

REDIS BASICS

- **HOW TO STORE STRINGS IN REDIS.**

1. Redis is used to store JSON files.
2. It only contains key: "value" pair, where value is a string.
3. To set a key we use, SET key(name) value (key value). For example: SET name Aditya
4. To get a key which is stored in the memory, we use GET key(name). For example: GET name
5. To delete a key, we use DEL key(name). For example: DEL key(name).
6. To check the number of keys stored, we use KEYS *.
7. To check if the key exists or not, we use, EXISTS key(name)
8. To delete all the keys, we use FLUSHALL.
9. To check the expiry time of the key, we use, ttl key(name).
Ttl(time to live).
10. To set the expiry time of the key, we use, expire key(name) time (in seconds).
11. To set the expiry of the key on the stage of declaring the key, we use, setex key(name) time (in seconds) value
12. To clear the console, we write clear.

```
127.0.0.1:6379> SET name rachit
OK
127.0.0.1:6379> GET name
"rachit"
127.0.0.1:6379> DEL name
(integer) 1
127.0.0.1:6379> KEYS *
(empty array)
127.0.0.1:6379> EXISTS name
(integer) 0
127.0.0.1:6379> FLUSHALL
OK
127.0.0.1:6379> ttl name
(integer) -2
127.0.0.1:6379> expire name 10
(integer) 0
127.0.0.1:6379> setex name 10 rachit
```



- **TO MAKE AN ARRAY IN REDIS**

1. To make an array we use lpush key(name) element (Aditya). This lpush operation adds the element (Aditya) in the beginning of the array.
2. To push the element at the end of the array we use rpush operation.
3. To delete an element from the beginning of the array we use lpop key(name).
4. To delete an element from the end of the array we use rpop key(name).
5. To print the array items, we use lrange key(name) startValue(0) endValue(-1).
6. To check the keys present we use keys *.

