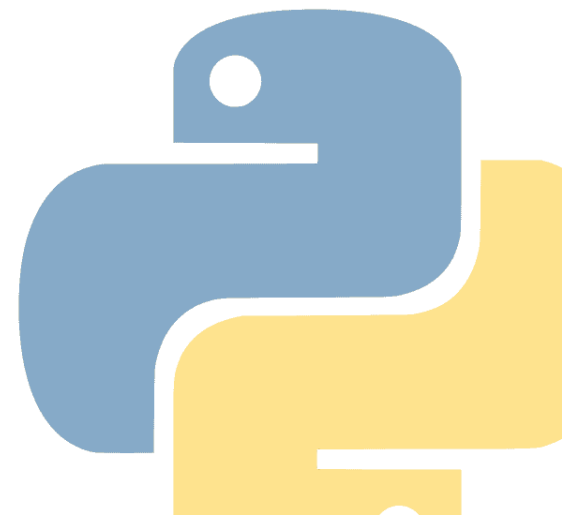


웹/파이썬 멘토링

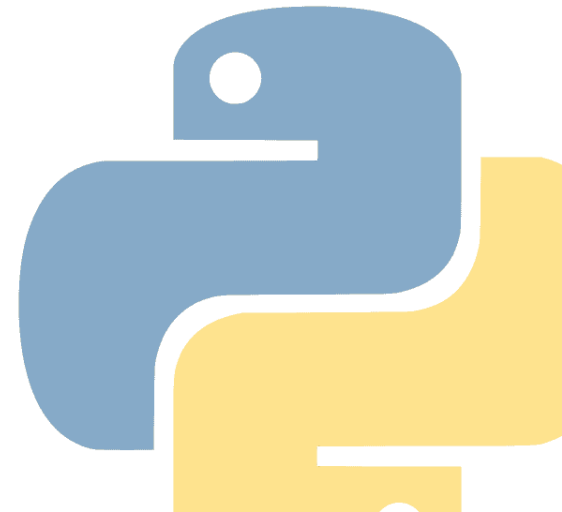
<List & Loops>

강유미 김진환





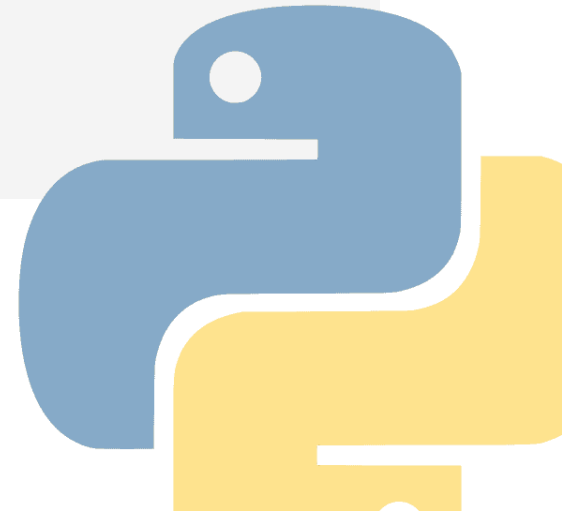
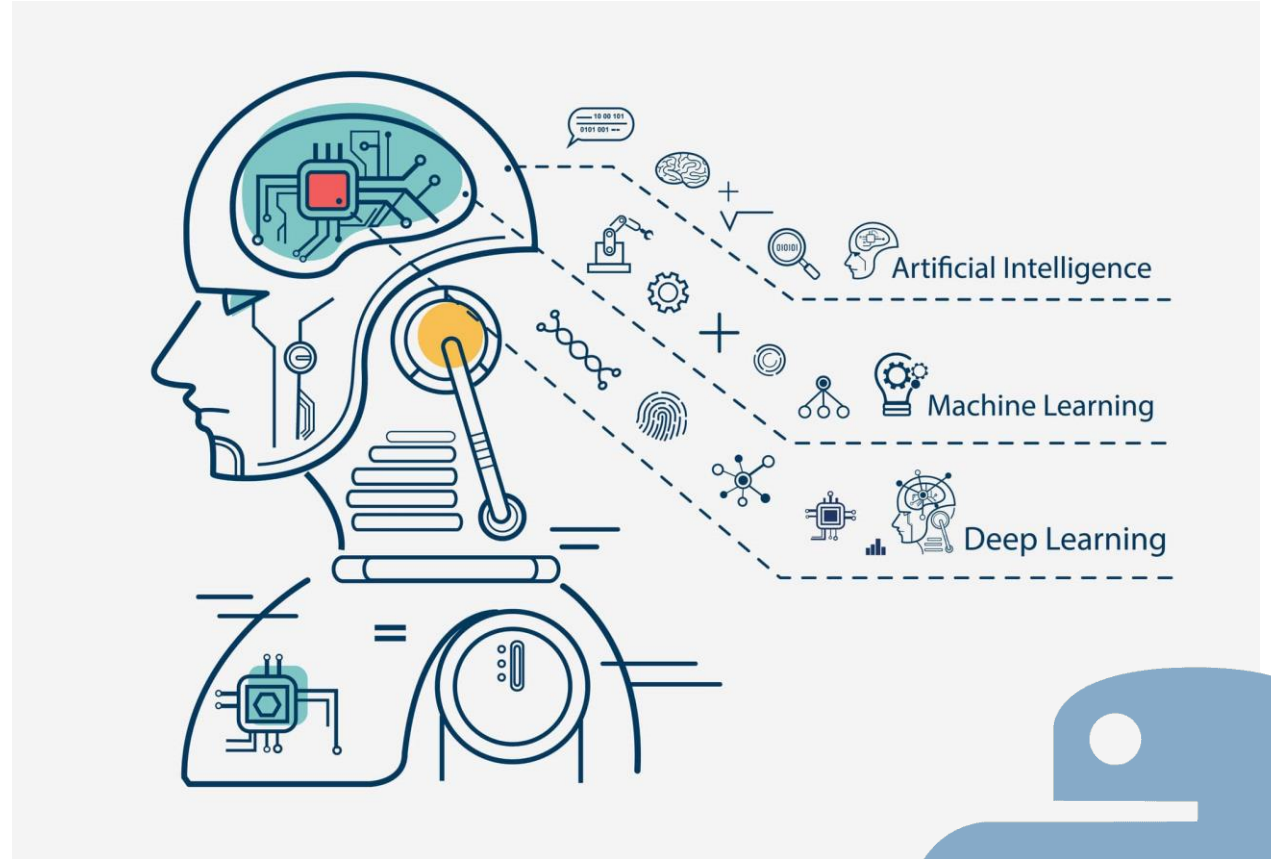
Loops



