_1type/_3int.py

```python
# --- Operatos ---

print(2 + 3)   # total
print(3 - 1)   # diff
print(2 * 3)   # product
print(3 / 2)   # division
print(3 ** 2)  # exp
print(3 % 2)   # remainder
print(7 // 2)  # floor division

# --- Math Operators ---
print(abs(-5.5))               #5.5
print(max(10, 20, 30))         #30
print(min(-10, 0, 10))         #-10
print(pow(2, 3))               #8    x**y elevado
print(round(3.14159, 2))       #3.14 2 decimales
number = 3.142448
print(f"{number:.4f}")

print(sum([1, 2, 3, 4, 5]))    #15
print(divmod(10, 3))           #(3, 1) division= (cociente , resto)
print(len([1, 2, 3, 4, 5]))    #5

# --- Funciones matemáticas ---
import math

techo = math.ceil(3.1)         # 4  Redondea hacia arriba.
piso = math.floor(3.9)         # 3  Redondea hacia abajo.
truncado = math.trunc(3.14159) # 3   Trunca eliminando los decimales

raiz_cuadrada = math.sqrt(16)  # 4.0 Raiz cuadrada
logaritmo = math.log(100, 10)  # 2.0 Logaritmo en base x (e por defecto)

# --- Aleatorios ---
import random

aleatorio = random.randint(1, 10)     # Entero aleatorio entre 1 y 10
aleatorio_fl = random.uniform(1, 10)  # Flotante aleatorio entre 1 y 10

# --- Manejo de excepciones ---
try:
    resultado = 10 / 0
except ZeroDivisionError:
    print("No se puede dividir por cero")

# --- Manipulación binaria ---
num = 10  # 1010 en binario
bitwise_and = num & 1  # 1010 & 0001 = 0000 -> 0
bitwise_or = num | 1   # 1010 | 0001 = 1011 -> 11
bitwise_xor = num ^ 1  # 1010 ^ 0001 = 1011 -> 11
bitwise_not = ~num     # -(1010 + 1) = -1011 -> -11
```

```
left_shift = num << 1  # 1010 << 1 = 10100 -> 20
right_shift = num >> 1 # 1010 >> 1 = 101 -> 5

# --- Conversión de base ---
num = 255
binario = bin(num)     # '0b11111111'
octal = oct(num)       # '0o377'
hexadecimal = hex(num) # '0xff'

"""-----------------------Func----------------------------------------"""
# Euclidean distance formula: sqrt((x2 - x1)^2 + (y2 - y1)^2)
import math
point1=(2,3)
point2=(10,8)
euclidean=math.sqrt((point2[0]- point1[0])**2 + (point2[1] - point1[1])**2)
print(f"Euclidean Distance between {point1} and {point2}: {euclidean}")

#e**x    e=Constante Euler 2.71828
resultado = math.exp(7)
```