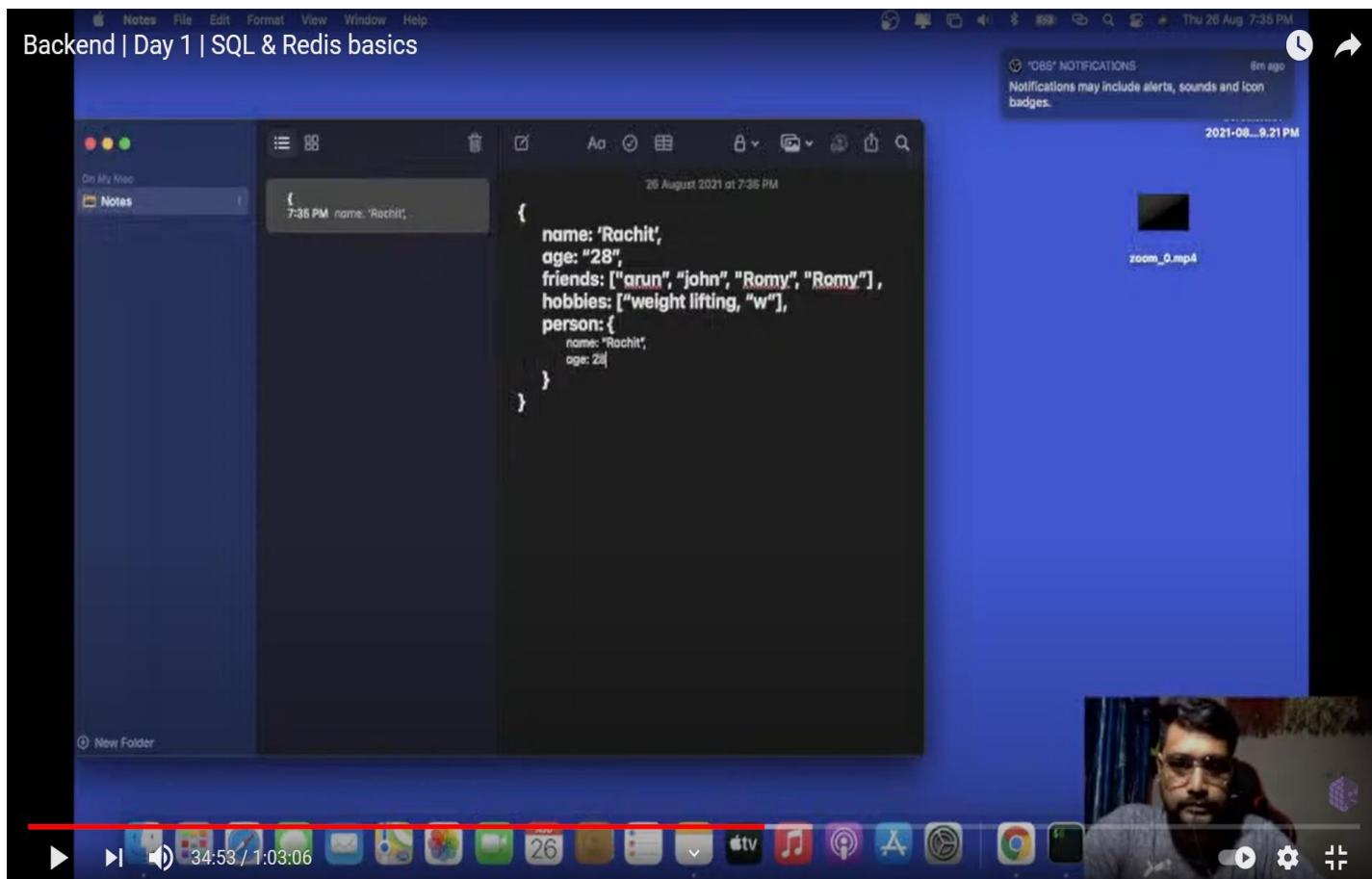
```
adityasaini_025@DESKTOP-LETFFQ: ~  
127.0.0.1:6379> lpush name aditya  
(integer) 1  
127.0.0.1:6379> lpush name dhruv  
(integer) 2  
127.0.0.1:6379> rpush name aryan  
(integer) 3  
127.0.0.1:6379> lpush name vinit  
(integer) 4  
127.0.0.1:6379> rpush name soumya  
(integer) 5  
127.0.0.1:6379> lrange name 0 -1  
1) "vinit"  
2) "dhruv"  
3) "aditya"  
4) "aryan"  
5) "soumya"  
127.0.0.1:6379> lpop name  
"vinit"  
127.0.0.1:6379> rpop name  
"soumya"  
127.0.0.1:6379> lpop name  
"dhruv"  
127.0.0.1:6379> keys *  
1) "hobbies"  
2) "name"  
127.0.0.1:6379> del hobbies  
(integer) 1  
127.0.0.1:6379> keys *  
1) "name"  
127.0.0.1:6379>
```

- **TO MAKE A SET IN REDIS**

1. To declare a set in Redis, we use keyword SADD. For example: sadd hobbies "Body Building", sadd hobbies skating, sadd hobbies coding. When we have to the space in between the string I.e., Body Building we put it in double quotes else it will automatically take it as a string.
2. In set, no two similar elements can be added as it will show (integer) 0 which means it cannot be added in set. For example, we cannot add coding twice in a set.
3. To print the elements in the set, we use SMEMBERS keyword. This prints the elements of the set. For example, smembers hobbies.

```
adityasaini_025@DESKTOP-LETFFFQ: ~  
127.0.0.1:6379> sadd hobbies "Body Building"  
(integer) 1  
127.0.0.1:6379> sadd hobbies coding  
(integer) 1  
127.0.0.1:6379> sadd hobbies skating  
(integer) 1  
127.0.0.1:6379> sadd hobbies coding  
(integer) 0  
127.0.0.1:6379> smembers hobbies  
1) "coding"  
2) "Body Building"  
3) "skating"  
127.0.0.1:6379> keys *  
1) "hobbies"  
127.0.0.1:6379> _
```

- TO ADD AN OBJECT IN REDIS (HASHES), EK KEY K ANDAR EK AUR KEY



1. To add an object inside another object we use objects (also called as hashes). To add a key inside the object we use HSET keyword (as we have used set, get etc. in strings, similarly we add prefix H. So, to add a key, hset key(person) field(name) value (Aditya).
2. To print the key and its value we use HGET keyword. For example, hget person name.
3. To delete the key, we use HDEL keyword to delete an item from the object. For example,