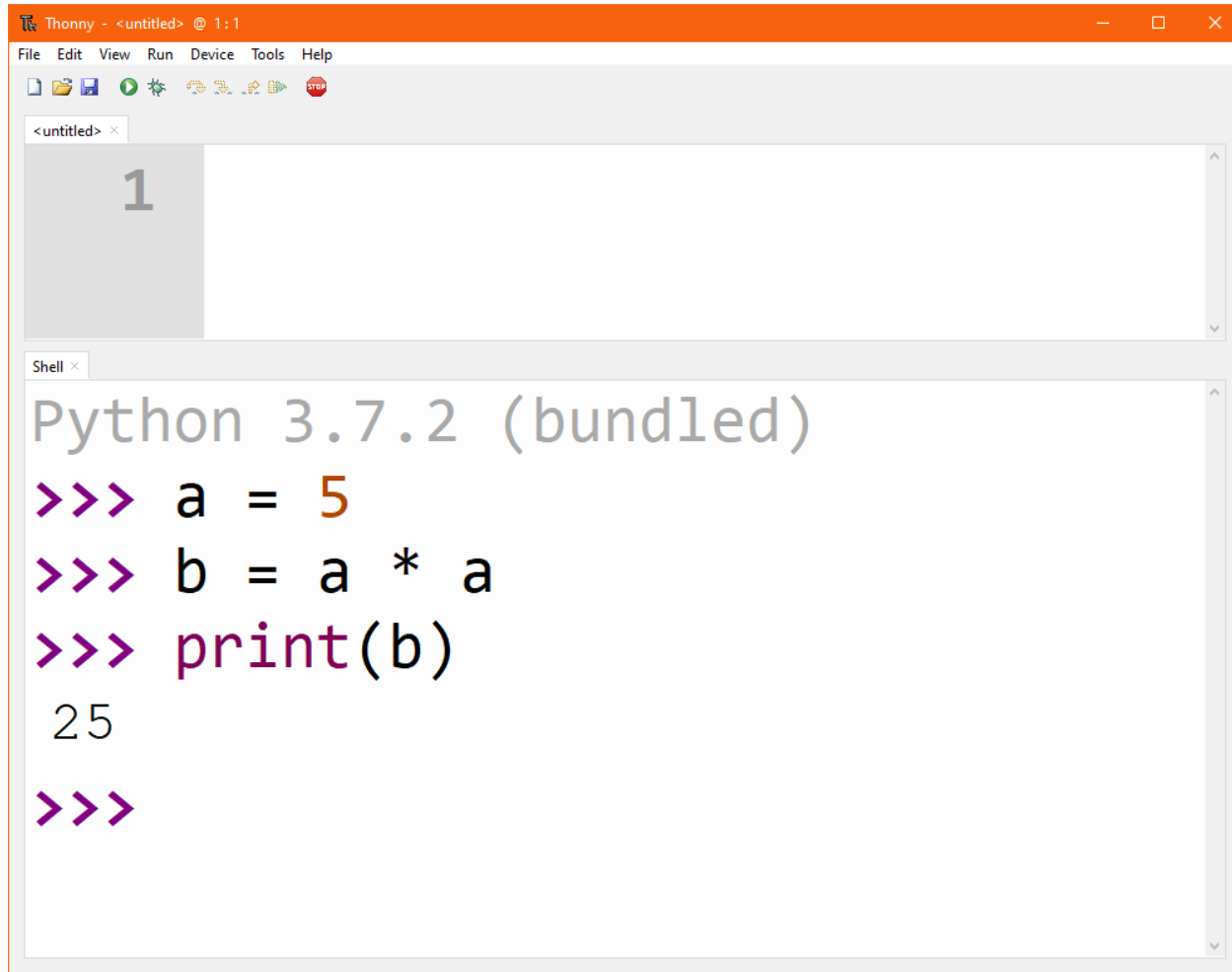
Ethernet adapter Ethernet:

    Connection-specific DNS Suffix  . : kornet
    Link-local IPv6 Address . . . . . : fe80::1ccf:9670:6c89:a76b%8
    IPv4 Address. . . . . : 192.168.0.124
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 192.168.0.1

C:\Users\lumer>echo 안녕하세요!
안녕하세요!

