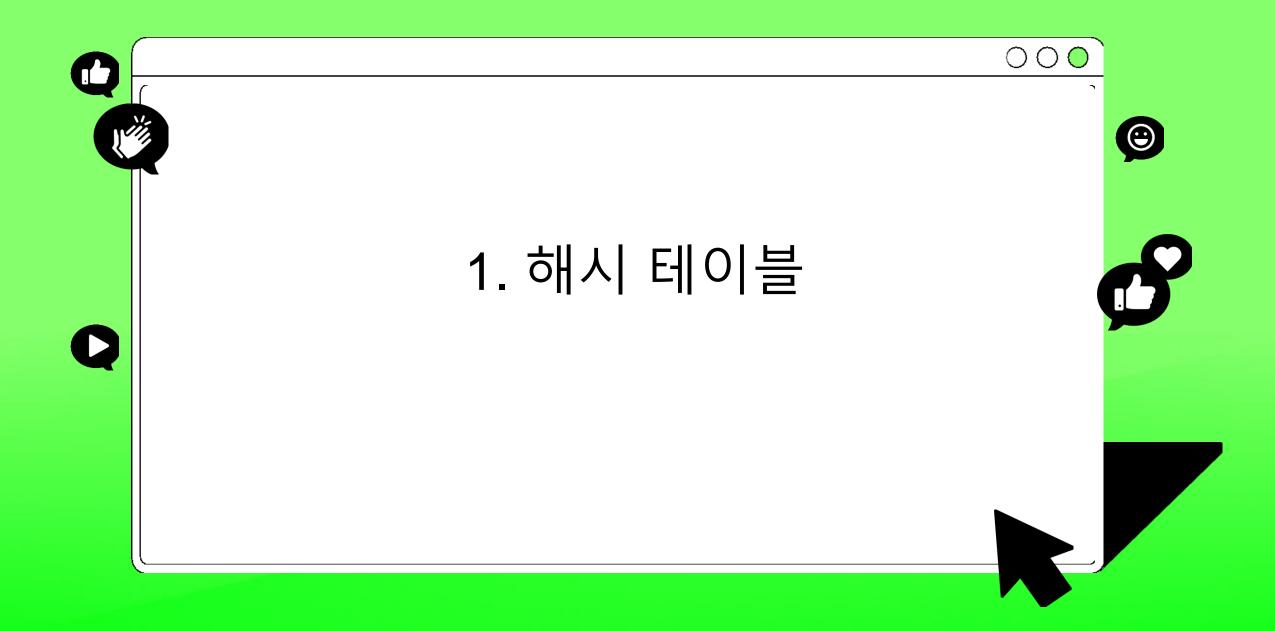


2. 딕셔너리 기본 문 법 3. 딕셔너리 메서드







파이썬에는 딕셔너리(dict) 자료구조가 내장 되어 있다.

Non-sequence & Key-Value

```
"name": "kyle",
"gender": "male",
"address": "Seoul",
}
```

Key는 immutable(변경 불가



Key: Value가 저장되는 원리가 무엇일까? 일단 리스트를 이용해 Key: Value를 저장해보자.

```
Key Value

( 210701, "haley" )

( 211209, "alex" )

( 210218, "justin" )

( 210405, "kyle" )
```

List

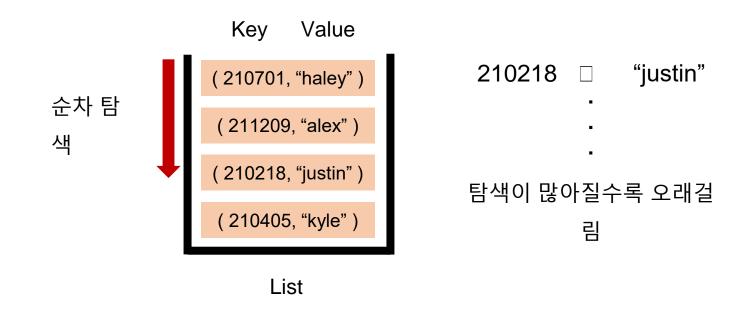


Key: Value가 저장되는 원리가 무엇일까? 일단 리스트를 이용해 Key: Value를 저장해보자.



