



# Fundamentos de Programación con Python

## Unidad VII Funciones – Programación Funcional

Docente: T.S.U Gerardo Alí Ferraro Schelijasch  
[gerferr83@soltecferr.com](mailto:gerferr83@soltecferr.com)  
<https://soltecferr.com>

# Unidad VII - Funciones

---

## Resumen

- Definición de función en programación
- Algoritmo de Divide y Vencerás (Divide et impera)
- Funciones en Python
  - Uso de argumentos
  - Funciones Lambdas

# Unidad VII – Funciones

---

## Funciones

### Definición:

Una función en programación es un bloque de código reutilizable que realiza una tarea específica. Las funciones reciben una entrada, llamada parámetros o argumentos, procesan esos datos, y luego devuelven un resultado o ejecutan una acción. Son esenciales para estructurar el código de manera modular, mejorar la legibilidad y evitar la repetición.

Las funciones permiten encapsular tareas, facilitando el mantenimiento y la extensión del código.

```
def nombre_funcion(parametro1, parametro2):  
    # Cuerpo de la función  
    return resultado
```

# Unidad VII – Funciones

---

## Algoritmo divide y vencerás (Divide et impera):

El algoritmo **Divide y Vencerás(Divide et impera)** es una técnica que consiste en resolver un problema grande dividiéndolo en subproblemas mas pequeños, resolviendo estos subproblemas y luego combinando estas soluciones se resuelve el problema original. Esta técnica se constituye de tres fases fundamentales:

1. Dividir: Dividir el problema grande en varios subproblemas más pequeños de la misma naturaleza.
2. Vencer (Resolver): Resolver los subproblemas de forma recursiva. Si los subproblemas son lo suficientemente simples, resolverlos directamente.
3. Combinar: Combinar las soluciones de los subproblemas para obtener la solución del problema original.

# Unidad VII – Funciones

---

## Algoritmo divide y vencerás

El algoritmo **Divide y Vencerás** está estrechamente relacionado con el uso de funciones en programación, ya que el enfoque funcional, depende directamente de la implementación de funciones. En cada paso, el problema se descompone en subproblemas más pequeños, que son resueltos a través de funciones.

Al final, las soluciones a cada uno de los subproblemas se combinan para dar una solución al problema original.

# Unidad VII – Funciones

---

## Funciones en Python

En las unidades anteriores ya se ha hecho uso de funciones, de hecho para poder crear el juego de piedra papel o tijera se hizo uso de las funciones de la librería `random` como `choice()` y `randint()` que nos permitían generar valores aleatorios, de la misma forma se usaron la funciones `sum(lista)`, `max(lista)`, `min(lista)` para poder ubicar respectivamente la suma, el valor mas alto y el valor mas bajo de una lista.

Muy a menudo ocurre que un cierto fragmento de código se repite muchas veces en un programa. Se repite de manera literal o, con algunas modificaciones menores, empleando algunas otras variables dentro del programa. También ocurre que un programador ha comenzado a copiar y pegar ciertas partes del código en más de una ocasión en el mismo programa.

si un fragmento de código comienza a aparecer en más de una ocasión, considera la posibilidad de aislarlo en la forma de una función invocando la función desde el lugar en el que originalmente se encontraba.

# Unidad VII – Funciones

---

## Ventajas

Puede suceder que el algoritmo que se desea implementar sea tan complejo que el código comience a crecer de manera incontrolada y, de repente, ya no se puede navegar por él tan fácilmente.

Un buen desarrollador divide el código (o mejor dicho: el problema) en piezas aisladas, y codifica cada una de ellas en la forma de una función. Esto simplifica considerablemente el trabajo del programa, debido a que cada pieza se codifica por separado, y consecuentemente se prueba por separado. A este proceso se le llama comúnmente descomposición. Esta descomposición continúa hasta que se obtiene un conjunto de funciones cortas, fáciles de comprender y probar.



# Unidad VII – Funciones

---

## Uso y generalidad:

Para definir una función en Python se utiliza la palabra clave o reservada `def`. A continuación viene el nombre o identificador de la función que es el que se utiliza para invocarla. Después del nombre hay que incluir los paréntesis y una lista opcional de parámetros. Por último, la cabecera o definición de la función termina con dos puntos.

```
def function_name():  
    cuerpo de la función
```



# Unidad VII – Funciones

---

## Ejemplo:

Crea una función en Python que permita suma 2 números a y b:

```
def suma (a, b):  
    return a + b
```

