

Clase 3 Parte 1: Programación

Programación Modular.

Temario

- Programación modular.
 - Creación de módulos
 - Funciones
 - Parámetros



Módulos y funciones

Módulos

¿Qué es un módulo?

Definición: Archivo que contiene funciones, clases y variables.

- Se puede usar en varios programas.
- Python ya trae módulos incorporados.
- Ejemplo: `math`, `random`, `datetime`.

Sentencia import

Con **import** se importan módulos. De esta forma, podemos acceder a su contenido.

Forma general

```
import nombre_modulo
```

¿Qué significa “importar” un módulo?

Se tiene “acceso” a los datos y las funciones definidas dentro del módulo.

Funciones en Python

Una función es un bloque de código que realiza una tarea específica.

Sintaxis:

```
def nombre_funcion(parametros):  
    # instrucciones  
    return valor
```

Ejemplo:

```
def saludo(nombre):  
    return "Hola, " + nombre + "!"  
  
print(saludo("Diego")) # → Hola, Diego!
```

Pasaje de parámetros

```
def nombreFuncion(parámetros):  
    sentencias  
    return <expresion>
```

Los parámetros se pueden pasar por copia o referencia.

Los parámetros deben pasarse respetando cantidad y posición.

Pasaje de parámetros

Tipos inmutables (int, float, str) → se comportan como *paso por copia* → los cambios dentro de la función **no afectan** al original.

Tipos mutables (list, dict, set) → se comportan como *paso por referencia* → los cambios dentro de la función **sí afectan** al original.

Pasaje de parámetros

```
def suma(x,y):  
    return x + y
```

¿Cómo lo
usamos?

```
print(suma(3,4) )
```

→ Imprime
7

```
def cuad(x,y=2):  
    return x **y
```

¿Cómo lo
usamos?

```
print(cuad(3))
```

→ imprime
9

```
def suma(x,y):  
    return x + y
```

¿Cómo lo usamos?

```
print(suma(y=5, x=4) )
```

→ Imprime
9

Variables locales y globales

- Local: existe solo dentro de la función.
- Global: definida fuera de la función, accesible en todo el programa.

```
x = 10    # variable global

def prueba():
    x = 5    # variable local
    print("Dentro:", x)

prueba()
print("Fuera:", x)
```

Salida

```
Dentro: 5
Fuera: 10
```

Importar un módulo real

```
import math
```

```
print(math.sqrt(16))    # Raíz cuadrada → 4.0
```

```
print(math.pow(2, 3))   # Potencia → 8.0
```

```
print(math.pi)         # Constante PI → 3.14159...
```

También podemos importar sólo lo que necesitamos:

```
from math import sqrt, pi
```

```
print(sqrt(25))         # → 5.0
```

```
print(pi)               # → 3.14159...
```

Crear nuestro propio módulo

Crear un archivo llamado `operaciones.py`:

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

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

```
def cuadrado(x):  
    return x * x
```

Usarlo desde otro archivo:

```
import operaciones  
  
print(operaciones.suma(5, 3))      # 8  
print(operaciones.resta(10, 4))   # 6  
print(operaciones.cuadrado(7))    # 49
```

Ejemplo

- Definir una función que reciba una lista de precios y retorne el precio promedio.

```
def promedio(lisprecios):  
    suma = 0  
    cant = len(lisprecios)  
    for j in range(cant):  
        suma += lisprecios[j]  
    return suma / cant
```

```
lista = [200, 450, 180]  
print(promedio(lista)) # → 276.666...
```