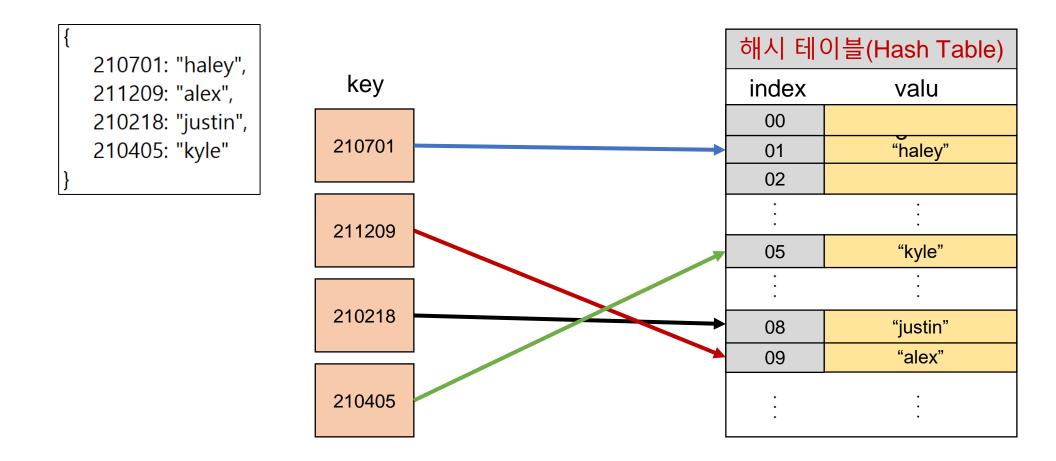


Key: Value가 저장되는 원리가 무엇일까? 일단 리스트를 이용해 Key: Value를 저장해보자.



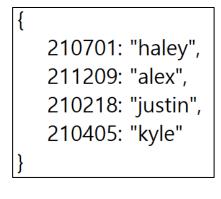


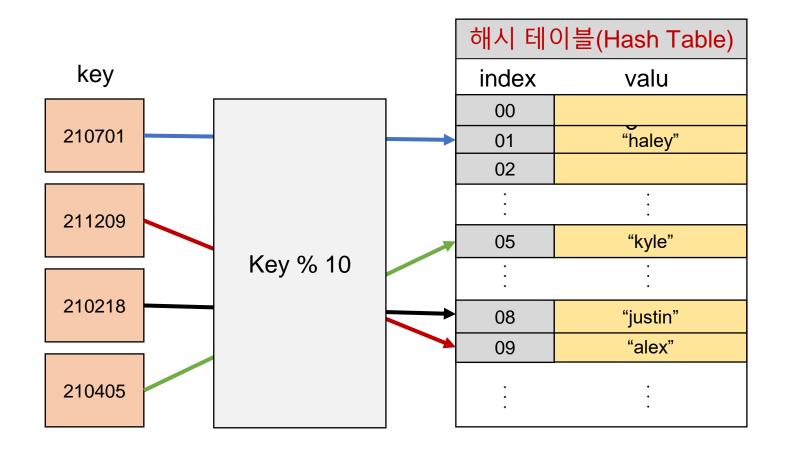
딕셔너리는 해시 테이블(Hash Table)을 이용하여 Key: Value를 저장





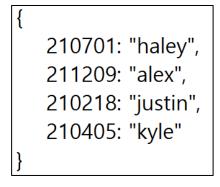
딕셔너리는 해시 테이블(Hash Table)을 이용하여 Key: Value를 저장

