SET

```
SET <key> <value> [EX|FX|NX|XX]
```

Sowie der Schlüssel als auch der Wert bilden je eine beliebige Zeichenkette. Sind Leerzeichen enthalten, müssen Hochkomma verwendet werden, wobei nicht zwischen " und ' unterschieden wird.
EX -> Anzahl der Sekunden die der Datensatz (DS) erhalten bleibt
PX -> Anzahl der Millisekunden die der DS erhalten bleibt
NX -> Setzt DS nur, wenn der Schlüssel noch nicht existiert
XX -> Setzt DS nur, wenn der Schlüssel schon existiert

GET

```
GET <key>
```

Der Schlüssel ist wieder eine beliebige Zeichenkette, kann dieser nicht gefunden werden, wird die spezielle Kette 'nil' zurückgegeben.

EXISTS

```
EXISTS <key>
```

Überprüft, ob der angegebene Schlüssel in der DB vorhanden ist.
Gibt 1 für WAHR und 0 für FALSCH zurück.

Hash

HSET <hashname> <key> <value>
HGETALL <hashname>
HMSET <hashname> <key> <value> <key> <value> [<key> <value> ...]
HGET <hashname> <key>
HINCRBY <hashset> <key> <count>
HDEL <hashset> <key>
HDECRBY <hashset> <key> <count>
Schlüssel und Wert sind beide Zeichenketten.

DEL

```
DEL <key>
```

Ähnlich wie EXISTS, nur dass dabei der DS gelöscht wird, wenn er gefunden wird. Rückgabewerte sind dieselben.

INCR | INCRBY

```
INCR <key>
INCRBY <key> <count>
```

Erhöht den Wert des Schlüssels um eins, oder im Fall von INCRBY um die beigefügte Zahl.
Ist der Wert keine ganzzahlige Zahl, wird ein Fehler ausgegeben.
Existiert der Schlüssel noch nicht, wird er neu erstellt. Bei INCR wäre der Wert dann automatisch 1.
8ung: Stellt eine *atomic operation* dar -> thread-safe

Lists

RPUSH <key> <value> [<value> ...]
LPUSH <key> <value> [<value> ...]
LLEN <key>
LRANGE <key> <start> <end>
LPOP <key>
RPOP <key>
RPUSH fügt den Wert (od. die Werte) am Ende der Liste ein.
LPUSH fügt den Wert (od. die Werte) am Anfang der Liste ein.
LLEN gibt die Länge der Liste zurück. (Anzahl der Werte)
LRANGE gibt alle Werte zwischen <start> und <end> zurück. <end> kann auch -1 und -2 sein, wobei -1 für das Ende der Liste und -2 für den vorletzten Eintrag in der Liste steht.
LPOP entfernt den ersten Eintrag der Liste und gibt dessen Wert zurück.
RPOP entfernt den letzten Eintrag der Liste und gibt dessen Wert zurück.

EXPIRE | TTL | PEXPIRE | PTTL

```
EXPIRE <key> <seconds>
TTL <key>
```

EXPIRE setzt die Sekunden, die eine Schlüssel existieren darf, läuft die Zeit ab, wird er (der gesamte DS) gelöscht.
TTL gibt die Sekunden zurück, welche ein Schlüssel noch existiert, oder -1 falls er kein "Ablaufdatum" besitzt, oder -2 falls der Schlüssel nicht existiert.
PEXPIRE und PTTL haben die selbe Funktionalität, nur mit Millisekunden.

By **deudaz15**
cheatography.com/deudaz15/

Published 9th November, 2020.
Last updated 9th November, 2020.

PERSIST

```
PERSIST <key>
```

Entfernt das Ablaufdatum für einen DS.

Set

SADD <key> <value> [<value> ...]
SREM <key> <value>
SISMEMBER <key> <value>
SMEMBERS <key>
SUNION <key> <key>
SPOP <key> <count>
Werte können nur einmal vorkommen.
SISMEMBER gibt 1 zurück, wenn der Wert enthalten ist und 0 wenn nicht.
SMEMBERS gibt alle Elemente des Sets zurück.
SUNION kombiniert zwei Sets und gibt deren Elemente zusammen zurück.
SPOP eliminiert <count> Elemente vom Set und gibt diese zurück.

Sorted Set

Ähnlich wie Set
statt SADD <key> <value> -> ZADD <key> <score> <value>
Mittels Score wird das Set geordnet.