

Curso de Desarrollo en Lenguaje Python para Inteligencia Artificial (Málaga)

M.374.001.001

23 de marzo 2020 09:30-13:30

Módulo 1 - Tema 6.3

Carmen Bartolomé



Unión Europea
Fondo Social Europeo
El FSE invierte en tu futuro

 polo de contenidos digitales



Ayuntamiento de Málaga

Módulo 1: Aprendiendo Python. El core de Python

Desarrollo en lenguaje Python

Año de realización: 2021

PROFESORA

Carmen Bartolomé Valentín-Gamazo



Unión Europea
Fondo Social Europeo
Iniciativa de Empleo Juvenil
El FSE invierte en tu futuro

EOI Escuela de
organización
industrial

Módulo 1: el core de Python

Índice

1. Funciones de orden superior
2. Módulo functools
3. Función filter
4. Función map
5. Función reduce
6. Reduce en acción
7. Llamada a reduce
8. Parámetros en reduce
9. Cuándo NO usar reduce
10. Cuándo usar reduce
11. Funciones integradas





Unión Europea
Fondo Social Europeo
El FSE invierte en tu futuro

Tema 6-3

Funciones de orden superior e integradas



Funciones “de orden superior”

- Se conoce como “funciones de orden superior” a aquellas que admiten una función como parámetro y/o devuelven una función como resultado.
- Ya hemos visto algunos casos generales, como las funciones decoradoras.
- Existen unas funciones integradas de orden superior que son especialmente útiles: map, filter
- Se suele usar lambda a menudo como argumento funcional



Unión Europea
Fondo Social Europeo
El FSE invierte en tu futuro

EOI Escuela de
organización
industrial

Módulo functools

- En la librería estándar de Python, tenemos el módulo functools, que aporta funcionalidades especiales a funciones de orden superior
- Ya hemos visto la función wraps, para funciones decoradoras
- Veremos la función reduce en esta lección.





Unión Europea
Fondo Social Europeo
El FSE invierte en tu futuro

EOI Escuela de
organización
industrial

Función filter

- Como su nombre indica, nos va a servir para filtrar
- Admite, como parámetros, una función condicional y un iterable
- Devuelve una nueva colección con los datos filtrados según la condición que marcaba la función que pasamos como argumento.

map aplica un criterio de validación a cada elemento del iterable y guarda solo los que devuelven True.



Unión Europea
Fondo Social Europeo
El FSE invierte en tu futuro

EOI Escuela de
organización
industrial

Función map

- La función map actúa sobre todos los elementos del iterable que se le pasa como argumento
- Ejecuta la función que se le pasa como argumento sobre los elementos del iterable
- El resultado también es un iterable, de tipo map

map aplica una misma instrucción a todos los elementos de un iterable, guardando el resultado





Función reduce

- La función reduce utiliza la técnica “fold” propia de funciones de orden superior.
- Se aplica a un iterable y lo reduce a un valor acumulativo
- En Python 3 ha sido sacada de las funciones integradas (built-in) al módulo functools.
- Guido Van Rossum [explica](#) por qué no está del todo convencido del uso de reduce de forma general. Pero la comunidad presenta una fuerte controversia al respecto.
- Su bucle interno (escrito en C) es más rápido que el bucle for equivalente en Python.

reduce devuelve un valor que es pasado de elemento a elemento



Unión Europea
Fondo Social Europeo
El FSE invierte en tu futuro

EOI Escuela de
organización
industrial

Reduce en acción

La técnica de la función reduce consiste en estos tres pasos:

- Aplica una función a los dos primeros elementos de un iterable y obtiene un resultado parcial
- Utiliza ese resultado parcial junto con el tercer elemento del iterable para actualizar el resultado parcial
- Repite el proceso hasta agotar el iterable y devuelve el valor acumulativo final





Unión Europea
Fondo Social Europeo
El FSE invierte en tu futuro

EOI Escuela de
organización
industrial

Llamada a reduce

```
import functools  
  
functools.reduce()
```

o

```
from functools import reduce  
  
reduce()
```



Parámetros en reduce

La función admite dos parámetros:

