

파이썬 기본

4. 반복문



python

01. 반복의 종류

- 횟수 반복(for 문): 정해진 횟수만큼 반복한다.
- 조건 반복(while 문): 특정한 조건이 성립되는 동안 반복한다.

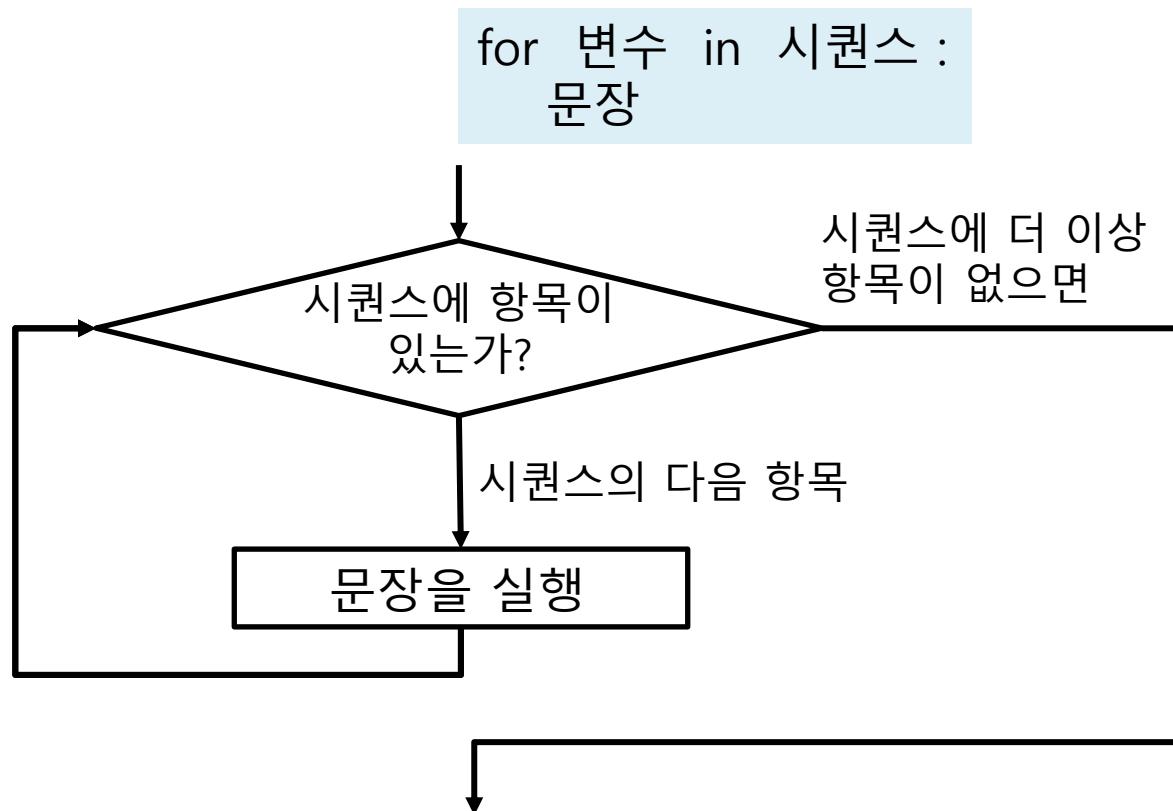




01. 반복의 종류