C:\Users\lumer>_
```


대화 프로그램



The image shows a screenshot of the Thonny Python IDE. The window title is "Thonny - <untitled> @ 1:1". The menu bar includes File, Edit, View, Run, Device, Tools, and Help. The toolbar contains icons for opening files, saving, running, and other functions. The main editor area shows a single line of code: `1`. Below the editor is a shell window titled "Shell x". The shell window displays the output of a Python script, showing the prompt `>>>` followed by the code `a = 5`, `b = a * a`, and `print(b)`. The output of the script is `25`, and the prompt `>>>` is shown again.

```
Thonny - <untitled> @ 1:1
File Edit View Run Device Tools Help
<untitled> x
1
Shell x
Python 3.7.2 (bundled)
>>> a = 5
>>> b = a * a
>>> print(b)
25
>>>
```

대화 프로그램

“뭘 도와드릴까요?”

“메뉴는 뭐가 있나요?”

“콜라, 사이다, 환타가 있습니다. 또 뭘 도와드릴까요?”

“콜라 주세요.”

“네 알겠습니다, 콜라 주문하셨습니다. 또 도와드릴까요?”

“주문 그만 할게요.”

“네 알겠습니다. 프로그램 종료.”

대화



같이 해보기

1. 대화 프로그램을 만들어보자.

이 프로그램에는 다음과 같이 두 가지 기능이 있다.

```
print()  
print("1. 선생님이 좋아하는 음식 출력")  
print("2. 프로그램 종료")
```

```
print()  
print("1. 선생님이 좋아하는 음식 출력")  
print("2. 프로그램 종료")
```

기능 소개

```
print()  
print("1. 선생님이 좋아하는 음식 출력")  
print("2. 프로그램 종료")  
while True:
```

반복해서 계속 받아야 하므로 반복문

```
print()
print("1. 선생님이 좋아하는 음식 출력")
print("2. 프로그램 종료")
while True:
    print()
    a = input("뭘 도와드릴까요?")
```

2. 명령어를 받음

```
print()
print("1. 선생님이 좋아하는 음식 출력")
print("2. 프로그램 종료")
while True:
    print()
    a = input("뭘 도와드릴까요?")
    if a == "1":
        print("치킨...")
```

11번 2점3점


```
print()
print("1. 선생님이 좋아하는 음식 출력")
print("2. 프로그램 종료")
while True:
    print()
    a = input("뭘 도와드릴까요?")
    if a == "1":
        print("치킨...")
    elif a == "2":
        print("이용해주셔서 감사합니다.")
        break
```

2번 명령어

break로 반복문 탈출

```
>>> %Run teacher.py
```

1. 선생님이 좋아하는 음식 출력
2. 프로그램 종료

뵘 도와드릴까요?1
치킨...

뵘 도와드릴까요?2
이용해주셔서 감사합니다.

```
>>>
```

같이 해보기

2. 대화 프로그램을 만들어보자.

멍멍이는 개껌, 뼈다귀, 사료 중 하나를 먹으면 먹는 소리를 낸다. 입맛이 까다로워 다른 음식은 먹지 않는다.