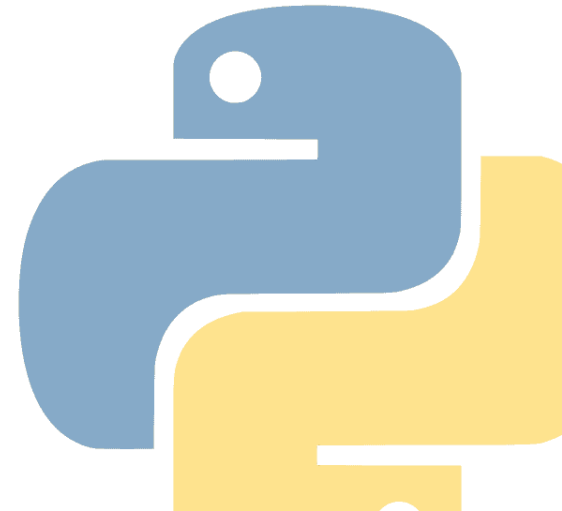
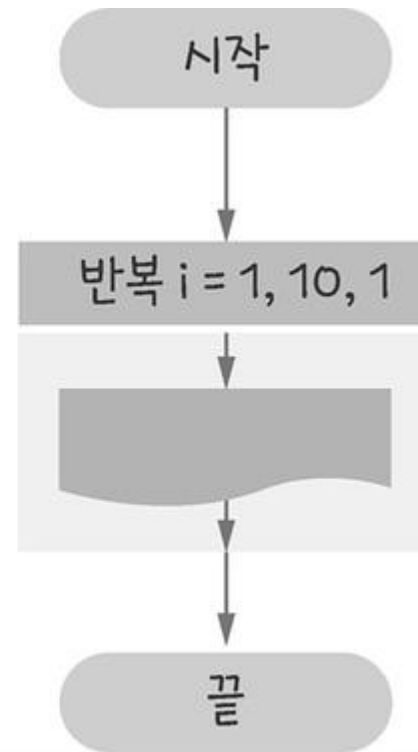
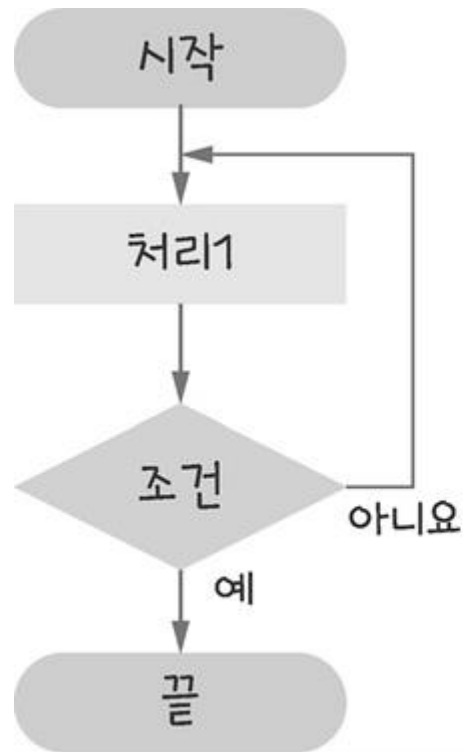
반복문이란?

Repetition – Loops

계산 같은 건 컴퓨터한테 맡겨!



반복문이란?