➤ 횡수 반복

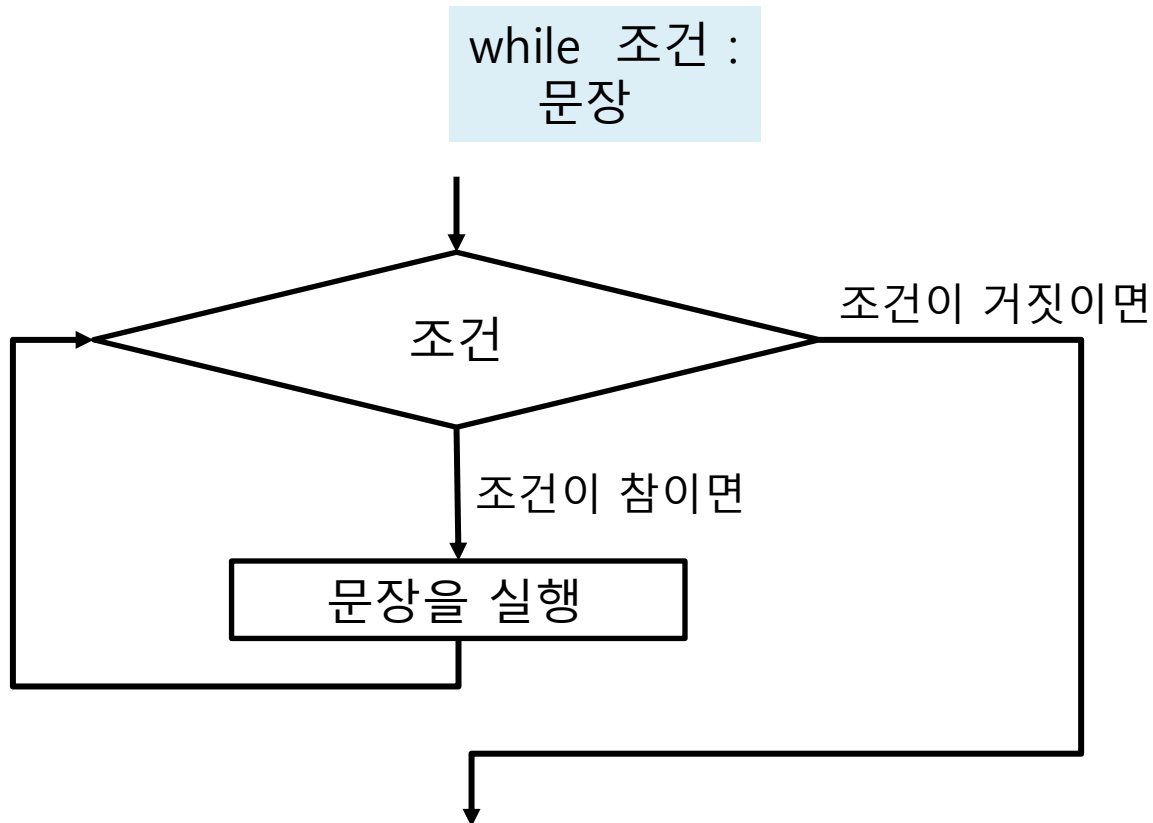
반복의 횡수를 미리 아는 경우에 사용



01. 반복의 종류

➤ 조건 반복

특정한 조건이 만족되는 동안 계속 반복

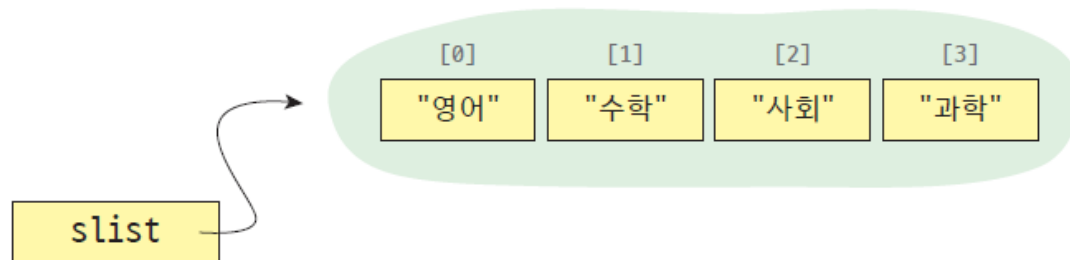




02. 리스트란?