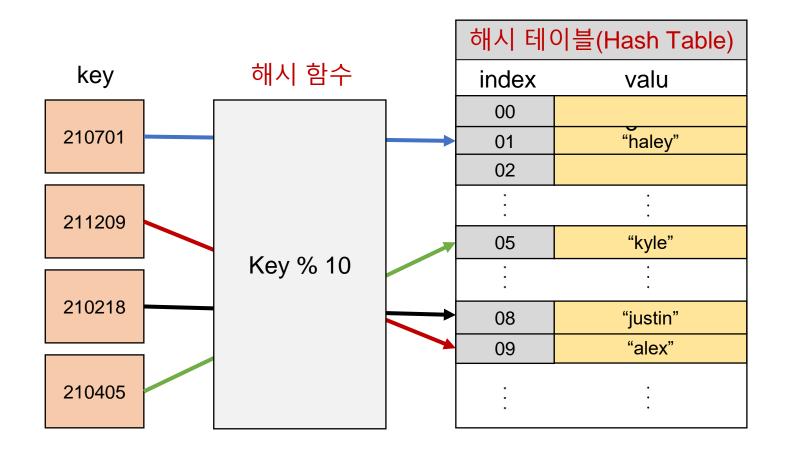






딕셔너리는 해시 테이블(Hash Table)을 이용하여 Key: Value를 저장



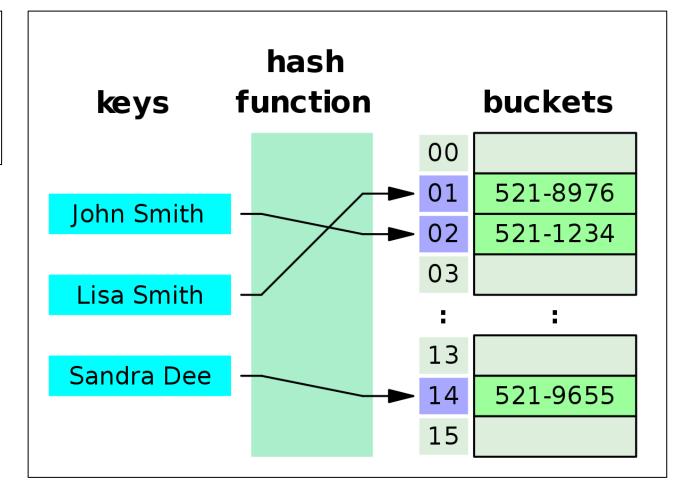




해시 함수 : 임의 길이의 데이터를 고정 길이의 데이터로 매핑하는 함수

해시 : 해시 함수를 통해 얻어진 값

{
 "John Smith": "521-1234",
 "Lisa Smith": "521-8976",
 "Sandra Dee": "521-9655"
}



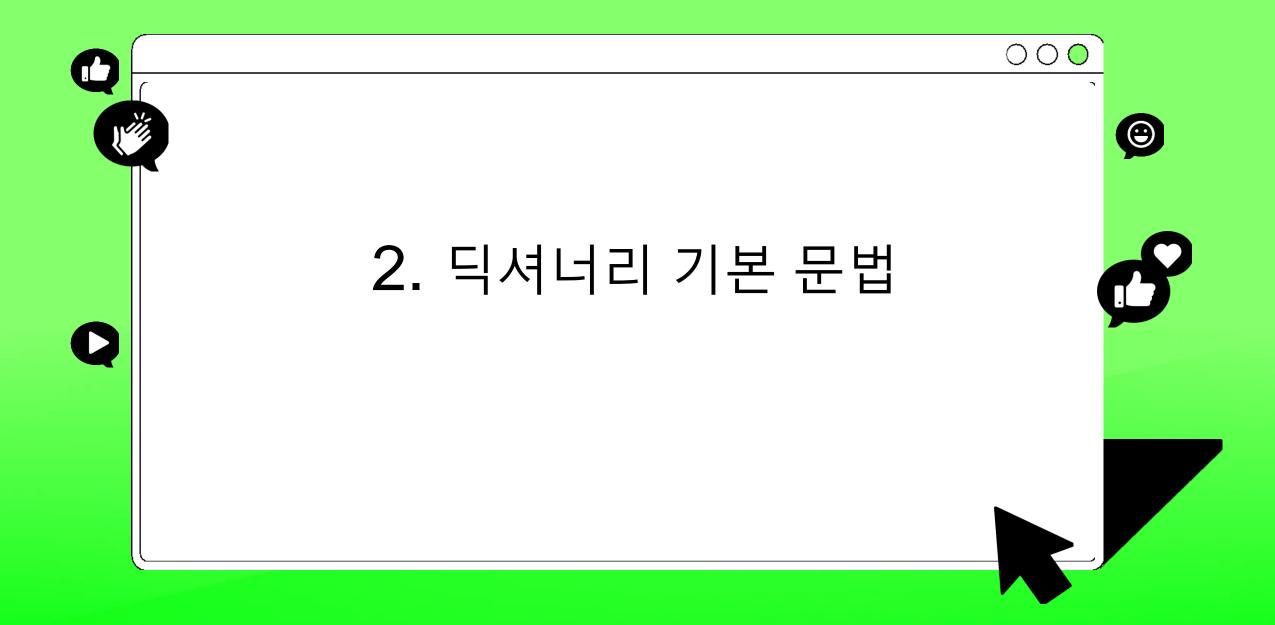


파이썬의 딕셔너리(Dictionary)는

해시 함수와 해시 테이블을 이용하여

삽입, 삭제, 수정, 조회 연산의 속도가 리스트보다 빠르다.







