



## Funciones avanzadas



# Funciones lambda



## Índice

- Fundamentos
- Reglas de construcción
- Función filter()
- Función map()



## Fundamentos

- Una función o expresión lambda es la definición en una línea de código de una función anónima, es decir que no tiene nombre.
- Son funciones que ejecutan tareas sencillas, formadas por una **única expresión**.
- Se pueden referenciar a través de variables
- Ejemplo:

```
cuadrado=lambda numero: numero*numero
#uso
print(cuadrado(10))
```



## Reglas de construcción

- El formato de la función lambda es el siguiente:

```
lambda lista_parametros:expresion
```

- *lista\_parametros* es la lista de parámetros de la función, separados por comas
- *expresion* es la instrucción de la función, que puede devolver o no un resultado



## Función filter()

- Filter permite aplicar una condición a una lista de elementos, devolviendo aquellos que la cumplan
- Su formato es:  
`filter(funcion, lista)`
- El primer parámetro representa la función a aplicar, habitualmente, en forma de función lambda.
- Ejemplo: Recupera los números pares de una lista:

```
numeros=[3, 5, 1, 6, 9, 11, 15, 10]  
pares=list(filter(lambda n:n%2==0, numeros))  
print(pares) # [6,10]
```



## Función map()

➤ Transforma una lista de datos en otra lista diferente, aplicando una función de transformación sobre cada uno de los elementos de la lista principal

➤ Su formato es el mismo que el de filter:

```
map(funcion, lista)
```

➤ El contenido de la nueva lista estará formado por los resultados de aplicar la función sobre cada elemento de la lista.

➤ Ejemplo: Genera una lista con los cuadrados de los números de otra lista

```
numeros=[3, 5, 1, 6, 9, 11, 15, 10]
```

```
cuadrados=list(map(lambda n:n*n, numeros))
```

```
print(cuadrados) # [9, 25, 1, 36, 81, 121, 225, 100]
```