- 하나 이상의 데이터(항목)를 묶어서 하나의 이름으로 저장하는 데이터 구조
- 대량의 데이터를 효율적이고 간편하게 처리 가능
- 대괄호로 묶어서 표현 - [항목, 항목, 항목,...]
- 각 항목은 인덱스를 사용하여 접근

```
slist = [ "영어", "수학", "사회", "과학" ]
```

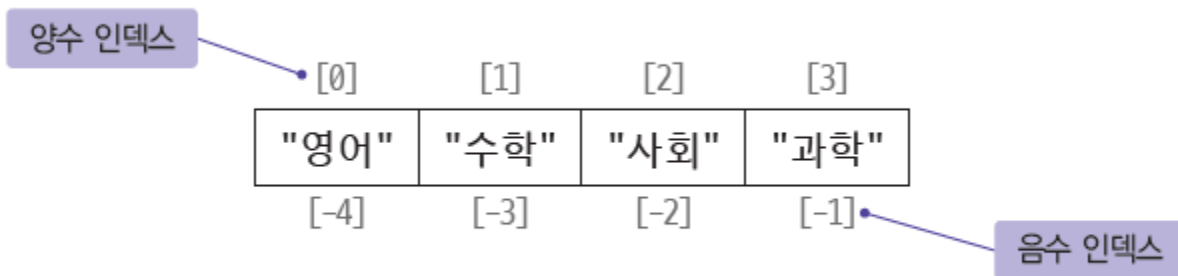




03. 리스트의 인덱스

- 인덱스는 0부터 시작한다.
- 첫 번째 항목의 인덱스는 0이고 두 번째 항목의 인덱스는 1, 세 번째 항목의 번호는 2인 것이다.

```
>>> slist = [ "영어", "수학", "사회", "과학" ]  
>>> slist[0]
```





04. 리스트 사용

- 파이썬에서는 다양한 종류의 데이터를 하나의 리스트 안에 저장할 수 있다

```
>>> myList = [ 1, 'computer', 3.4]
```

```
>>> myList = ['apple', [8, 4, 6]]
```

```
>>> myList[0] = 'banana'
```

```
>>> numbers = [ 7, 12, 33, 777 ]
```

```
>>> heroes = [ ] ➔ 빈 리스트 생성
```

```
>>> heroes.append('아이언맨')
```

```
>>> heroes.append('슈퍼맨')
```

```
>>> print(heroes)
```

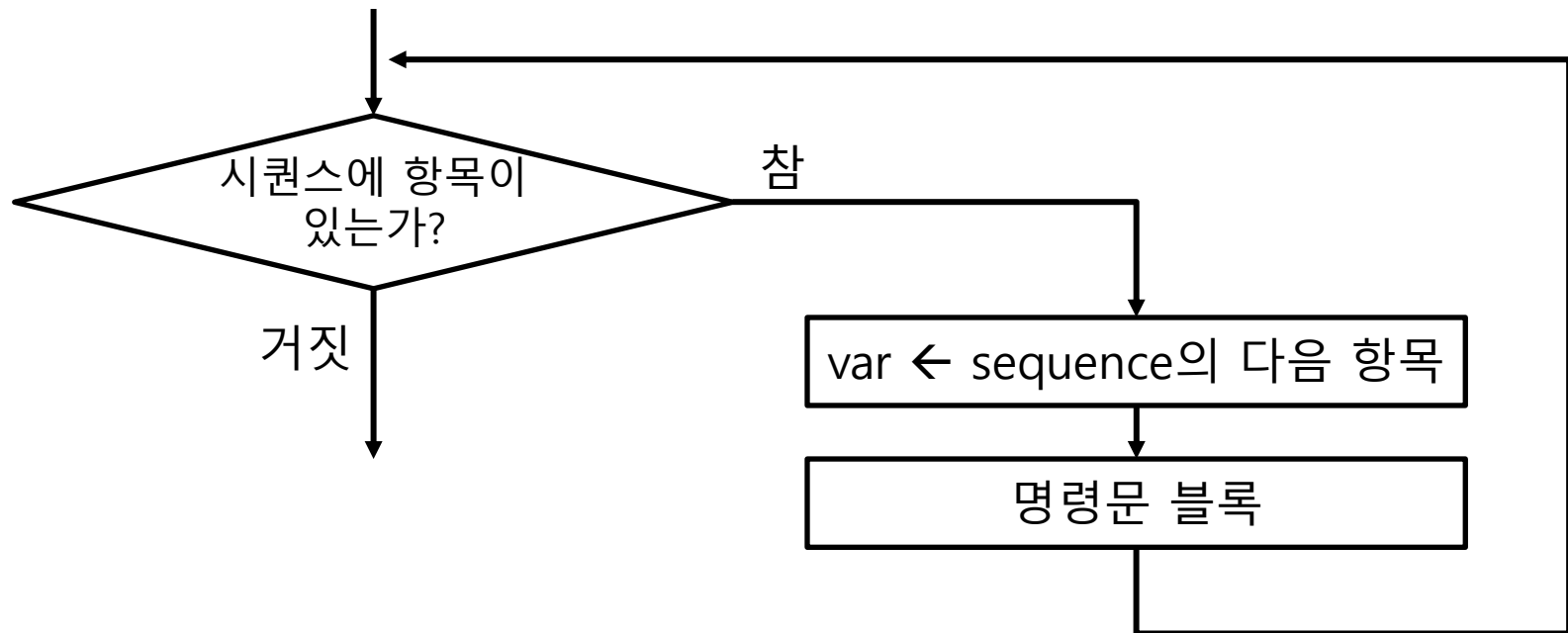
```
>>> numbers.append('a')
```

```
>>> numbers
```

```
>>> numbers.append([10,20])
```

