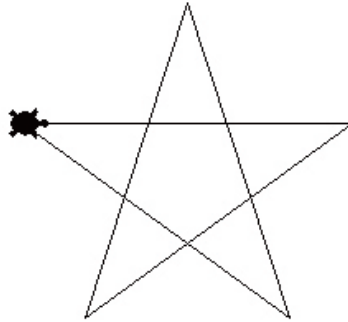




6장 반복해봅시다.

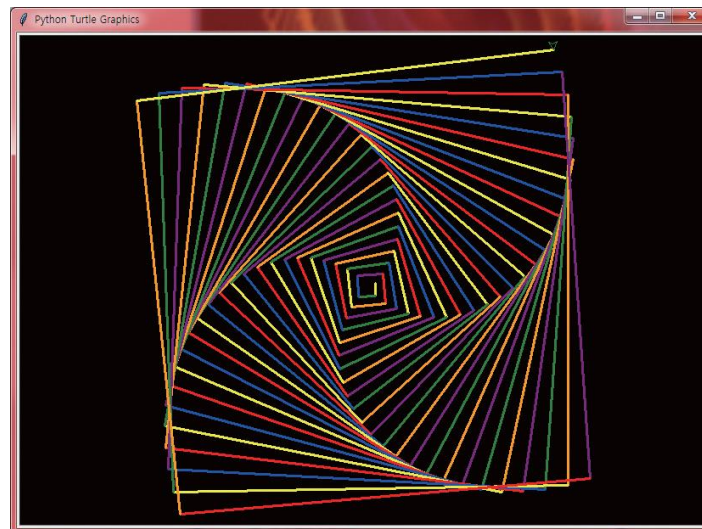
이번 장에서 만들 프로그램

(1) 반복을 이용해서 별을 그려보자



Run Python

(2) 반복을 이용하여 스파이럴을 그려보자.



Run Python

반복이란?

- 반복(iteration)은 동일한 문장을 여러 번 반복시키는 구조
- 컴퓨터는 인간과 다르게 반복적인 작업을 실수 없이 빠르게 할 수 있다. 이것이 컴퓨터의 가장 큰 장점이다.



왜 반복이 중요한가?

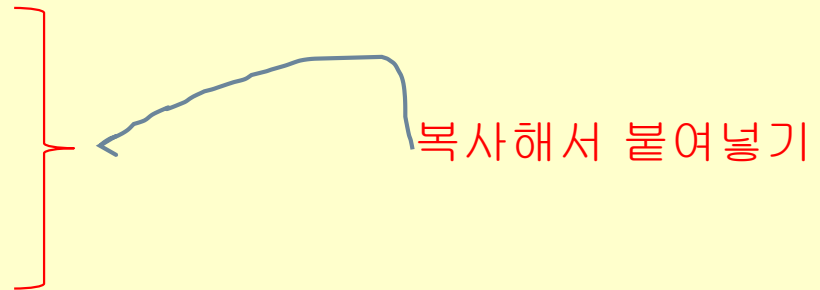
- 하나의 예로 화면에 회사에 중요한 손님이 오셔서 대형 전광판에 '방문을 환영합니다!'를 5번 출력한다고 하자.

방문을 환영합니다!
방문을 환영합니다!
방문을 환영합니다!
방문을 환영합니다!
방문을 환영합니다!



왜 반복이 중요한가?

```
print("방문객을 환영합니다!")  
print("방문객을 환영합니다!")  
print("방문객을 환영합니다!")  
print("방문객을 환영합니다!")  
print("방문객을 환영합니다!")
```



만약 1000번 반복해야 한다면?

```
print("방문을 환영합니다!")  
print("방문을 환영합니다!")  
print("방문을 환영합니다!")  
print("방문을 환영합니다!")  
print("방문을 환영합니다!")  
...  
...  
print("방문을 환영합니다!")
```

1000번 복사해서 붙여넣기 ?



만약 1000번 반복해야 한다면?

