



한국IT진흥부설

정보보호교육학원 아이섹

파이썬 기초

반복문, 탈출문

By SoonGu Hong



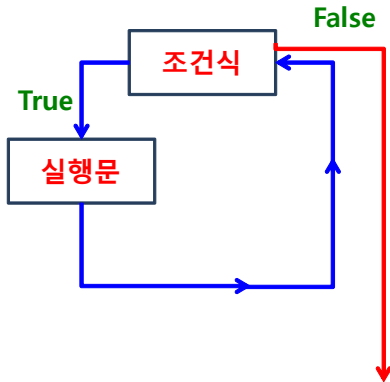
한국IT진흥부설

정보보호교육학원 아이섹

파이썬 기초

1. 반복문 while

1-1-1. 반복문 while



- while문은 조건식을 검사하여 참일 경우 블록 내부의 코드를 실행하며 실행이 끝날 때마다 반복적으로 조건식을 검사하여 False가 나올 때까지 반복합니다.

- while문도 if와 마찬가지로 논리값이 도출되는 조건식이나 함수를 사용합니다.

- while문은 반복 횟수를 제어하기 위한 증감식이나 탈출문이 필요합니다.

1-1-2. while문 사용법

```
user_num = 1 # 제어변수 선언(begin)
while user_num <= 10: # 논리형 조건식(end)
    print(user_num, "번 회원님 안녕하세요~~") # 반복 실행할 코드
    user_num += 1 # 제어변수의 증감식(step)

print("반복문 종료!!")
```

1. 반복문의 시작점이 되는 값을 저장하는 제어변수를 선언합니다.
2. while문의 조건식 자리에 반복문이 끝나는 시점을 조건식이나 함수로 표현합니다.
3. 반복실행할 코드를 적고 1번에서 만든 제어변수의 증감식을 적어 반복문이 언젠가 false가 되어 종료될 수 있게 합니다.

1-1-3. Quiz

```
정수 1: 5  
정수 2: 8  
5~8까지의 총합: 26
```

문제) 위와 같은 결과가 나오는 프로그램을 코딩하세요.

- 요구사항

1. 사용자는 정수 2개를 각각 입력할 수 있어야 함.
2. 프로그램은 2번째 정수를 입력하고 엔터를 누르면 1번째 정수부터 2번째 정수까지의 총합(2번째 정수포함)을 출력해야 함.
3. 반복문 while을 사용할 것!

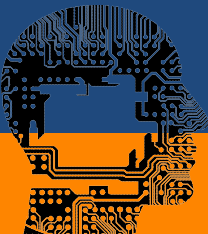


한국IT진흥부설

정보보호교육학원 아이섹

파이썬 기초

2. 반복문 for



2-1-1. 반복문 for

- for ~ in 반복문
- 나열 가능한 자료에서 자료를 모두 소비할 때까지 처리

- ❖ 나열 가능한 자료에서 하나씩 빼내서 저장할 변수 또는 변수의 목록입니다.
- ❖ for 구문 블록에서 사용합니다.
- ❖ 단일 변수 또는 여러 개 변수가 올 수 있습니다.

- ❖ 나열 가능한 자료를 저장한 변수의 이름 또는 실행 결과입니다.
- ❖ 리스트, 튜플, 딕셔너리, 셋 등이 될 수 있습니다.

for 변수 in 나열가능한자료 :

변수의 값을 처리할 문장 1

변수의 값을 처리할 문장 2

...

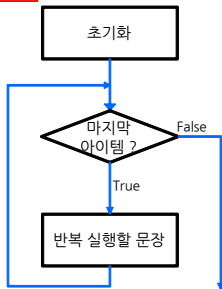
else :

- ❖ else 구분은 선택사항입니다.

- ❖ 필요 없을 경우에는 사용하지 않아도 됩니다.

변수에 값이 할당되지 않는 경우 실행할 문장

- ❖ 들여쓰기 되어 있는 문장들을 블록이라고 합니다.
- ❖ 들여쓰기 간격은 같아야 합니다.
- ❖ 일반적으로 공백 4개 또는 탭(Tab)을 사용합니다.



2-1-2. for문 사용법

```
for user_num in [1,2,3,4,5,6,7,8,9,10]:  
    print(user_num, "번 회원님 안녕하세요!!")
```

1. for 키워드와 함께 값을 저장하는 제어변수를 선언한 후 in 키워드를 붙입니다.
2. in 뒤에 나열가능한 데이터(리스트, 튜플 등)를 배치합니다.
3. 공백(들여쓰기 4칸)과 함께 실행문을 작성합니다.

2-1-3. range() 함수

- for 문장의 items 객체 위치에 range(*start*, *stop*, *step*) 함수를 이용하여 반복문을 실행시킬 수 있음. range() 함수를 이용하면 인덱스 위주의 반복을 실행

```
for i in range(0, 10):  
    print(i, end="Wt")
```

0 1 2 3 4 5 6 7 8 9

- range() 함수의 *start*가 생략되면 0부터 시작.
- range() 함수의 *step*은 얼마씩 증가시킨 값을 갖게 할 것인지 결정

```
favorite_hobby = ['reading', 'fishing', 'shopping']  
for hobby in favorite_hobby :  
    print('%s is my favorite hobby' % hobby)
```

```
reading is my favorite hobby  
fishing is my favorite hobby  
shopping is my favorite hobby
```

2-1-4. Quiz

구구단 단수 : 5

구구단 5 단

=====

5 x 1 = 5

5 x 2 = 10

5 x 3 = 15

5 x 4 = 20

5 x 5 = 25

5 x 6 = 30

5 x 7 = 35

5 x 8 = 40

5 x 9 = 45

문제) 왼쪽과 같은 결과가 나오는 프로그램을 코딩하세요.