반복문 활용

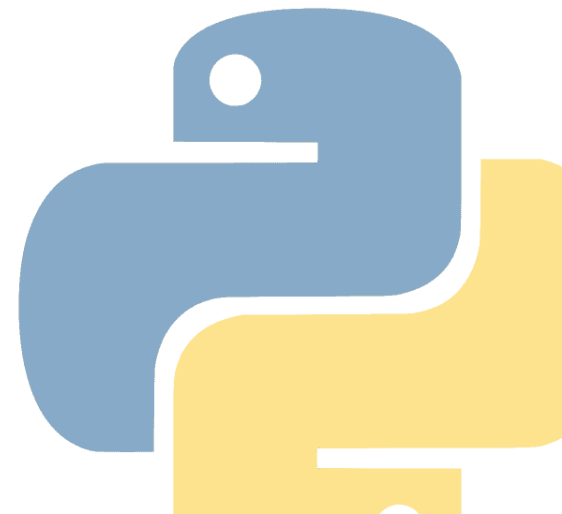
```
1번 선수 배승호!!!!
2번 선수 최원준!!!!
3번 선수 박선홍!!!!
4번 선수 유영빈!!!!
5번 선수 김진환!!!!
```

```
0 1 2 3 4 5 6 7 8
10 11 12 13 14 15 16 17 18
20 21 22 23 24 25 26 27 28
30 31 32 33 34 35 36 37 38
40 41 42 43 44 45 46 47 48
50 51 52 53 54 55 56 57 58
60 61 62 63 64 65 66 67 68
70 71 72 73 74 75 76 77 78
80 81 82 83 84 85 86 87 88
90 91 92 93 94 95 96 97 98
```

```
a = [10, 11, 12, 13, 14, 15, 16, 17, 18, 19]
```

```
10 + 11 + 12 + 13 + 14 + 15 + 16 + 17 + 18 + 19
```

연산, 입력, 출력 등 다양한 작업 반복



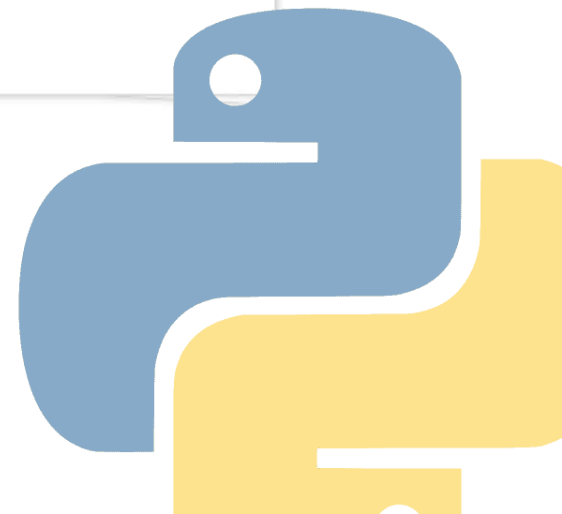
while 문

`while` <<조건문>>:
 <<반복 실행 블록>>
 <<반복문 탈출조건>>

```
while True:  
while 1 < x < 5:  
while a != "코카콜라"
```

```
In [4]: i = 0  
while i < 10:  
    i += 1  
    print("{}번째 출력".format(i))
```