- 반복 구조를 사용하여야 한다.

```
for i in range(1000):  
    print("방문을 환영합니다!")
```

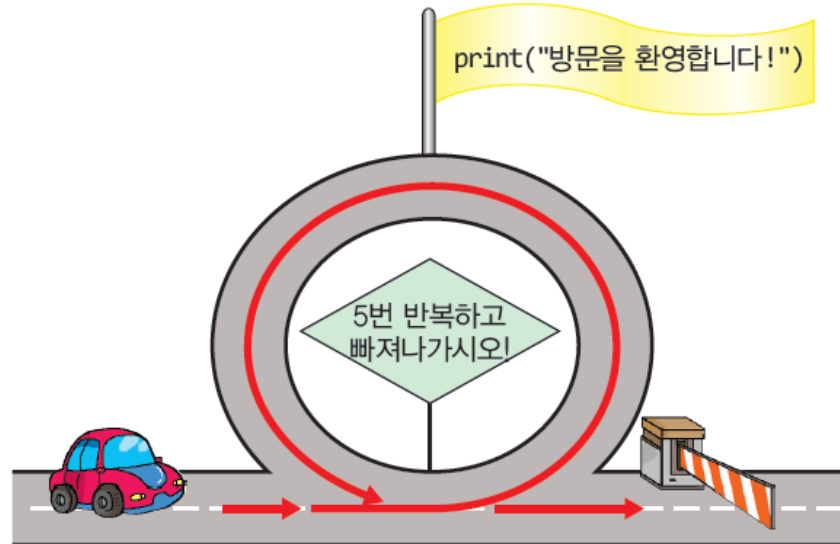
1000번 반복시키는 구조

횟수 제어 반복

- 파이썬에서 횟수 제어 반복은 for 루프라고도 한다.

```
for i in [1, 2, 3, 4, 5]:  
    print("방문을 환영합니다.")
```

끝에 :이 있어야 함
들여쓰기하여야 함



횟수 제어 반복

```
for i in [1, 2, 3, 4, 5]:      # 끝에 :이 있어야 함  
    print("방문을 환영합니다.") # 들여쓰기 하여야 함
```

방문을 환영합니다!
방문을 환영합니다!
방문을 환영합니다!
방문을 환영합니다!
방문을 환영합니다!

i가 1부터 5까지 변경되면서 반복된다.

i의 값을 출력해보자.

```
for i in [1, 2, 3, 4, 5]:  
    print("i=", i)
```

```
i= 1  
i= 2  
i= 3  
i= 4  
i= 5
```

구구단을 출력해보자.

```
for i in [1, 2, 3, 4, 5]:  
    print("9*", i, "=", 9*i)
```

```
9* 1 = 9  
9* 2 = 18  
9* 3 = 27  
9* 4 = 36  
9* 5 = 45
```

range() 함수

for 문

```
for 변수 in range(종료 값) :
```

```
    문장
```

0에서 (종료 값-1)까지의 숫자를 반환한다.

반복되는 문장으로
들여쓰기 하여야 한다.

range() 함수

```
for i in range(5):  
    print("방문을 환영합니다!")
```

```
방문을 환영합니다!  
방문을 환영합니다!  
방문을 환영합니다!  
방문을 환영합니다!  
방문을 환영합니다!
```

range() 함수

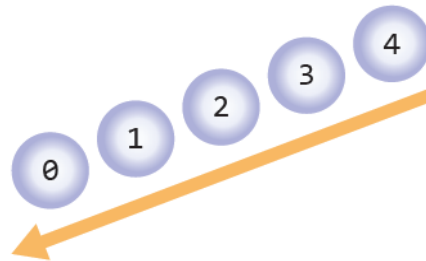
range() 함수

시작값이다.

종료 값이지만 stop은
포함되지 않는다.

한 번에 증가되는 값이다.

`range(start=0, stop, step=1)`



range() 함수의 사용

- 만약 1부터 시작하여서 5까지 반복하고 싶다면 어떻게 하면 될까?

