

기초 PYTHON 프로그래밍

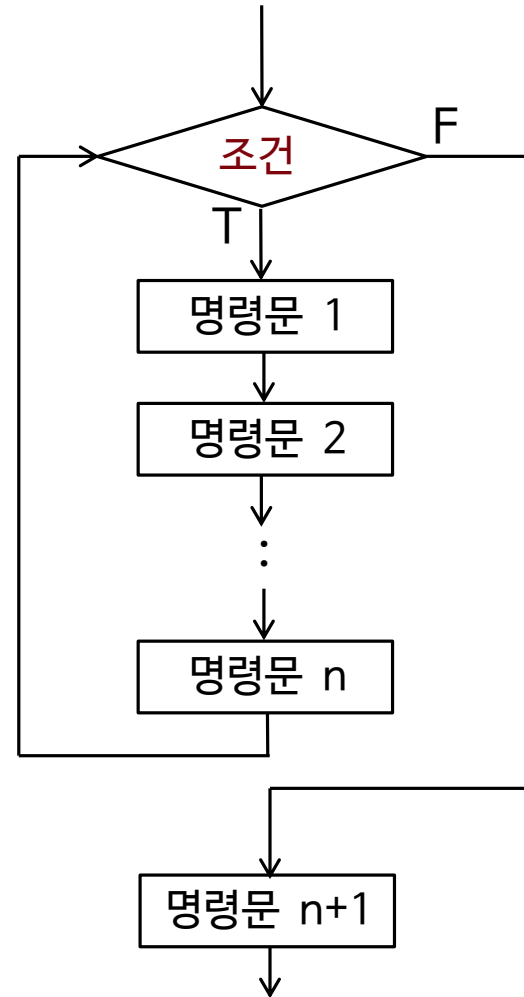
7. 반복문 - while

1. 반복 논리
2. 파이썬의 두 가지 반복문
3. while 반복문
4. 중첩된 while 반복문

1. 반복 논리

◆ 반복 논리

- 특정 명령 또는 연산을 반복적으로 수행해야 하는 경우
- 반복을 제어할 조건문이 있어야 한다.



2. 파이썬의 두 가지 반복문

◆ while 반복문

while 조건식 :

명령어 1

명령어 2

... ..

명령어 n

else:

◆ for 반복문

for x **in** 데이터 :

명령어 1

명령어 2

... ..

명령어 n

else:

3. while 반복문

- ◆ 예제 1 : 1부터 5까지 출력하기

```
a = 1
while a <= 5 :
    print(a)
    a += 1
```

```
1
2
3
4
5
```

3. while 반복문

◆ 예제 2 : 1부터 10까지의 합

$$1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9 + 10 = ?$$

```
a = 1
```

```
sum = 0
```

```
while a <= 10 :
```

```
    sum += a
```

```
    a += 1
```

```
print('sum : ', sum)
```

```
sum(range(11))
```

a	sum
1	0
2	1
3	3
4	6
5	10
6	15
7	21
8	28
9	36
10	45
11	55

3. while 반복문

◆ 예제 3 : 1부터 10까지 짝수의 합 구하기

방법 1 : 2부터 시작하여 2씩 증가시켜 나가면서 합을 구한다.

방법 2 : 1부터 시작하여 1씩 증가시켜 나가면서 짝수인지 판단하여 짝수이면 합을 구한다.

```
a = 2
```

```
sum = 0
```

```
while a <= 10 :
```

```
    sum += a
```

```
    a += 2
```

```
print('sum :', sum)
```

```
a = 1
```

```
sum = 0
```

```
while a <= 10 :
```

```
    if a % 2 == 0 :
```

```
        sum += a
```