기본적인 딕셔너리 사용법 (선언)

```
변수 = { key1: value1, key2: value2 ... }
```

```
a = {
    "name": "kyle",
    "gender": "male",
    "address": "Seoul"
}
print(a)
{'name': 'kyle', 'gender': 'male', 'address': 'Seoul'}
```



기본적인 딕셔너리 사용법 (삽입/수정)

딕셔너리[key] = value

내부에 해당 key가 없으면 삽입, 있으면 수정

```
a = {
    "name": "kyle",
    "gender": "male",
    "address" "Seoul"
a["iob"] = "coach"
print(a)
{'name': 'kyle', 'gender': 'male', 'address': 'Seoul', 'job': 'coach'}
```

```
a = {
   "name" "kyle",
   "gender": "male",
   "address" "Seoul"
a["name"] = "justin"
print(a)
```

{'name': 'justin', 'gender': 'male', 'address': 'Seoul'}



기본적인 딕셔너리 사용법 (삭제)

딕셔너리.pop(key)

내부에 존재하는 key에 대한 value 삭제 및 반환, 존재하지 않는 key에 대해서는 KeyError 발생

```
a = {
    "name": "kyle",
    "gender": "male",
    "address": "Seoul"
}
gender = a.pop("gender")
print(a)
print(gender)
{'name': 'kyle', 'address': 'Seoul'}
male
```

```
a = {
    "name": "kyle",
    "gender": "male",
    "address": "Seoul"
}
phone = a.pop("phone")
print(a)
print(phone)
```



기본적인 딕셔너리 사용법 (삭제)

딕셔너리.pop(key, default)

두 번째 인자로 default(기본)값을 지정하여 KeyError 방지 가능

```
a = {
    "name": "kyle",
    "gender": "male",
    "address": "Seoul"
}

phone = a.pop("phone", "010-1234-5678")

print(a)
print(phone)

{'name': 'kyle', 'gender': 'male', 'address': 'Seoul'}
010-1234-5678
```



기본적인 딕셔너리 사용법 (조회)

key에 해당하는 value 반환

딕셔너리[key]

```
a = {
    "name": "kyle",
    "gender": "male",
    "address": "Seoul",
}
print(a["name"])
```

kyle

딕셔너리.get(key)

```
a = {
    "name": "kyle",
    "gender": "male",
    "address": "Seoul",
}
print(a.get("name"))
```

kyle



기본적인 딕셔너리 사용법 (조회)

딕셔너리[key]

VS

딕셔너리.get(key,

```
a = {
    "name": "kyle",
    "gender": "male",
    "address": "Seoul",
}
print(a["phone"])
```

```
KeyError
~#AppData#Local#Temp#ipyker
5 }
6
----> 7 print(a["phone"])
```

KeyError: 'phone'

```
a = {
    "name": "kyle",
    "gender": "male",
    "address": "Seoul",
}
print(a.get("phone"))
```

None

```
a = {
        "name": "kyle",
        "gender": "male",
        "address": "Seoul",
}
print(a.get("phone", "없음"))
```

