

二

07 附录：更多字符串操作命令

键值对过期操作

a.添加键值对并设置过期时间

语法：set key value [expiration EX seconds|PX milliseconds] [NX|XX] 示例：

```
127.0.0.1:6379> set k1 val1 ex 1000
OK
```

设置键值对 k1=val1，过期时间为 1000 秒。查询键的过期时间可以使用 ttl key，如下代码所示：

```
127.0.0.1:6379> ttl k1
(integer) 997
```

b.赋值字符串，并设置过期时间(单位/秒)

语法：setex key seconds value 示例：

```
127.0.0.1:6379> setex k1 1000 v1
OK
127.0.0.1:6379> ttl k1
(integer) 999
127.0.0.1:6379> get k1
"v1"
```

如果 key 已经存在，setex 命令将会覆写原来的旧值。

c.赋值字符串，并设置过期时间(单位/毫秒)

与 `setex` 用法类似，只不过 `psetex` 设置的单位是毫秒。语法：`psetex key milliseconds value` 示例：

```
127.0.0.1:6379> psetex k1 100000 v11
OK
127.0.0.1:6379> ttl k1
(integer) 97
127.0.0.1:6379> get k1
"v11"
```

字符串操作进阶

a.根据指定的范围截取字符串

语法：`getrange key start end` 示例：

```
127.0.0.1:6379> get hello
"hello world"
127.0.0.1:6379> getrange hello 0 4
"hello"
127.0.0.1:6379> getrange hello 0 -1
"hello world"
127.0.0.1:6379> getrange hello 0 -2
"hello worl"
```

负数表示从字符串最后开始计数，`-1` 表示最后一个字符，`-2` 表示倒数第二个，以此类推。

b.设置字符串新值并返回旧值

语法：`getset key value` 示例：

```
127.0.0.1:6379> get db
"redis"
127.0.0.1:6379> getset db mysql
"redis"
127.0.0.1:6379> get db
"mysql"
```

使用 `getset` 命令时，如果 `key` 不为字符串会报错，如下效果所示：

```
127.0.0.1:6379> type myset
```

```
set
127.0.0.1:6379> getset myset v1
(error) WRONGTYPE Operation against a key holding the wrong kind of value
```

根据 type 命令可以查询出 key 所对应的数据类型为非字符串，在使用 getset 命令就会报错。

c.赋值(创建)键值对，当 key 不存在时

如果 key 已经存在，则执行命令无效，不会修改原来的值，否则会创建新的键值对。语法：setnx key value 示例：

```
127.0.0.1:6379> setnx k9 v9
(integer) 1
127.0.0.1:6379> get k9
"v9"
127.0.0.1:6379> setnx k9 v99
(integer) 0
127.0.0.1:6379> get k9
"v9"
```

d.设置一个或多个键值，当所有键值都不存在时

语法：msetnx key value [key value ...] 示例：

```
127.0.0.1:6379> msetnx k5 v5 k6 v6
(integer) 1
127.0.0.1:6379> mget k5 k6
1) "v5"
2) "v6"
```

注意：msetnx 是一个原子操作，当一个操作失败时，其他操作也会失败。例如，如果有一个已经存在的值，那么全部键值都会设置失败，效果如下：

```
127.0.0.1:6379> get k1
"val1"
127.0.0.1:6379> get k8
(nil)
127.0.0.1:6379> msetnx k1 v1 k8 v8
(integer) 0
127.0.0.1:6379> get k1
"val1"
127.0.0.1:6379> get k8
(nil)
```

```
(nil)
```

e. 截取字符串并赋值

语法：setrange key offset value 示例：

```
127.0.0.1:6379> get hello
"hello java"
127.0.0.1:6379> setrange hello 6 redis
(integer) 11
127.0.0.1:6379> get hello
"hello redis"
```

如果待截取的键不存在，会当作空白字符串处理，效果如下：

```
127.0.0.1:6379> setrange mystr 3 mystring
(integer) 11
127.0.0.1:6379> get mystring
(nil)
```

以上这些命令基本涵盖了所有的字符串操作，有些不常用，但很好用，例如 `setnx key value` 命令，当 key 已经存在，则执行命令无效，并不会覆盖原有的值，如果没有此 key 则会新创建一个键值对。

[上一页](#)

[下一页](#)