



단국대학교
SW중심대학

창의적 사고와 코딩

Lecture 4-3. 중첩반복

- 만약 입력될 데이터의 정확한 개수가 미리 알려지지 않거나 데이터가 너무 많아서 개수를 알기가 어려운 경우에는 어떻게 하는 것이 좋을까? 이런 경우에는 데이터의 끝에다 끝을 알리는 특수한 데이터를 놓으면 된다. ↳ 특한 값 넣기 !!
- 보초값은 일반적인 데이터값에서는 절대 등장할 수 없는 값으로 선택하는 것이 좋다. (끝은 알림)



- ❖ 사용자로부터 임의의 개수의 성적을 받아서 평균을 계산한 후에 출력하는 프로그램을 작성하여 보자. 센티넬로는 음수의 값을 사용하자.

```
종료하려면 음수를 입력하시오
성적을 입력하시오: 90
성적을 입력하시오: 89
성적을 입력하시오: 87
성적을 입력하시오: 100
성적을 입력하시오: 70
성적을 입력하시오: 92
성적을 입력하시오: -1
성적의 평균은 88.000000입니다.
```

```
# while 문을 이용한 성적의 평균 구하기 프로그램  
# 필요한 변수들을 초기화한다.
```

```
n = 0  
total = 0  
score = 0
```

```
print("종료하려면 음수를 입력하시오")
```

```
while score >= 0 :  
    score = int(input("성적을 입력하시오: "))  
    total = total + score  
    n = n + 1
```

```
# 평균을 계산하고 화면에 출력한다.
```

```
if n > 0 :  
    average = total / n  
print("성적의 평균은 %f입니다." % average)
```

```
# while 문을 이용한 성적의 평균 구하기 프로그램  
# 필요한 변수들을 초기화한다.
```

```
SENTINEL = -1 #보초값  
n = 0  
total = 0  
score = 0
```

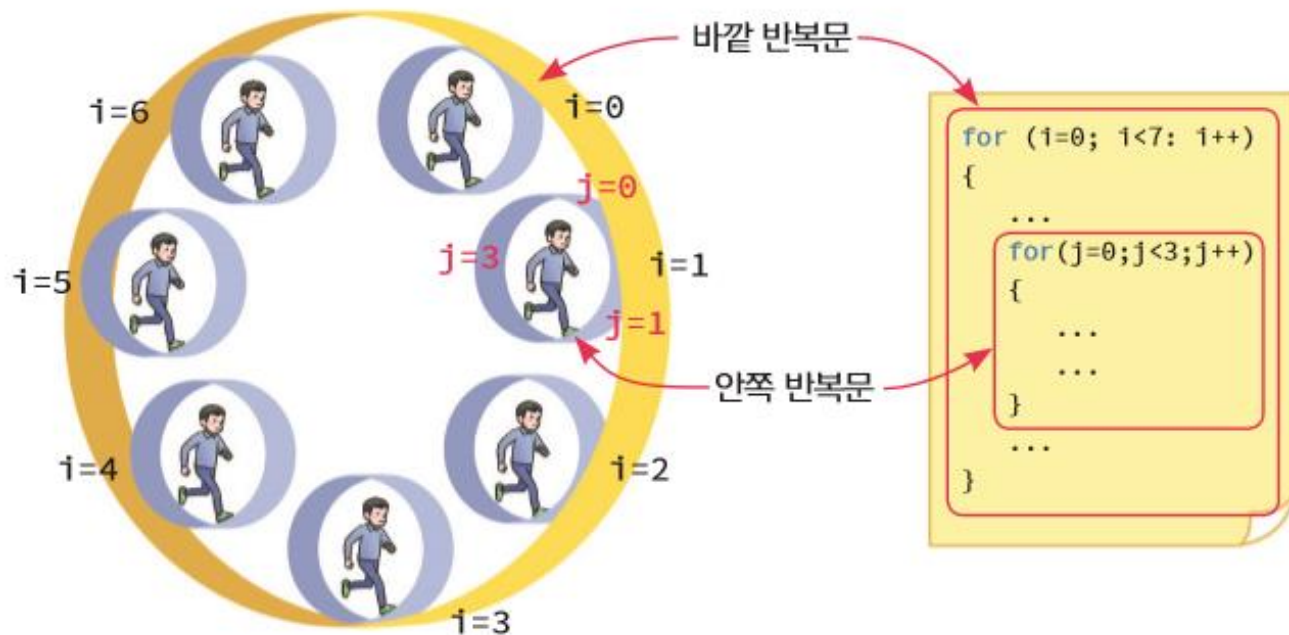
```
print("종료하려면 음수를 입력하시오")
```

```
while score != SENTINEL :  
    score = int(input("성적을 입력하시오: "))  
    if score >= 0:  
        total = total + score  
        n = n + 1
```

```
# 평균을 계산하고 화면에 출력한다.
```

```
if n > 0 :  
    average = total / n  
print("성적의 평균은 %.2f입니다." % average)
```

