

6. Funciones

Las funciones son bloques de código que realizan una tarea específica y se pueden utilizar una y otra vez en un programa. Las funciones nos permiten dividir nuestro código en tareas más pequeñas y fáciles de manejar, lo que hace que nuestro código sea más organizado, más legible y más fácil de mantener.

a) Definición y llamada de funciones

Para definir una función en Python, se utiliza la palabra reservada **def**, seguida del nombre de la función y los parámetros entre paréntesis. El cuerpo de la función se define con sangría. Para llamar a una función, simplemente escriba su nombre seguido de paréntesis y cualquier argumento que se le pase.

Ejemplo de definición y llamada de función:

```
def calcular_precio(cantidad, precio_unitario):
```

```
    total = cantidad * precio_unitario
```

```
    return total
```

```
precio_total = calcular_precio(10, 5.5)
```

```
print(precio_total)
```

```
> 55
```

b) Parámetros y argumentos

Los parámetros son variables que se utilizan para pasar información a una función. Los argumentos son los valores que se pasan a los parámetros cuando se llama a una función. Los parámetros se definen entre paréntesis en la definición de la función, mientras que los argumentos se pasan entre paréntesis cuando se llama a la función.

Ejemplo de parámetros y argumentos:

```
def saludar(nombre):
```

```
    print("Hola, " + nombre + "!")
```

```
saludar("Juan")
```

```
> Hola, Juan!
```

c) Retorno de valores desde funciones

Las funciones pueden devolver un valor utilizando la palabra clave **return**. El valor devuelto se puede almacenar en una variable o utilizar directamente en otra parte del programa.

Ejemplo de retorno de valores desde funciones:

<pre>def sumar(a, b): return a + b</pre>
<pre>resultado = sumar(3, 5) print(resultado)</pre>
<pre>> 8</pre>

d) Ámbito de variables

Las variables definidas dentro de una función se consideran variables locales y solo están disponibles dentro de la función. Las variables definidas fuera de una función se consideran variables globales y están disponibles en todo el programa. Si una variable local y una variable global tienen el mismo nombre, la variable local tendrá prioridad dentro de la función.

Ejemplo de ámbito de variables:

<pre>x = 10</pre>
<pre>def mi_funcion(): x = 5 print("Dentro de la función, x = ", x)</pre>
<pre>mi_funcion() print("Fuera de la función, x = ", x)</pre>
<pre>> Dentro de la función, x = 5 > Fuera de la función, x = 10</pre>

En conclusión, las funciones son una herramienta fundamental en Python que permiten agrupar un conjunto de instrucciones que pueden ser ejecutadas múltiples veces a lo largo del programa, permitiendo un código más organizado y modular. Además, las

funciones pueden recibir parámetros de entrada y devolver valores de salida, lo que las hace muy versátiles y útiles en la programación. Es importante tener en cuenta el alcance de las variables, para evitar errores y problemas en el código. Por lo tanto, es necesario entender cómo funcionan las funciones en Python para poder aprovechar al máximo su potencial en la programación.