없음



딕셔너리 기본 문법 정리

선언 변수 = { key1: value1, key2: value2 ... }

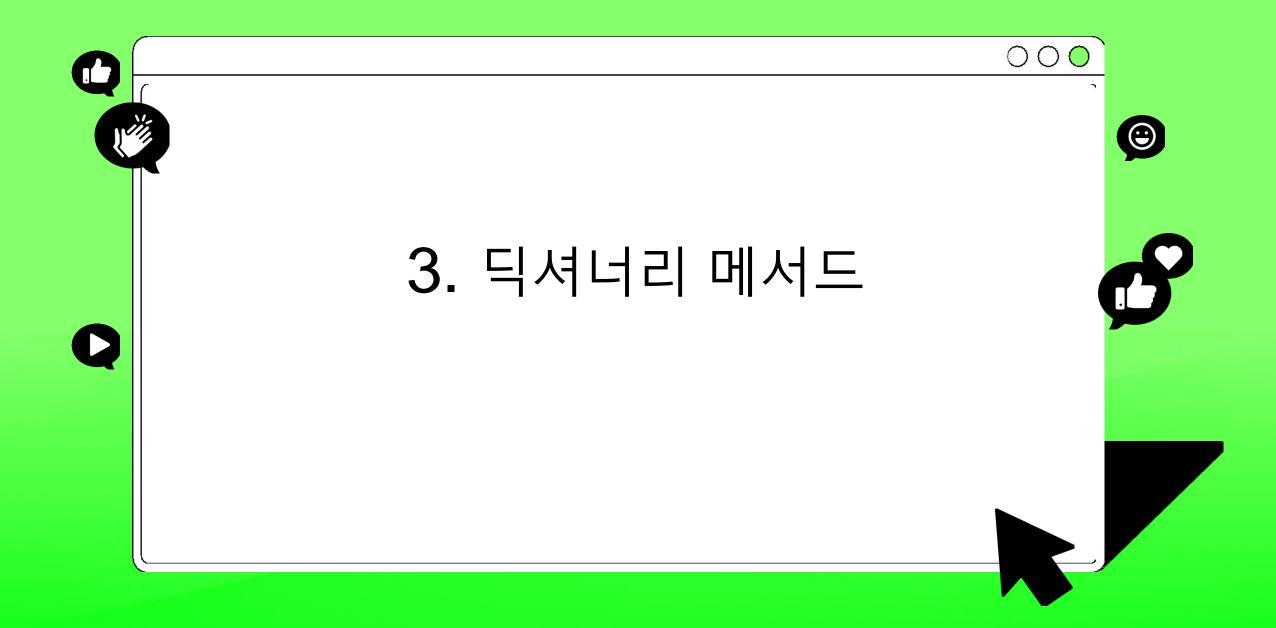
삽입/수 딕셔너리[key] = 정 value

삭제 딕셔너리.pop(key, default)

조회 딕셔너리[key]

딕셔너리.get(key, default)







- 1) .keys()
- 2) .values()
- 3) .items()



1) .keys()

딕셔너리의 key 목록이 담긴 dict_keys 객체 반환

```
a = {
    "name": "kyle",
    "gender": "male",
    "address": "Seoul",
}
print(a.keys())
dict_keys(['name', 'gender', 'address'])
```

```
a = {
    "name": "kyle",
    "gender": "male",
    "address": "Seoul",
}

for key in a.keys():
    print(key)
```

name gender address

```
a = {
    "name": "kyle",
    "gender": "male",
    "address": "Seoul",
}

for key in a:
    print(key)
```

name gender address



2) .values()

딕셔너리의 value 목록이 담긴 dict_values 객체 반환

```
a = {
    "name": "kyle",
    "gender": "male",
    "address": "Seoul",
}
print(a.values())
dict_values(['kyle', 'male', 'Seoul'])
```

```
a = {
    "name": "kyle",
    "gender": "male",
    "address": "Seoul",
}

for value in a.values():
    print(value)
```

kyle male Seoul



3) .items()

딕셔너리의 (key, value) 쌍 목록이 담긴 dict_items 객체 반환

```
a = {
    "name": "kyle",
    "gender": "male",
    "address": "Seoul",
}
print(a.items())
dict_items([('name', 'kyle'), ('gender', 'male'), ('address', 'Seoul')])
```

```
a = {
    "name": "kyle",
    "gender": "male",
    "address": "Seoul",
}

for item in a.items():
    print(item)

('name', 'kyle')
('gender', 'male')
('address', 'Seoul')
```

```
a = {
    "name": "kyle",
    "gender": "male",
    "address": "Seoul",
}

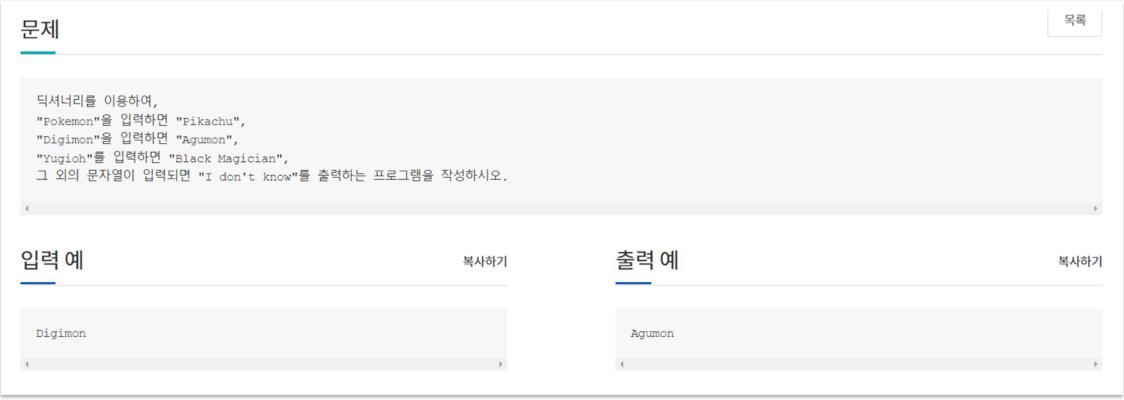
for key, value in a.items():
    print(key, value)

name kyle
gender male
```

address Seoul



딕셔너리 활용 연습 - 1



http://jungol.co.kr/bbs/board.php?bo_table=pbank&wr_id=4372&sca=pyd0



딕셔너리 활용 연습 - 2



http://jungol.co.kr/bbs/board.php?bo_table=pbank&wr_id=4373&sca=pyd0



딕셔너리 활용 연습 - 3



http://jungol.co.kr/bbs/board.php?bo_table=pbank&wr_id=4380&sca=pyd0