- 반복문은 중첩하여 사용될 수 있다. 즉 반복문 안에 다른 반복문이 포함될 수 있다.



- ❖ '*'를 한주에 10개씩 5줄로 나타내는 프로그램을 작성해보자.

```
*****  
*****  
*****  
*****  
*****
```

중첩 for 문을 이용하여 *기호를 사각형 모양으로 출력하는 프로그램

```
for y in range(5): #외부 반복문(outer loop)  
    for x in range(10): #내부 반복문(inner loop)  
        print("*", end="")  
    print("")      # 내부 반복문이 종료될 때마다 실행
```

❖ 다음과 같이 출력되는 프로그램을 작성해보자

```
*  
**  
***  
****  
*****  
*****  
*****  
*****  
*****  
*****  
*****
```

Solution: 파일명 print_star1.py

```
for line in range(10):  
    for star in range(line+1):  
        print('*', end='')  
    print()                # 줄 바꿈
```

```
for line in range(10):  
    print('*' * (line+1))
```

while문 사용시

```
line = 1  
while line <= 10:  
    star = 1  
    while star <= line:  
        print('*', end='')  
    star += 1  
    print()  
    line += 1
```

line 변수의 값	star 변수의 값	출력
0	0	*
1	0, 1	**
2	0, 1, 2	***
3	0, 1, 2, 3	****
4	0, 1, 2, 3, 4	*****
5	0, 1, 2, 3, 4, 5	*****
6	0, 1, 2, 3, 4, 5, 6	*****
7	0, 1, 2, 3, 4, 5, 6, 7	*****
8	0, 1, 2, 3, 4, 5, 6, 7, 8	*****
9	0, 1, 2, 3, 4, 5, 6, 7, 8, 9	*****

❖ 다음과 같이 출력되는 프로그램을 작성해보자

```
*****  
*****  
*****  
*****  
*****  
*****  
*****  
*****  
*****  
*****
```

Solution: 파일명 print_star2.py



❖ 다음과 같이 출력되는 프로그램을 작성해보자

```
      *  
     **  
    ***  
   ****  
  *****  
 *****  
*****  
*****  
*****  
*****
```

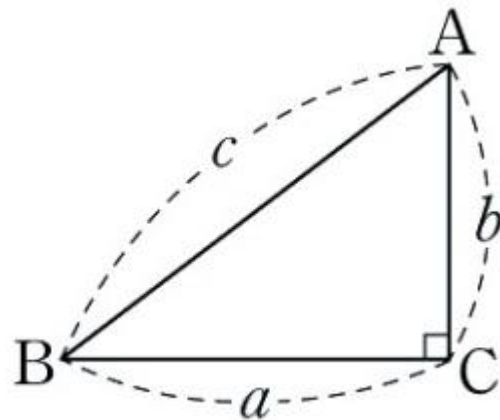
Solution: print_star3.py



- ❖ 피타고라스의 정리는 직각 삼각형에서 직각을 낀 두 변의 길이를 a , b 라고 하고, 빗변의 길이를 c 라고 하면 $a^2 + b^2 = c^2$ 의 수식이 성립한다는 것이다. 각 변의 길이가 100보다 작은 삼각형 중에서 피타고라스의 정리가 성립하는 직각 삼각형은 몇 개나 있을까?

