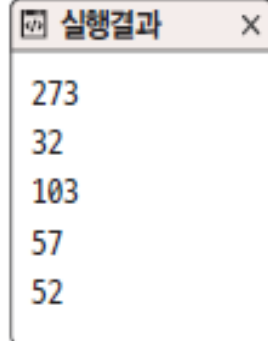


## for 반복문 : 리스트와 함께 사용하기

- 문자열, 리스트, 딕셔너리 등과 조합하여 for 반복문을 사용

```
for 반복자 in 반복할 수 있는 것:  
    코드
```

```
01  # 리스트를 선언합니다.  
02  array = [273, 32, 103, 57, 52]  
03  
04  # 리스트에 반복문을 적용합니다.  
05  for element in array:  
06      # 출력합니다.  
07      print(element)
```



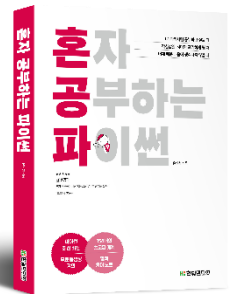
실행결과

```
273  
32  
103  
57  
52
```

## for 반복문 : 딕셔너리와 함께 사용하기

- for 반복문과 딕셔너리의 조합

```
for 키 변수 in 딕셔너리:  
    코드
```



## for 반복문 : 딕셔너리와 함께 사용하기

```
01 # 딕셔너리를 선언합니다.
02 dictionary = {
03     "name": "7D 건조 망고",
04     "type": "당절임",
05     "ingredient": ["망고", "설탕", "메타중아황산나트륨", "치자황색소"],
06     "origin": "필리핀"
07 }
08
09 # for 반복문을 사용합니다.
10 for key in dictionary:
11     # 출력합니다.
12     print(key, ":", dictionary[key])
```

### 실행결과

```
name : 7D 건조 망고
type : 당절임
ingredient : ['망고', '설탕', '메타중아황산나트륨', '치자황색소']
origin : 필리핀
```

## 확인문제

- 다음 빈칸을 채워서 numbers 내부에 들어있는 숫자가 몇 번 등장하는지를 출력하는 코드를 작성해보세요.

# 숫자는 무작위로 입력해도 상관 없습니다.

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

```
counter = {}
```

```
for number in numbers:
```

```
    # 빈칸에 코드를 작성하세요
```

# 최종 출력

```
print(counter)
```

실행결과

```
{1: 3, 2: 4, 6: 1, 8: 2, 4: 3, 3: 3, 9: 3, 5: 2, 7: 2}
```

# 확인문제

# 딕셔너리를 선언합니다.

```
character = {  
    "name": "기사",  
    "level": 12,  
    "items": {  
        "sword": "불꽃의 검",  
        "armor": "풀플레이트"  
    },  
    "skill": ["베기", "세게 베기", "아주 세게 베기"]  
}
```

# for 반복문을 사용합니다.

```
for key in character:
```

실행결과

```
name : 기사  
level : 12  
sword : 불꽃의 검  
armor : 풀플레이트  
skill : 베기  
skill : 세게 베기  
skill : 아주 세게 베기
```

## 확인문제

- 빈칸을 채워 키와 값으로 이루어진 각 리스트를 조합해 하나의 딕셔너리를 만들어 보세요.

```
# 숫자는 무작위로 입력해도 상관없습니다.  
key_list = ["name", "hp", "mp", "level"]  
value_list = ["기사", 200, 30, 5]  
character = {}
```

```
# 최종 출력  
print(character)
```

실행결과

```
{'name': '기사', 'hp': 200, 'mp': 30, 'level': 5}
```

## 확인문제

- 1부터 숫자를 하나씩 증가시키면서 더하는 경우를 생각해 봅시다. 몇을 더할 때 1000을 넘는지 구해 보세요. 그리고 그때의 값도 출력해보세요. 다음은 1000이 넘는 경우를 구한 예입니다.

1, 1 + 2 = 3, 1 + 2 + 3 = 6, 1 + 2 + 3 + 4 = 10...

```
limit = 10000
```

```
i = 1
```

```
# sum은 파이썬 내부에서 사용하는 식별자이므로 sum_value라는 변수 이름을 사용합니다.
```

```
print("{}를 더할 때 {}을 넘으며 그때의 값은 {}입니다.".format(i, limit, sum_value))
```

실행결과

142를 더할 때 10000을 넘으며 그때의 값은 10011입니다.