string

{'APPEND': [{'# 对不存在的 key 执行 APPEND\n\nredis> EXISTS myphone # 确保 myphone 不存在\n(integer) 0\n\nredis> APPEND myphone "nokia" # 对不存在的 key 进行 APPEND ，等同于 SET myphone "nokia"\n(integer) 5 # 字符长度\n\n\n# 对已存在的字符串进行 APPEND\n\nredis> APPEND myphone " - 1110" # 长度从 5 个字符增加到 12 个字符\n(integer) 12\n\nredis> GET myphone\n"nokia - 1110"'}, {'http://doc.redisfans.com/string/append.html'}], 'BITCOUNT': [{'redis> BITCOUNT bits\n(integer) 0\n\nredis> SETBIT bits 0 1 # 0001\n(integer) 0\n\nredis> BITCOUNT bits\n(integer) 1\n\nredis> SETBIT bits 3 1 # 1001\n(integer) 0\n\nredis> BITCOUNT bits\n(integer) 2'}, {'http://doc.redisfans.com/string/bitcount.html'}], 'BITOP': [{'redis> SETBIT bits-1 0 1 # bits-1 = 1001\n(integer) 0\n\nredis> SETBIT bits-1 3 1\n(integer) 0\n\nredis> SETBIT bits-2 0 1 # bits-2 = 1011\n(integer) 0\n\nredis> SETBIT bits-2 1 1\n(integer) 0\n\nredis> SETBIT bits-2 3 1\n(integer) 0\n\nredis> BITOP AND and-result bits-1 bits-2\n(integer) 1\n\nredis> GETBIT and-result 0 # and-result = 1001\n(integer) 1\n\nredis> GETBIT and-result 1\n(integer) 0\n\nredis> GETBIT and-result 2\n(integer) 0\n\nredis> GETBIT and-result 3\n(integer) 1'}, {'http://doc.redisfans.com/string/bitop.html'}], 'DECR': [{'# 对存在的数字值 key 进行 DECR\n\nredis> SET failure\_times 10\nOK\n\nredis> DECR failure\_times\n(integer) 9\n\n\n# 对不存在的 key 值进行 DECR\n\nredis> EXISTS count\n(integer) 0\n\nredis> DECR count\n(integer) -1\n\n\n# 对存在但不是数值的 key 进行 DECR\n\nredis> SET company YOUR\_CODE\_SUCKS.LLC\nOK\n\nredis> DECR company\n(error) ERR value is not an integer or out of range'}, {'http://doc.redisfans.com/string/decr.html'}], 'DECRBY': [{'# 对已存在的 key 进行 DECRBY\n\nredis> SET count 100\nOK\n\nredis> DECRBY count 20\n(integer) 80\n\n\n# 对不存在的 key 进行DECRBY\n\nredis> EXISTS pages\n(integer) 0\n\nredis> DECRBY pages 10\n(integer) -10'}, {'http://doc.redisfans.com/string/decrby.html'}], 'GET': [{'# 对不存在的 key 或字符串类型 key 进行 GET\n\nredis> GET db\n(nil)\n\nredis> SET db redis\nOK\n\nredis> GET db\n"redis"\n\n\n# 对不是字符串类型的 key 进行 GET\n\nredis> DEL db\n(integer) 1\n\nredis> LPUSH db redis mongodb mysql\n(integer) 3\n\nredis> GET db\n(error) ERR Operation against a key holding the wrong kind of value'}, {'http://doc.redisfans.com/string/get.html'}], 'GETBIT': [{'# 对不存在的 key 或者不存在的 offset 进行 GETBIT， 返回 0\n\nredis> EXISTS bit\n(integer) 0\n\nredis> GETBIT bit 10086\n(integer) 0\n\n\n# 对已存在的 offset 进行 GETBIT\n\nredis> SETBIT bit 10086 1\n(integer) 0\n\nredis> GETBIT bit 10086\n(integer) 1'}, {'http://doc.redisfans.com/string/getbit.html'}], 'GETRANGE': [{'redis> SET greeting "hello, my friend"\nOK\n\nredis> GETRANGE greeting 0 4 # 返回索引0-4的字符，包括4。