

Actividad de Reforzamiento: Liquidación de Sueldo con AFP y FONASA

Instrucciones:

1. Crea un archivo Python llamado "liquidacion_sueldo.py".
2. Solicita al usuario su nombre y el sueldo bruto usando un bucle while y manejo de errores con try y except.
3. Calcula los descuentos de AFP y FONASA, y el total de descuentos.
4. Calcula el sueldo líquido restando los descuentos al sueldo bruto.
5. Muestra en pantalla el desglose de los cálculos usando print() y f-strings.
6. Crea un archivo de texto con el nombre del usuario y guarda los datos de la liquidación usando write() y saltos de línea.
7. Cierra el archivo usando close().
8. Ejecuta el programa, verifica los resultados y sube el código a GitHub.

Opcional: Agregar cálculo de horas extras

1. Solicita al usuario la cantidad de horas extras trabajadas.
2. Calcula el pago por hora normal y hora extra (50% adicional).
3. Calcula el monto total de horas extras multiplicando las horas extras por el pago por hora extra.
4. Suma el monto de horas extras al sueldo bruto para obtener el sueldo bruto total.
5. Realiza los cálculos de descuentos y sueldo líquido con el sueldo bruto total.
6. Muestra y guarda el desglose de los cálculos, incluyendo las horas extras.

Liquidación de Sueldo

Sueldo Bruto: \$800,000

Horas Extras: 50.0

Monto Horas Extras: \$333,333

Sueldo Bruto Total: \$1,133,333

Descuento AFP (10%): \$113,333

Descuento FONASA (7%): \$79,333

Total Descuentos: \$192,667

Sueldo Líquido: \$940,667

1. `input()`:

- Descripción: La función `input()` se utiliza para solicitar al usuario que ingrese datos por teclado.
- Uso: `nombre_usuario = input("Ingrese su nombre: ")`. Esto mostrará el mensaje "Ingrese su nombre: " y esperará que el usuario ingrese un valor, el cual se almacenará en la variable `nombre_usuario`.

2. Bucle `while`:

- Descripción: El bucle `while` se utiliza para repetir un bloque de código mientras se cumpla una condición específica.
- Uso: `while True: ...`. Esto creará un bucle infinito que se ejecutará hasta que se encuentre una instrucción `break` o se deje de cumplir la condición.

3. `try` y `except`:

- Descripción: Las palabras clave `try` y `except` se utilizan para manejar errores y excepciones en Python. El código dentro del bloque `try` se ejecuta y, si ocurre un error, se captura en el bloque `except` para evitar que el programa se detenga abruptamente.

```
- Uso:  
```python  
try:
 sueldo_bruto = float(input("Ingrese el sueldo bruto: "))
except ValueError:
 print("Error: Ingrese un valor numérico válido.")
```
```

Esto intentará convertir la entrada del usuario a un número flotante. Si el usuario ingresa un valor no numérico, se capturará el error `ValueError` y se mostrará un mensaje de error.

4. `print()`:

- Descripción: La función `print()` se utiliza para mostrar mensajes o valores en la consola.
- Uso: `print("Liquidación de Sueldo")`. Esto mostrará el mensaje "Liquidación de Sueldo" en la consola.

5. Formato de cadenas (f-strings):

- Descripción: Los f-strings son una forma conveniente de formatear cadenas en Python. Permiten incluir variables y expresiones dentro de una cadena utilizando la sintaxis `{variable}`.
- Uso: `print(f"Sueldo Bruto: ${sueldo_bruto}")`. Esto mostrará el mensaje "Sueldo Bruto: " seguido del valor de la variable `sueldo_bruto`.

6. `open()`:

- Descripción: La función `open()` se utiliza para abrir un archivo en Python. Permite especificar el nombre del archivo y el modo de apertura (lectura, escritura, etc.).
- Uso: `archivo = open("liquidacion.txt", "w")`. Esto abrirá un archivo llamado "liquidacion.txt" en modo de escritura ("w"). Si el archivo no existe, se creará.

7. `write()`:

- Descripción: El método `write()` se utiliza para escribir contenido en un archivo abierto.
- Uso: `archivo.write("Línea de texto\n")`. Esto escribirá la cadena "Línea de texto" seguida de un salto de línea (`\n`) en el archivo.

8. `close()`:

- Descripción: El método `close()` se utiliza para cerrar un archivo abierto y liberar los recursos del sistema.
- Uso: `archivo.close()`. Esto cerrará el archivo previamente abierto.

9. `float()`:

- Descripción: La función `float()` se utiliza para convertir una cadena o un número a un número de punto flotante.
- Uso: `sueldo_bruto = float("1000")`. Esto convertirá la cadena "1000" en el número flotante 1000.0.

10. Operadores aritméticos:

- Descripción: Los operadores aritméticos se utilizan para realizar operaciones matemáticas en Python.
- Uso:
 - `+`: Suma. Ejemplo: `total = subtotal + impuestos`.
 - `-`: Resta. Ejemplo: `sueldo_liquido = sueldo_bruto - descuentos`.
 - `*`: Multiplicación. Ejemplo: `monto = cantidad * precio`.
 - `/`: División. Ejemplo: `mitad = total / 2`.