1 2 3 4 5

range() 함수의 사용

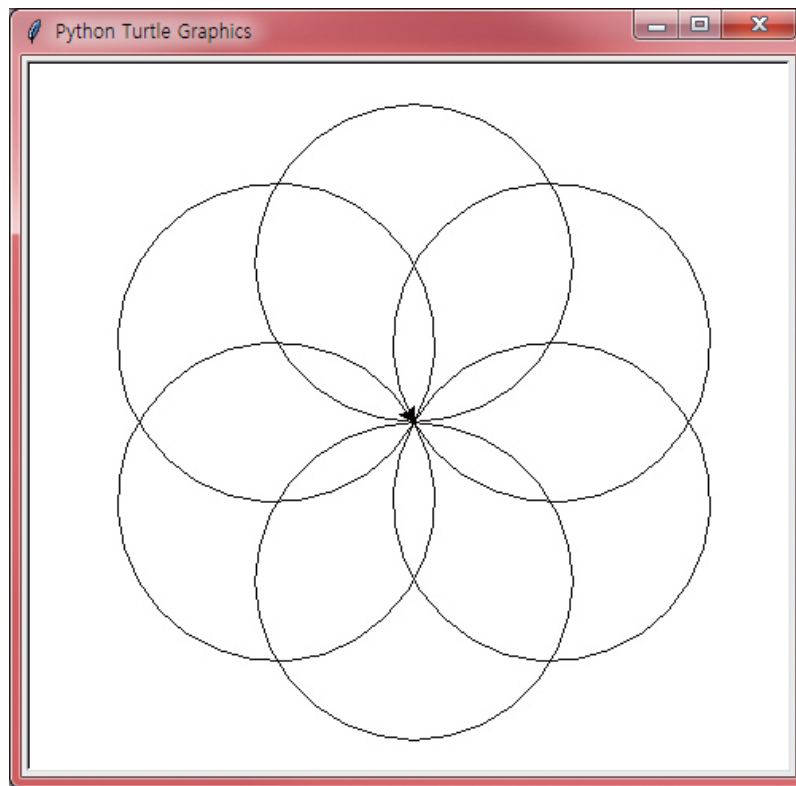
- 만약 1부터 시작하여서 5까지 반복하고 싶다면 어떻게 하면 될까?

