

## Actividad Python ejercicios

1. Programa que permita multiplicar 3 números.

```
1  #funcion multiplicar tres numeros
2  def multiplicar_tres_numeros(num1,num2,num3):
3      return num1*num2*num3
4
5  # Solicitar al usuario que ingrese los tres números
6  try:
7      numero1 = float(input("Ingrese el primer número: "))
8      numero2 = float(input("Ingrese el segundo número: "))
9      numero3 = float(input("Ingrese el tercer número: "))
10
11     # Llamar a la función y mostrar el resultado
12     resultado = multiplicar_tres_numeros(numero1, numero2, numero3)
13     print(f"La multiplicación de {numero1}, {numero2} y {numero3} es: {resultado}")
14 except ValueError:
15     print("Por favor, ingrese solo números válidos.")
16
```

2. Solicitar al usuario un número y calcular cuál es el cuadrado del número ( $2^2=4$ ).

```
C: > Users > gabri > Downloads > # Solicitar al usuario que ingrese un nú.py > ...
1  # Solicitar al usuario que ingrese un número
2  try:
3      numero = float(input("Ingrese un número: "))
4
5      # Calcular el cuadrado del número
6      cuadrado = numero ** 2
7
8      # Mostrar el resultado
9      print(f"El cuadrado de {numero} es: {cuadrado}")
10 except ValueError:
11     print("Por favor, ingrese un número válido.")
12
```

- 3.

3 programa para calcular la distancia recorrida en un movimiento rectilíneo. Recuerde  $x=v*t$  donde  $v$  es velocidad y  $t$  es tiempo. Solicitar al usuario velocidad en kilómetros por hora (Km/h) y tiempo en horas (h) para obtener la distancia en kilómetros (Km).

```
C: > Users > gabri > Downloads > # Solicitar al usuario que ingrese un nú.py > ...  
1  # Solicitar al usuario la velocidad en km/h  
2  try:  
3      velocidad = float(input("Ingrese la velocidad en km/h: "))  
4  
5      # Solicitar al usuario el tiempo en horas  
6      tiempo = float(input("Ingrese el tiempo en horas: "))  
7  
8      # Calcular la distancia  
9      distancia = velocidad * tiempo  
10  
11     # Mostrar el resultado  
12     print(f"La distancia recorrida es: {distancia} km")  
13 except ValueError:  
14     print("Por favor, ingrese valores numéricos válidos.")  
15
```

4 programa que permita calcular la edad de una persona conociendo el año actual y el usuario digita su año de nacimiento.

```
C: > Users > gabri > Downloads > # Solicitar al usuario que ingrese un nú.py > ...  
1  # Solicitar al usuario el año actual  
2  try:  
3      anio_actual = int(input("Ingrese el año actual: "))  
4  
5      # Solicitar al usuario su año de nacimiento  
6      anio_nacimiento = int(input("Ingrese su año de nacimiento: "))  
7  
8      # Calcular la edad  
9      edad = anio_actual - anio_nacimiento  
10  
11     # Mostrar el resultado  
12     print(f"Su edad es: {edad} años")  
13 except ValueError:  
14     print("Por favor, ingrese años válidos en formato numérico.")  
15  
16
```

5 Programa para calcular el 20% de cualquier número de entrada.

```

C: > Users > gabri > Downloads > # Solicitar al usuario que ingrese un nú.py > ...
1  # Solicitar al usuario que ingrese un número
2  try:
3      numero = float(input("Ingrese un número: "))
4
5      # Calcular el 20% del número
6      porcentaje_20 = numero * 0.20
7
8      # Mostrar el resultado
9      print(f"El 20% de {numero} es: {porcentaje_20}")
10 except ValueError:
11     print("Por favor, ingrese un número válido.")
12

