

HINTS

PARÁMETROS EN LA FUNCIÓN

Al pasar los valores que corresponden a los parámetros de una función, es indispensable hacerlo en el orden correcto, pues al contrario obtendremos resultados incorrectos.

Sin embargo, cuando conocemos el nombre de las variables que recibe como parámetros una función, podemos pasar los parámetros usando el nombre de estas variables y su valor; de esta forma ya no es indispensable pasar los parámetros en el orden exacto.

Ejemplo:

```
1 def estudiante(nombre, apellido):
2     print(nombre, apellido)
3
4
5 estudiante(nombre='Pedro', apellido='Pérez')
6 estudiante(apellido='Pérez', nombre='Pedro')
7
8 Resultado:
9 Pedro Pérez
10 Pedro Pérez
```

FUNCIÓN DENTRO DE FUNCIÓN

Es posible declarar una función dentro de otra función. Una función que se define dentro de otra se conoce como función interna, o anidada. Las funciones anidadas pueden acceder a las variables del ámbito que las rodea.

Las funciones internas se utilizan para que puedan estar protegidas de todo lo que ocurre fuera de la función donde se creó. Es decir, como ésta solo funcionará dentro de la función donde fue creada, es imposible detectarla o usarla en cualquier otro sitio, de la misma forma que una variable local.

FUNCIÓN RECURSIVA

En Python es posible crear funciones recursivas, que son aquellas que se llaman a sí mismas para poder realizar las operaciones que requieren, hasta que alguna condición evite que siga

ejecutándose el llamado. Esto es de práctica común en la solución de algunos problemas matemáticos, o en algoritmos de búsqueda de datos.

De esta manera, primero se resolverá la última llamada, esta devolverá el resultado a la llamada anterior la cual realizará el cálculo respectivo, y así sucesivamente, hasta que todos los cálculos de todas las llamadas se hayan hecho.

Por ejemplo: usemos una función recursiva para el cálculo de un número factorial. La función se llamará a ella misma hasta que el resultado sea 1.

```
1 def factorial(x):
2     if x == 1:
3         return 1
4     else:
5         return (x * factorial(x-1))
6
7
8 num = 3
9 print("El factorial de", num, "es", factorial(num))
10
11 #Resultado:
12 El factorial de 3 es 6
```