

T08__IterablesMapFilter

January 29, 2021

1 Python de cero a experto

Autor: Luis Miguel de la Cruz Salas

Python de cero a experto by Luis M. de la Cruz Salas is licensed under Attribution-NonCommercial-NoDerivatives 4.0 International

1.1 Pythonico es más bonito: Pensando como pythonista (intermedio)

1.1.1 Iterables

- La mayoría de los objetos contenedores se pueden recorrer usando un ciclo **for ... in ...**.
- Estos contenedores se conocen como iterables (objetos iterables, secuencias iterables, contenedores iterables, conjunto iterable, ...).

Por ejemplo:

```
[1]: mi_cadena = "123"
mi_lista = [1,2,3]

print('\ncadena:', end=' ')
for char in mi_cadena:
    print(char, end=' ')

print('\nlista:', end=' ')
for element in mi_lista:
    print(element, end=' ')
```

cadena: 1 2 3

lista: 1 2 3

Mapeo (map)

En análisis matemático, un *Mapeo* es una regla que asigna a cada elemento de un primer conjunto, un único elemento de un segundo conjunto:

map

$$\begin{bmatrix} s_1 \\ s_2 \\ \vdots \\ s_{n-1} \end{bmatrix} \longrightarrow \begin{bmatrix} t_1 \\ t_2 \\ \vdots \\ t_{n-1} \end{bmatrix}$$

Definición.

`map(function, sequence)`

map() es una función que toma dos argumentos:

1. Una función.
2. Una secuencia iterable.

map() aplica la función a todos los elementos de la secuencia y regresa una nueva lista con los elementos transformados por la función.

Por ejemplo:

$$f(x) = x^2$$

$$\begin{bmatrix} 0 \\ 1 \\ 2 \\ 3 \\ 4 \end{bmatrix} \longrightarrow \begin{bmatrix} 0 \\ 1 \\ 4 \\ 9 \\ 16 \end{bmatrix}$$

```
[2]: def square(x):
      return x**2

x = [0,1,2,3,4]

xs = map(square, x)

print(type(xs))
print(xs)
print(list(xs))
```

```
<class 'map'>
<map object at 0x7febc8374fa0>
[0, 1, 4, 9, 16]
```

Ejercicio 1.

- Convertir grados Fahrenheit a Celsius y viceversa:

```
[3]: def toFahrenheit(T):
      return (9/5)*T + 32

def toCelsius(T):
      return (5/9)*(T-32)
```

```
c = [0, 22.5, 40, 100]
```

```
[4]: fmap = map(toFahrenheit, c)
```

```
[5]: print(fmap)
```

```
<map object at 0x7febc8374bb0>
```

```
[6]: f = list(fmap)
```

```
print(c)
print(f)
```

```
[0, 22.5, 40, 100]
```

```
[32.0, 72.5, 104.0, 212.0]
```

- Convertir Fahrenheit a Celsius en una sola línea.

```
[7]: print(list(map(toFahrenheit, c)))
```

```
[32.0, 72.5, 104.0, 212.0]
```

Observación: `map()` se puede aplicar a más de un conjunto iterable, siempre y cuando los iterables tengan la misma longitud.

```
[8]: def suma(x,y,z):
      return x+y+z
```

```
a = [1,2,3,4]
```

```
b = [5,6,7,8]
```

```
c = [9,10,11,12]
```

```
list(map(suma, a,b,c))
```

```
[8]: [15, 18, 21, 24]
```

1.1.2 Filtrado (`filter`)

- Filtrar es un procedimiento para seleccionar cosas de un conjunto o para impedir su paso libremente.
- En matemáticas, un filtro es un subconjunto especial de un conjunto parcialmente ordenado.

`filter`

$$\begin{array}{c} \left[\begin{array}{c} s_1 \\ s_2 \\ s_3 \\ s_4 \\ s_{n-1} \end{array} \right] \xrightarrow{\text{True}} \left[\begin{array}{c} - \\ f_1 \\ - \\ f_2 \\ f_{m-1} \end{array} \right] \end{array}$$

Definición.

`filter(function, sequence)`

`filter()` es una función que toma dos argumentos:

1. Una función que regrese un valor *Booleano* (`True/False`)
2. Una secuencia iterable

La función se aplica a cada elemento de la secuencia y solo cuando esta función regresa `True`, el elemento se incluirá en el subconjunto resultante.

Por ejemplo: Encontrar los números pares en una lista:

```
[9]: def esPar(n):  
      if n%2 == 0:  
          return True
```

```
[10]: print(esPar(9))
```

None

```
[11]: numeros = range(20)  
      print(type(numeros))  
      print(list(numeros))
```

```
<class 'range'>  
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19]
```

```
[12]: print(list(filter(esPar, numeros)))
```

```
[0, 2, 4, 6, 8, 10, 12, 14, 16, 18]
```

Ejercicio 2. Encontrar los números pares en una lista que contiene elementos de muchos tipos.

```
[18]: lista = ['Hola', 4, 3.1416, 3, 8, ('a',2), 10, {'x':1.5, 'y':5} ]  
      print(lista)
```

```
['Hola', 4, 3.1416, 3, 8, ('a', 2), 10, {'x': 1.5, 'y': 5}]
```

- Paso 1: escribir una función que verifique si una entrada es de tipo `int`.

```
[13]: def esEntero(i):  
      if type(i) == int:  
          return True  
      else:  
          return False
```

```
[14]: print(esEntero("Hola"))
```

False

- Paso 2: usar la función esPar para encontrar los pares de la lista

Ejercicio 3. Encontrar los números primos en el conjunto $\{2, \dots, 50\}$.

```
[15]: def noPrimo():
        np_list = []
        for i in range(2,8):
            for j in range(i*2, 50, i):
                np_list.append(j)
        return np_list

no_primo = noPrimo()
#print(no_primo)

def esPrimo(number):
    np_list = noPrimo()
    if(number not in np_list):
        return True

numeros = list(range(2,50))

#print(numeros)
primo = list(filter(esPrimo, numeros))
print(primo)
```

[2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47]

```
[ ]:
```