```

6 Programa que permita calcular el 30%, el 60% y el 90% de cualquier número.

```

C: > Users > gabri > Downloads > # Solicitar al usuario que ingrese un nú.py > ...
1  # Solicitar al usuario que ingrese un número
2  try:
3      numero = float(input("Ingrese un número: "))
4
5      # Calcular los porcentajes
6      porcentaje_30 = numero * 0.30
7      porcentaje_60 = numero * 0.60
8      porcentaje_90 = numero * 0.90
9
10     # Mostrar los resultados
11     print(f"El 30% de {numero} es: {porcentaje_30}")
12     print(f"El 60% de {numero} es: {porcentaje_60}")
13     print(f"El 90% de {numero} es: {porcentaje_90}")
14 except ValueError:
15     print("Por favor, ingrese un número válido.")
16
17

```

7 Programa para calcular el área de un cuadrado, la longitud de un lado la ingresa el usuario.

```

C: > Users > gabri > Downloads > # Solicitar al usuario que ingrese un nú.py > ...
1  # Solicitar al usuario que ingrese la longitud del lado
2  try:
3      lado = float(input("Ingrese la longitud de un lado del cuadrado: "))
4
5      # Calcular el área del cuadrado
6      area = lado ** 2
7
8      # Mostrar el resultado
9      print(f"El área del cuadrado es: {area}")
10 except ValueError:
11     print("Por favor, ingrese un número válido.")
12
13
14
15

```

8 Programa que permita ingresar 5 números y calcular el promedio.

```
C: > Users > gabri > Downloads > # Solicitar al usuario que ingrese un nú.py > ...  
1  # Inicializar la suma  
2  suma = 0  
3  
4  # Ingresar 5 números  
5  for i in range(5):  
6      numero = float(input(f"Ingrese el número {i + 1}: "))  
7      suma += numero # Sumar el número a la suma total  
8  
9  # Calcular el promedio  
10 promedio = suma / 5  
11  
12 # Mostrar el resultado  
13 print(f"El promedio de los 5 números ingresados es: {promedio}")  
14
```

9 Programa que permita a una tienda saber el valor que pagara un cliente por la compra de varios elementos de la misma referencia. Debe tener como entradas el valor unitario, la cantidad de productos comprados y al valor final se debe adicionar el 16% correspondiente al IVA.

```
C: > Users > gabri > Downloads > # Solicitar al usuario que ingrese un nú.py > ...  
1  # Solicitar al usuario el valor unitario del producto  
2  valor_unitario = float(input("Ingrese el valor unitario del producto: "))  
3  
4  # Solicitar la cantidad de productos comprados  
5  cantidad = int(input("Ingrese la cantidad de productos comprados: "))  
6  
7  # Calcular el subtotal  
8  subtotal = valor_unitario * cantidad  
9  
10 # Calcular el IVA (16% del subtotal)  
11 iva = subtotal * 0.16  
12  
13 # Calcular el valor total  
14 total = subtotal + iva  
15  
16 # Mostrar el resultado  
17 print(f"Subtotal: ${subtotal:.2f}")  
18 print(f"IVA (16%): ${iva:.2f}")  
19 print(f"Total a pagar: ${total:.2f}")  
20
```

10 Programa que permita determinar el salario a pagar a un empleado, teniendo como entradas el salario diario y el número de días trabajados. Tenga en cuenta que al empleado se le debe descontar el 10% por concepto de pensión y 15% por concepto de salud.

```

C: > Users > gabri > Downloads > # Solicitar al usuario que ingrese un nú.py > ...
1  # Solicitar al usuario el salario diario
2  salario_diario = float(input("Ingrese el salario diario del empleado: "))
3
4  # Solicitar el número de días trabajados
5  dias_trabajados = int(input("Ingrese el número de días trabajados: "))
6
7  # Calcular el salario bruto
8  salario_bruto = salario_diario * dias_trabajados
9
10 # Calcular los descuentos
11 descuento_pension = salario_bruto * 0.10
12 descuento_salud = salario_bruto * 0.15
13
14 # Calcular el salario neto
15 salario_netto = salario_bruto - (descuento_pension + descuento_salud)
16
17 # Mostrar los resultados
18 print(f"Salario bruto: ${salario_bruto:.2f}")
19 print(f"Descuento por pensión (10%): ${descuento_pension:.2f}")
20 print(f"Descuento por salud (15%): ${descuento_salud:.2f}")
21 print(f"Salario neto a pagar: ${salario_netto:.2f}")
22