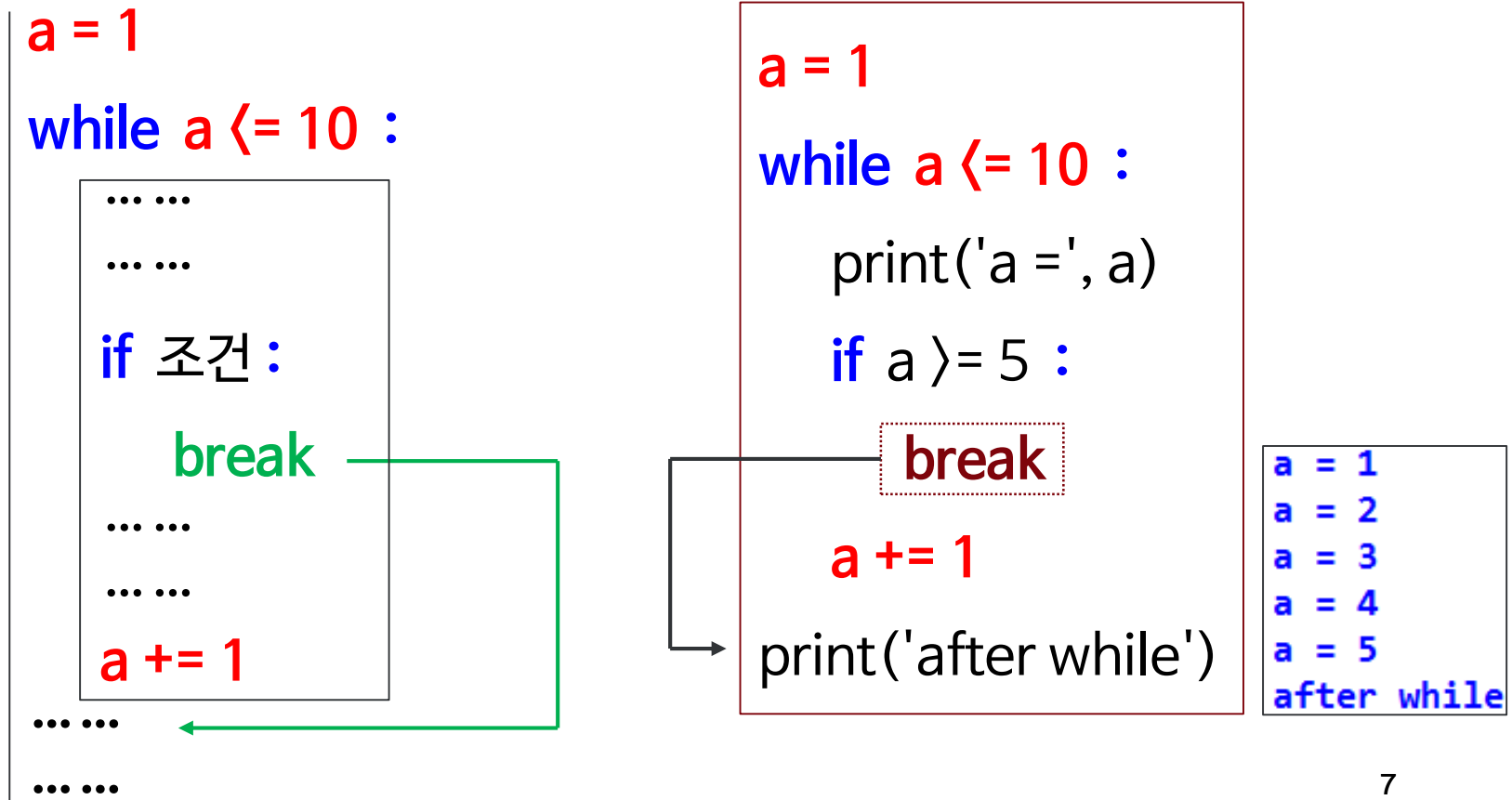
```
    a += 1
```

```
print('sum :', sum)
```

4. while 반복문 제어하기

◆ while 구문에 break 사용하기

- break 는 반복문 안에서 사용하면 반복문을 끝내도록 한다.



4. while 반복문 제어하기

◆ 무한루프와 break

```
a = 1
```

```
while True:
```

```
    print(a)
```

```
    if a == 5 : break
```

```
    a += 1
```