05. 횡수 제어 반복

for 변수 in 시퀀스 :
반복 실행할 명령어





05. 횟수 제어 반복

```
>>> for i in [1, 2, 3, 4, 5] :  
    print("Good morning!")  
  
>>> for i in [1, 1, 1, 1, 1] :  
    print("Good morning!")  
  
>>> for i in [1, 2, 3, 4, 5] :  
    print(i, "Good morning")  
  
>>> for i in [1, 2, 3, 4, 5, 6, 7, 8, 9] :  
    print("9 *", i, "=", 9 * i)  
  
  
>>> for i in "string" :  
    print("Hello")  
  
>>> for i in "string" :  
    print(i, end=' ')
```



05. 횟수 제어 반복

- range() 함수

start 에서 stop -1 까지 step의 간격으로 정수 생성

start 를 생략하면 0, step를 생략하면 1

```
range( [ start=0, ] stop [ , step=1 ] )
```

```
>>> range(10) ➔ 0, 1, 2, 3, 4, 5, 6, 7, 8, 9
```

```
>>> range(1,5) ➔ 1, 2, 3, 4
```

```
>>> range(0, 10, 2) ➔ 0, 2, 4, 6, 8
```

```
>>> range(10, 0, -1) ➔ 10, 9, 8, 7, 6, 5, 4, 3, 2, 1
```

```
>>> range(20, 10, -2) ➔ 20, 18, 16, 14, 12
```

```
>>> list(range(10))
```

```
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```



05. 횡수 제어 반복

```
for 변수 in range(종료값):  
    문장
```

```
>>> for i in range(10):
```

```
    print("Hello")
```

```
>>> for i in range(10):
```

```
    print(i, end=' ')
```

```
>>> for i in range(1, 10):
```

```
    print('9 x %d = %2d' %(i, 9*i))
```

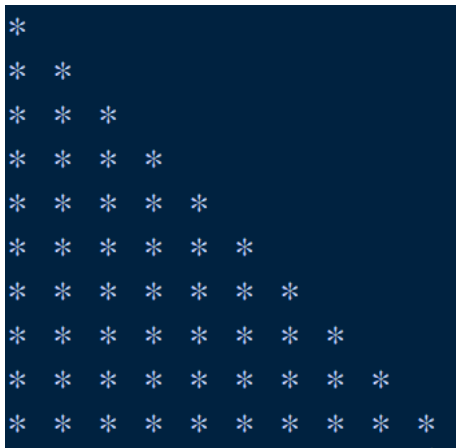
```
>>> for num in range(0, 21, 3):
```