1 2 3 4 5

```
for i in range(1, 6, 1):  
    print(i, end=" ")
```


예제

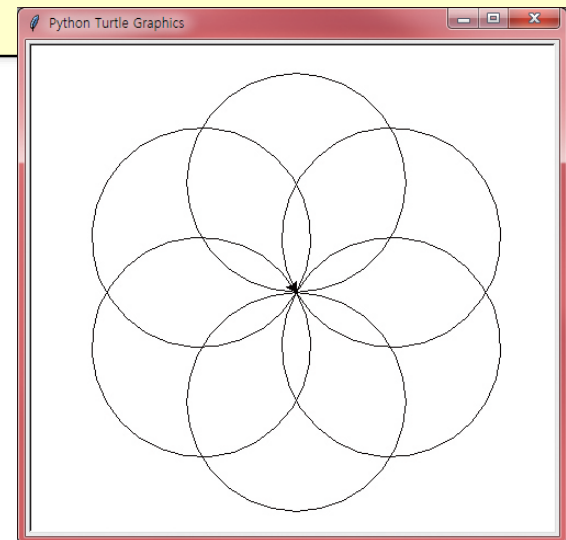
□ 6개의 원 그리기



예제 소스

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

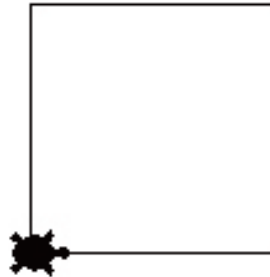
for count in range(6):
    t.circle(100)
    t.left(360/6)
```



Lab: 반복을 사용하여 도형 그리기



- 정삼각형과 정사각형을 반복을 이용하여 화면에 그려 보자.



Run Python

Solution

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

# 정삼각형 그리기
for i in range(3):
    t.forward(100)
    t.left(360/3)

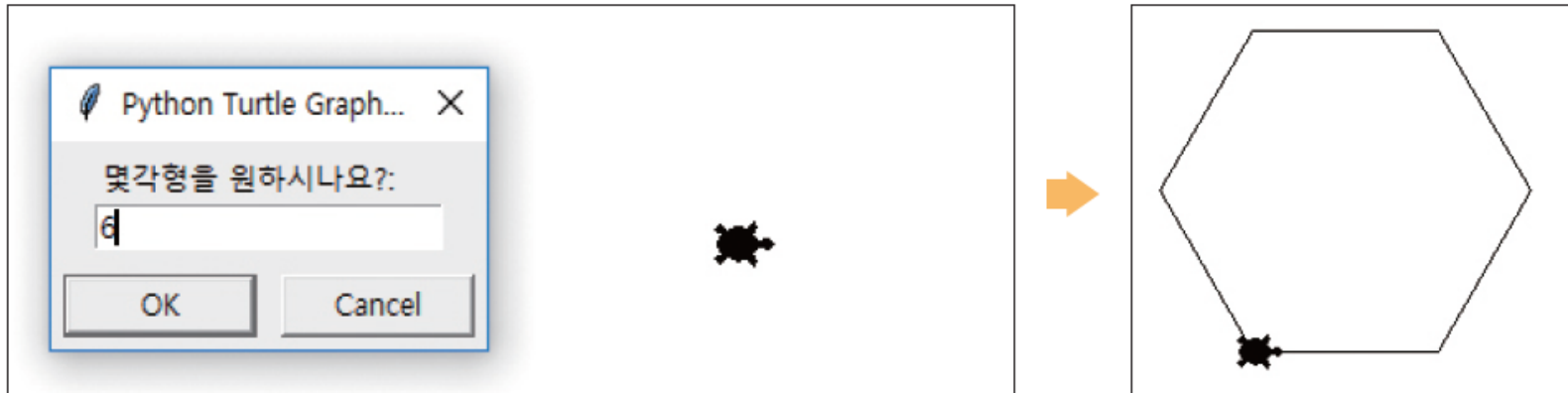
# 이동하기
t.penup()
t.goto(200, 0)
t.pendown()

# 정사각형 그리기
for i in range(4):
    t.forward(100)
    t.left(360/4)
```

Lab: n-각형 그리기



- 사용자로부터 정수 n 을받아서 n -각형을 그리는 프로그램을 작성할 수 있는가?



Solution

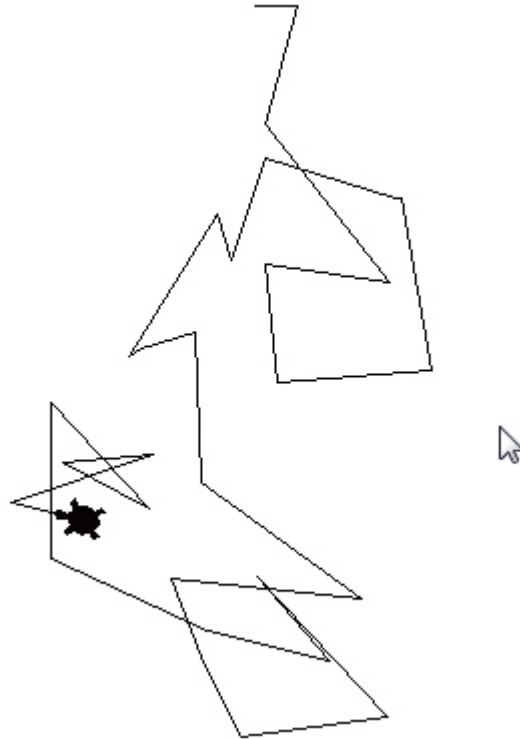
```
import turtle
t = turtle.Turtle()
t.shape("turtle")

s = turtle.textinput("", "몇각형을 원하시나요?:")
n=int(s)

for i in range(n):
    t.forward(100)
    t.left(360/n)
```

Lab: 거북이를 랜덤하게 움직이게 하자

- 터틀 그래픽에서 거북이가 술에 취한 것처럼 랜덤하게 움직이게 해 보자.



