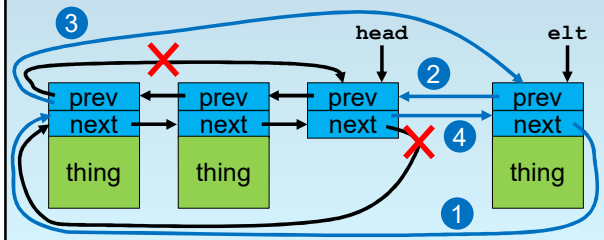
Second Two Insertion Steps Change List

```
head->next->prev = elt;  // step 3
head->next = elt;        // step 4
```



17

Complete Code for List Insertion

Together, we have:

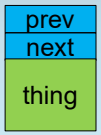
```
void dl_insert (double_list_t* head,
                double_list_t* elt)
{
    elt->next = head->next;
    elt->prev = head;
    head->next->prev = elt;
    head->next = elt;
}
```


18

The "Thing" Must Have a `double_list_t` Field First

But what IS the "thing"?

Any data structure, as follows:




```
struct thing_t {
    double_list_t dl;
    // the rest of the structure
};
```

`double_list_t` must be first in structure.

19

To Insert, Use the Address of the `double_list_t` Field

Once we create a list and a thing:

```
double_list_t my_list =
                {&my_list, &my_list};
struct thing_t my_thing;
```

We can insert the thing into the list as follows:

```
dl_insert (&my_list, &my_thing.dl);
```

Same address as `&my_thing`.

20