3 4 5
4 3 5
5 12 13
...

$$a^2 + b^2 = c^2$$



```
for a in range(1, 101, 1):  
    for b in range(1, 101, 1):  
        for c in range(1, 101, 1):  
            if( a*a+b*b==c*c ):  
                print(a, b, c)
```

- ✧ 정수를 1부터 10까지 1씩 증가시키면서 n-제곱값을 테이블 형식으로 출력하는 프로그램을 작성하여 보자.

x^{**1}	x^{**2}	x^{**3}	x^{**4}	x^{**5}
1	1	1	1	1
2	4	8	16	32
3	9	27	81	243
4	16	64	256	1024
5	25	125	625	3125
6	36	216	1296	7776
7	49	343	2401	16807
8	64	512	4096	32768
9	81	729	6561	59049
10	100	1000	10000	100000

Solution: 파일명 power.py

```
for y in range(1, 6):  
    print("x**%d" % y , end="Wt")  
print()  
  
for x in range(1, 11):  
    for y in range(1, 6):  
        print(x**y, end="Wt")  
    print()
```




단국대학교
SW중심대학

창의적 사고와 코딩

반복문 예제

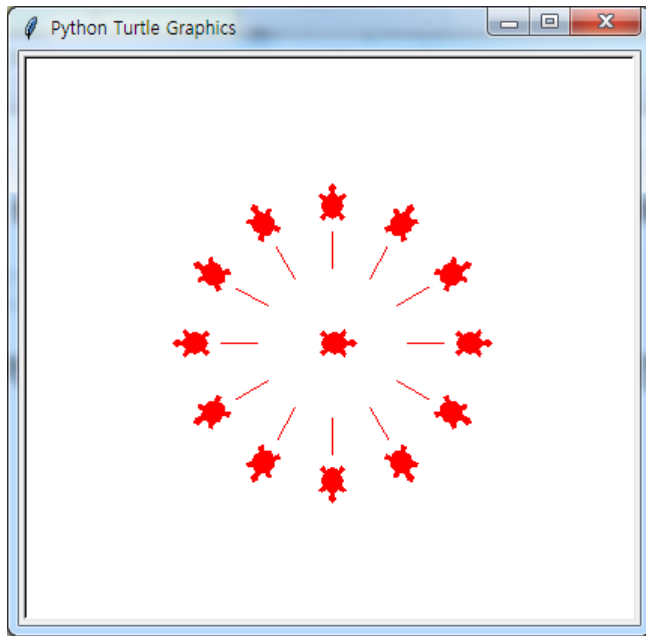
- ✧ 1부터 n까지의 숫자 중 홀수와 짝수를 구분하여 각각의 합을 구한다.

```
n의 값은?100  
1부터 100 까지의 짝수합: 2550 홀수합: 2500  
>>>
```

Solution: 파일명



❖ stamp()함수 이용



Solution: 파일명 turtle_timer_draw.py



- ① 총 5회에 걸쳐 사칙연산의 정답을 알아 맞춘다.
- ② 5회 중 중간에 정답을 맞추면 프로그램은 '정답입니다'라는 문자열을 출력하고 종료된다.
- ③ 정답을 맞추지 못한 경우 '정답이 아닙니다'라는 문자열을 출력하고 다음 문제로 넘어간다. 5회 모두 틀린 경우 '사칙연산 공부를 하세요.'라는 문자열을 출력한다.
- ④ 사칙연산의 숫자는 100이하의 숫자 2개를 난수를 이용하고 연산자도 임의의 연산자를 적용한다.

