

Actividad relacionada con la lección 4:

```
23 def max_sublists(sublists=None):  sublists: [[2, 4, 1], [1, 2, 3, 4, 5, 6, 7, 8], [100, 250, 43]]
24     n_max_list = []  n_max_list: []
25     if sublists is not None:
26         for sublist in sublists:  sublist: [2, 4, 1]
27             n_max = sublist[0]  n_max: 4
28             for element in sublist:  element: 1
29                 if element > n_max:
30                     n_max = element
31     n_max_list.append(n_max)
32     return n_max_list
```

```
23 def max_sublists(sublists=None):  sublists: [[2, 4, 1], [1, 2, 3, 4, 5, 6, 7, 8], [100, 250, 43]]
24     n_max_list = []  n_max_list: [4, 8]
25     if sublists is not None:
26         for sublist in sublists:  sublist: [1, 2, 3, 4, 5, 6, 7, 8]
27             n_max = sublist[0]  n_max: 8
28             for element in sublist:  element: 8
29                 if element > n_max:
30                     n_max = element
31     n_max_list.append(n_max)
32     return n_max_list
```

```
23 def max_sublists(sublists=None):  sublists: [[2, 4, 1], [1, 2, 3, 4, 5, 6, 7, 8], [100, 250, 43]]
24     n_max_list = []  n_max_list: [4, 8]
25     if sublists is not None:
26         for sublist in sublists:  sublist: [100, 250, 43]
27             n_max = sublist[0]  n_max: 250
28             for element in sublist:  element: 43
29                 if element > n_max:
30                     n_max = element
31     n_max_list.append(n_max)
32     return n_max_list
```

Conclusión: Como se puede observar se recorren las sublistas y se pueden apreciar los diferentes valores de cada variable durante el proceso de iteración.

Marlo Lebrón Pareja