1번째 출력
2번째 출력
3번째 출력
4번째 출력
5번째 출력
6번째 출력
7번째 출력
8번째 출력
9번째 출력
10번째 출력



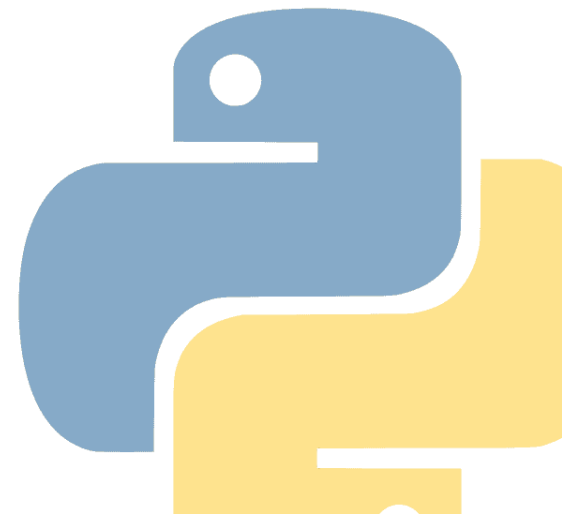
반복문의 필수 요소

- 반복 실행 조건
- 반복문 탈출 조건

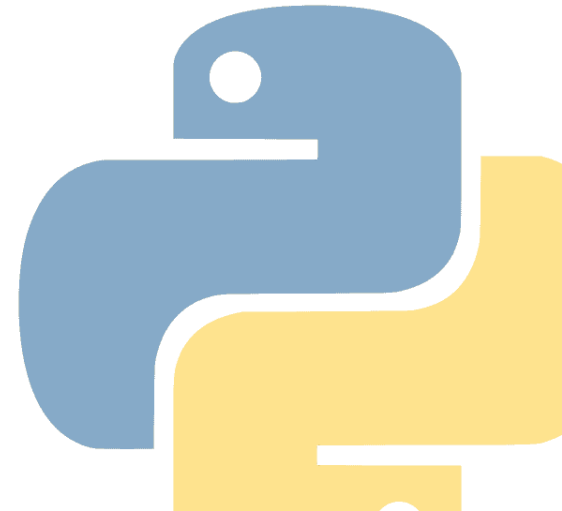
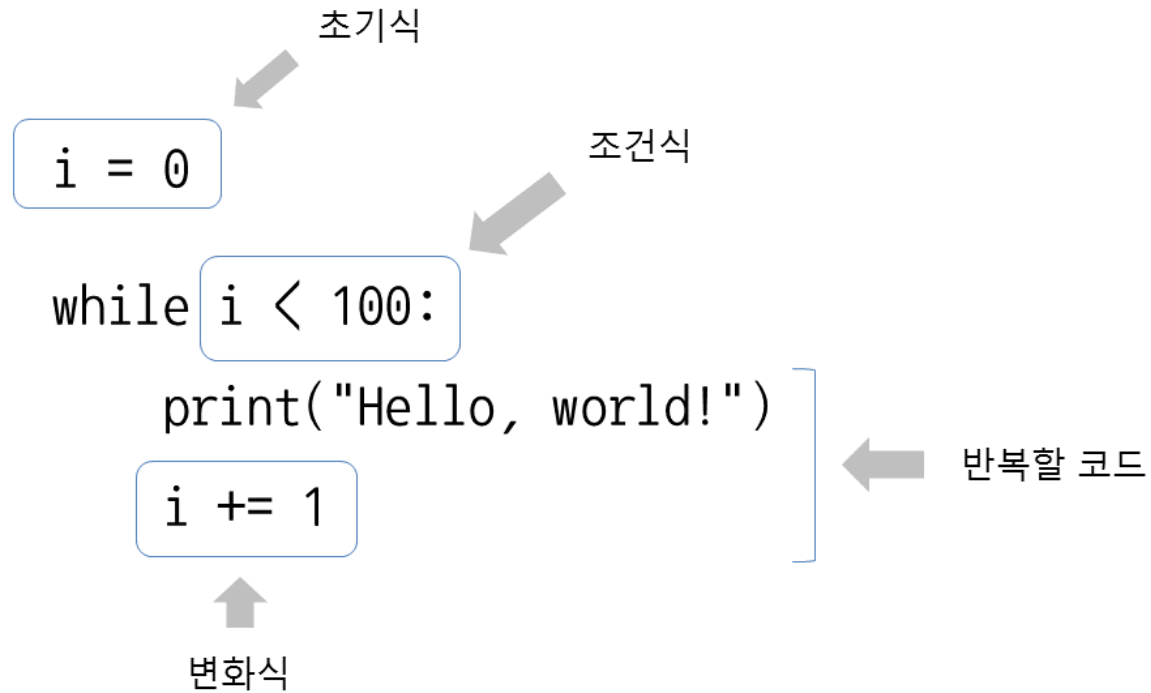
“무한루프”

```
while True:  
    print("진환선배 오늘 교육 언제 끝나나요?")
```

-> 조건문을 거짓으로 만들기



반복문의 필수 요소



Quiz #1

진환 선배에게 밥을 사달라고 졸라보자.

진환 선배는 5번 조르면 넘어온다.

선배를 5번 조르면 선배는 **그래 가자!** 를 외친다.

출력 결과

진환 선배 밥 사주세요!

진환 선배 밥 사주세요!

진환 선배 밥 사주세요!

진환 선배 밥 사주세요!

진환 선배 밥 사주세요!

그래 가자!



for 문

for <<변수>> in <<리스트>>:
 <<반복 실행 블록>>
~~<<반복문 탈출조건>>~~

For문은 반복문 탈출 조건이 자동으로 정해짐

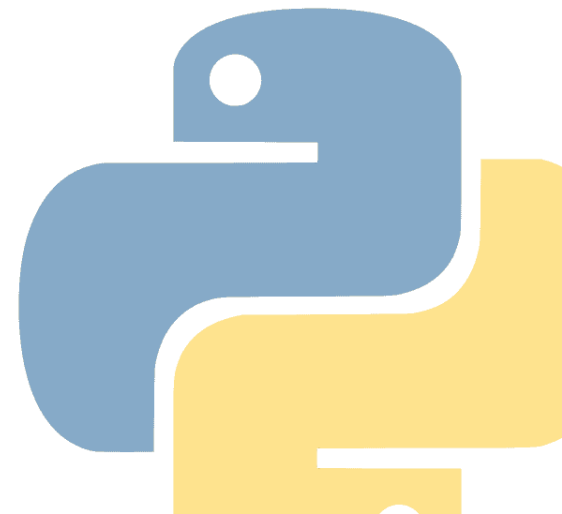
-> 마지막 블록 수행 뒤 자동종료

```
In [1]: for i in [1, 2, 3]:
        print(i) # 1, 2, 3 출력

        for i in (1.5, 33, 8):
            print(i) # 1.5, 33, 8 출력

        for i in "Python":
            print(i) # P y t h o n 출력
```

```
1
2
3
1.5
33
8
P
y
t
h
o
n
```



range(n)