```
    print(num, end = ' ')
```



실습문제

1. 반복문을 이용하여 아래 보이는 것 처럼 2가지 모양으로 출력해보세요



2. 1 부터 100 까지의 합을 계산하여 출력하는 프로그램

예) 1부터 100 까지의 합은 : 5050

3. 숫자를 입력 받고, 1부터 입력 받은 수까지의 합을 계산하여 출력하는 프로그램

예) 합을 계산할 숫자를 입력하세요 : 10

1부터 10 까지의 합은 : 55



06. 조건 제어 반복

- 어떤 조건이 만족되는 동안(조건이 참인 동안) 반복

```
while 조건식 :
```

```
    문장1
```

```
    문장2
```

참이나 거짓으로 계산되는 조건식,
관계 연산자 `==` `!=` `<` `>` `<=` `>=`를 사용한다.

```
while money < TARGET :
```

콜론(:)은 복합문을 의미한다.

```
    money = money + money * rate
```

```
    year = year + 1
```

반복되는 문장은 동일하게
들여쓰기가 되어야 한다.

조건이 참이면 반복되는 문장들

실습. 투자금이 2배가 되는데 걸리는 시간 계산

(P.173)

```
# money.py
```

```
TARGET = 2000          # 목표 금액
money = 1000            # 초기 자금
year = 0                # 연도
rate = 0.07             # 이자율
```

```
# 현재 금액이 목표 금액보다 작으면 반복한다.
```

```
while money < TARGET :
    money = money + money * rate
    year = year + 1
```

