

_1type/_01ejemplos.py

```
1 class Animal:
2     def __init__(self, nombre):
3         self.nombre = nombre
4         print(f"Creando un nuevo animal llamado {nombre}")
5
6     def sonido(self):
7         return "Hago algún ruido"
8
9 class Perro(Animal):
10     def sonido(self):
11         return "Guau Guau"
12
13 gato = Animal("Whiskers")
14 perro = Perro("Fido")
15
16 def imprime_sonido(animal):
17     print(f"El sonido del animal es: {animal.sonido()}")
18
19 if isinstance(gato, Animal):
20     print(f"{gato.nombre} es un animal.")
21
22 for i in range(3):
23     print("Iteración", i)
24
25 for i in range(5):
26     print("Valor de i en el segundo bucle:", i)
27     if i == 2:
28         break
29
30 for i in range(5):
31     if i == 2:
32         continue
33     print("Valor de i en el tercer bucle:", i)
34
35 def funcion_vacia():
36     pass
37
38 x = 5
39 assert x > 0, "x no puede ser negativo"
40 print("La aserción pasó correctamente.")
41
42 try:
43     resultado = 10 / 0
44 except ZeroDivisionError:
45     print("Error: División por cero")
46 finally:
47     print("Finalizando el bloque try-except")
48
49 del gato
50 print("La variable 'gato' ha sido eliminada.")
51
```

```
52 doble = lambda x: x * 2
53 print("El doble de 5 es:", doble(5))
54
55 contador = 0
56 def incrementa_contador():
57     global contador
58     contador += 1
59 incrementa_contador()
60 print("El valor del contador es:", contador)
61
62 def externa():
63     x = 10
64     def interna():
65         nonlocal x
66         x += 1
67         print("Valor de x dentro de la función interna:", x)
68     interna()
69     print("Valor de x fuera de la función interna:", x)
70 print("Llamando a la función externa...")
71 externa()
72
73 def suma(a, b):
74     return a + b
75 print("La suma de 3 y 4 es:", suma(3, 4))
76
77 #Recursividad para factorizar (multiplicar todos los nº por si mismo hasta alcanzar el nº)
78 #Forma 1
79 def factorial(n):
80     if n == 1:
81         return 1
82     else: n* factorial(n-1)
83 #Forma 2
84 import math
85 # Ejemplo: Factorial de 5
86 print(math.factorial(5)) # Salida: 120
87 #Forma 3
88 def factorial(n):
89     resultado = 1
90     for i in range(1, n + 1):
91         resultado *= i
92     return resultado
93
94 # Ejemplo: Factorial de 5
95 print(factorial(5)) # Salida: 120
96
97 def sumar (a,b):
98     """
99     Calcula la suma de dos argumentos.
100
101     Args:
102         a(parametro)
103         b(parametro)
104
105     Retorna:
```

```
106         int: La suma de a y b.
107
108     """
109     return(a+b)
110
111 print(sumar(5,7))
112
113
114 #Gestión de Excepciones:
115 """try: x = 1 / 0
116     #Inicia un bloque de código donde se manejan las excepciones.
117 except ZeroDivisionError: print("División por cero")
118 #Captura y maneja una excepción específica.
119 raise ValueError("Valor inválido")
120 #Genera una excepción específica.
121 assert x > 0, "x debe ser mayor que cero"#Verifica si una condición es verdadera y genera
una excepción si no lo es.
122 finally: print("Operación finalizada") #Define un bloque de código que siempre se ejecuta,
123 independientemente de si se genera una excepción o no."""
124
125 class Perro:
126     especie="Canis Spaniel"
127     def ladrar(self):
128         print("GUAU")
129     def __init__(self,nombre,edad):
130         self.nombre=nombre
131         self.edad=edad
132
133 mi_perro= Perro("Trufa",14)
134 print(mi_perro.nombre)
135 print(mi_perro.edad)
136
137 class Torra:
138     pass
139
140 try:
141     numero=int(input("Ingrese un numero"))
142     division = 10/numero
143     print(division)
144 except:
145     print("elemento no apropiado")
146
147
148 def verificar_tipos(valor):
149     if isinstance(valor, (int, float, complex)):
150         return "Es un número"
151     elif isinstance(valor, str):
152         return "Es una cadena"
153     elif isinstance(valor, list):
154         return "Es una lista"
155     else:
156         return "Tipo desconocido"
```