1 번째 $27 + 64 =$
정답을 입력하시오 20
정답이 아닙니다
2 번째 $44 + 12 =$
정답을 입력하시오 6
정답이 아닙니다
3 번째 $37 + 25 =$
정답을 입력하시오 10
정답이 아닙니다
4 번째 $74 / 31 =$
정답을 입력하시오 7
정답이 아닙니다
5 번째 $74 * 33 =$
정답을 입력하시오 10
정답이 아닙니다
사칙연산 공부를 하세요

1 번째 $39 + 52 =$
정답을 입력하시오 91
정답입니다

1 번째 $59 * 30 =$
정답을 입력하시오 11
정답이 아닙니다
2 번째 $26 - 64 =$
정답을 입력하시오 -38
정답입니다

Solution: four_arithmetic_perations.py



2 4 6 8 10

12 14 16 18 20

22 24 26 28 30

32 34 36 38 40

42 44 46 48 50

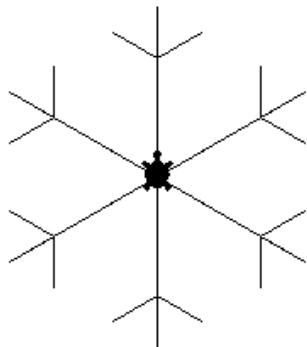
52 54 56 58 60

62 64 66 68 70

72 74 76 78 80

82 84 86 88 90

92 94 96 98 100



1. 거북이를 왼쪽으로 90도 회전
2. 6번 반복
 1. 앞으로 100이동
 2. 앞으로 -30이동
 3. 왼쪽으로 60도 회전
 4. 앞으로 30이동
 5. 앞으로 -30이동
 6. 오른쪽으로 120도 회전
 7. 앞으로 30이동
 8. 앞으로 -30이동
 9. 왼쪽으로 60도 회전
 10. 앞으로 -70 이동
 11. 왼쪽으로 60도 회전

```
import turtle
```

```
t = turtle.Turtle()  
t.shape("turtle")  
t.left(90)
```

```
for i in range (1,7):  
    t.forward(100)  
    t.forward(-30)  
    t.left(60)  
    t.forward(30)  
    t.forward(-30)
```

```
t.right(120)  
t.forward(30)  
t.forward(-30)
```

```
t.left(60)  
t.forward(-70)  
t.left(60)
```

❖ 사각형의 개수를 입력받아 다음과 같은 프로그램을 작성하시오.

- 사각형 한번의 길이는 [50, 200]에서 난수값을 생성하여 그린다.

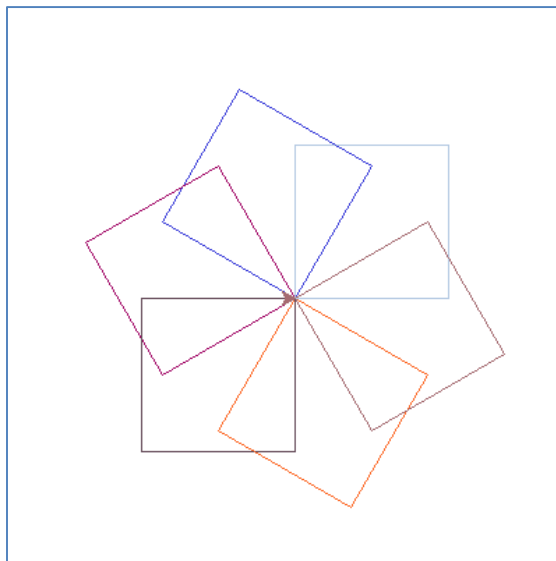
- 사각형을그릴때마다

그리는 선의 색은 0과1사이의 실수값에서 난수값을 생성하여 RGB값을 지정하여 원을 그릴때 마다 다른색으로 그리도록 설정한다.

```
t.color(random.random(), random.random(), random.random())
```

0~1 실수값

사각형의 개수를 입력: 6



Solution: 파일명 turtle_rect.py