```

11 Programa que solicite un número al usuario y permita calcular la raíz cuadrada del mismo (sin usar función).

```

C: > Users > gabri > Downloads > # Solicitar al usuario que ingrese un nú.py > ...
1  # Solicitar al usuario que ingrese un número
2  try:
3      numero = float(input("Ingrese un número: "))
4
5      # Verificar que el número no sea negativo
6      if numero < 0:
7          print("No se puede calcular la raíz cuadrada de un número negativo.")
8      else:
9          # Inicializar variables
10         aproximacion = numero / 2 # Suponer una aproximación inicial
11         tolerancia = 0.0001 # Tolerancia para la aproximación
12
13         # Método de aproximación (bisección)
14         while True:
15             raiz_cuadrada = (aproximacion + numero / aproximacion) / 2 # Promedio de la aproximación y el
16
17             # Verificar si la diferencia es menor que la tolerancia
18             if abs(raiz_cuadrada - aproximacion) < tolerancia:
19                 break
20
21             aproximacion = raiz_cuadrada # Actualizar la aproximación
22
23         # Mostrar el resultado
24         print(f"La raíz cuadrada de {numero} es aproximadamente: {raiz_cuadrada:.4f}")
25 except ValueError:
26     print("Por favor, ingrese un número válido.")

```

12 Calcular la hipotenusa con el Teorema de Pitágoras (sin usar funciones).

```
C: > Users > gabri > Downloads > # Solicitar al usuario que ingrese un nú.py > ...
1  # Solicitar al usuario la longitud de los dos catetos
2  try:
3      cateto_a = float(input("Ingrese la longitud del primer cateto (a): "))
4      cateto_b = float(input("Ingrese la longitud del segundo cateto (b): "))
5
6      # Calcular la hipotenusa utilizando el Teorema de Pitágoras
7      hipotenusa = (cateto_a ** 2 + cateto_b ** 2) ** 0.5
8
9      # Mostrar el resultado
10     print(f"La longitud de la hipotenusa es: {hipotenusa:.2f}")
11 except ValueError:
12     print("Por favor, ingrese valores numéricos válidos.")
13
```

13 Solicitar tiempo en segundos y transformar a horas y minutos.

```
C: > Users > gabri > Downloads > # Solicitar al usuario que ingrese un nú.py > ...
1  # Solicitar al usuario el tiempo en segundos
2  try:
3      segundos = int(input("Ingrese el tiempo en segundos: "))
4
5      # Calcular horas y minutos
6      horas = segundos // 3600
7      minutos = (segundos % 3600) // 60
8
9      # Mostrar el resultado
10     print(f"{segundos} segundos son equivalentes a {horas} horas y {minutos} minutos.")
11 except ValueError:
12     print("Por favor, ingrese un número entero válido.")
13
```

14 Solicitar al usuario una distancia en metros y transformar a km., cm. y mm.

```
C: > Users > gabri > Downloads > # Solicitar al usuario que ingrese un nú.py > ...
1  # Solicitar al usuario la distancia en metros
2  try:
3      metros = float(input("Ingrese la distancia en metros: "))
4
5      # Transformar a otras unidades
6      kilometers = metros / 1000
7      centimeters = metros * 100
8      millimeters = metros * 1000
9
10     # Mostrar los resultados
11     print(f"{metros} metros son equivalentes a:")
12     print(f"{kilometers:.3f} kilómetros")
13     print(f"{centimeters:.2f} centímetros")
14     print(f"{millimeters:.2f} milímetros")
15 except ValueError:
16     print("Por favor, ingrese un número válido.")
17
```