

A graphic consisting of four stacked, 3D rectangular blocks in purple, orange, yellow, and blue, forming an arrow shape pointing to the right.

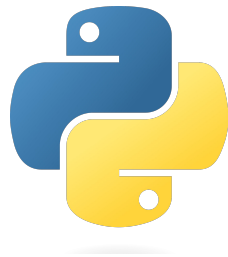
Agencia de
Aprendizaje
a lo largo
de la vida

CODO A CODO INICIAL

Clase 18

Python 6

Funciones II: Argumentos



Les damos la bienvenida

Vamos a comenzar a grabar la clase

Clase 17

Python 5 - Funciones I: Introducción

- Definición
- Propósito. Ventajas
- Sintaxis para declarar funciones.
- Ejemplos de funciones sencillas.

Clase 18

Python 6 - Funciones II: Argumentos

- Parámetros
- Argumentos
- Ejemplos de funciones con parámetros y argumentos.
- Funciones integradas en el lenguaje (len(), print(), input(), etc.)

Clase 19

Python 7 - Funciones III: Retorno de valores

- Uso de return para devolver valores desde una función.
- Argumentos por posicionales y por nombre.
- Parámetros con valores por defecto.

Parámetros de la función

Un **parámetro** es una variable que se utiliza en la definición de la función para representar un dato que la ésta espera recibir cuando es llamada.

Los parámetros **permiten que una función acepte valores externos** y los utilice dentro de su bloque de código. Estos valores, que son proporcionados al llamar a la función, se denominan "**argumentos**".

Los argumentos son obligatorios, si definimos n cantidad de parámetros debemos llamar la función con n cantidad de argumentos.

Funciones | Parámetros y Argumentos

Los **parámetros** son las variables que ponemos cuando se define una función. En la siguiente función tenemos un parámetro: *numero*:

```
def multiplicarPor5(numero):  
    print(numero * 5)
```

Los **argumentos** son los valores que se pasan a la función cuando ésta es invocada, “8” en el ejemplo:

```
# Programa principal  
num = 8  
multiplicarPor5(num) # 40
```

Dentro de la función, los **argumentos** se copian en los parámetros y son usados por ésta para realizar la tarea.

Funciones | Argumentos fijos y variables


Los **argumentos** que se utilizan al invocar la función pueden ser valores fijos o variables (un valor introducido por el usuario o calculado):

Ejemplo 1:

```
def saludarUsuario(nombre):  
    print(f'Hola {nombre}')
```

Programa principal

```
saludarUsuario("Juan Pablo")  
nom = input("Ingrese su nombre: ")  
saludarUsuario(nom)
```




```
Hola Juan Pablo  
Ingrese su nombre: Agustina  
Hola Agustina
```

Ejemplo 2:

```
def calcularCuadrado(base):  
    print(f'El cuadrado de {base} es  
{base**2}')
```

Programa principal

```
calcularCuadrado(3)  
num = int(input("Ingrese un número: "))  
calcularCuadrado(num)
```



```
El cuadrado de 3 es 9  
Ingrese un número: 4  
El cuadrado de 4 es 16
```

Funciones | Argumentos múltiples

En una función se pueden colocar **uno o más argumentos**, cuidando que en la invocación, los **parámetros** se coloquen en el mismo orden:

```
def generarMail(nombre, apellido, anio):  
    mail = f'{apellido.lower()}.{nombre[0].lower()}_{anio}@empresa.com'  
    print(f'Tu email es: {mail}')
```

```
# Programa principal  
nombre = input("Ingrese su nombre: ")  
apellido = input("Ingrese su apellido: ")  
anio = int(input("Ingrese su año de nacimiento: "))  
generarMail(nombre, apellido, anio)
```

Esta función genera un mail a partir de los datos ingresados por el usuario

```
Ingrese su nombre: Diego  
Alejandro  
Ingrese su apellido: Garcia  
Ingrese su año de nacimiento:  
1988  
Tu email es:  
garcia.d_1988@empresa.com
```


Funciones | Argumentos múltiples | Ejemplo

```
def obtenerDescuento(precio, porcentaje):  
    descuento = precio * porcentaje/100  
    precioNuevo = precio - descuento  
    print(f'Precio anterior: $  
{precio:.2f}\nDescuento: $ {descuento:.2f}\nPrecio  
nuevo: $ {precioNuevo:.2f}')
```

Programa principal

```
obtenerDescuento(200, 15)
```

Precio anterior: \$ 200.00
Descuento: \$ 30.00
Precio nuevo: \$ 170.00

Esta función obtiene el precio con descuento, recibiendo como argumentos el precio del producto y el porcentaje (número entero) de descuento.

Funciones integradas en el lenguaje

Python trae integradas una serie de funciones listas para usar, algunas ya las conocemos:

Función	Significado	Ejemplo
print	Muestra información en la consola.	<code>print("Hola, mundo!")</code>
input	Recibe la entrada del usuario desde la consola.	<code>nombre = input("Ingrese su nombre: ")</code>
len	Devuelve el número de elementos de un objeto, como una cadena de texto.	<code>longitud_cadena = len("Python")</code>
type	Devuelve el tipo de un objeto.	<code>tipo_numero = type(42)</code>

Funciones integradas en el lenguaje

Función	Significado	Ejemplo
int	Convierte un valor a un número entero.	<code>entero = int("42")</code>
float	Convierte un valor a un número de punto flotante.	<code>flotante = float("3.14")</code>
str	Convierte un valor a una cadena de texto.	<code>cadena = str(123)</code>

Desafíos

Desafío 1: Área de un triángulo

Desarrollar una función que reciba la longitud de la base y la altura de un triángulo, y muestre en la pantalla estos dos valores junto al área de la figura.

Tip: el área de cualquier triángulo se calcula multiplicando la longitud de la base por la altura, y dividiendo el resultado entre dos.

Desafío 2: Tabla de multiplicar

Desarrollar una función que reciba un número entero de 0 a 10, y que muestre por la terminal la tabla de multiplicar por ese número.

Algunas ideas que puedes implementar:

- Verificar que el valor recibido está en el rango correcto.
- Poner un título antes de comenzar a mostrar la tabla solicitada.

Desafío 3: Generador de ticket para cine

Desarrollar dos funciones: la primera imprimirá los datos de la película (nombre, día y hora, sala), solicitados por teclado. La segunda función obtendrá el valor del ticket, recibiendo como parámetros la cantidad de entradas, el valor de la entrada y un descuento en % sobre el total si la cantidad de entradas es mayor o igual a 5. El formato de salida podrá ser similar a este:

```
*****  
WONKA  
Jueves 30/01 16 hs  
Sala 3  
*****  
Valor de la entrada: $ 5000.00  
Cantidad de entradas: 5  
Valor del ticket: $ 20000.00
```

Material extra

Artículos de interés

Material extra:

- [Funciones predefinidas en Python 3.10](#) | Sitio oficial de Python
- [Guia de estilo PEP8](#), o cómo escribir buen código Python
- Variables en Python | Sitio oficial de

Videos:

- [Funciones en Python](#) | Tutoriales sobre Ciencia y Tecnología
- [Curso de Python](#), en Píldoras informáticas

No te olvides de dar el presente

Recordá:

- Revisar la Cartelera de Novedades.
- Hacer tus consultas en el Foro.
- Realizar el Ejercicio de Repaso.

Todo en el Aula Virtual.

Muchas gracias por tu atención.

Nos vemos pronto