특정 구간의 정수 범위를 반환

특정 횟수 반복문 작성 시 용이

```
print(list(range(10)))  
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

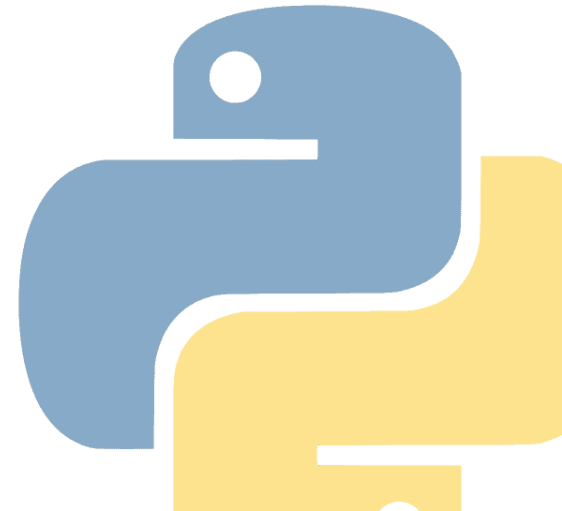
```
print(list(range(3,10)))  
[3, 4, 5, 6, 7, 8, 9]
```

```
print(list(range(0, 10, 2)))  
[0, 2, 4, 6, 8]
```

```
range(stop)  
-> [0:stop)
```

```
range(start, stop)  
-> [start : stop)
```

```
range(start, stop, step)  
-> [start : stop) step 간격
```

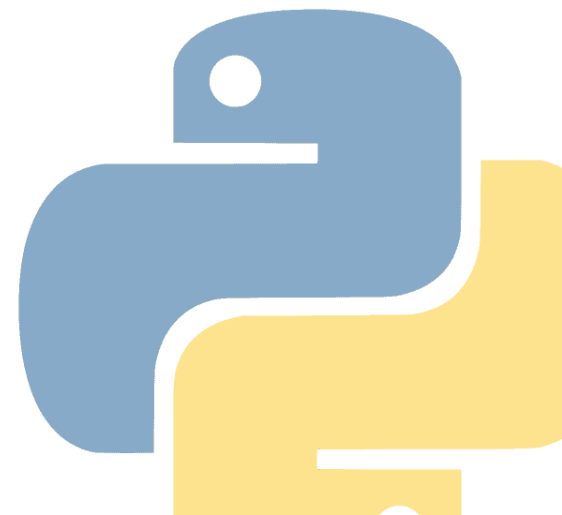


for i in range(n)

```
In [1]: ▶ # 1 ~ 100 까지의 합 구하기  
  
total = 0  
for i in range(1, 100+1):  
    total += i # (total = total + i)  
  
print(total) # 5050 값 출력
```

5050

그냥 외우세요

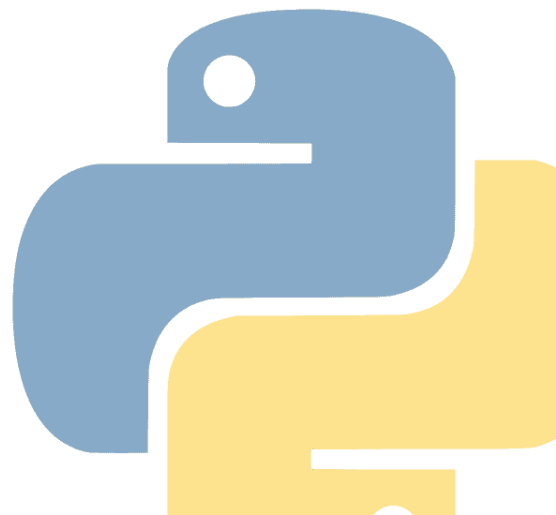


다중 for 문

```
In [3]: ▶ # 리스트 요소로 반복문 사용
sample = [[11,12,13],[21,22,23],[31,32,33],[41,42,43],[51,52,53]]

