

En Python, una función lambda es una pequeña función anónima que puede tener cualquier número de argumentos, pero solo puede tener una expresión. Son muy útiles cuando necesitas una función para una operación corta y no deseas definir una función usando `def`.

La sintaxis general de una función lambda es la siguiente:

```
lambda argumentos: expresión
```

Por ejemplo, podríamos definir una función lambda que tome un argumento `x` y devuelva `x` multiplicado por 2:

```
lambda x: x * 2
```

Esta función es equivalente a la siguiente función definida con `def`:

```
def duplicar(x):  
    return x * 2
```

Las funciones lambda son más útiles cuando se usan con funciones que toman otras funciones como argumentos, como `map()`, `filter()`, `sorted()`, etc.

Entonces, en el caso de `filter()`, usamos una función lambda para proporcionar la condición de filtrado. Por ejemplo:

```
lego_parts = [8980, 7323, 5343, 82700, 92232, 1203, 7319, 8903, 2328, 1279, 679, 589]  
  
filtered_parts = filter(lambda x: x > 500, lego_parts)  
  
print("Números mayores que 500:")  
for part in filtered_parts:  
    print(part)
```

En este caso, `lambda x: x > 500` es una función lambda que devuelve `True` si `x` es mayor que 500 y `False` en caso contrario. La función `filter()` luego usa esta función lambda para filtrar los elementos de `lego_parts` y devuelve un iterador que produce solo los elementos que cumplen la condición.

Espero que esto sea lo que estabas buscando. Si necesitas más ayuda o alguna otra conversión, ¡házmelo saber!