```
print(year, "년")
```



Lab. 구구단 출력(p.176)

- 구구단 9단을 반복문(while 문)을 이용하여 출력

```
# gugu.py
```

```
dan = int(input('원하는 단은 : '))
```

```
i = 1
```

```
while i <= 9 :
```

```
    print('%s x %s = %s' %(dan, i, dan*i))
```

```
    i = i + 1
```



실습. 사용자가 입력하는 숫자의 합 계산

- 사용자가 입력한 숫자를 더해서 합을 계산하는 프로그램. 입력을 계속할 건지 물어서 yes 를 입력하면 다음숫자 입력, no를 입력하면 합을 출력

예) 숫자를 입력하세요 : 30

계속 (yes/no) ? Yes

숫자를 입력하세요 : 40

계속 (yes/no) ? No

입력한 숫자들의 합 : 70



실습문제. 성적계산

4. 학생들의 성적을 입력 받아 합계와 평균을 구하시오. 성적 입력을 끝내고 싶을 때는 점수에 음수를 입력한다. 평균은 수수 둘째자리까지 나타낸다

예) 학생들의 성적 합계와 평균을 계산합니다.

입력을 종료하려면 음수를 입력하세요

성적을 입력하세요 : 88

성적을 입력하세요 : 98

성적을 입력하세요 : 76

성적을 입력하세요 : -1

합계 : 262 평균 : 87.33



Lab. 숫자 맞추기 게임(업 앤 다운,p.177)

- 프로그램이 가지고 있는 정수를 사용자가 알아맞히는 게임. 프로그램이 1 부터 100사이로 정한 숫자와 입력한 값을 비교하여 입력한 값이 높은 지 낮은 지 알려준다. 정답을 맞추면 몇 번 만에 맞추었는지 시도 횟수를 알려준다

예) 1 부터 100 사이의 숫자를 맞추시오.

숫자를 입력하세요 : 50

낮음!!! 더 큰 숫자를 입력하세요

숫자를 입력하세요 : 75

낮음!!! 더 큰 숫자를 입력하세요

숫자를 입력하세요 : 90

높음!!! 더 작은 숫자를 입력하세요

숫자를 입력하세요 : 82

축하합니다. 정답입니다!!! 시도횟수 = 4



LAB. 로그인 프로그램 (p.180)

알고리즘

1. 암호 = ""
2. 암호가 'pythonisfun'이 아니면 다음을 반복한다
 사용자로부터 암호를 입력 받는다
3. '로그인 성공'을 출력한다

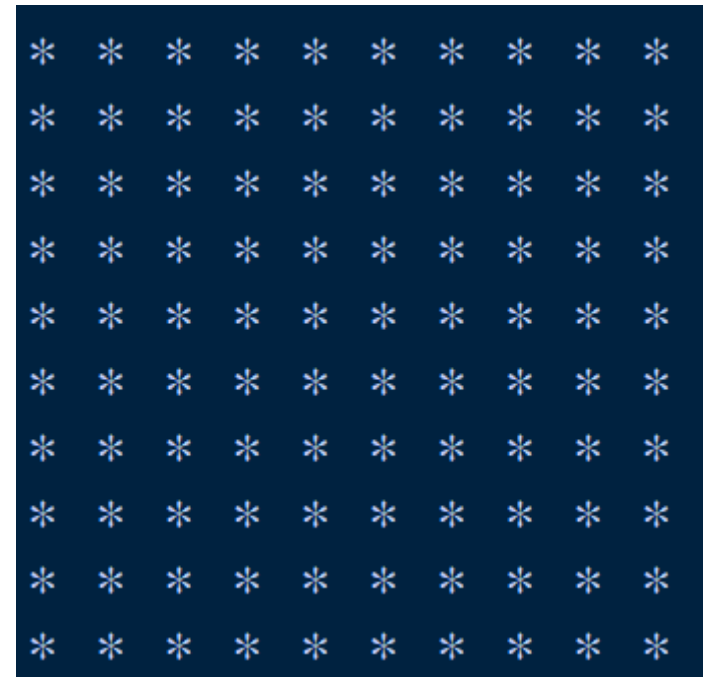
```
# password.py
password = ""
while password != 'pythonisfun':
    password = input('암호를 입력하세요 : ')
print('로그인 성공')
```



07. 중첩 반복문

- 반복문 안에 다른 반복문이 포함된 구문
- 외부 반복문(outer loop) , 내부 반복문(inner loop)로 구분한다.
- 내부 반복문은 외부 반복문이 실행될 때 마다 새롭게 실행된다

```
for x in range(10):  
    for y in range(10):  
        print('*', end = '')  
    print('')
```





07. 중첩 반복문

```
# nested3.py
```

```
adj = ["small", "medium", "large"]
```

```
nouns = ["apple", "banana", "grape"]
```

```
for x in adj :
```

```
    for y in nouns :
```

```
        print(x, y)
```



실습문제

5. 중첩 반복문을 이용하여 아래 보이는 것 처럼 2가지 모양으로 출력해보세요

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

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

6. 2단부터 9단까지 구구단을 출력해보세요

| | | | | | | | |
|--------|--------|--------|--------|--------|--------|--------|--------|
| 2x1= 2 | 3x1= 3 | 4x1= 4 | 5x1= 5 | 6x1= 6 | 7x1= 7 | 8x1= 8 | 9x1= 9 |
| 2x2= 4 | 3x2= 6 | 4x2= 8 | 5x2=10 | 6x2=12 | 7x2=14 | 8x2=16 | 9x2=18 |
| 2x3= 6 | 3x3= 9 | 4x3=12 | 5x3=15 | 6x3=18 | 7x3=21 | 8x3=24 | 9x3=27 |
| 2x4= 8 | 3x4=12 | 4x4=16 | 5x4=20 | 6x4=24 | 7x4=28 | 8x4=32 | 9x4=36 |
| 2x5=10 | 3x5=15 | 4x5=20 | 5x5=25 | 6x5=30 | 7x5=35 | 8x5=40 | 9x5=45 |
| 2x6=12 | 3x6=18 | 4x6=24 | 5x6=30 | 6x6=36 | 7x6=42 | 8x6=48 | 9x6=54 |
| 2x7=14 | 3x7=21 | 4x7=28 | 5x7=35 | 6x7=42 | 7x7=49 | 8x7=56 | 9x7=63 |
| 2x8=16 | 3x8=24 | 4x8=32 | 5x8=40 | 6x8=48 | 7x8=56 | 8x8=64 | 9x8=72 |
| 2x9=18 | 3x9=27 | 4x9=36 | 5x9=45 | 6x9=54 | 7x9=63 | 8x9=72 | 9x9=81 |



실습문제

7. 피타고라스 정리는 직각 삼각형에서 직각을 낀 두 변을 a, b 라 하고 빗변을 c 라 하면, $a^2 + b^2 = c^2$ 의 수식이 성립한다는 것이다. 각 변의 길이가 100보다 작은 삼각형 중에 피타고라스 정리가 성립하는 삼각형의 각 변의 길이를 출력하고 모두 몇 개인지 출력하라

예) 3 4 5

4 3 5

...

피타고라스 정리를 만족하는 삼각형의 수 : 104 개



08. 무한 루프와 break, continue

- break : 루프를 강제로 빠져 나올 때 사용하는 문장
- continue : 반복문의 처음으로 다시 돌아가게 한다