for i in range(5) :
    for j in range(3):
        print(sample[i][j], end=" ") # 1차원 리스트 한 줄씩 출력
    print() # 한 줄 띄기
```

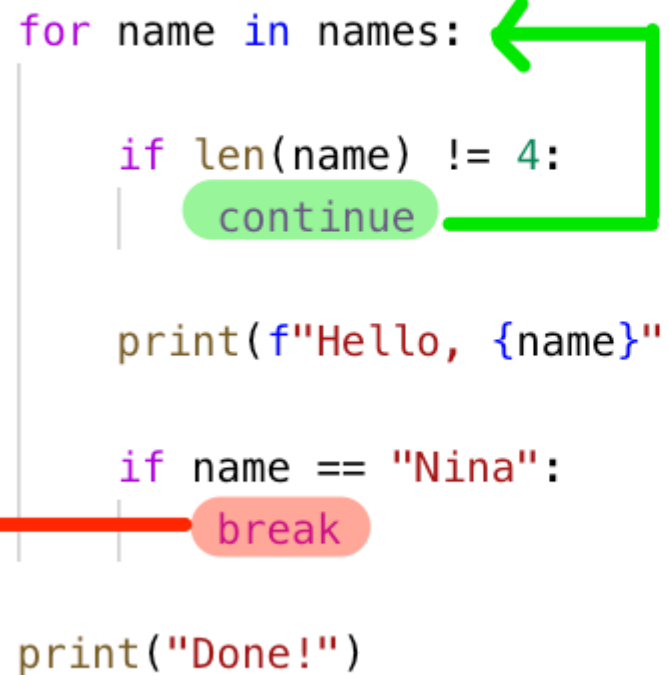
```
11 12 13
21 22 23
31 32 33
41 42 43
51 52 53
```



break / continue

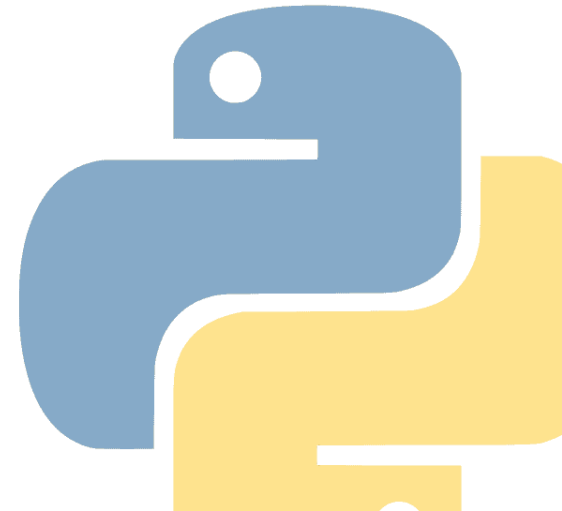
```
names = ["Jimmy", "Rose", "Max", "Nina", "Phillip"]
```

```
for name in names:
    if len(name) != 4:
        continue
    print(f"Hello, {name}")
    if name == "Nina":
        break
print("Done!")
```



특정 조건, 예외 처리에 많이 사용

