Redis Commands

This is not a comprehensive list of every single redis command, however we cover most of these. This will help you as a reference. Otherwise feel free to look at http://redis.io/commands for every command available.

Installation

Strongly recommended to use a **UNIX** based system. Windows has a download, however it is not official and not extremely stable, but you can still use it for learning.

Install in Ubuntu/Debian:
$ sudo apt-get install redis-server
$ sudo service redis-server start

Redis CLI
$ redis-cli

Monitor Redis
$ redis-cli monitor

Strings

Setting
> set name jesse
> set age 30

Getting
> get juicy
> get age

Appending
> append name " is my name!"

increment by 1
> incr age

increment by any
> incrby age 5

decrement by 1

> decr age

decrement by any
> decyby 3
Get string length
Command: strlen [key]
> strlen name

Set multiple values
Command: mset [key] [value] [..key] [..value]
> mset mammal whale insect spider

Set multiple values if key does not exists (nx)
Command: msetnx [key] [value] [..key] [..value]
> msetnx mammal whale insect spider

Keys

List all keys in a database
> keys *

Get the DataType of a Key
> type name
> type age

Delete a key
Command: del [key]
> del name

Does a key exists
Command: exists [key]
> exists name

Get Multiple Keys
Command: mget [key] [..key]
> mget name age

Set Expiration in UNIX epoch
> expire name 1504566531

Expire a key in a number of milliseconds
> pexpire name 100000

Check the Time to Live for an expiring Key
> ttl name

Remove Expiration from a key
> persist name

Name a key
> rename name person

Rename a key if the new one does not exist
> renamenx person someone

Hashes

Set a Hash
Command: hset [hash] [member] [value]
> hset people john 23
> hset people teddy 30

Set Multiple Hashes
Command: hmset [hash] [member] [value] [..member] [..value]
> hmset people barny 50 graham 55 chuck 60

Get Hash Data
> hgetall people

Get Single Hash Value Data
> hget people john

Get Multiple Hash Values
> hmget people barny graham

Delete a member from a hash
> hdel people john teddy

Check if a hash exists
> hexists people jake

Lists

```
# Create a new list (Or add to one)
> lpush colors red  (Prepends)
> rpush colors green  (Appends)

# Push an item to a list only if the list exists
> lpushx animals whale
> rpushx animals dog

# Set a List Item to a different value based on the Index
> lset colors 1 yellow

# Inserts an item into a list based of a value
Command: linsert colors [before|after] [value] [new_value]
> linsert colors before green yellow

# Get the List Length
> llen colors

# Remove the First or last item
> lpop colors (first item)
> rpop colors (last item)

# Get the Value of a List Index
> lindex colors 2

# Show all List Items
> lrange colors 0 -1

# Show List items with indexes from 0 to 5
> lrange colors 0 5
```

Sets

```
# Create/Add Set
Command: sadd [key] [value]
> sadd art pencil

# Create/Add Multiple to Set
Command: sadd [key] [value] [..value]
> sadd art charcoal paint oink

# Remove a set item based on value
```

```
> srem art oink

# Get all the members of a set
> smembers art
# Check if an item is a member
> sismember art pencil

# Get a random member
> srandmember art 2

# Remove a random member from a Set
> spop art

# See how many items ar ein a set
> scard art
```

Sorted Sets

```
# Add/Create Sorted Set
Command: zadd [key] [score] [member]
> zadd bowling 50 jane

# Add/Create Multiple Values to a Sorted Set
Command: zadd [key] [score] [member] [..score] [..member]
> zadd bowling 75 mark 100 tim 150 daniel

# Count how many items are in a Sorted Set
> zcard bowling

# Count how many items are in a Sorted Set between a value of scores
Command: zcount [key] [min] [max]
zcount bowling 100 200

zrange key start stop
zrange key start stop 'withscore'

zrank key member
zrem key member [..member]

zrevrange
zrevrangebyscore
zrevrank
```

zscore

HyperLogLog

```
> pfadd key element [..element]
> pfcount key [..key]
> pfmerge new_key key [..key] [..key]
```

Pub/Sub

```
(Three Terminals)
Terminal (1)
> subscribe chatroom

Terminal (2)
> subscribe chatroom

Terminal (3)
> publish chatroom hello
```

Terminal (1) * You can use unsubscribe from a Client Wrapper, Terminal is Ctrl+C to unsubscribe.
```
> unsubscribe chatroom

Terminal (3)
> publish chatroom How is life?
```

Transactions

```
# Begin a Transaction
> MULTI
> set what "A Series of Commands"
> set is "that require all of them to work"
> set this "in order to process a transaction"

# Complete Transaction (Always happens after MULTI is called)
> EXEC

# Cancel Transaction (Always happens after MULTI is called)
> DISCARD
```

Connection

```
# Ping the Server
> ping

# Select a Database (0 is the default)
> select 1

# Echo a String
> echo "Hello World"

# Leave Redis
> quit

# Require authorization to run commands
# This is under /etc/redis/redis.conf ' titled: requirepass your-password
# Requires a Redis Server Restart.
> auth password
```

Server

```
# Get server info
> info

# Set a Clients Connection name
> client setname samuel

# Get a Clients Connection name
> client getname

# Get a a of Client Connections
> client list

# Kill a Client by a Set ID
> client kill id samuel

# Kill a Client by UID (If a Client name was not set, this data will be in the > client list)
> client kill addr ip:port
```

Remove all values from the current database
> flushdb

Remove all values from all databases (Dangerous)
> flushall

Save a Snapshot of the Redis Database to an RDB file
> save

Save a Snapshot of the Redis Database to an RDB file in the background (Asynchronous Forked while other processes are still running)
> bgsave

See the last time the database was saved to an RDB file
> lastsave

Get the server time as a UNIX timestamp and Microseconds within the current timestamp
> time

Stop all Clients and force save
> shutdown save

Stop all Clients and prevent a save
> shutdown nosave

Where are RDB files saved?
$ cat /etc/redis/redis.conf | grep "rdb"