알고리즘

30번 반복

- * $[1, 100]$ 사이의 난수를 발생하여 변수 `length`에 저장한다.
- * 거북이를 `length`만큼 움직인다.
- * $[-180, 180]$ 사이의 난수를 발생하여 변수 `angle`에 저장한다.
- * 거북이를 `angle`만큼 회전시킨다.

Solution

```
① import turtle
② import random
③ t = turtle.Turtle()
④ t.shape("turtle")
⑤
⑥ for i in range(30):
⑦     length = random.randint(1, 100)
⑧     t.forward(length)
⑨     angle = random.randint(-180, 180)
⑩     t.right(angle)
```

Lab: 팩토리얼 계산하기



- for문을 이용하여서 팩토리얼을 계산해보자.
- 팩토리얼 $n!$ 은 1부터 n 까지의 정수를 모두 곱한 것을 의미한다. 즉, $n! = 1 \times 2 \times 3 \times \dots \times (n-1) \times n$ 이다.

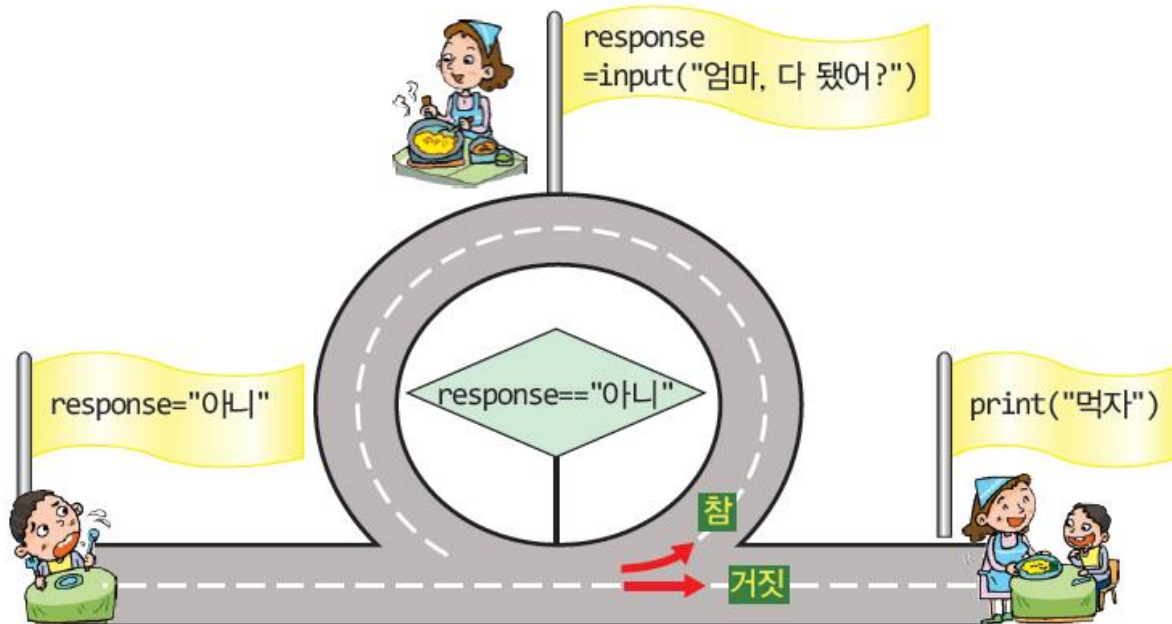
정수를 입력하시오: 10
 $10!$ 은 3628800이다.