\n"hello"\n\nredis> GETRANGE greeting -1 -5 # 不支持回绕操作\n""\n\nredis> GETRANGE greeting -3 -1 # 负数索引\n"end"\n\nredis> GETRANGE greeting 0 -1 # 从第一个到最后一个\n"hello, my friend"\n\nredis> GETRANGE greeting 0 1008611 # 值域范围不超过实际字符串，超过部分自动被符略\n"hello, my friend"'}, {'http://doc.redisfans.com/string/getrange.html'}], 'GETSET': [{'redis> GETSET db mongodb # 没有旧值，返回 nil\n(nil)\n\nredis> GET db\n"mongodb"\n\nredis> GETSET db redis # 返回旧值 mongodb\n"mongodb"\n\nredis> GET db\n"redis"'}, {'http://doc.redisfans.com/string/getset.html'}], 'INCR': [{'redis> SET page\_view 20\nOK\n\nredis> INCR page\_view\n(integer) 21\n\nredis> GET page\_view # 数字值在 Redis 中以字符串的形式保存\n"21"'}, {'http://doc.redisfans.com/string/incr.html'}], 'INCRBY': [{'# key 存在且是数字值\n\nredis> SET rank 50\nOK\n\nredis> INCRBY rank 20\n(integer) 70\n\nredis> GET rank\n"70"\n\n\n# key 不存在时\n\nredis> EXISTS counter\n(integer) 0\n\nredis> INCRBY counter 30\n(integer) 30\n\nredis> GET counter\n"30"\n\n\n# key 不是数字值时\n\nredis> SET book "long long ago..."\nOK\n\nredis> INCRBY book 200\n(error) ERR value is not an integer or out of range'}, {'http://doc.redisfans.com/string/incrby.html'}], 'INCRBYFLOAT': [{'# 值和增量都不是指数符号\n\nredis> SET mykey 10.50\nOK\n\nredis> INCRBYFLOAT mykey 0.1\n"10.6"\n\n\n# 值和增量都是指数符号\n\nredis> SET mykey 314e-2\nOK\n\nredis> GET mykey # 用 SET 设置的值可以是指数符号\n"314e-2"\n\nredis> INCRBYFLOAT mykey 0 # 但执行 INCRBYFLOAT 之后格式会被改成非指数符号\n"3.14"\n\n\n# 可以对整数类型执行\n\nredis> SET mykey 3\nOK\n\nredis> INCRBYFLOAT mykey 1.1\n"4.1"\n\n\n# 后跟的 0 会被移除\n\nredis> SET mykey 3.0\nOK\n\nredis> GET mykey # SET 设置的值小数部分可以是 0\n"3.0"\n\nredis> INCRBYFLOAT mykey 1.000000000000000000000 # 但 INCRBYFLOAT 会将无用的 0 忽略掉，有需要的话，将浮点变为整数\n"4"\n\nredis> GET mykey\n"4"'}, {'http://doc.redisfans.com/string/incrbyfloat.html'}], 'MGET': [{'redis> SET redis redis.com\nOK\n\nredis> SET mongodb mongodb.org\nOK\n\nredis> MGET redis mongodb\n1) "redis.com"\n2) "mongodb.org"\n\nredis> MGET redis mongodb mysql # 不存在的 mysql 返回 nil\n1) "redis.com"\n2) "mongodb.org"\n3) (nil)'}, {'http://doc.redisfans.com/string/mget.html'}], 'MSET': [{'redis> MSET date "2012.3.30" time "11:00 a.m." weather "sunny"\nOK\n\nredis> MGET date time weather\n1) "2012.3.30"\n2) "11:00 a.m."\n3) "sunny"\n\n\n# MSET 覆盖旧值例子\n\nredis> SET google "google.hk"\nOK\n\nredis> MSET google "google.com"\nOK\n\nredis> GET google\n"google.com"'}, {'http://doc.redisfans.com/string/mset.html'}], 'MSETNX': [{'# 对不存在的 key 进行 MSETNX\n\nredis> MSETNX rmdbs "MySQL" nosql "MongoDB" key-value-store "redis"\n(integer) 1\n\nredis> MGET rmdbs nosql key-value-store\n1) "MySQL"\n2) "MongoDB"\n3) "redis"\n\n\n# MSET 的给定 key 当中有已存在的 key\n\nredis> MSETNX rmdbs "Sqlite" language "python" # rmdbs 键已经存在，操作失败\n(integer) 0\n\nredis> EXISTS language # 因为 MSET 是原子性操作，language 没有被设置\n(integer) 0\n\nredis> GET rmdbs # rmdbs 也没有被修改\n"MySQL"'}, {'http://doc.redisfans.com/string/msetnx.html'}], 'PSETEX': [{'redis> PSETEX mykey 1000 "Hello"\nOK\n\nredis> PTTL mykey\n(integer) 999\n\nredis> GET mykey\n"Hello"'}, {'http://doc.redisfans.com/string/psetex.html'}], 'SET': [{'# 对不存在的键进行设置\n\nredis 127.0.0.1:6379> SET key "value"\nOK\n\nredis 127.0.0.1:6379> GET key\n"value"\n\n\n# 对已存在的键进行设置\n\nredis 127.0.0.1:6379> SET key "new-value"\nOK\n\nredis 127.0.0.1:6379> GET key\n"new-value"\n\n\n# 使用 EX 选项\n\nredis 127.0.0.1:6379> SET key-with-expire-time "hello" EX 10086\nOK\n\nredis 127.0.0.1:6379> GET key-with-expire-time\n"hello"\n\nredis 