

Pdb de act4.py

Haciendo uso del pdb he analizado el archivo act4.py.

```
python act4.py
```

Creando un punto de parada en la línea donde he realizado la compresión de listas:

```
1 import pdb
2 pdb.set_trace()
3
4 enteros = [[2,4,1], [1,2,3,4,5,6,7,8], [100,250,43]]
5 maximos = [num for num in sublista if num==max(sublista)] for sublista in enteros]
6
7 print(maximos)
```

```
(Pdb) break 5
```

Ejecucion

Continué con el programa hasta llegar al punto de parada

```
(Pdb) continue
```

Donde probé a mostrar las distintas variables

```
> c:\users\pmart\python-workspace\buenas_practicas_programacion\leccion4\act4.py(5)<module>()
-> maximos = [[num for num in sublista if num==max(sublista)] for sublista in enteros]
(Pdb) p num
*** NameError: name 'num' is not defined
(Pdb) p sublista
*** NameError: name 'sublista' is not defined
(Pdb) p enteros
[[2, 4, 1], [1, 2, 3, 4, 5, 6, 7, 8], [100, 250, 43]]
(Pdb) p maximos
*** NameError: name 'maximos' is not defined
```

Fue entonces cuando noté que el pdb sigue actuando como si fuera un for normal, es decir todavía no había insertado el valor a ninguna variable porque el breakpoint se había detenido al principio del for y no había ejecutado toda la línea como un único comando. Hice pruebas línea a línea hasta que por fin encontré que ya asigno valor a una variable.

```
(Pdb) next
> c:\users\pmart\python-workspace\buenas_practicas_programacion\leccion4\act4.py(5)<listcomp>()
-> maximos = [[num for num in sublista if num==max(sublista)] for sublista in enteros]
(Pdb) p enteros
[[2, 4, 1], [1, 2, 3, 4, 5, 6, 7, 8], [100, 250, 43]]
(Pdb) p sublista
[2, 4, 1]
```

Conclusiones

Después de realizar esta serie de pruebas he notado una gran dificultad a la hora de debuggear el código, puesto que limita enormemente el uso de puntos de parada y tienes que hacer un gran uso del comando next para ir viendo lo que pasa. Del modo tradicional sería mucho más sencillo debuggear el código en caso de que saltase algún error, ya que es más legible y más fácil localizar un posible fallo mediante el uso de breakpoints. Aunque bien es cierto que de esta manera a nivel experto el problema de la legibilidad desaparece y puedes concentrar varias líneas de código en una sola haciendo tu algoritmo más compacto