```
Hello, Rose
Hello, Nina
Done!
```



Quiz #2

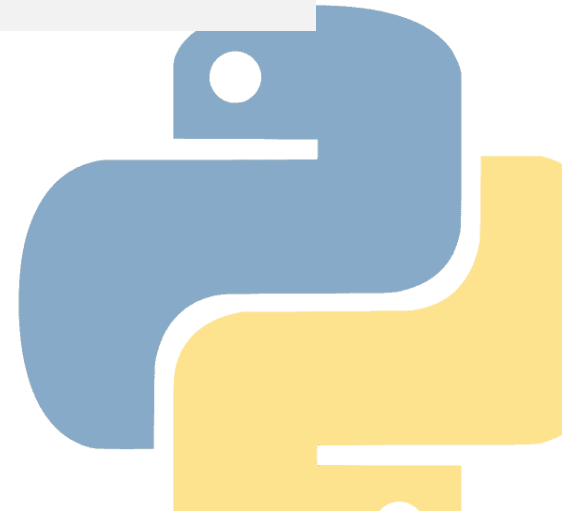
다가오는 크리스마스를 위해

10단 짜리 크리스마스 트리를 만들어 보자

출력 결과

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

Hint! Print 시 공백 개수와 별의 개수가 어떻게 증감하는지 확인

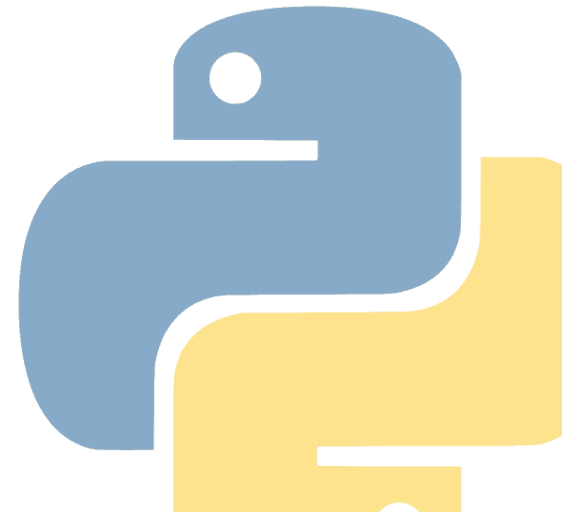


while

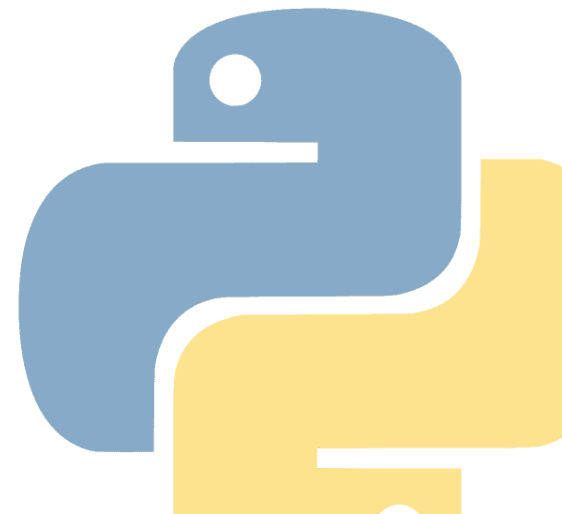
- 특수한 조건, 경우의 수가 많은 경우
- 실행 횟수가 정해져 있지 않은 경우

for

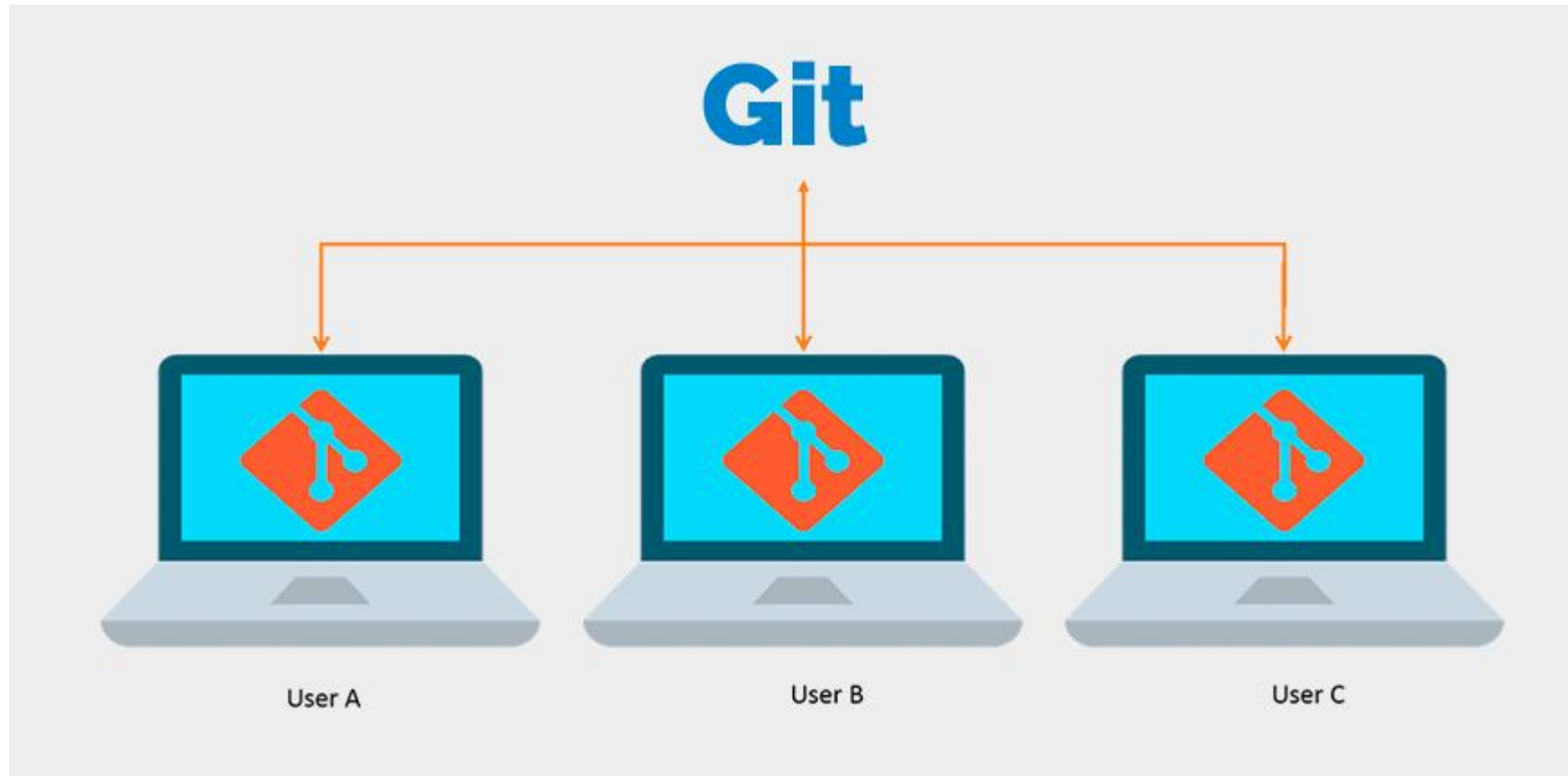
- 실행 횟수가 정해져 있는 경우
- 리스트, 문자열, 튜플의 요소를 이용하는 경우
- 행렬을 만드는 경우

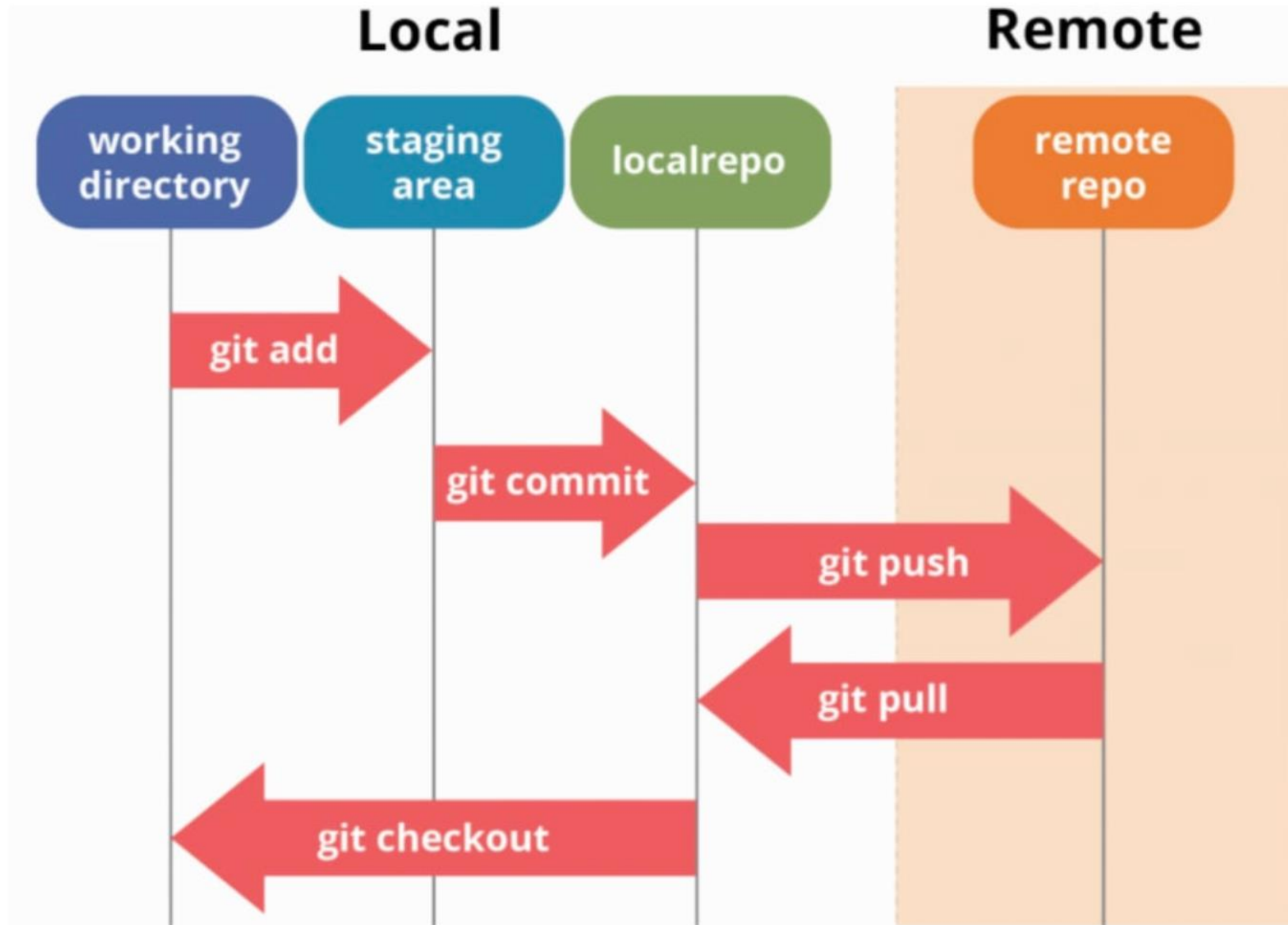


Q & A



