



```
SET resource:lock "Redis Demo"  
EXPIRE resource:lock 120
```

This causes the key *resource:lock* to be deleted in 120 seconds. You can

```
TTL resource:lock => 113  
// after 113s  
TTL resource:lock => -2
```

The -2 for the TTL of the key means that the key does not exist (anymore)

```
SET resource:lock "Redis Demo 1"  
EXPIRE resource:lock 120  
TTL resource:lock => 119  
SET resource:lock "Redis Demo 2"  
TTL resource:lock => -1
```

The SET command is actually able to accept further arguments in order to perform an atomic operation:

```
SET resource:lock "Redis Demo 3" EX 5  
TTL resource:lock => 5
```

It is also possible to cancel the time to live of a key removing the expire

```
SET resource:lock "Redis Demo 3" EX 5  
PERSIST resource:lock  
TTL resource:lock => -1
```



- NoSQL-Datenbank zur Speicherung strukturierter Key/Value-Paare
- Darüber hinaus Unterstützung für viele weitere Datentypen wie
 - **Binärstrings**
 - Listen
 - Sortierte und unsortierte Mengen
 - Hashes (Objekte bzw. Dictionaries in Python)
 - Bit-Arrays
 - **Streams**
- Alle Daten liegen komplett Hauptspeicher (In-Memory-Datenbank)
- Die Datenbankinhalte werden regelmäßig aber auch auf „Platte“ gesichert, damit sie bei einem Serverneustart nicht verloren gehen.