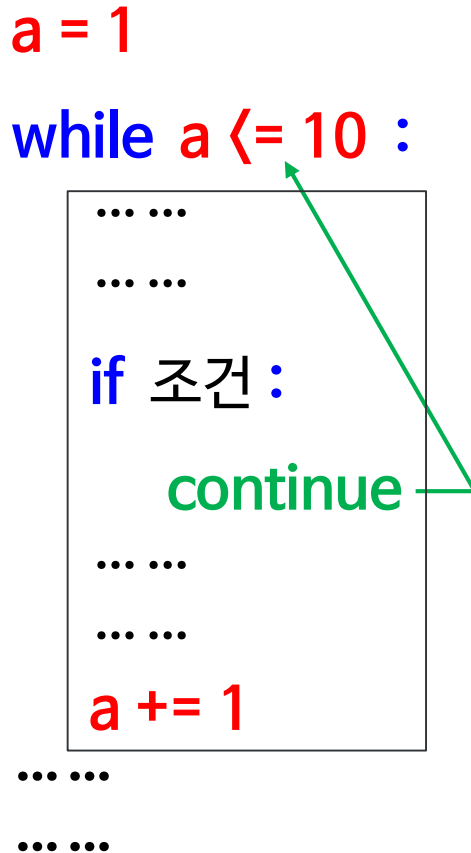
1
2
3
4
5

4. while 반복문 제어하기

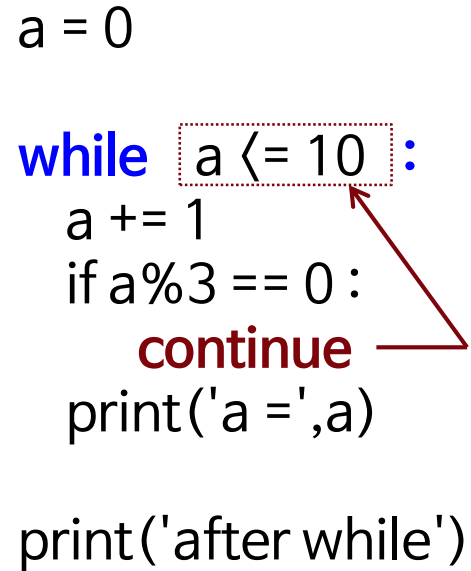
◆ while 반복문에 continue 사용하기

- continue 는 반복문의 조건으로 제어가 가도록 한다.

```
a = 1
while a <= 10 :
    ... ..
    ... ..
    if 조건 :
        continue
    ... ..
    ... ..
    a += 1
... ..
... ..
```



```
a = 0
while a <= 10 :
    a += 1
    if a%3 == 0 :
        continue
    print('a =',a)
print('after while')
```



```
a = 1
a = 2
a = 4
a = 5
a = 7
a = 8
a = 10
a = 11
after while
```

continue 조건이 참인 경우 continue
아래 부분은 실행하지 않는다.

5. else 갖는 while 반복문

◆ while 반복문에 else 구문 사용하기

- while 구문의 조건이 False가 되어 반복문이 끝날 때, else 구문이 있으면 else 구문을 수행한다.

```
a = 1
while a <= 3:
    print(a)
    print('-----')
    a += 1
else:
    print('I am else')
print('while finished')
```

```
1
-----
2
-----
3
-----
I am else
while finished
```

5. else 갖는 while 반복문

◆ while 반복문에 else 구문 사용하기

```
a = 1
while a <= 3:
    print(a)
    if a == 5: break
    print('-----')
    a += 1
else:
    print('I am else')

print('while finished')
```

```
1
-----
2
-----
3
-----
I am else
while finished
```

```
a = 1
while a <= 10:
    print(a)
    if a == 5: break
    print('-----')
    a += 1
else:
    print('I am else')

print('while finished')
```

```
1
-----
2
-----
3
-----
4
-----
5
while finished
```

6. 중첩된 while 반복문

- ◆ 반복문 안에 다시 반복이 필요한 경우에는 중첩된 반복문을 사용한다.
- ◆ 예제 4 : 구구단 출력하기 (2단 ~ 9단)

2	4	6	8	10	12	14	16	18	20
3	6	9	12	15	18	21	24	27	30
4	8	12	16	20	24	28	32	36	40
5	10	15	20	25	30	35	40	45	50
6	12	18	24	30	36	42	48	54	60
7	14	21	28	35	42	49	56	63	70
8	16	24	32	40	48	56	64	72	80
9	18	27	36	45	54	63	72	81	90

6. 중첩된 while 반복문

dan = 2

while dan <= 9 :

n = 1

while n <= 10 :

value = dan * n

print('%3d' % (value), end='')

n += 1

print()

dan += 1

2	4	6	8	10	12	14	16	18	20
3	6	9	12	15	18	21	24	27	30
4	8	12	16	20	24	28	32	36	40
5	10	15	20	25	30	35	40	45	50
6	12	18	24	30	36	42	48	54	60
7	14	21	28	35	42	49	56	63	70
8	16	24	32	40	48	56	64	72	80
9	18	27	36	45	54	63	72	81	90