

08

Loop

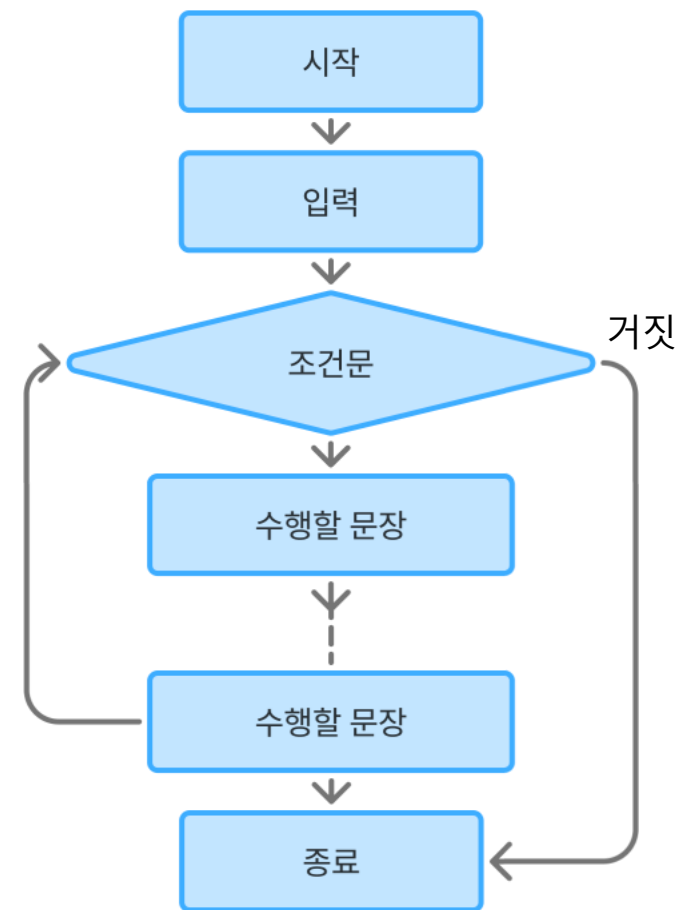
Loop

- 반복문은 같은 코드를 여러 번 실행할 수 있게 해주는 구조
- for, while 두 가지가 있음

while 조건:
실행할 코드

while문

- 주어진 조건이 참일 동안 계속 반복
- 조건이 거짓이 되면 반복 종료



Loop

- 조건이 True인 동안 들여쓰기 된 부분 반복
- 조건을 바꾸는 코드 필요

```
>>> i = 1
>>> while i <= 5:
...     print(i)
...     i += 1
...
1
2
3
4
5
```

Loop

- 조건이 항상 True이면 종료되지 않음
- break로 종료 필요

```
>>> while True:  
...     print("Repeating")  
...  
Repeating  
Repeating  
Repeating  
Repeating  
Repeating  
Repeating
```

Loop

- 조건이 항상 True이면 종료되지 않음
- $i += 1$ 없음 \rightarrow 무한 루프

```
>>> i = 1
>>> while i < 5:
...     print(i)
...
1
1
1
1
1
```

Loop

- 콘솔에서는 Ctrl + C로 중단

```
Traceback (most recent call last):  
  File "<stdin>", line 1, in <module>  
KeyboardInterrupt
```

- 쥬피터 노트북에서는 커널이 과부하될 수 있음
- Interrupt the kernel로 중단 가능



Loop

- input()을 이용해 반복문에서 나올 수 있음
- 문자열이 일치할 경우 break

```
while True:
    text = input("끝내려면 q 입력: ")
    if text == 'q':
        break
```

```
끝내려면 q 입력: Hello
끝내려면 q 입력: Python
끝내려면 q 입력: q
```

Loop

- 반복문을 활용한 예제

```
prompt = ""  
1. Add  
2. Del  
3. List  
4. Quit  
  
Enter number: ""
```

```
number = 0  
while number != 4:  
    print(prompt)  
    number = int(input())
```

```
1. Add  
2. Del  
3. List  
4. Quit
```

```
Enter number:  
3
```

```
1. Add  
2. Del  
3. List  
4. Quit
```

```
Enter number:  
4
```


Loop

- while문을 사용해 0부터 4까지의 수 중 짝수만 출력
- i는 한 번 반복할 때마다 1씩 증가

```
i = 0

while i < 5:
    if i % 2 == 0:
        print(i, "짝수")
    i += 1
```

```
0 짝수
2 짝수
4 짝수
```

Loop

```
i = 0

while i < 5:
    if i % 2 == 0:
        print(i, "짝수")
    i += 1
```

