S06. 제어문

■ if문

```
--------------------------------------------------------------------------
# if - else 문 예시

signal_color = input('색을 영문으로 입력하세요: ')       # 색 입력 요청

if signal_color == 'blue':                                  # 파란색인 경우
    print('신호등은 파란색입니다. 건너세요.')
else:                                                       # 빨간색인 경우
    print('신호등은 빨간색입니다. 기다리세요.')



--------------------------------------------------------------------------
# if - elif - else 문 예시

signal_color = input('색을 영문으로 입력하세요: ')       # 색 입력 요청

if signal_color == 'blue':                                  # 파란색인 경우
    print('신호등은 파란색입니다. 건너세요.')
elif signal_color == 'red':                                 # 빨간색인 경우
    print('신호등은 빨간색입니다. 기다리세요.')
else:                                                       # 모르는 색인 경우
    print('잘못된 색입니다.')
```

```
# 중첩 if 문 예시

signal_color = input('색을 영문으로 입력하세요: ')          # 색 입력 요청

if signal_color == 'blue':                                  # 파란색인 경우
    print('신호등은 파란색입니다. 건너세요.')

    is_pass = input('건널 준비가 되었나요? (yes/no)')     # 건널 준비가 되었는지 확인
    if is_pass == 'yes':                                    # 준비가 된 경우
        print('건너겠습니다!!')
    else:                                                   # 준비가 안 된 경우
        print('다음 번에 건너겠습니다.')

elif signal_color == 'red':                                 # 빨간색인 경우
    print('신호등은 빨간색입니다. 기다리세요.')
else:                                                       # 모르는 색인 경우
    print('잘못된 색입니다.')
```

■ while문

```
# while 문 예시

signal_color = ''                                                    # 변수 선언 및 초기화

while signal_color != 'blue' and signal_color != 'red':       # 반복문 시작

    signal_color = input('색을 영문으로 입력하세요: ')        # 표준 입력문

    if signal_color == 'blue':                                       # 파란색인 경우
        print('신호등은 파란색입니다. 건너세요.')
    elif signal_color == 'red':                                      # 빨간색인 경우
        print('신호등은 빨간색입니다. 기다리세요.')
    else:                                                                # 모르는 색인 경우
        print('잘못된 색입니다. 다시 입력 하세요.')

print('프로그램을 종료합니다.')
```

■ for 문

```
# for 문 예시

signals = 'blue', 'yellow', 'red'         # 3가지 색에 대한 튜플 생성

for signal in signals:                      # for 문 실행
    print(signal, len(signal))            # 튜플의 값, 길이 출력
```

```
--------------------------------------------------------------------------------
# for 문과 range() 함수 예시

signals = 'blue', 'yellow', 'red'                # 3가지 색에 대한 튜플 생성

for x in range(len(signals)):                      # range() 함수를 통한 for 문 수행
    print(x, signals[x], len(signals[x]))        # 튜플의 색인, 값, 길이 출력


--------------------------------------------------------------------------------
# break 문 예시

signals = 'blue', 'yellow', 'red'              # 3가지 색에 대한 튜플 생성

for x in range(len(signals)):                    # range() 함수를 통한 for 문 수행
    print(x, signals[x], '루프 시작!')          # 튜플의 색인, 값, 루프 시작 메시지 출력
    if signals[x] == 'yellow':
        break                                        # 반복문 수행 종료
    print(x, signals[x], '루프 종료!!')         # 튜플의 색인, 값, 루프 종료 메시지 출력

print('프로그램 종료!!')                          # 프로그램 종료 메시지 출력


--------------------------------------------------------------------------------
# continue 문 예시

signals = 'blue', 'yellow', 'red'              # 3가지 색에 대한 튜플 생성

for x in range(len(signals)):                    # range() 함수를 통한 for 문 수행
    print(x, signals[x], '루프 시작!')          # 튜플의 색인, 값, 루프 시작 메시지 출력
    if signals[x] == 'yellow':
        continue                                     # 루프 수행 종료
    print(x, signals[x], '루프 종료!!')         # 튜플의 색인, 값, 루프 종료 메시지 출력

print('프로그램 종료!!')                          # 프로그램 종료 메시지 출력
```

x값에는 0,1,2 index값이 들어간다.

```
--------------------------------------------------------------------------------
# pass 문 예시

signals = 'blue', 'yellow', 'red'               # 3가지 색에 대한 튜플 생성

for x in range(len(signals)):                    # range() 함수를 통한 for 문 수행
    print(x, signals[x], '루프 시작!')           # 튜플의 색인, 값, 루프 시작 메시지 출력
    if signals[x] == 'yellow':
        pass                                     # 아무 작업 하지 않기
    print(x, signals[x], '루프 종료!!')          # 튜플의 색인, 값, 루프 종료 메시지 출력

print('프로그램 종료!!')                          # 프로그램 종료 메시지 출력



--------------------------------------------------------------------------------
# 중첩 for문

# 3x2 2차원 배열을 중첩리스트로 선언
nest = [[1,2,3], [4,5,6], [7,8,9]]

for x in range(3):                       # 행을 위한 바깥쪽 for문
    for y in range(3):                   # 열을 위한 안쪽 for문
        print('nest[', x, '][', y, ']:', nest[x][y])  #항목 출력
```