Redis cheat sheet v2.5

Test

```
$ redis-cli
127.0.0.1:6379> ping
127.0.0.1:6379> auth mypassword
127.0.0.1:6379> set somekey somevalue
127.0.0.1:6379> get somekey
```

Common commands

Meta

- `KEYS *` (or `KEYS` "pattern with `*`'s" – `*`'s are wildcards)
- `TYPE key`
- `DEL key`

Atomic

- SET/MSET key val (key2 val2 key3 val3. . .)
- GET/MGET key (key2 key3. . .)
- EXISTS key
- GETRANGE key i1 i2 (`substr`; indices inclusive; neg indices from end)
- For numeric only:
  - INCR/DECR key
  - INCRBY/DECRBY key num
  - INCRBYFLOAT key num

Sets

- SADD setname elem1 elem2 elem3. . .
- SREM setname elem
- SMEMBERS setname
- SCARD setname (# of elements)
- SDIFF setname1 setname2
- SINTER setname1 setname2
- SUNION setname1 setname2
- SPOP setname (removes a random element)
- SRANDMEMBER setname (returns a random element)

(Linked) Lists

- LPUSH/RPUSH listname val1 val2 val3. . .
- LPOP/RPOP listname
- LRANGE listname i1 i2 (to view elements of list)
- LTRIM listname i1 i2 (to crop list to only sublist of values)
- LLEN listname
- LINDEX listname index (get by index)
- LSET listname index val (set by index)
- LREM listname cnt val (remove (first cnt matches) by val)

Hashes (dicts)

- `HSET` hashname key value
- `HGET` hashname key
- `HEXISTS` hashname key
- `HLEN` hashname (number of key/val pairs)
- `HKEYS` hashname (gets all keys)
- `HVALS` hashname (gets all values)
- `HGETALL` hashname (gets all keys and values in one alternating list)

Sorted sets

A sorted set is a set whose members have numeric "scores," and which can be queried in various ways to take advantage of this. By default, as in golf, the item with the top/best "rank" (*i.e.*, rank 0) is the one with the *lowest* score, and the bottom rank (*i.e.*, rank n-1, where n is the cardinality of the set) is the one with the highest score. Adding the letters "`REV`" after the `Z` for the range commands will **rev**erse this, giving rank 0 to the item with the *highest* score instead.

- `ZADD` ssname score elem (note: score comes before element)
- `ZSCORE` ssname elem (get score for element)
- `ZRANK` ssname elem (get rank of element)
- `ZRANGEBYSCORE` ssname score score (get element(s) for score)
- `ZINCRBY` ssname increment elem (increment score for an element)
- `ZCARD` ssname (# elements in sset)
- `ZCOUNT` ssname min max (# elements with scores in range)
- `ZRANGE/ZRANGEBYSCORE` ssname min max (range of elements by rank or score) (can also include WITHSCORES as a third argument to get the scores as well as the elements)
- `ZREM` ssname elem
- `ZREMRANGEBYRANK/ZREMRANGEBYSCORE` ssname min max (bulk remove)

Tips on combining data structures

If you want a bunch of objects that are hashes (say, a bunch of baseball players), best way is to make the top-level key a naming convention, like so:

```
127.0.0.1:6379> hmset player:ruth uni 3 pos OF team yanks
127.0.0.1:6379> hmset player:gwynn uni 19 pos OF team padres
```

You can retrieve the keys with `KEYS player:*`.

Python support

Connecting

Install `redis` package (with `pip`), then:

```
import redis

r = redis.Redis(password="yourpassword")    # Can set hostname, port, etc
```

....or....

```
r = redis.StrictRedis(password="yourpassword", charset="utf-8",
    decode_responses=True)   # If you want autocoded responses
```

Example usage

```
r.keys()                              <- get all keys in entire database
r.keys("person:*")                    <- get all keys that start with a prefix
r.exists("name")                      <- True/False
r.set("name","stephen")
r.get("name")                         <- "stephen"
r.getrange("name",2,3)                <- "tep"

r.sadd("davieses","Stephen")        <- add one/multiple members to a set
r.sadd("davieses","Rae","Lizzy","TJ","Johnny")

r.rpush("holidays","halloween")     <- add one/multiple elems to a list
r.rpush("holidays","thanksgiving","christmas","new years")
r.lrange("holidays",1,3)              <- retrieve slice of list (inclusive)
r.ltrim("holidays",1,3)              <- permanently trim list (inclusive)

r.hset("soccer","ball","round")
r.hget("soccer","ball")
r.hkeys("soccer")                     <- get all keys of hash
r.hvalues("soccer")                   <- get all values of hash

r.zadd("golf",{'tiger':63,'xander':67,'phil':66})
r.zscore("golf",'phil')               <- get phil's golf score
r.zrank("golf",'phil')                <- get phil's rank (from lowest, from 0)
r.zincrby("golf",3,'xander')        <- incr xander's golf score by 3
r.zrange("golf",3,6)                  <- get players ranked 3 through 6
r.zrange("golf",0,-1)                 <- get all players, in rank order
r.zrange("golf",0,-1,withscores=True)   <- include scores in the above
```