* [APPEND](http://redis.io/commands/append) key valueAppend a value to a key
* [AUTH](http://redis.io/commands/auth) passwordAuthenticate to the server
* [BGREWRITEAOF](http://redis.io/commands/bgrewriteaof)Asynchronously rewrite the append-only file
* [BGSAVE](http://redis.io/commands/bgsave)Asynchronously save the dataset to disk
* [BITCOUNT](http://redis.io/commands/bitcount) key [start] [end]Count set bits in a string
* [BITOP](http://redis.io/commands/bitop) operation destkey key [key ...]Perform bitwise operations between strings
* [BITPOS](http://redis.io/commands/bitpos) key bit [start] [end]Find first bit set or clear in a string
* [BLPOP](http://redis.io/commands/blpop) key [key ...] timeoutRemove and get the first element in a list, or block until one is available
* [BRPOP](http://redis.io/commands/brpop) key [key ...] timeoutRemove and get the last element in a list, or block until one is available
* [BRPOPLPUSH](http://redis.io/commands/brpoplpush) source destination timeoutPop a value from a list, push it to another list and return it; or block until one is available
* [CLIENT KILL](http://redis.io/commands/client-kill) ip:portKill the connection of a client
* [CLIENT LIST](http://redis.io/commands/client-list)Get the list of client connections
* [CLIENT GETNAME](http://redis.io/commands/client-getname)Get the current connection name
* [CLIENT PAUSE](http://redis.io/commands/client-pause) timeoutStop processing commands from clients for some time
* [CLIENT SETNAME](http://redis.io/commands/client-setname) connection-nameSet the current connection name
* [CONFIG GET](http://redis.io/commands/config-get) parameterGet the value of a configuration parameter
* [CONFIG REWRITE](http://redis.io/commands/config-rewrite)Rewrite the configuration file with the in memory configuration
* [CONFIG SET](http://redis.io/commands/config-set) parameter valueSet a configuration parameter to the given value
* [CONFIG RESETSTAT](http://redis.io/commands/config-resetstat)Reset the stats returned by INFO
* [DBSIZE](http://redis.io/commands/dbsize)Return the number of keys in the selected database
* [DEBUG OBJECT](http://redis.io/commands/debug-object) keyGet debugging information about a key
* [DEBUG SEGFAULT](http://redis.io/commands/debug-segfault)Make the server crash
* [DECR](http://redis.io/commands/decr) keyDecrement the integer value of a key by one
* [DECRBY](http://redis.io/commands/decrby) key decrementDecrement the integer value of a key by the given number
* [DEL](http://redis.io/commands/del) key [key ...]Delete a key
* [DISCARD](http://redis.io/commands/discard)Discard all commands issued after MULTI
* [DUMP](http://redis.io/commands/dump) keyReturn a serialized version of the value stored at the specified key.
* [ECHO](http://redis.io/commands/echo) messageEcho the given string
* [EVAL](http://redis.io/commands/eval) script numkeys key [key ...] arg [arg ...]Execute a Lua script server side
* [EVALSHA](http://redis.io/commands/evalsha) sha1 numkeys key [key ...] arg [arg ...]Execute a Lua script server side
* [EXEC](http://redis.io/commands/exec)Execute all commands issued after MULTI
* [EXISTS](http://redis.io/commands/exists) keyDetermine if a key exists
* [EXPIRE](http://redis.io/commands/expire) key secondsSet a key's time to live in seconds
* [EXPIREAT](http://redis.io/commands/expireat) key timestampSet the expiration for a key as a UNIX timestamp
* [FLUSHALL](http://redis.io/commands/flushall)Remove all keys from all databases
* [FLUSHDB](http://redis.io/commands/flushdb)Remove all keys from the current database
* [GET](http://redis.io/commands/get) keyGet the value of a key
* [GETBIT](http://redis.io/commands/getbit) key offsetReturns the bit value at offset in the string value stored at key
* [GETRANGE](http://redis.io/commands/getrange) key start endGet a substring of the string stored at a key
* [GETSET](http://redis.io/commands/getset) key valueSet the string value of a key and return its old value
* [HDEL](http://redis.io/commands/hdel) key field [field ...]Delete one or more hash fields
* [HEXISTS](http://redis.io/commands/hexists) key fieldDetermine if a hash field exists
* [HGET](http://redis.io/commands/hget) key fieldGet the value of a hash field
* [HGETALL](http://redis.io/commands/hgetall) keyGet all the fields and values in a hash
* [HINCRBY](http://redis.io/commands/hincrby) key field incrementIncrement the integer value of a hash field by the given number
* [HINCRBYFLOAT](http://redis.io/commands/hincrbyfloat) key field incrementIncrement the float value of a hash field by the given amount
* [HKEYS](http://redis.io/commands/hkeys) keyGet all the fields in a hash
