

## Design to Allow Changes to List During Callback

Here we have design choices.

Our choice:

- one **can safely change the list** in the callback function,
- **EXCEPT** that one **cannot remove the “thing”** passed to the callback from the list.

(The alternative choices can be confusing to use.)

33

## Callback Returns a Number Saying What to Do

So, again, what should the callback return?

An enumerated constant!

Let's call it `dl_execute_response_t`.

34

## Meanings of Possible Callback Return Values

```
typedef enum {
    DL_CONTINUE,
    DL_STOP_AND_RETURN,
    DL_REMOVE_AND_CONTINUE,
    DL_REMOVE_AND_STOP,
    DL_FREE_AND_CONTINUE
} dl_execute_response_t;
```

Keep going.

Return this “thing.”

Remove “thing” and continue with next one.

35

## Meanings of Possible Callback Return Values

```
typedef enum {
    DL_CONTINUE,
    DL_STOP_AND_RETURN,
    DL_REMOVE_AND_CONTINUE,
    DL_REMOVE_AND_STOP,
    DL_FREE_AND_CONTINUE
} dl_execute_response_t;
```

Remove “thing” and return it.

Free “thing” and continue with next one.

36