

# 9\_Clases y Funciones

June 18, 2025

*Creado por:*

*Isabel Maniega*

## 1 Clases y Funciones

Nos permiten ordenar el código, como por ejemplo código repetido, etc

### 1.0.1 Ejemplo 1: 1 Clase - 2 funciones

```
[1]: class Clase1:

    def funcion1():
        print("Estamos en: Clase1 - Funcion1")

    def funcion2():
        print("Estamos en: Clase1 - Funcion2")
```

```
[2]: # funcion1() #--> no funciona, no encuentra la funcion
```

```
[3]: # se define primero la clase + función
Clase1.funcion1()
```

Estamos en: Clase1 - Funcion1

```
[4]: Clase1.funcion2()
```

Estamos en: Clase1 - Funcion2

### Ejemplo 1

```
[5]: class Clase1:

    def imprimir(y):
        print("Estamos en: ")
        print("Clase Python - funcion imprimir, argumento y: ", y)

    def funcion_suma():
        x = 1
        z = 3
```

```
y = x + z

print(f"La suma de {x} más {z} es {y}")
```

```
[6]: # con argumento
Clase1.imprimir(6)
```

Estamos en:

Clase Python - funcion imprimir, argumento y: 6

```
[7]: # sin argumento
Clase1.funcion_suma()
```

La suma de 1 más 3 es 4

## Ejemplo 2

```
[8]: class Clase1:

    def funcion1():
        print("Estamos en: Clase1 - Funcion1")

    def funcion2():
        print("Estamos en: Clase1 - Funcion2")
        print("\n")
        funcion1() # No accede a la función se necesita declarar la clase
```

```
[9]: Clase1.funcion1()
```

Estamos en: Clase1 - Funcion1

```
[10]: Clase1.funcion2()
```

Estamos en: Clase1 - Funcion2

```
-----
NameError                                Traceback (most recent call last)
Cell In[10], line 1
----> 1 Clase1.funcion2()

Cell In[8], line 9, in Clase1.funcion2()
      7 print("Estamos en: Clase1 - Funcion2")
      8 print("\n")
----> 9 funcion1()

NameError: name 'funcion1' is not defined
```

```
[11]: class Clase1:

    def funcion1():
        print("Estamos en: Clase1 - Funcion1")

    def funcion2():
        print("Estamos en: Clase1 - Funcion2")
        print("\n")
        Clase1.funcion1()
```

```
[12]: Clase1.funcion2()
```

```
Estamos en: Clase1 - Funcion2
```

```
Estamos en: Clase1 - Funcion1
```

### Ejemplo 3

```
[13]: class Clase1:

    def imprimir(y):
        print("Estamos en: ")
        print("Clase Python - funcion imprimir, argumento y: ", y)

    def funcion_suma():
        x = 1
        z = 3
        y = x + z

        Clase1.imprimir(y)
```

```
[14]: Clase1.funcion_suma()
```

```
Estamos en:
Clase Python - funcion imprimir, argumento y:  4
```

```
[15]: Clase1.imprimir(10)
```

```
Estamos en:
Clase Python - funcion imprimir, argumento y:  10
```

#### 1.0.2 2 Clases con la misma función

- Una misma función puede estar en varias clases diferentes

Dado que tenemos que llamar previamente a la clase...

entonces: \* Sólo se ejecutará la función dentro de la clase

```
[16]: class Clase1:

    def funcion1():
        print("Estamos en: Clase1 - Funcion1")

class Clase2:

    def funcion1():
        print("Estamos en: Clase2 - Funcion1")
```

```
[17]: Clase1.funcion1()
```

Estamos en: Clase1 - Funcion1

```
[18]: Clase2.funcion1()
```

Estamos en: Clase2 - Funcion1

### Ejemplo

```
[19]: class Frutas:
    def suma(x, y):
        suma = x + y
        print("la suma de frutas que tenemos es: ", suma)

class Verduras:
    def suma(x, y):
        suma = x + y
        print("la suma de verduras que tenemos es: ", suma)
```

```
[20]: Frutas.suma(3, 2)
```

la suma de frutas que tenemos es: 5

```
[21]: Verduras.suma(6, 2)
```

la suma de verduras que tenemos es: 8

### 1.0.3 1 clase que recibe otra clase

```
[22]: class Clase1:
    def funcion1():
        print("Estamos en: Clase1 - Funcion1")

class Clase2(Clase1):
    def funcion2():
        print("Estamos en: Clase2 - Funcion2")

    # def funcion1():
    #     print("Estamos en: Clase1 - Funcion1")
```

```
[23]: Clase1.funcion1()
```

Estamos en: Clase1 - Funcion1

```
[24]: Clase2.funcion2()
```

Estamos en: Clase2 - Funcion2

```
[25]: # heredado con la clase --> HERENCIA
Clase2.funcion1()
```

Estamos en: Clase1 - Funcion1

### Ejemplo práctico

```
[26]: class Frutas:
    def manzanas():
        suma_manzanas = 6
        return suma_manzanas

class Alimentos(Frutas):
    def peras():
        suma_peras = 7
        return suma_peras
    #def manzanas():
    #    suma_manzanas = 6
    #    return suma_manzanas
```

```
[27]: Frutas.manzanas()
```

```
[27]: 6
```

```
[28]: Alimentos.peras()
```

```
[28]: 7
```

```
[29]: Alimentos.manzanas()
```

```
[29]: 6
```

También podemos “llamar” variables

```
[30]: class Clase1:
    x = 5

class Clase2:
    def funcion():
        print("Estamos en la Clase2")
        print("\n")
        print("Ahora llamamos a la variable de la Clase1")
```

```
print("Valor: ", Clase1.x)
#print("Valor: ", x) # Necesario añadir la clase, sino error de
↪variable No definida
```

```
[31]: Clase2.funcion()
```

Estamos en la Clase2

Ahora llamamos a la variable de la Clase1

Valor: 5

```
[32]: class Clase1:
      x = 5

      class Clase2:
          def funcion():
              x = 6
              print("Estamos en la Clase2")
              print("\n")
              print("Ahora llamamos a la variable de la Clase1")
              print("Valor: ", Clase1.x)
              print("Valor: ",x) # Necesario añadir la clase, sino error de variable
              ↪No definida
```

```
[33]: Clase2.funcion()
```

Estamos en la Clase2

Ahora llamamos a la variable de la Clase1

Valor: 5

Valor: 6

```
[34]: class Clase1:
      x = 6
      def funcion1(x):
          return x
```

```
[35]: Clase1.funcion1(Clase1.x)
```

```
[35]: 6
```

```
[36]: class Clase1:
      y = 6
      def funcion1(y):
```

```
return y
```

```
[37]: Clase1.funcion1(y)
```

```
-----  
NameError                                Traceback (most recent call last)  
Cell In[37], line 1  
----> 1 Clase1.funcion1(y)  
  
NameError: name 'y' is not defined
```

```
[38]: x = 6  
def funcion1(x):  
    return x
```

```
[39]: funcion1(x)
```

```
[39]: 6
```

*Creado por:*

*Isabel Maniega*