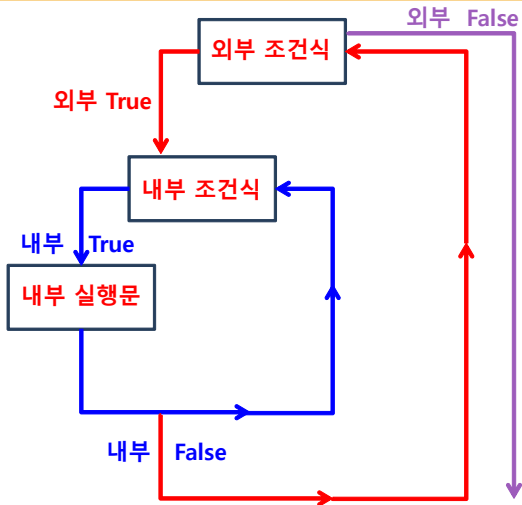
- 요구사항

1. 프로그램 실행 시 구구단 단수를 입력할 수 있게 함.

2. 단수에 맞는 구구단이 출력되어야 함.

2. for문과 while문으로 각각 구현하세요.

2-2-1. 중첩 반복문



2-2-2. 중첩 반복문 예시

```
for dan in range(2, 10):  
    print("# 구구단 {}".format(dan))  
    print("=" * 40)  
  
    for hang in range(1, 10):  
        print("{} x {} = {}".format(dan, hang, dan*hang))  
    print("=" * 40)
```

```
2 x 1 = 2  
2 x 2 = 4  
2 x 3 = 6  
2 x 4 = 8  
2 x 5 = 10  
2 x 6 = 12  
2 x 7 = 14  
2 x 8 = 16  
2 x 9 = 18
```

...

```
9 x 1 = 9  
9 x 2 = 18  
9 x 3 = 27  
9 x 4 = 36  
9 x 5 = 45  
9 x 6 = 54  
9 x 7 = 63  
9 x 8 = 72  
9 x 9 = 81
```

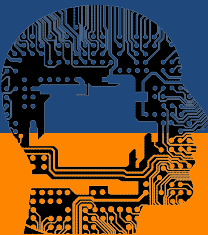


한국IT진흥부설

정보보호교육학원 아이섹

파이썬 기초

3. 탈출문



3-1-1. 탈출문 break

for / while :

break



- break문은 for문이나 while문에서 사용되며 반복문이 실행 중 break를 만나는 순간 **반복문을 해당 시점에서 종료**합니다.

- break를 만나는 순간 이하의 **반복문의 남은 코드는 모두 실행되지 않으며** 블록을 탈출합니다.

- 대개 if문과 함께 사용되며 조건에 따라 반복문을 종료할 때 사용합니다.

3-1-2. break문 사용 예시

```
for a in range(1, 11): #[1,2,3,4,...,9,10]
    if a == 6:
        break
    print(a, end=" ")
print("\n반복문 종료!")
```



```
1 2 3 4 5
반복문 종료!
```

while True:

if (...):

break

- 무한 루프는 반복문의 **반복 횟수**를 개발자가 **사전에** 정확하게 **인지하지 못하는 상황**에서 사용하며 특정 조건 하에서 반복문을 강제로 종료하는 형태로 구성합니다.

- 프로그램이 **중단되지 않게 유지**할 때도 무한루프를 사용합니다.

- 무한 루프는 일반적으로 while문을 사용하며 while의 조건식 자리에 논리값 True를 적으면 무한 루프를 구성할 수 있습니다.

3-1-4. 무한루프 사용 예시

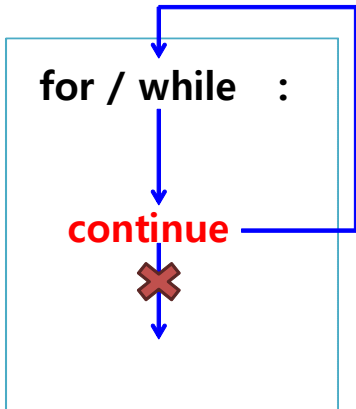
```
print("# 먹고 싶은 과일을 입력하세요.")
print("[입력을 중지하려면 '메롱'이라고 입력하세요!]")

while True:
    fruit = input("> ")
    if fruit == "메롱":
        print("입력을 중지합니다.")
        break
```



```
# 먹고 싶은 과일을 입력하세요.
[입력을 중지하려면 '메롱'이라고 입력하세요!]
> 오렌지
> 자몽
> 메롱
입력을 중지합니다.
```

3-2-1. 탈출문 continue



- **continue**문은 for문이나 while문에서 사용되며 반복문이 실행 중 **continue**를 만나는 순간 **for문의 경우 증감식**, **while문의 경우 조건식**으로 이동합니다.

- **continue**는 **break**와 달리 반복문을 종료하지 않고 계속 수행합니다.

- 조건부로 특정 반복회차를 건너뛸 때 사용합니다.

3-2-2. continue문 사용 예시

```
for x in range(1, 11):  
    if x == 6: continue  
    print(x, end=" ")  
print("\n반복문 종료!")
```



```
1 2 3 4 5 7 8 9 10  
반복문 종료!
```

감사합니다
THANK YOU