0 짝수
2 짝수
4 짝수

i	조건문	조건 판단	조건문	조건 판단	수행하는 문장	while문
0	$0 < 5$	참	$0 \% 2 == 0$	참	0 짝수	반복
1	$1 < 5$	참	$1 \% 2 == 0$	거짓		반복
2	$2 < 5$	참	$2 \% 2 == 0$	참	2 짝수	반복
3	$3 < 5$	참	$3 \% 2 == 0$	거짓		반복
4	$4 < 5$	참	$4 \% 2 == 0$	참	4 짝수	반복
5	$5 < 5$	거짓				종료

Loop

- while문을 사용해 리스트 nums의 모든 요소를 순서대로 출력
- 인덱스 i를 이용해 리스트에 접근
- i가 리스트 길이보다 작을 때 반복

```
nums = [10, 20, 30]
i = 0

while i < len(nums):
    print(nums[i])
    i += 1
```

```
10
20
30
```

Loop

```
i = 1
total = 0

while i <= 5:
    total += i
    i += 1
print(total)
```

15

```
i = 1
total = 0

while i <= 5:
    total += i
    i += 1
    print(total)
```

1
3
6
10
15

break

- break는 반복문 내에서 조건을 만족할 때 즉시 반복을 종료하는 데에 사용
- 보통 while True와 함께 사용하여 원하는 조건에서만 빠져나오는 구조로 활용

```
while True:
    n = int(input("음수 입력 시 종료: "))
    if n < 0:
        break
```

```
음수 입력 시 종료: 2
음수 입력 시 종료: 5
음수 입력 시 종료: 0
음수 입력 시 종료: -1
```

continue

- 반복문 안에서 continue가 있을 경우 그 아래 코드를 건너뛰고 다음 반복으로 이동
- 예제에서는 i가 3일 때 print(i)를 건너뛰므로 3은 출력되지 않음

```
i = 0

while i < 5:
    i += 1
    if i == 3:
        continue
    print(i)
```

```
1
2
4
5
```

pass

- pass는 코드 블록이 필요하지만 실제로는 아무 작업도 하지 않을 때 사용
- 주로 아직 작성하지 않은 부분에 넣음

```
while True:
    user_input = input("입력: ")
    if user_input == "":
        pass # 아무것도 입력하지 않으면 그냥 넘어감
    else:
        print("입력됨:", user_input)
```

```
입력:
입력:
입력:
입력: Hello
입력됨: Hello
```

Loop

- while문이 정상적으로 종료되면 else 블록 실행
- break로 빠져나가면 else 블록은 실행되지 않음

```
i = 0

while i < 3:
    print(i)
    i += 1
else:
    print("반복 완료")
```

```
0
1
2
반복 완료
```


Loop

- while문이 정상적으로 종료되면 else 블록 실행
- break로 빠져나가면 else 블록은 실행되지 않음

```
i = 0

while i < 3:
    print(i)
    if i == 2:
        break
    i += 1
else:
    print("반복 완료")
```

0
1
2

Loop

```
coffee = 10
while True:
    money = int(input("돈을 넣어주세요: "))
    if money == 300:
        print("커피를 줍니다.")
        coffee = coffee - 1
    elif money > 300:
        print("거스름돈 %d를 주고 커피를 줍니다." % (money-300))
        coffee -= 1
    else:
        # money < 300
        print("돈을 다시 돌려주고 커피를 주지 않습니다.")
        print("남은 커피의 양은 %d개입니다." % coffee)
    if not coffee:
        print("커피가 다 떨어졌습니다. 판매를 중지합니다.")
        break
```

돈을 넣어주세요: 300

커피를 줍니다.

돈을 넣어주세요: 350

거스름돈 50를 주고 커피를 줍니다.

돈을 넣어주세요: 270

돈을 다시 돌려주고 커피를 주지 않습니다.

남은 커피의 양은 8개입니다.

⋮

돈을 넣어주세요: 300

커피를 줍니다.

커피가 다 떨어졌습니다. 판매를 중지합니다.

Loop

```
coffee = 10
while True:
    money = int(input("돈을 넣어주세요: "))
    if money == 300:
        print("커피를 줍니다.")
        coffee = coffee - 1
    elif money > 300:
        print("거스름돈 %d를 주고 커피를 줍니다." % (money-300))
        coffee -= 1
    else:
        print("돈을 다시 돌려주고 커피를 주지 않습니다.")
        print("남은 커피의 양은 %d개입니다." % coffee)
    if not coffee:
        print("커피가 다 떨어졌습니다. 판매를 중지합니다.")
        break
```

커피 개수 정의
무한 반복 → break 필요

넣은 돈 == 커피 값

남은 개수 -1
넣은 돈 > 커피 값
거스름돈 계산

남은 개수 -1
넣은 돈 < 커피 값
개수 변화 x

커피 매진 시

반복문 break

8주차 과제

- 08_While_Loop.ipynb