```
print("왈왈!")  
print("맛있는거 주세요!")  
print("먹을 것: 개껌, 뼈다귀, 사료")
```

```
print("왈왈!")  
print("먹을 것: 개껌, 뼈다귀, 사료")
```

기능 소개

```
print("왈왈!")  
print("먹을 것: 개껌, 뼈다귀, 사료")  
while True:
```

계속 먹을 수 있어야 함으로 무한반복문

```
print("왈왈!")  
print("먹을 것: 개껌, 뼈다귀, 사료")  
while True:  
    print()  
    a = input("뭘 먹일까?")
```

먹을 음식 이름을 입력받음

```
print("왈왈!")  
print("먹을 것: 개껌, 뼈다귀, 사료")  
while True:  
    print()  
    a = input("뭘 먹일까?")  
    if a == "개껌":  
        print("질경질경~")
```

개껌 먹는 기능 구현

```
print("왈왈!")  
print("먹을 것: 개껌, 뼈다귀, 사료")  
while True:  
    print()  
    a = input("뭘 먹일까?")  
    if a == "개껌":  
        print("질경질경~")  
    elif a == "뼈다귀":  
        print("아작아작...")
```

뼈다귀 먹는 기능 구현


```
print("왈왈!")  
print("먹을 것: 개껌, 뼈다귀, 사료")  
while True:  
    print()  
    a = input("뭘 먹일까?")  
    if a == "개껌":  
        print("질경질경~")  
    elif a == "뼈다귀":  
        print("아작아작...")  
    elif a == "사료":  
        print("챗챗챗!")
```

사료 먹는 기능 구현

```
print("왈왈!")
print("먹을 것: 개껌, 뼈다귀, 사료")
while True:
    print()
    a = input("뭘 먹일까?")
    if a == "개껌":
        print("질경질경~")
    elif a == "뼈다귀":
        print("아작아작...")
    elif a == "사료":
        print("챗챗챗!")
    else:
        print("아무거나 먹지 않아요!")
```

모르는 것은 먹지 않기!

```
>>> %Run hey.py
```

왈왈!

먹을 것: 개껌, 뼈다귀, 사료

뭘 먹일까? **개껌**
질경질경~

뭘 먹일까? **사료**
잡잡잡!

뭘 먹일까?

```
print("왈왈!")  
print("먹을 것: 개껌, 뼈다귀, 사료")  
while True:  
    print()  
    a = input("뭘 먹을까?")  
    if a == "개껌":  
        print("질경질경~")  
    elif a == "뼈다귀":  
        print("아작아작...")  
    elif a == "사료":  
        print("챗챗챗!")  
    else:  
        print("아무거나 먹지 않아요!")
```

이 코드의 문제점이 뭘까?

```
print("왈왈!")  
print("먹을 것: 개껌, 뼈다귀, 사료")  
while True:  
    print()  
    a = input("뭘 먹일까?")  
    if a == "개껌":
```

break가 없어서 무한반복문을 나갈수가 없다!

```
elif a == "뼈다귀":  
    print("아작아작...")  
elif a == "사료":  
    print("챇챇챇!")  
else:  
    print("아무거나 먹지 않아요!")
```

직접 해보기

1. **같이 해보기 2번**의 코드를 수정해서 멍멍이가 그만 먹을 수 있게 하자.

힌트: elif 문을 추가하고 그 블록 안에 break 를 사용하자.

```
>>> %Run hey.py
왈왈!
먹을 것: 개껌, 뼈다귀, 사료

뭘 먹일까? 그만
>>>
```

직접 해보기

2. 용돈기입장 프로그램을 만들어보자.

힌트: 다음 슬라이드의 코드에서 시작하자.

```
>>> %Run hey.py
용돈기입장입니다.
1. 돈 넣기
2. 돈 빼기
3. 잔액 확인
4. 종료
명령을 입력해주세요.
```

```
print("용돈기입장입니다.")
```

```
print("1. 돈 넣기")
```

```
print("2. 돈 빼기")
```

```
print("3. 잔액 확인")
```

```
print("4. 종료")
```

```
잔액 = 0
```

```
while True:
```

```
    a = input("명령을 입력해주세요.")
```

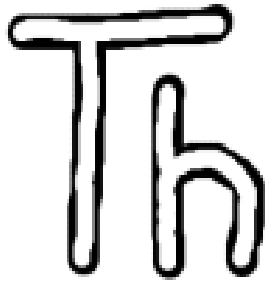
```
    if a == "1":
```

