

# 循环链表接口的定义

---

这里介绍的是单向循环链表。因此，我们只需要考虑维护尾元素和头元素的关系，使得尾元素的`next`指针指向头元素。

## clist\_init

---

```
void clist_init(CList *list, void (*destroy)(void *data));
```

- 返回值 无
- 描述 用来初始化由参数`list`所指定的循环链表
- 复杂度  $O(1)$

## clist\_destroy

---

```
void clist_destroy(CList *list);
```

- 返回值 无
- 描述 用来销毁由参数`list`所指定的循环链表
- 复杂度  $O(n)$  这里 $n$ 表示链表里面元素的个数

## clist\_ins\_next

---

```
int clist_ins_next(CList *list, CListElmt *element, const void *data);
```

- 返回值 如果插入成功返回0，否则返回-1
- 描述 将元素插入由`list`指定的循环链表中`element`元素的后面。
- 复杂度  $O(1)$

## clist\_rem\_next

---

```
int clist_rem_next(CList *list, CListElmt *element, void **data);
```

- 返回值 移除成功返回0，否则返回-1
- 描述 移除`element`的后面的元素
- 复杂度  $O(1)$

## clist\_size

---

```
int clist_size(const CList *list);
```

- **返回值** 链表中元素的个数
- **描述** 这是一个宏，用来计算由参数`list`所指定的链表中元素的个数
- **复杂度**  $O(1)$

## clist\_head

---

```
CListElmt *clist_head(const CList *list);
```

- **返回值** 返回链表的头元素
- **描述** 这是一个宏，返回链表的头元素
- **复杂度**  $O(1)$

## clist\_data

---

```
void *clist_data(const CListElmt *element);
```

- **返回值** 返回由`element`所指定的链表元素的数据域
- **描述** 这是一个宏，返回由`element`所指定的链表元素的数据域
- **复杂度**  $O(1)$

## clist\_next

---

```
CListElmt *clist_next(const CListElmt *element);
```

- **返回值** 返回由`element`所指定元素的后继元素
- **描述** 这是一个宏，返回由`element`所指定元素的后继元素
- **复杂度**  $O(1)$