* [HLEN](http://redis.io/commands/hlen) keyGet the number of fields in a hash
* [HMGET](http://redis.io/commands/hmget) key field [field ...]Get the values of all the given hash fields
* [HMSET](http://redis.io/commands/hmset) key field value [field value ...]Set multiple hash fields to multiple values
* [HSET](http://redis.io/commands/hset) key field valueSet the string value of a hash field
* [HSETNX](http://redis.io/commands/hsetnx) key field valueSet the value of a hash field, only if the field does not exist
* [HVALS](http://redis.io/commands/hvals) keyGet all the values in a hash
* [INCR](http://redis.io/commands/incr) keyIncrement the integer value of a key by one
* [INCRBY](http://redis.io/commands/incrby) key incrementIncrement the integer value of a key by the given amount
* [INCRBYFLOAT](http://redis.io/commands/incrbyfloat) key incrementIncrement the float value of a key by the given amount
* [INFO](http://redis.io/commands/info) [section]Get information and statistics about the server
* [KEYS](http://redis.io/commands/keys) patternFind all keys matching the given pattern
* [LASTSAVE](http://redis.io/commands/lastsave)Get the UNIX time stamp of the last successful save to disk
* [LINDEX](http://redis.io/commands/lindex) key indexGet an element from a list by its index
* [LINSERT](http://redis.io/commands/linsert) key BEFORE|AFTER pivot valueInsert an element before or after another element in a list
* [LLEN](http://redis.io/commands/llen) keyGet the length of a list
* [LPOP](http://redis.io/commands/lpop) keyRemove and get the first element in a list
* [LPUSH](http://redis.io/commands/lpush) key value [value ...]Prepend one or multiple values to a list
* [LPUSHX](http://redis.io/commands/lpushx) key valuePrepend a value to a list, only if the list exists
* [LRANGE](http://redis.io/commands/lrange) key start stopGet a range of elements from a list
* [LREM](http://redis.io/commands/lrem) key count valueRemove elements from a list
* [LSET](http://redis.io/commands/lset) key index valueSet the value of an element in a list by its index
* [LTRIM](http://redis.io/commands/ltrim) key start stopTrim a list to the specified range
* [MGET](http://redis.io/commands/mget) key [key ...]Get the values of all the given keys
* [MIGRATE](http://redis.io/commands/migrate) host port key destination-db timeout [COPY] [REPLACE]Atomically transfer a key from a Redis instance to another one.
* [MONITOR](http://redis.io/commands/monitor)Listen for all requests received by the server in real time
* [MOVE](http://redis.io/commands/move) key dbMove a key to another database
* [MSET](http://redis.io/commands/mset) key value [key value ...]Set multiple keys to multiple values
* [MSETNX](http://redis.io/commands/msetnx) key value [key value ...]Set multiple keys to multiple values, only if none of the keys exist
* [MULTI](http://redis.io/commands/multi)Mark the start of a transaction block
* [OBJECT](http://redis.io/commands/object) subcommand [arguments [arguments ...]]Inspect the internals of Redis objects
* [PERSIST](http://redis.io/commands/persist) keyRemove the expiration from a key
* [PEXPIRE](http://redis.io/commands/pexpire) key millisecondsSet a key's time to live in milliseconds
* [PEXPIREAT](http://redis.io/commands/pexpireat) key milliseconds-timestampSet the expiration for a key as a UNIX timestamp specified in milliseconds
* [PING](http://redis.io/commands/ping)Ping the server
* [PSETEX](http://redis.io/commands/psetex) key milliseconds valueSet the value and expiration in milliseconds of a key
* [PSUBSCRIBE](http://redis.io/commands/psubscribe) pattern [pattern ...]Listen for messages published to channels matching the given patterns
* [PUBSUB](http://redis.io/commands/pubsub) subcommand [argument [argument ...]]Inspect the state of the Pub/Sub subsystem
* [PTTL](http://redis.io/commands/pttl) keyGet the time to live for a key in milliseconds
* [PUBLISH](http://redis.io/commands/publish) channel messagePost a message to a channel
* [PUNSUBSCRIBE](http://redis.io/commands/punsubscribe) [pattern [pattern ...]]Stop listening for messages posted to channels matching the given patterns
* [QUIT](http://redis.io/commands/quit)Close the connection
* [RANDOMKEY](http://redis.io/commands/randomkey)Return a random key from the keyspace
* [RENAME](http://redis.io/commands/rename) key newkeyRename a key
* [RENAMENX](http://redis.io/commands/renamenx) key newkeyRename a key, only if the new key does not exist
* [RESTORE](http://redis.io/commands/restore) key ttl serialized-valueCreate a key using the provided serialized value, previously obtained using DUMP.