```
        elif a == "2":
```

```
        elif a == "3":
```

```
        elif a == "4":
```

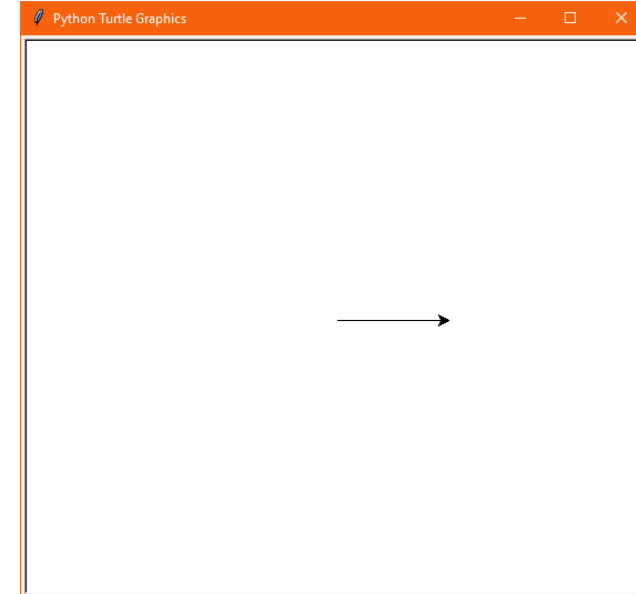

복습



turtle.py

거북이로 그림을 그려주는 모듈

```
import turtle as t  
t.forward(100)
```



복습

▲ t.forward(100)

100만큼 앞으로 가기

▼ t.backward(50)

50만큼 뒤로 가기

◀ t.left(90)

왼쪽으로 90도 돌기

▶ t.right(90)

오른쪽으로 90도 돌기

♡ t.write("안녕!")

"안녕!"이라고 쓰기

○ t.circle(50)

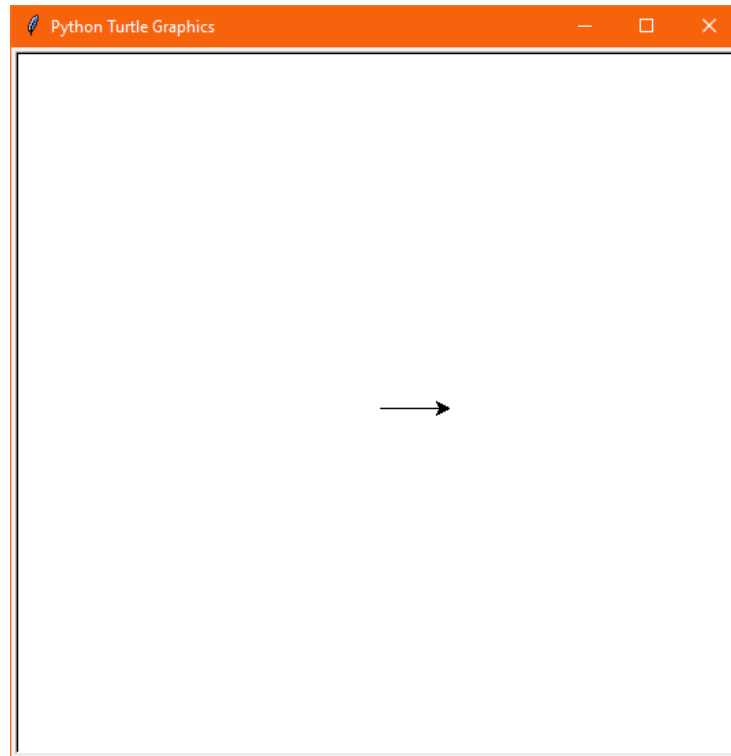
크기 50의 원 그리기

☆ t.reset()

모두 지우고 처음으로 돌아간다.

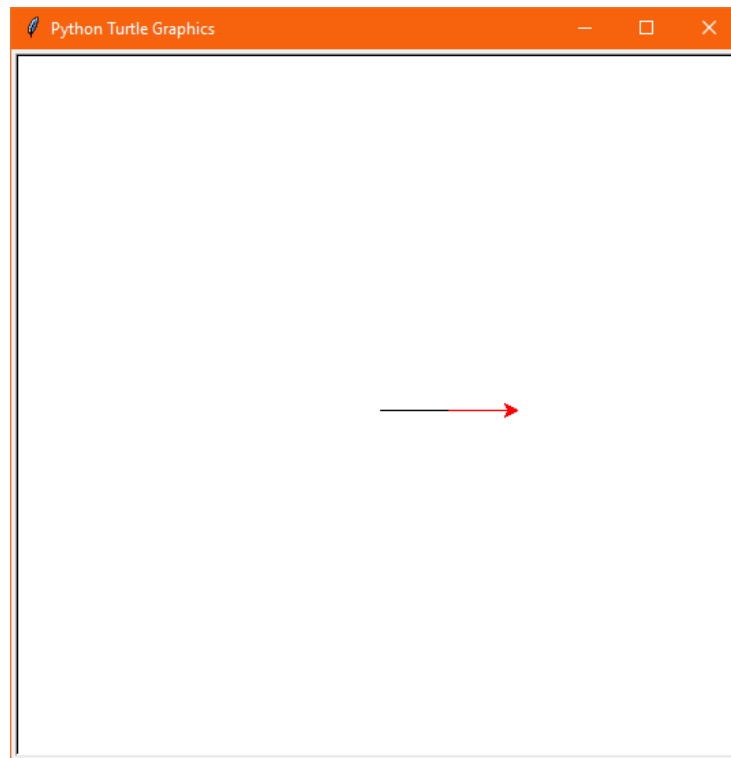
같이 해보기