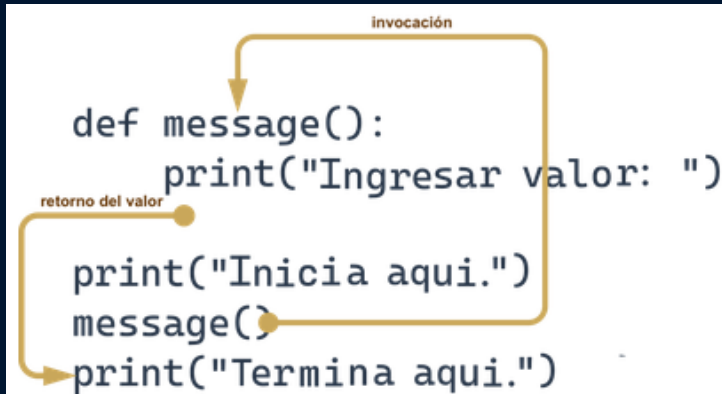
Características Principales:

- Siempre comienza con la palabra reservada def (que significa definir)
- después de def va el nombre de la función (las reglas para darle nombre a las funciones son las mismas que para las variables)

# Unidad VII – Funciones

## Características Principales:

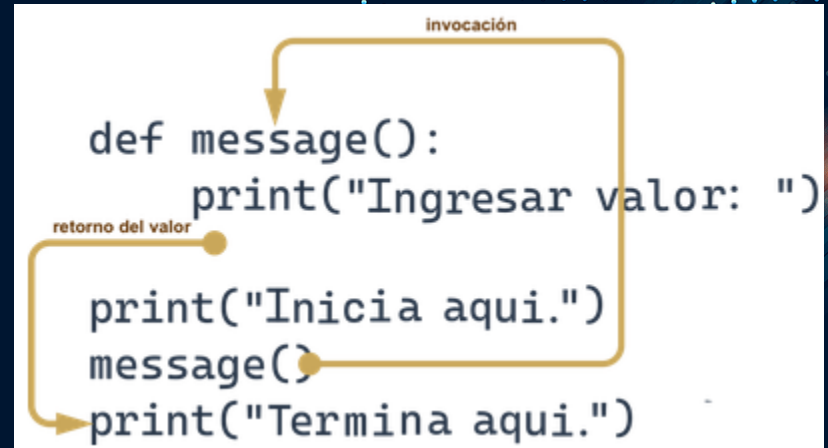
- después del nombre de la función, hay un espacio para un par de paréntesis (ahorita no contienen algo, pero eso cambiará pronto)
- la línea debe de terminar con dos puntos;
- Cuerpo de la función, secuencia de instrucciones que realizan la operación requerida



# Unidad VII – Funciones

## Como funcionan las funciones:

- cuando se invoca una función, Python recuerda el lugar donde esto ocurre y salta hacia dentro de la función invocada;
- el cuerpo de la función es entonces ejecutado;
- al llegar al final de la función, Python regresa al lugar inmediato después de donde ocurrió la invocación.



# Unidad VII – Funciones

---

## Como funcionan las funciones:

Tras los dos puntos se incluye el cuerpo de la función (con un sangrado mayor, generalmente cuatro espacios) que no es más que el conjunto de instrucciones que se encapsulan en dicha función y que le dan significado. En último lugar y de manera opcional, se añade la instrucción con la palabra reservada `return` para devolver un resultado.

Para usar o invocar a una función, simplemente hay que escribir su nombre como si de una instrucción más se tratara. Eso sí, pasando los argumentos necesarios según los parámetros que defina la función.

# Unidad VII – Funciones

---

## Como funcionan las funciones:

### Sentencia return:

La sentencia return te ayuda a que las funciones se puedan comunicar con el exterior. Esto se da gracias a la devolución de valores. En Python tienes la posibilidad de retornar valores múltiples separados en comas. A su vez, se puede registrar a distintas variables y valores de la tupla inmutable.

### Parámetros de la función:

Un parámetro es un valor que la función espera recibir cuando sea llamada (invocada), a fin de ejecutar acciones en base al mismo. Una función puede esperar uno o más parámetros (que irán separados por una coma) o ninguno.

# Unidad VII – Funciones

---

## Como funcionan las funciones:

### Parámetros de la función:

Los parámetros, se indican entre los paréntesis, a modo de variables, a fin de poder utilizarlos como tales, dentro de la misma función. Los parámetros que una función espera, serán utilizados por ésta, dentro de su algoritmo, a modo de variables de ámbito local. Es decir, que los parámetros serán variables locales, a las cuáles solo la función podrá acceder.

Si quisiéramos acceder a esas variables locales, fuera de la función, obtendríamos un error: Al llamar a una función, siempre se le deben pasar sus argumentos en el mismo orden en el que los espera. Pero esto puede evitarse, haciendo uso del paso de argumentos como keywords

## Unidad VII – Funciones

---

### Términos de la licencia.

- This work is licensed under the creative commons Attribution-shareAlike License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-sa/4.0> or send a letter to Creative Commons, 559 Nathan Abbott Way, Stanford, California 94305, USA,
- Este trabajo se otorga bajo los términos de la licencia Creative Commons Attribution-shareAlike License. Para obtener una copia de esta licencia visita <http://creativecommons.org/licenses/by-sa/4.0> o envía una carta a la dirección Creative Commons, 559 Nathan Abbott Way, Stanford, California 94305, USA.