Solution

```
n = int(input("정수를 입력하시오: "))  
  
fact = 1  
for i in range(1, n+1):  
    fact = fact * i  
print(n, "!은", fact, "이다.")
```

조건 제어 반복

- 조건 제어 반복은 어떤 조건이 만족되는 동안 반복하는 구조



while 문

while 루프

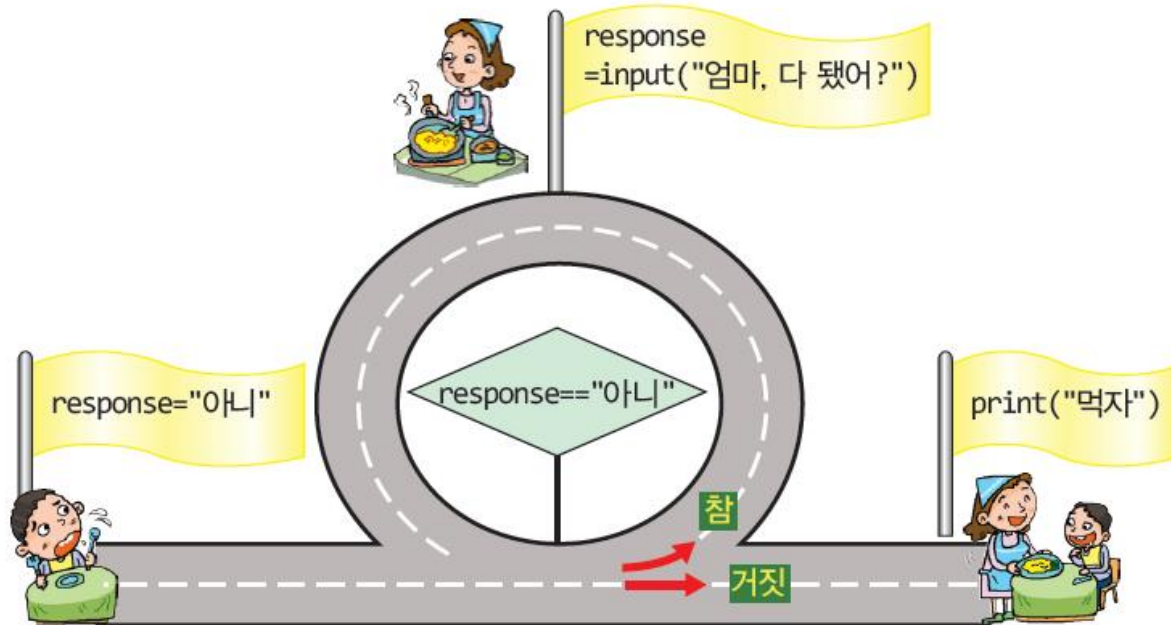
while 조건 :

반복 문장

반복을 하는 조건이다. 조건이 참이면 반복을 계속한다.

반복되는 문장이다.

while 문

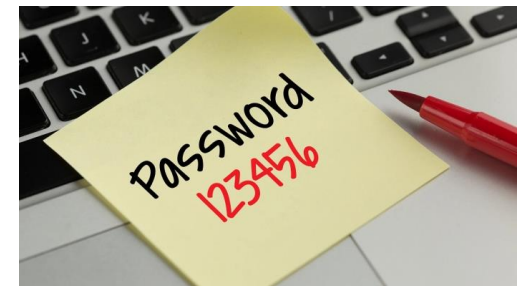


```
response = "아니"
while response == "아니":
    response = input("엄마, 다 됐어?");
print("먹자")
```

예제

- 사용자가 암호를 입력하고 프로그램에서 암호가 맞는지
를 체크한다고 하자.

```
암호를 입력하시오: idontknow  
암호를 입력하시오: 12345678  
암호를 입력하시오: password  
암호를 입력하시오: pythonisfun  
로그인 성공
```



예제

```
password = ""  
while password != "pythonisfun":  
    password = input("암호를 입력하시오: ")  
print("로그인 성공")
```


예제

- 예를 들어서 1부터 10까지의 합을 계산하는 예제를 while 루프로 작성해 보자.