```
>>> import turtle as t  
>>> t.forward(50)
```



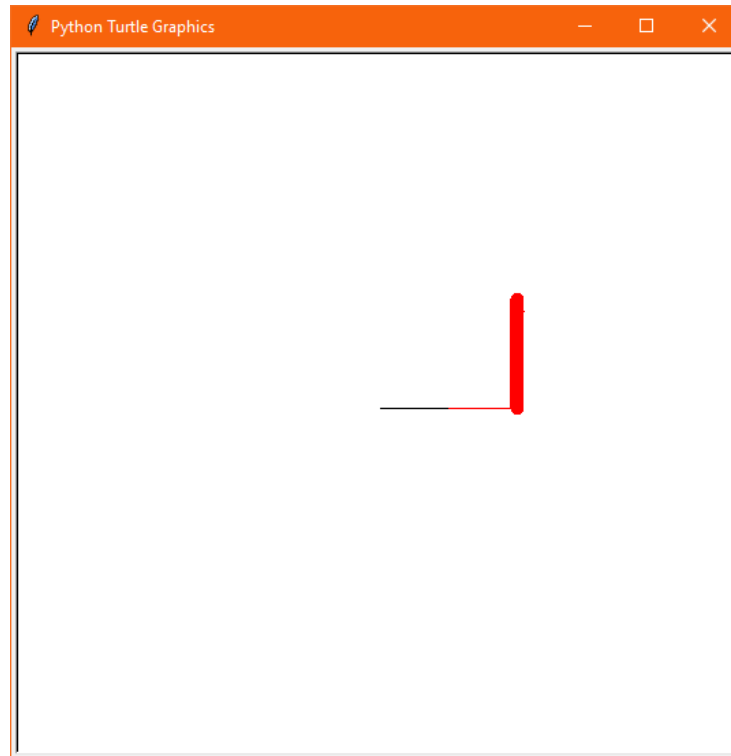
같이 해보기