* [RPOP](http://redis.io/commands/rpop) keyRemove and get the last element in a list
* [RPOPLPUSH](http://redis.io/commands/rpoplpush) source destinationRemove the last element in a list, append it to another list and return it
* [RPUSH](http://redis.io/commands/rpush) key value [value ...]Append one or multiple values to a list
* [RPUSHX](http://redis.io/commands/rpushx) key valueAppend a value to a list, only if the list exists
* [SADD](http://redis.io/commands/sadd) key member [member ...]Add one or more members to a set
* [SAVE](http://redis.io/commands/save)Synchronously save the dataset to disk
* [SCARD](http://redis.io/commands/scard) keyGet the number of members in a set
* [SCRIPT EXISTS](http://redis.io/commands/script-exists) script [script ...]Check existence of scripts in the script cache.
* [SCRIPT FLUSH](http://redis.io/commands/script-flush)Remove all the scripts from the script cache.
* [SCRIPT KILL](http://redis.io/commands/script-kill)Kill the script currently in execution.
* [SCRIPT LOAD](http://redis.io/commands/script-load) scriptLoad the specified Lua script into the script cache.
* [SDIFF](http://redis.io/commands/sdiff) key [key ...]Subtract multiple sets
* [SDIFFSTORE](http://redis.io/commands/sdiffstore) destination key [key ...]Subtract multiple sets and store the resulting set in a key
* [SELECT](http://redis.io/commands/select) indexChange the selected database for the current connection
* [SET](http://redis.io/commands/set) key value [EX seconds] [PX milliseconds] [NX|XX]Set the string value of a key
* [SETBIT](http://redis.io/commands/setbit) key offset valueSets or clears the bit at offset in the string value stored at key
* [SETEX](http://redis.io/commands/setex) key seconds valueSet the value and expiration of a key
* [SETNX](http://redis.io/commands/setnx) key valueSet the value of a key, only if the key does not exist
* [SETRANGE](http://redis.io/commands/setrange) key offset valueOverwrite part of a string at key starting at the specified offset
* [SHUTDOWN](http://redis.io/commands/shutdown) [NOSAVE] [SAVE]Synchronously save the dataset to disk and then shut down the server
* [SINTER](http://redis.io/commands/sinter) key [key ...]Intersect multiple sets
* [SINTERSTORE](http://redis.io/commands/sinterstore) destination key [key ...]Intersect multiple sets and store the resulting set in a key
* [SISMEMBER](http://redis.io/commands/sismember) key memberDetermine if a given value is a member of a set
* [SLAVEOF](http://redis.io/commands/slaveof) host portMake the server a slave of another instance, or promote it as master
* [SLOWLOG](http://redis.io/commands/slowlog) subcommand [argument]Manages the Redis slow queries log
* [SMEMBERS](http://redis.io/commands/smembers) keyGet all the members in a set
* [SMOVE](http://redis.io/commands/smove) source destination memberMove a member from one set to another
* [SORT](http://redis.io/commands/sort) key [BY pattern] [LIMIT offset count] [GET pattern [GET pattern ...]] [ASC|DESC] [ALPHA] [STORE destination]Sort the elements in a list, set or sorted set
* [SPOP](http://redis.io/commands/spop) keyRemove and return a random member from a set
* [SRANDMEMBER](http://redis.io/commands/srandmember) key [count]Get one or multiple random members from a set
* [SREM](http://redis.io/commands/srem) key member [member ...]Remove one or more members from a set
* [STRLEN](http://redis.io/commands/strlen) keyGet the length of the value stored in a key
* [SUBSCRIBE](http://redis.io/commands/subscribe) channel [channel ...]Listen for messages published to the given channels
