

Ordenando mediante decoradores

Escribe un decorador que permita ordenar el resultado de cualquier función (*siempre que devuelva un iterable*) tomando un **parámetro opcional** que indique si la ordenación es *ascendente* o *descendente*.

Notas:

- Nombre de función: **sort**
- El decorador debe ordenar – por defecto – de forma *ascendente*.
- No es necesario comprobar que la función decorada devuelva un iterable.

Ejemplo:

Primer ejemplo en el que ordenamos de forma **ascendente** (*por defecto*) en base a otra función que devuelve los números pares.

```
1 @sort() # por defecto ascendente
2 def evens(values: list) -> list:
3     return [v for v in values if v % 2 == 0]
4
5 evens([8, 7, 6, 5, 4, 3, 2, 1])
6 # [2, 4, 6, 8]
```

Segundo ejemplo en el que ordenamos de forma **descendente** en base a otra función que devuelve los números pares.

```
1 @sort(asc=False) # descendente
2 def evens(values: list) -> list:
3     return [v for v in values if v % 2 == 0]
4
5 evens([1, 2, 3, 4, 5, 6, 7, 8])
6 # [8, 6, 4, 2]
```