Git





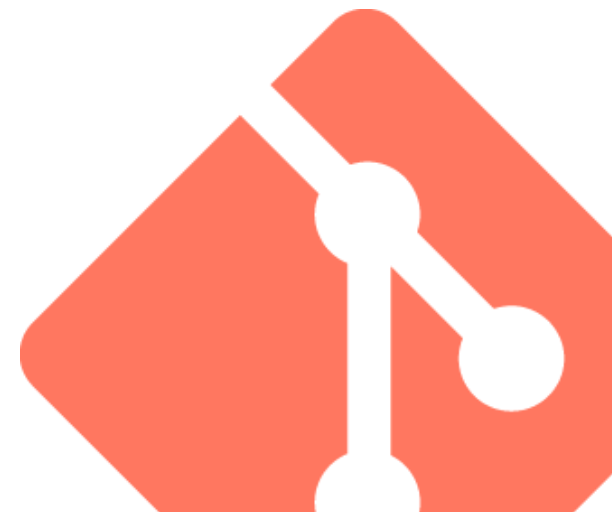
Git

- `git init`
- `git config --global user.email <<email>>`
- `git config --global user.name <<name>>`
- `git remote add origin <<Github 레포 주소>>`
- `git add <<파일명>>`
- `git commit -m "<<Commit message>>"`
- `git push origin master`



Git

- ~~git init~~
- ~~git config --global user.email <<email>>~~
- ~~git remote add origin <<Github 레포 주소>>~~
- ~~git config --global user.name <<name>>~~
- git add <<파일명>>
- git commit -m "<<Commit message>>"
- git push origin master



Q & A

