

Lección 4

```
(venv) (base) python@pop-os:~/Python/Buenas_Practicas/04$ python Leccion_4.py
> /home/python/Python/Buenas_Practicas/04/Leccion_4.py(10)activity_1()
-> result = [reduce(lambda n1, n2: n1 if n1 > n2 else n2, list)
(Pdb) break 10
Breakpoint 1 at /home/python/Python/Buenas_Practicas/04/Leccion_4.py:10
(Pdb) list
5      list2 = [3, 4, 8, 5, 5, 22, 13]
6
7
8      def activity_1():
9          pdb.set_trace()
10 B->     result = [reduce(lambda n1, n2: n1 if n1 > n2 else n2, list)
11             for list in list1]
12         print(result)
13
14
15     def is_prime(n):
(Pdb) |
```

Una vez creado el *breakpoint*, podemos ver los valores de las variables.

```
(Pdb) continue
> /home/python/Python/Buenas_Practicas/04/Leccion_4.py(10)<listcomp>()
-> result = [reduce(lambda n1, n2: n1 if n1 > n2 else n2, auxList)
(Pdb) p auxList
[2, 4, 1]
(Pdb) continue
> /home/python/Python/Buenas_Practicas/04/Leccion_4.py(10)<lambda>()
-> result = [reduce(lambda n1, n2: n1 if n1 > n2 else n2, auxList)
(Pdb) p n1
2
(Pdb) p n2
4
(Pdb) |
```

La variable *auxList* obtiene el valor de la variable *list1[0]*. Una vez continuado, vemos como las variables *n1* y *n2* obtienen los primeros 2 valores de la lista para obtener el mayor (en este caso 4).

```
(Pdb) p n2
4
(Pdb) continue
> /home/python/Python/Buenas_Practicas/04/Leccion_4.py(10)<lambda>()
-> result = [reduce(lambda n1, n2: n1 if n1 > n2 else n2, auxList)
(Pdb) p n1
4
(Pdb) p n2
1
(Pdb) |
```

Cuando continuamos, vemos como la variable *n1* es el mayor de la comparación anterior y *n2* el valor siguiente de la lista.

Buenas prácticas de programación en Python

```
(Pdb) continue
> /home/python/Python/Buenas_Practicas/04/Leccion_4.py(10)<listcomp>()
-> result = [reduce(lambda n1, n2: n1 if n1 > n2 else n2, auxList)
(Pdb) p auxList
[1, 2, 3, 4, 5, 6, 7, 8]
(Pdb) |
```

Si continuamos vemos como ahora la variable *auxList* coge el siguiente valor, *list1[1]*. Esto se seguirá repitiendo hasta haber cogido el valor mayor de las 3 sublistas.

```
(Pdb) continue
> /home/python/Python/Buenas_Practicas/04/Leccion_4.py(12)activity_1()
-> print(result)
(Pdb) p result
[4, 8, 250]
(Pdb) |
```

Vemos como, una vez termino, la variable *result* obtiene el mayor valor de sus 3 sublistas.