- Cualquier objeto “invocable”, como una función, una clase o método. Debe aceptar dos argumentos
- Un iterable (secuencia, tupla, lista, diccionario, ...) o un iterador. En caso de pasar un iterador, será necesario llegar a su agotamiento para obtener el valor final.
- También tenemos un argumento opcional: **initializer**. Hace de semilla, siendo el primer valor con el que se iniciaría el proceso de reduce o será el resultado por defecto cuando el iterable esté vacío



Unión Europea
Fondo Social Europeo
El FSE invierte en tu futuro

EOI Escuela de
organización
industrial

Cuándo NO usar reduce

Existen funciones integradas o en la librería estándar que aportan más sencillez y legibilidad:

- Suma acumulativa de valores numéricos
- Producto acumulativo de valores numéricos
- Encontrar valores máximo o mínimo de una secuencia numérica.





Unión Europea
Fondo Social Europeo
El FSE invierte en tu futuro

EOI Escuela de
organización
industrial

Cuándo usar reduce

Utilizar reduce tiene especial sentido en los casos:

- Suma acumulativa de valores numéricos según un cierto patrón
- Producto acumulativo de valores numéricos según un cierto patrón
- Encontrar valores máximo o mínimo teniendo en cuenta alguna particularidad.
- Encontrar si algún elemento de una secuencia es True





Unión Europea
Fondo Social Europeo
El FSE invierte en tu futuro

EOI Escuela de
organización
industrial

Funciones integradas

- La distribución de Python lleva incorporadas varias funciones predefinidas e integradas de forma que podemos llamarlas directamente.
- Son funciones que se utilizan con bastante frecuencia y esta facilidad en su llamada contribuye a un código más simple y legible.



	Built-in Functions			
abs()	delattr()	hash()	memoryview()	set()
all()	dict()	help()	min()	setattr()
any()	dir()	hex()	next()	slice()
ascii()	divmod()	id()	object()	sorted()
bin()	enumerate()	input()	oct()	staticmethod()
bool()	eval()	int()	open()	str()
breakpoint()	exec()	isinstance()	ord()	sum()
bytearray()	filter()	issubclass()	pow()	super()
bytes()	float()	iter()	print()	tuple()
callable()	format()	len()	property()	type()
chr()	frozenset()	list()	range()	vars()
classmethod()	getattr()	locals()	repr()	zip()
compile()	globals()	map()	reversed()	__import__()
complex()	hasattr()	max()	round()	



Unión Europea
Fondo Social Europeo
El FSE invierte en tu futuro

EOI Escuela de
organización
industrial

isinstance()

`isinstance(object, type)`

Devuelve “True” si el objeto es del tipo que pasamos como argumento type.





Unión Europea
Fondo Social Europeo
El FSE invierte en tu futuro

EOI Escuela de
organización
industrial

enumerate()

`enumerate(iterable, start)`

Crea un objeto enumerate a partir de una colección (el parámetro iterable)

El parámetro start por defecto es 0. A través de él podemos indicar a partir de qué número empezar el índice de la secuencia



Unión Europea
Fondo Social Europeo
El FSE invierte en tu futuro

EOI Escuela de
organización
industrial

eval()

`enumerate(expression, global, locals)`

Evalúa una expresión dada, y si es admitida por Python, la ejecuta

Los parámetros *global* y *local* son opcionales, y son diccionarios de tipo global o local respectivamente

Advertencia: no es aconsejable vincularla a input porque es una puerta abierta a que se ejecute código perjudicial sin ningún filtro.



Unión Europea
Fondo Social Europeo
El FSE invierte en tu futuro

EOI Escuela de
organización
industrial

sorted()

`sorted(iterable, key=key, reverse=True or False)`

Dado un iterable, devuelve una lista ordenada

key es opcional, y es una función que indica el criterio de orden

reverse es de tipo booleano, y también es opcional. Por defecto es False y corresponde a orden ascendente.

Si ponemos reverse = True, el orden pasa a ser descendente



Unión Europea
Fondo Social Europeo
El FSE invierte en tu futuro

EOI Escuela de
organización
industrial

carmenbvg@gmail.com

```
1  def gratitude():  
2      print("Thank you.")  
3
```