

Función filter()

Tal como su nombre indica filter significa filtrar, y es una de mis funciones favoritas, ya que a partir de una lista o iterador y una función condicional, es capaz de devolver una nueva colección con los elementos filtrados que cumplan la condición.

Por ejemplo, supongamos que tenemos una lista varios números y queremos filtrarla, quedándonos únicamente con los múltiplos de 5...

```
def multiple(numero):    # Primero declaramos una función condicional
    if numero % 5 == 0:  # Comprobamos si un numero es múltiple de cinco
        return True      # Sólo devolvemos True si lo es

numeros = [2, 5, 10, 23, 50, 33]

filter(multiple, numeros)
```

```
<filter at 0x257ac84abe0>
```

Si ejecutamos el filtro obtenemos un objeto de tipo filtro, pero podemos transformarlo en una lista fácilmente haciendo un cast (conversión):

```
list( filter(multiple, numeros) )
```

```
[5, 10, 50]
```

Por tanto cuando utilizamos la función **filter()** tenemos que enviar una función condicional, pero como recordarán, no es necesario definirla, podemos utilizar una función anónima lambda:

```
list( filter(lambda numero: numero%5 == 0, numeros) )
```

```
[5, 10, 50]
```

Así, en una sola línea hemos definido y ejecutado el filtro utilizando una función condicional anónima y una lista de numeros.

Filtrando objetos

Sin embargo, más allá de filtrar listas con valores simples, el verdadero potencial de **filter()** sale a relucir cuando necesitamos filtrar varios objetos de una lista.

Por ejemplo, dada una lista con varias personas, nos gustaría filtrar únicamente las que son menores de edad:

```
class Persona:

    def __init__(self, nombre, edad):
        self.nombre = nombre
        self.edad = edad

    def __str__(self):
        return "{} de {} años".format(self.nombre, self.edad)

personas = [
    Persona("Juan", 35),
    Persona("Marta", 16),
    Persona("Manuel", 78),
    Persona("Eduardo", 12)
]
```

Para hacerlo nos vamos a servir de una función lambda, comprobando el campo edad para cada persona:

```
menores = filter(lambda persona: persona.edad < 18, personas)

for menor in menores:
    print(menor)
```

```
Marta de 16 años
Eduardo de 12 años
```

Sé que es un ejemplo sencillo, pero estoy seguro que os puede servir como base para realizar filtrados en muchos de vuestros proyectos.