* [SUNION](http://redis.io/commands/sunion) key [key ...]Add multiple sets
* [SUNIONSTORE](http://redis.io/commands/sunionstore) destination key [key ...]Add multiple sets and store the resulting set in a key
* [SYNC](http://redis.io/commands/sync)Internal command used for replication
* [TIME](http://redis.io/commands/time)Return the current server time
* [TTL](http://redis.io/commands/ttl) keyGet the time to live for a key
* [TYPE](http://redis.io/commands/type) keyDetermine the type stored at key
* [UNSUBSCRIBE](http://redis.io/commands/unsubscribe) [channel [channel ...]]Stop listening for messages posted to the given channels
* [UNWATCH](http://redis.io/commands/unwatch)Forget about all watched keys
* [WATCH](http://redis.io/commands/watch) key [key ...]Watch the given keys to determine execution of the MULTI/EXEC block
* [ZADD](http://redis.io/commands/zadd) key score member [score member ...]Add one or more members to a sorted set, or update its score if it already exists
* [ZCARD](http://redis.io/commands/zcard) keyGet the number of members in a sorted set
* [ZCOUNT](http://redis.io/commands/zcount) key min maxCount the members in a sorted set with scores within the given values
* [ZINCRBY](http://redis.io/commands/zincrby) key increment memberIncrement the score of a member in a sorted set
* [ZINTERSTORE](http://redis.io/commands/zinterstore) destination numkeys key [key ...] [WEIGHTS weight [weight ...]] [AGGREGATE SUM|MIN|MAX]Intersect multiple sorted sets and store the resulting sorted set in a new key
* [ZRANGE](http://redis.io/commands/zrange) key start stop [WITHSCORES]Return a range of members in a sorted set, by index
* [ZRANGEBYSCORE](http://redis.io/commands/zrangebyscore) key min max [WITHSCORES] [LIMIT offset count]Return a range of members in a sorted set, by score
* [ZRANK](http://redis.io/commands/zrank) key memberDetermine the index of a member in a sorted set
* [ZREM](http://redis.io/commands/zrem) key member [member ...]Remove one or more members from a sorted set
* [ZREMRANGEBYRANK](http://redis.io/commands/zremrangebyrank) key start stopRemove all members in a sorted set within the given indexes
* [ZREMRANGEBYSCORE](http://redis.io/commands/zremrangebyscore) key min maxRemove all members in a sorted set within the given scores
* [ZREVRANGE](http://redis.io/commands/zrevrange) key start stop [WITHSCORES]Return a range of members in a sorted set, by index, with scores ordered from high to low
* [ZREVRANGEBYSCORE](http://redis.io/commands/zrevrangebyscore) key max min [WITHSCORES] [LIMIT offset count]Return a range of members in a sorted set, by score, with scores ordered from high to low
* [ZREVRANK](http://redis.io/commands/zrevrank) key memberDetermine the index of a member in a sorted set, with scores ordered from high to low
* [ZSCORE](http://redis.io/commands/zscore) key memberGet the score associated with the given member in a sorted set
* [ZUNIONSTORE](http://redis.io/commands/zunionstore) destination numkeys key [key ...] [WEIGHTS weight [weight ...]] [AGGREGATE SUM|MIN|MAX]Add multiple sorted sets and store the resulting sorted set in a new key
* [SCAN](http://redis.io/commands/scan) cursor [MATCH pattern] [COUNT count]Incrementally iterate the keys space
* [SSCAN](http://redis.io/commands/sscan) key cursor [MATCH pattern] [COUNT count]Incrementally iterate Set elements
* [HSCAN](http://redis.io/commands/hscan) key cursor [MATCH pattern] [COUNT count]Incrementally iterate hash fields and associated values
* [ZSCAN](http://redis.io/commands/zscan) key cursor [MATCH pattern] [COUNT count]Incrementally iterate sorted sets elements and associated scores

.