

제4장 4-2 딕셔너리와 반복문 마무리

▶ 3가지 키워드로 정리하는 핵심 포인트

1. ()는 키를 기반으로 여러 자료를 저장하는 자료형입니다.
2. ()는 딕셔너리 내부에서 값에 접근할 때 사용하는 것입니다.
3. ()은 딕셔너리 내부에 있는 각각의 내용을 의미합니다.

정답 : 1. 딕셔너리, 2. 키, 3. 값

▶ 확인문제

1. 다음 표에서 dict_a의 결과가 나오도록 빈칸을 채워보세요.

dict_a의 값	dict_a에 적용할 코드	dict_a의 결과
{}	()	{"name": "구름"}
{"name": "구름"}	()	{}

정답 : dict_a["name"]="구름", del dict_a["name"]

2. 딕셔너리와 리스트를 조합하면 다음 코드의 변수 pets처럼 다양한 정보를 축적할 수 있습니다. 빈칸에 반복문과 print() 함수를 조합해 보세요.

```
pets = [  
    {"name": "구름", "age": 5},  
    {"name": "초코", "age": 3},  
    {"name": "아지", "age": 1},  
    {"name": "호랑이", "age": 1}  
]
```

```
print("# 우리 동네 애완 동물들")  
()
```

정답

```
for pet in pets:  
    print(pet["name"], str(pet["age"]) + "살")
```

3. 다음 빈칸을 채워서 numbers 내부에 들어 있는 숫자가 몇 번 등장하는지를 출력하는 코드를 작성해 보세요.

```
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:  
    ()
```

```
print(counter)
```

정답

```
if number in counter:  
    counter[number] = counter[number] + 1  
else:  
    counter[number] = 1
```

4. 파이썬은 다음과 같은 방법으로 특정 값이 어떤 자료형인지 확인할 수 있습니다.

```
type("문자열") is str  
type([]) is list  
type({}) is dict
```

이를 활용해 다음 빈칸을 채우시오.

```
character = {  
    "name": "기사",
```

```
"level": 12,  
"items": {  
    "sword": "불꽃의 검",  
    "armor": "플플레이트"  
},  
"skill": ["베기", "세게 배기", "아주 세게 배기"]  
}
```

```
for key in character:  
    ()
```

정답

```
if type(character[key]) is dict:  
    for small_key in character[key]:  
        print(small_key, ":", character[key][small_key])  
elif type(character[key]) is list:  
    for item in character[key]:  
        print(key, ":", item)  
else:  
    print(key, ":", character[key])
```