```
>>> t.color("red")  
>>> t.forward(50)
```



같이 해보기

```
>>> t.left(90)
>>> t.pensize(10)
>>> t.forward(80)
```



직접 해보기

1. 거북이로 다음 도형을 그려보자.

`t.forward(100)`

`t.left(90)`

`t.right(90)`

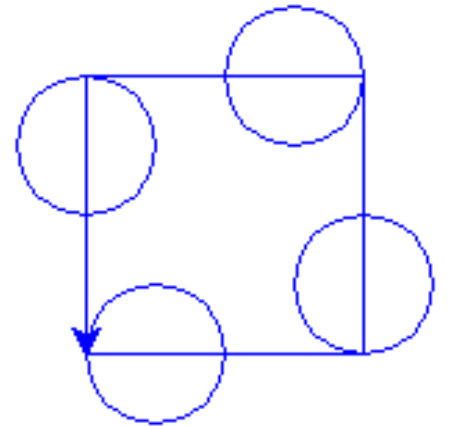
`t.circle(25)`

200만큼 앞으로 가기

왼쪽으로 90도 돌기

오른쪽으로 90도 돌기

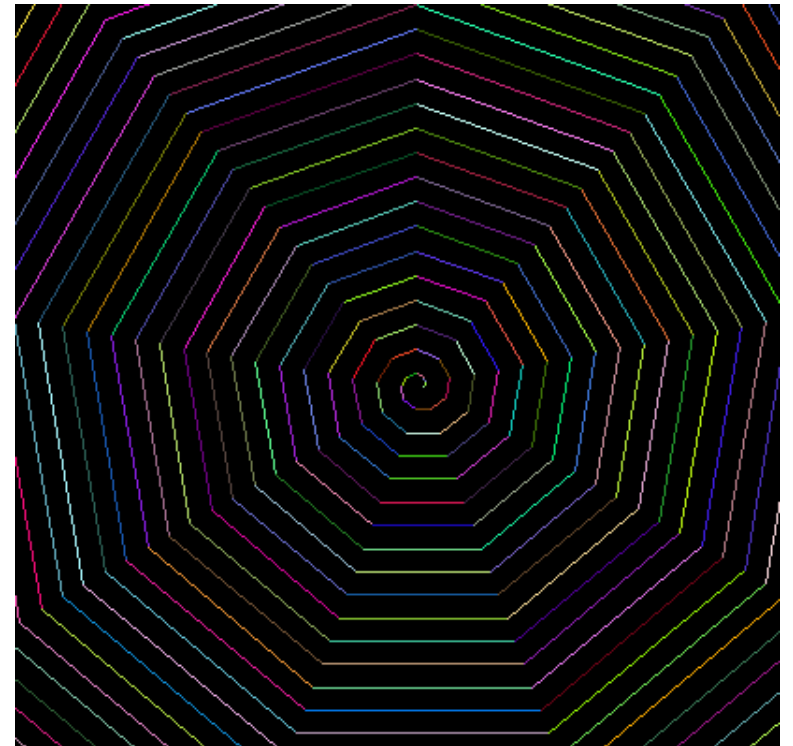
크기 25의 원 그리기



직접 해보기

2. 아래 코드를 자유롭게 수정해 모양을 만들어보자.

```
import turtle as t
import random as r
t.speed(1000)
t.bgcolor("black")
for a in range(0, 1000):
    t.color(r.random(),
            r.random(),
            r.random())
    t.forward(???)
    t.left(???)
```



10, 30, 60, 90, 120, a , $a*2$, $a+50$, $a/5$ 등을 넣어보자.



파이썬과 친해지기

while 반복문, 대화 프로그램, turtle 모듈 활용

9주차

References

왕초보를 위한 Python 2.7
뱀 인형 이미지
Stock Photos

<https://wikidocs.net/145>
<https://bit.ly/2WINL65>
<https://unsplash.com/>