

IMF Smart Education
IMF Smart Education

Funciones © IMF Smart Education

IMF Smart Education
IMF Smart Education

IMF Smart Education
IMF Smart Education

-tion

-tion

Índice

Funciones	3
I. Introducción	3
II. Objetivos	3
III. Funciones	4
IV. Resumen	5
Ejercicios	6
Caso práctico 1	6
Se pide	6
Solución	6
Caso práctico 2	6
Se pide	6
Solución	6
Caso práctico 3	6
Se pide	7
Solución	7
Caso práctico 4	7
Se pide	7
Solución	7
Caso práctico 5	8
Se pide	8
Solución	8
Caso práctico 6	8
Se pide	8
Solución	9

Funciones

I. Introducción

En esta unidad se trabajará con los distintos tipos de funciones que se pueden crear y manejar en Python. El primer paso será declarar las funciones y cuál es su estructura básica. El nombre genérico de este tipo de funciones es “de primer orden”. Además de saber crear esas funciones, se debe saber cómo llamarlas dentro del código de Python.

A continuación, se analizará el tipo de argumentos que se pueden pasar a estas funciones y se verán dos conceptos nuevos, “valores por defecto” y “parámetros arbitrarios”. Para estos dos nuevos términos, se realizarán ejemplos para que se entienda cuándo y dónde se pueden usar esos parámetros.

Otras funciones que se puede usar en Python, además de las de primer orden, son las conocidas como funciones anónimas LAMBDA. Aprenderemos mediante ejemplo como trabajar con ellas. Por último, en este módulo trabajaremos con las funciones más importantes conocidas como de “alto orden”. Se hará uso de las funciones map, filter y reduce.

II. Objetivos

1

Definir las funciones de primer orden.

Funciones

2

Aprender la llamada de las funciones dentro del código.

3

Saber cuándo y dónde usar los valores por defecto.

4

Saber cuándo y dónde usar los parámetros arbitrarios.

5

Conocer qué es una función anónima.

6

Manejar la función anónima Lambda.

7

Conocer las funciones de alto orden.

8

Manejar mediante ejemplos las funciones map, filter, reduce.

III. Funciones



Funciones

Puedes ver las diapositivas relacionas con este vídeo en el siguiente enlace: [./Funciones.pdf](#).

IV. Resumen

En esta unidad se ha aprendido cómo se definen las funciones de primer orden y como sería su llamada en el código de Python. Se ha practicado el caso de funciones con valores por defecto, mezclándolo con parámetros arbitrarios. Para poder completar los conceptos vistos en esta unidad, se han manejado funciones con valores por argumento y aplicando funciones de alto orden.

Ejercicios

Caso práctico 1

Se pide

Diseña un programa. Supongamos que tenemos una lista de nombre de personas `people = ['Amy', 'Alice', 'Bobby', 'Charlie', 'Connie', 'David']`. Lo que queremos es que se filtre la información y quedarnos solo con los que empiezan por C.

Crea una función que haga lo que se te pide.

Solución

```
people=['Amy', 'Alice', 'Bobby', 'Charlie', 'Connie', 'David']
def iniciales(names):
    print([[name,name[0]] for name in names ])
```

```
iniciales(people)
```

```
 [['Amy', 'A'], ['Alice', 'A'], ['Bobby', 'B'], ['Charlie', 'C'], ['Connie', 'C'], ['David', 'D']]
```

Pincha sobre la imagen para visualizarla a mayor resolución.

Caso práctico 2

Se pide

Crear una función que establezca como parámetros de la función lambda tres palabras y nos las muestre concatenadas.

Solución

```
mz = (lambda a = 'Lunes', b = 'Miercoles', c = 'Viernes': a + b + c)
print(mz())
```

```
LunesMiercolesViernes
```

Caso práctico 3

Se pide

Explica que devuelve cada caso de la función lambda:

$f = \text{lambda } x, y, z: (x+y) * z$

$f(5, 6)$

$f(5, 6, 7)$

Solución

$F(5, 6)$

$X=5$

$Y=6$

$Z=1 \rightarrow 11$

$F(5, 6, 7)$

$X=5$

$Y=6$

$Z=7 \rightarrow 77$

Caso práctico 4

Se pide

Haciendo uso de las funciones de alto orden, crea una función llamada “cuadrado” que calcule la raíz cuadrada de una lista de valores:

- Indica que función de alto nivel has usado.
- Realiza el ejercicio. Los valores y la lista os lo podéis inventar vosotros.

Solución

```
import math

lista=[1,2,3,4,5]
for i in lista:
    print((lambda x: math.sqrt(x))(i))
```

```
1.0
1.4142135623730951
1.7320508075688772
2.0
2.23606797749979
```

Caso práctico 5

Se pide

Se nos informa que tenemos una lista como la siguiente: lista5 = [-3, -2, 0, 1, 9, -5].

Queremos realizar una función llamada `verdad_mentira` que nos indique si cada valor de la lista cumple o no el criterio de ser menor a 0. Para ello, debemos recurrir a las funciones de alto orden.

- Indica que función de alto nivel has usado.
- Realiza el ejercicio.

Solución

```
lista5=[-3, -2, 0, 1, 9, -5]
def verdad_mentira(lista):
    return list(filter(lambda n:n<0,lista))

print(verdad_mentira(lista5))
```

```
[-3, -2, -5]
```

Caso práctico 6

Se pide

Ejercicio 1

Funciones

Crear una función a la que se le va a pasar una cadena y un número y debe pintar dicha cadena tantas veces como el número indica.

Ejercicio 2

Desarrollar una función que reciba tres enteros y retorne la suma de los dos más grandes.

def sumar mayores (x1, x2, x3):

Solución



Caso práctico: Funciones