



FILTER – EJEMPLO – DEPENDIENDO DE LONGITUD DE LOS ELEMENTOS

- ▶ Obtenemos los elementos con menos o 6 caracteres de longitud.
- ▶ Individualmente se realiza con la función "len" y comprobamos con el condicional.
- ▶ Si tenemos "Anartz" => len("Anartz") => 6 => if len("Anartz") <= 6 => True (incluye)



```
def only_short_texts(text): return len(text) <= 6
list_values = ["Python", "Javascript", "Typescript", "Kotlin", "Rust"]
list_values_two_mult = list(filter( only_short_texts, list_values))
# ['Python', 'Kotlin', 'Rust']

list_values_two_mult = list(filter( lambda x : len(x) <= 6, list_values))
# ['Python', 'Kotlin', 'Rust']
```



REDUCE

- ▶ Se usa cuando tenemos una colección y necesitamos obtener UN ÚNICO RESULTADO.
- ▶ La función nos permite ir reduciendo los elementos, pudiéndolo considerar un acumulador.
- ▶ Lo importante es aplicar la función. Esta debe de tener 2 parámetros obligatoriamente.
- ▶ Primer parámetro es el acumulador, variable que se modifica por cada elemento de la colección.
- ▶ Segundo parámetro hace referencia a cada elemento de la colección.

```
reduce(función a aplicar, objeto iterable)
```



- ▶ La función devuelve un valor que va añadiendo al acumulador que será un elemento