

Subprogramas

Los subprogramas, también conocidos como funciones, son bloques de código reutilizables que realizan una tarea específica. Se emplean para modularizar el código, haciéndolo más organizado, fácil de entender y de mantener. Además, permiten evitar la repetición innecesaria de código

```
def nombre_subprograma(parámetros):  
    """  
    Cuerpo del subprograma  
    """
```

Partes de la declaración de un subprograma

def: Es la palabra clave que indica que se está definiendo un subprograma.

nombre_subprograma: Es el nombre que se le da al subprograma. Debe ser un nombre descriptivo y fácil de recordar.

parámetros: Son los valores que se le pasan al subprograma cuando se llama. Se declaran entre paréntesis, separados por comas.

Cuerpo del subprograma: Es el código que se ejecuta cuando se llama al subprograma. Se escribe entre sangrías.

¿Cómo se llama un subprograma en Python?

```
nombre_subprograma(argumentos)
```

nombre_subprograma: Es el nombre del subprograma que se quiere llamar.

argumentos: Son los valores que se le pasan al subprograma cuando se llama. Se declaran entre paréntesis, separados por comas.

Ejemplo Subprograma

```
def saludar():  
    print("El cacho no mata")  
  
saludar() # Se llama al subprograma 'saludar'.
```

Ejemplo Subprograma con parámetros

Los subprogramas pueden recibir parámetros, que son valores que se les pasan cuando se llaman. Los parámetros se utilizan para proporcionar datos al subprograma y controlar su comportamiento.

```
def saludar(nombre):  
    print("Hola", nombre)  
  
saludar("Juan") # Se llama al subprograma 'saludar'  
                # con el parámetro 'Juan'.
```

Ejemplo Subprograma con parámetros

```
def saludar(nombre):  
    print("Hola", nombre)  
  
primer_nombre = input("Digite su primer nombre: ")  
  
saludar(primer_nombre)    # Se llama al subprograma  
                           #'saludar' con el parámetro  
                           #'primer_nombre '.
```

Ejemplo Subprograma con parámetros y retorno

Los subprogramas pueden devolver un valor cuando se ejecutan. El valor devuelto puede ser utilizado por el programa que llama al subprograma.

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

```
resultado = sumar(5, 3) # Se llama al subprograma 'sumar'  
                        # y se almacena el resultado en la  
                        # variable 'resultado'.
```

```
print(resultado) # Se imprime el valor de la  
                # variable 'resultado'.
```



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

```
numero1 = float(input("Digite su primer número: "))  
numero2 = float(input("Digite su segundo número: "))
```

```
# Se llama al subprograma 'sumar'  
# y se almacena el resultado en la  
# variable 'resultado'  
resultado = sumar(numero1, numero2)
```

```
print(resultado) # Se imprime el valor de la  
                 # variable 'resultado'.
```


Subprograma "Sumar"

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

5, 3

8

Estructura Principal

```
resultado = sumar(5, 3) # Se llama al subprograma 'sumar'  
                        # y se almacena el resultado en la  
                        # variable 'resultado'.
```

```
print("El resultado de la suma es: ", resultado)
```

Subprograma "Sumar"

```
def sumar(a, b):  
    suma = a + b  
    print("El resultado de la suma es: ", suma)
```

5, 3

12, 5

Estructura Principal

```
sumar(5, 3) # Se llama al subprograma 'sumar' por 1era vez
```

```
sumar(12, 5) # Se llama al subprograma 'sumar' por 2da vez
```