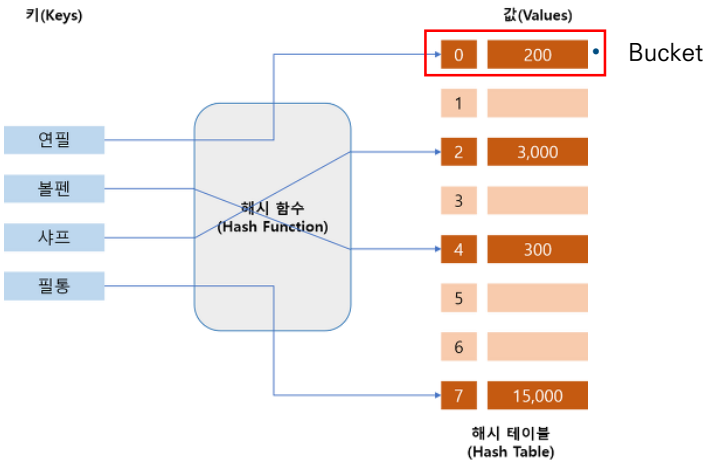
# HASH

## Hash

**Key : Value**

- 데이터를 다루는 기법
- 검색과 저장이 빠름
- key값이 배열의 인덱스로 변환




## 장 · 단점

- 해싱 된 키(Hash Key)를 가지고 배열의 인덱스로 사용하기 때문에 삽입, 삭제, 검색이 매우 빠르다.
- 해시 함수(Hash Function)를 사용하는데 추가적인 연산이 필요하다.
- 해시 테이블(Hash Table)의 크기가 유한적이고 해시 함수(Hash Function)의 특성상 해시 충돌(Hash Collision)이 발생할 수 밖에 없다.
- 충돌이 없거나 적으면 O(1)의 상수 시간에 가까워지고, 충돌이 발생하면 할수록 성능은 점점 O(n)에 가까워진다.
- Simple uniform hash : 충돌이 적은 좋은 해시 함수, 각각의 해시값들은 서로 연관성을 가지지 않고 독립성으로 생성될 것.

## Hash 사용

### Python

```
#d1 = {"apple", 1, “banana", 3}
#d1 = {key1:value1 , key2,value2 ….}
```

### JAVA

```
Map<String, Integer> fruits = new HashMap<>();
fruits.put("apple", 1);
```

### JS

```
Const m1 = new Map();
m1.set("apple", 1);
```

### JSON

```
[  {"name":"홍길동", "age":12},
{"name":"김병만", "age":32}   ]
```

- SHA256 해시 – 온라인 변환기 https://www.convertstring.com/ko/Hash/SHA256
- https://ato-planet.com/media / https://ato-planet.com/assets/json/mediaList.json / https://ato-planet.com/assets/js/media.js
- 비밀번호 저장 _ BCryptPasswordEncoder
- Hash vs list
- https://m.blog.naver.com/PostView.nhn?blogId=myca11&logNo=221373872927&proxyReferer=https:%2F%2Fwww.google.com%2F