127.0.0.1:6379> TTL key-with-expire-time\n(integer) 10069\n\n\n# 使用 PX 选项\n\nredis 127.0.0.1:6379> SET key-with-pexpire-time "moto" PX 123321\nOK\n\nredis 127.0.0.1:6379> GET key-with-pexpire-time\n"moto"\n\nredis 127.0.0.1:6379> PTTL key-with-pexpire-time\n(integer) 111939\n\n\n# 使用 NX 选项\n\nredis 127.0.0.1:6379> SET not-exists-key "value" NX\nOK # 键不存在，设置成功\n\nredis 127.0.0.1:6379> GET not-exists-key\n"value"\n\nredis 127.0.0.1:6379> SET not-exists-key "new-value" NX\n(nil) # 键已经存在，设置失败\n\nredis 127.0.0.1:6379> GEt not-exists-key\n"value" # 维持原值不变\n\n\n# 使用 XX 选项\n\nredis 127.0.0.1:6379> EXISTS exists-key\n(integer) 0\n\nredis 127.0.0.1:6379> SET exists-key "value" XX\n(nil) # 因为键不存在，设置失败\n\nredis 127.0.0.1:6379> SET exists-key "value"\nOK # 先给键设置一个值\n\nredis 127.0.0.1:6379> SET exists-key "new-value" XX\nOK # 设置新值成功\n\nredis 127.0.0.1:6379> GET exists-key\n"new-value"\n\n\n# NX 或 XX 可以和 EX 或者 PX 组合使用\n\nredis 127.0.0.1:6379> SET key-with-expire-and-NX "hello" EX 10086 NX\nOK\n\nredis 127.0.0.1:6379> GET key-with-expire-and-NX\n"hello"\n\nredis 127.0.0.1:6379> TTL key-with-expire-and-NX\n(integer) 10063\n\nredis 127.0.0.1:6379> SET key-with-pexpire-and-XX "old value"\nOK\n\nredis 127.0.0.1:6379> SET key-with-pexpire-and-XX "new value" PX 123321\nOK\n\nredis 127.0.0.1:6379> GET key-with-pexpire-and-XX\n"new value"\n\nredis 127.0.0.1:6379> PTTL key-with-pexpire-and-XX\n(integer) 112999\n\n\n# EX 和 PX 可以同时出现，但后面给出的选项会覆盖前面给出的选项\n\nredis 127.0.0.1:6379> SET key "value" EX 1000 PX 5000000\nOK\n\nredis 127.0.0.1:6379> TTL key\n(integer) 4993 # 这是 PX 参数设置的值\n\nredis 127.0.0.1:6379> SET another-key "value" PX 5000000 EX 1000\nOK\n\nredis 127.0.0.1:6379> TTL another-key\n(integer) 997 # 这是 EX 参数设置的值'}, {'http://doc.redisfans.com/string/set.html'}], 'SETBIT': [{'redis> SETBIT bit 10086 1\n(integer) 0\n\nredis> GETBIT bit 10086\n(integer) 1\n\nredis> GETBIT bit 100 # bit 默认被初始化为 0\n(integer) 0'}, {'http://doc.redisfans.com/string/setbit.html'}], 'SETEX': [{'SET key value\nEXPIRE key seconds # 设置生存时间'}, {'http://doc.redisfans.com/string/setex.html'}], 'SETNX': [{'redis> EXISTS job # job 不存在\n(integer) 0\n\nredis> SETNX job "programmer" # job 设置成功\n(integer) 1\n\nredis> SETNX job "code-farmer" # 尝试覆盖 job ，失败\n(integer) 0\n\nredis> GET job # 没有被覆盖\n"programmer"'}, {'http://doc.redisfans.com/string/setnx.html'}], 'SETRANGE': [{'# 对非空字符串进行 SETRANGE\n\nredis> SET greeting "hello world"\nOK\n\nredis> SETRANGE greeting 6 "Redis"\n(integer) 11\n\nredis> GET greeting\n"hello Redis"\n\n\n# 对空字符串/不存在的 key 进行 SETRANGE\n\nredis> EXISTS empty\_string\n(integer) 0\n\nredis> SETRANGE empty\_string 5 "Redis!" # 对不存在的 key 使用 SETRANGE\n(integer) 11\n\nredis> GET empty\_string # 空白处被"\\x00"填充\n"\\x00\\x00\\x00\\x00\\x00Redis!"'}, {'http://doc.redisfans.com/string/setrange.html'}], 'STRLEN': [{'# 获取字符串的长度\n\nredis> SET mykey "Hello world"\nOK\n\nredis> STRLEN mykey\n(integer) 11\n\n\n# 不存在的 key 长度为 0\n\nredis> STRLEN nonexisting\n(integer) 0'}, {'http://doc.redisfans.com/string/strlen.html'}]}