합계는 55



1

1



1+2

= 3



1+2+3

= 6



1+2+3+4

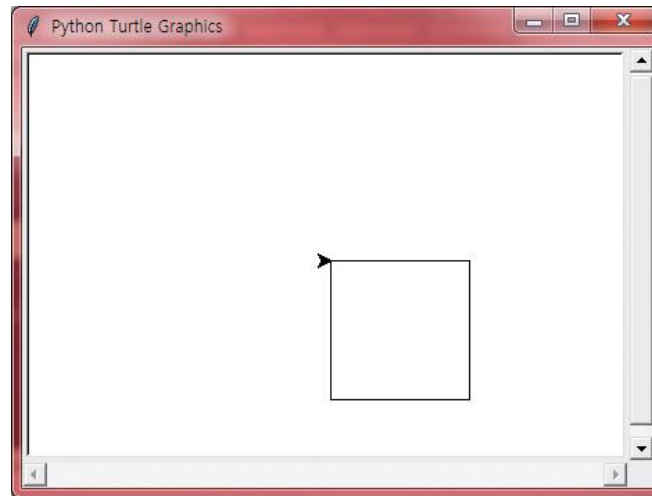
= 10

예제

```
count = 1
sum = 0
while count <= 10 :
    sum = sum + count
    count = count + 1
print("합계는", sum)
```

예제

- while 루프를 이용하여서 화면에 사각형을 그리는 코드를 작성해보자.



예제

```
import turtle
t = turtle.Turtle()
i = 0
while i < 4:
    t.forward(100)
    t.right(90)
    i = i + 1
```

Lab: 구구단 출력



- 구구단 중에서 9단을 반복문을 이용하여 출력해보자. $9*1$, $9*2$, $9*3$, ..., $9*9$ 까지 9번 반복시키면 출력하면 될 것이다.

원하는 단은: 9

$$9*1=9$$

$$9*2=18$$

$$9*3=27$$

$$9*4=36$$

$$9*5=45$$

$$9*6=54$$

$$9*7=63$$

$$9*8=72$$

$$9*9=81$$



Run Python

Solution

```
dan = int(input("원하는 단은: "))
i = 1

while i <= 9:
    print("%s*%s=%s" % (dan, i, dan*i))
    i = i + 1
```



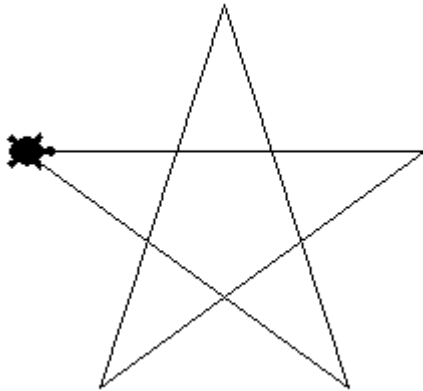
도전문제

구구단의 1단부터 9단까지를 모두 출력하도록 위의 프로그램을 수정해보자.

Lab: 별 그리기



- 반복문을 사용하여 별을 그려보자.
- 5번 반복하고 반복할 마다 거북이를 50픽셀만큼 전진시키고 오른쪽으로 144도 회전하면 별이 그려진다.



Solution

```
import turtle
t = turtle.Turtle()
t.shape("turtle")
i = 0
while i < 5:
    t.forward(200)
    t.right(144)
    i = i + 1
```


이번 장에서 배운 것

- 문장들을 반복 실행하려면 *for*나 *while*을 사용한다.
- 반복 실행되는 문장들을 들여쓰기 하여야 한다.
- *for* 문은 반복 회수를 정해져있을 때 유용하다.
- *while* 문은 반복 조건이 정해져 있을 때 유용하다.
- 반복문의 초입에서 조건식은 검사된다.



Q & A