```
while True :  
    if 조건 :  
        break           # 반복을 중단한다.  
    if 조건 :  
        continue       # 다음 반복을 시작한다.
```




08. 무한 루프와 break, continue

```
# break.py
while True :
    light = input("신호등 색상을 입력하시오. ")
    if light == 'green' :
        break
print('전진!!!...')
```

```
# continue.py
for i in range(1, 11) :
    if i%3 == 0 :
        continue
    print(i, end=' ')
```



08. 무한 루프와 break, continue

➤ 암호체크

```
password = ""
while password != 'pythonisfun':
    print('continue test...')
    password = input("암호를 입력하세요 : ")
    if password != 'pythonisfun':
        continue
    print('Good morning!!')
    print('Have a nice day~~')
```



실습문제

8. 학생들의 성적이 리스트 자료로 있다. 성적을 체크해서 합격한 학생의 결과만 출력하는 프로그램. 60점 이상 합격 (continue문 사용)

예) scores = [90, 88, 34, 56, 67, 78, 88, 60, 59, 100]

1번 학생 축하합니다. 합격입니다.

2번 학생 축하합니다. 합격입니다.

5번 학생 축하합니다. 합격입니다.

6번 학생 축하합니다. 합격입니다.

7번 학생 축하합니다. 합격입니다.

8번 학생 축하합니다. 합격입니다.

10번 학생 축하합니다. 합격입니다.



실습문제

9. 문자열을 조사하여 알파벳 문자의 개수, 숫자의 개수, 스페이스의 개수를 출력하는 프로그램

예) 문자열을 입력하세요 : I have three apples. Phone_number : 123-456-7890

알파벳 문자의 개수 = 27

숫자 문자의 개수 = 11

스페이스 문자의 개수 = 7



실습문제

```
statement = input('문자열을 입력하세요 : ')
```

```
alphas = 0
```

```
digits = 0
```

```
spaces = 0
```

```
for c in statement:
```

```
    if c.isalpha():
```

```
        alphas = alphas + 1
```

```
    if c.isdigit():
```

```
        digits = digits + 1
```

```
    if c.isspace():
```

```
        spaces = spaces + 1
```

```
print('알파벳 문자의 개수 = ', alphas)
```

```
print('숫자 문자의 개수 = ', digits)
```

```
print('스페이스 문자의 개수 = ', spaces)
```



실습문제

10. 사용자로부터 '-'이 포함된 계좌 번호를 입력 받아 '-'을 삭제한 문자열 출력

예) 계좌번호를 입력하세요 : 321-03-235123

32103235123



실습문제. 주사위 합 확률

11. 2개의 주사위를 던져서 나오는 합의 확률을 계산한다. 사용자로 부터 주사위를 던지는 횟수를 입력 받아서 각 숫자의 합이 나오는 확률을 계산한다.

예) 주사위 반복 실험 횟수 : 100000

2 : 2805

3 : 5512

:

:

11 : 5667

12 : 2697