

Programación de Algoritmos

PGY1121

Escuela de Informática y Telecomunicaciones

DuocUC



ESCUELA DE
INFORMÁTICA Y
TELECOMUNICACIONES



Experiencia de Aprendizaje N° 4

Clase N° 3

Funciones

Objetivos de la sesión

- Crear diferentes funciones definidas por el usuario, para dar respuesta a los requerimientos del caso.

- Son aquellas que dividen el programa en partes, considerando la parte principal y los diferentes métodos (tareas) que deben proporcionar resultados.
- Se ejecutan sólo cuando son llamadas.
- Pueden ser llamadas las veces que se requieran.

Sintaxis

```
def mi_funcion():  
    instrucciones  
    .....
```

La estructura de la función se compone de la siguiente forma:

- Def palabra reservada (con minúscula)
- Nombre descriptivo para la función
- Paréntesis
- Dos Puntos (:)
- Las instrucciones, las cuales cumplen la misma norma que si fuera una sentencia de control

Tipos de Funciones:

- Sin Argumentos y sin retorno
- Sin Argumentos y con retorno
- Con Argumentos y sin retorno
- Con Argumentos y con retorno

- Sin Argumentos y sin retorno



```
def saludo():  
    print("Saludando a mis estudiantes")
```

```
saludo()
```

```
Saludando a mis estudiantes
```



Función saludo(), sin argumentos y emite mensaje propio de la función



Llamada a función saludo(), no envía argumentos



Resultado de la llamada a la función saludo().

- Sin Argumentos y con retorno

```
def sumar():  
    num1 = 3  
    num2 = 5  
    return(num1 + num2)  
  
print("La suma es: ", sumar())
```

La suma es: 8

→ Función sumar() sin argumentos, posee dos variables con valor y retorna la suma de ellas.

→ Instrucción que invoca la función sumar() y espera el resultado para ser mostrado por pantalla.

→ Resultado del proceso de la función que proporciona un valor final.

- Con Argumentos y sin retorno

```
def sumar(a, b):  
    suma = a + b  
    print(f"La suma de los argumentos es: {suma}")
```

```
num1 = int(input("Ingrese primer numero: "))  
num2 = int(input("Ingrese segundo numero: "))  
sumar(num1, num2)
```

```
Ingrese primer numero: 3  
Ingrese segundo numero: 6  
La suma de los argumentos es: 9
```

→ Función sumar() con argumentos, posee dos variables con valor y no retorna la suma de ellas, sino que muestra el resultado en la misma función.

→ Solicitud de los argumentos numéricos.

→ Llamada a la Función sumar() con dos argumentos.

→ Ingreso de los argumentos numéricos.

→ Muestra resultado del proceso realizado por la Función sumar().

- Con Argumentos y con retorno

```
def sumar(a, b):  
    suma = a + b  
    return(suma)  
  
num1 = int(input("Ingrese primer numero: "))  
num2 = int(input("Ingrese segundo numero: "))  
print(f"El resultado de la suma es: ", sumar(num1, num2))
```

```
Ingrese primer numero: 5  
Ingrese segundo numero: 4  
El resultado de la suma es: 9
```

→ Función sumar() con argumentos, posee dos variables con valor y retorna la suma de ellas.

→ Solicitud de los valores a las variables.

→ Muestra mensaje y hace llamada a la Función sumar() con argumentos.

→ Ingresa los valores a los argumentos.

→ Muestra mensaje con el resultado final del proceso de la función sumar().

PGY1121

Ver video

<https://www.youtube.com/watch?v=hrv1ruHxiQY>

- Otras funciones

```
▶ def varios_valores(*args):  
    for arg in args:  
        print(arg)  
  
varios_valores(4.5, "Buen dia", [1,2,3,4,5])
```

```
4.5  
Buen dia  
[1, 2, 3, 4, 5]
```

→ Función `varios_valores()` que recibe una lista dinámica de argumentos y por ello, se debe incluir un `*`. Esto se llama recibir parámetros indeterminados por posición.

→ Llamada a la Función `varios_valores()` que envía diferentes argumentos.

→ Resultados del proceso de la Función `varios_valores()`.

- Otras funciones

```
▶ def mostrar_valores():  
    return("Buen dia", 15, [10,20,30])
```

```
mostrar_valores()
```

```
➤ ('Buen dia', 15, [10, 20, 30])
```

➡ Función `mostrar_valores()` que no recibe parámetros, sin embargo retorna valores múltiples, esto se llama “Retorno múltiple”.

➡ Llamada a la Función `mostrar_valores()` que no envía argumentos.

➡ Resultados del proceso de la Función `mostrar_valores()`.



Analiza con tus compañeros las siguientes funciones y comenta el resultado con tu docente.

Resultado

```
def resta(a, b):  
    return a - b  
  
resta(30, 10)
```

retorna 20

```
def resta(a, b):  
    return a - b  
  
resta(b=30, a=10)
```

retorna -20

```
def funcion():  
    return "Bienvenidos a Python"  
  
frase = funcion()  
print(frase)
```

retorna -"Bienvenidos a Python"
Output print "Bienvenidos a Python"

Resultado

```
def resta(a=None, b=None):  
    if a == None or b == None:  
        print("Error, faltan parámetros a la función")  
        return  
    return a - b  
  
resta()
```

retorno None
Output print "Error, faltan patametros"

```
def calculo(precio, descuento):  
    return precio - (precio * descuento / 100)  
  
datos = [10000, 10]  
print("El monto final a pagar es: ", calculo(*datos))
```

```
def saludo(nombre, mensaje='Python'):  
    print(mensaje, nombre)  
  
saludo(mensaje="Buen día", nombre="Pedro")
```


Se pide escribir las instrucciones necesarias para crear un menú con las opciones de:

- Calcular_Iva
- Descuento
- Calcular_Imc

Las cuales deben ser desarrolladas en funciones (métodos).

Donde:

- Calcular_Iva: Es el precio del producto, multiplicado por el 19% (0.19)
- descuento: Es el precio del producto menos el descuento por aplicar. Mostrar el valor final del producto.
- Calcular_Imc: Índice de masa corporal. Su fórmula es $\text{peso} / \text{estatura}^2$, además se debe mostrar el estado de la persona de acuerdo a la siguiente tabla.

< 18.5	Bajo peso
18.5 - 24.9	Adecuado
25.0 - 29.9	Sobrepeso
30.0 - 34.9	Obesidad grado 1
35.0 - 39.9	Obesidad grado 2
>40	Obesidad grado 2

Recuerda subir los archivos a GitHub

Revisar ejercicios de la Experiencia 4 Semana 3.

Programación de Algoritmos

PGY1121

Escuela de Informática y Telecomunicaciones

DuocUC



ESCUELA DE
INFORMÁTICA Y
TELECOMUNICACIONES