

Práctica 1

Al momento de entregar su solucionario, deben colocar todos sus soluciones en un único archivo .py

Variables:

Problema 1:

Escribir un programa que solicite su nombre de usuario por consola y después de que el usuario lo introduzca muestre por pantalla la cadena “¡Hola <nombre>!”, donde <nombre> es el nombre que el usuario haya introducido.

Problema 2:

En los Estados Unidos, se acostumbra dejar una propina a su mesero después de cenar en un restaurante, generalmente una cantidad igual al 15% o más del costo de su comida.

Escriba un programa que pregunte al usuario cuánto fue su consumo en el restaurante y que porcentaje de propina desea dejar al mesero. Su programa debe devolver la cantidad de dinero a dejar como propina.

Problema 3:

Una juguetería tiene mucho éxito en dos de sus productos: payasos y muñecas. Suele hacer venta por correo y la empresa de logística les cobra por peso de cada paquete así que deben calcular el peso de los payasos y muñecas que saldrán en cada paquete a demanda. Cada payaso pesa 112 g y cada muñeca 75 g. Escribir un programa que lea el número de payasos y muñecas vendidos en el último pedido y calcule el peso total del paquete que será enviado.

Problema 4:

Escribir un programa que lea un entero positivo, N, introducido por el usuario y después muestre en pantalla la suma de todos los enteros desde 1 hasta N. La suma de los N primeros enteros positivos puede ser calculada de la siguiente forma:

$$1 + 2 + 3 + \dots + n = \frac{n(n+1)}{2}$$

Condicionales:

Problema 5:

Escribir un programa que solicite al usuario que ingrese cuántos shows musicales ha visto en el último año y almacene ese número en una variable. A continuación, mostrar en pantalla un valor de verdad (True o False - tipo de datos booleanos) que indique si el usuario ha visto más de 3 shows.

Problema 6:

Escribir un programa para una empresa que tiene salas de juegos para todas las edades y quiere calcular de forma automática el precio que debe cobrar a sus clientes por entrar. El programa debe preguntar al usuario la edad del cliente y mostrar el precio de la entrada. Si el cliente es menor de 4 años puede entrar gratis, si tiene entre 4 y 18 años debe pagar \$5 y si es mayor de 18 años, \$10.

Problema 7:

Realiza un programa que lea dos números por teclado y permite elegir entre 3 opciones en un menú:

- Mostrar una suma de los dos números
- Mostrar una resta de los dos números (el primero menos el segundo)
- Mostrar una multiplicación de los dos números
- En caso de introducir una opción inválida, el programa informará de que no es correcta.

Problema 8:

Supongamos que estás en un país donde se acostumbra desayunar entre las 7:00 y las 8:00, almorzar entre las 12:00 y las 13:00 y cenar entre las 18:00 y las 19:00.

Implemente un programa que solicite al usuario una hora y le indique si es la hora del desayuno, la hora del almuerzo o la hora de la cena. Si no es hora de comer, no envíes nada.

Suponga que la entrada del usuario se formatea en formato de 24 horas como #:## o ##:##. Y suponga que el rango de tiempo de cada comida es inclusivo. Por ejemplo, ya sean las 7:00, las 7:01, las 7:59 o las 8:00, o en cualquier momento intermedio, es hora de desayunar.

Nota:

Recuerde que cuando solicitamos datos a través de un input este nos devuelve un str, quizás le sea más fácil realizar la comparación convirtiendo los datos a float.

Ejemplo :

Dato ingresado: "7:30" se puede interpretar como 7.5

Para ello, podemos usar el siguiente método de cadena para facilitar la obtención del dato. Puede revisar otros métodos de cadena en el [siguiente link](#).

```
time = input("hora")  
hours, minutes = time.split(":")
```

Colecciones de Datos:

Problema 9:

Escriba un programa que, dada una lista, devuelve una nueva lista cuyo contenido sea igual a la original pero invertida. Así, dada la lista ['Di', 'buen', 'día', 'a', 'papa'], deberá devolver ['papa', 'a', 'día', 'buen', 'Di'].

Problema 10:

Escriba un programa para imprimir una lista específica después de eliminar los elementos que se encuentran en la posición 0, 4 y 5.

lista de muestra: ['Rojo', 'Verde', 'Blanco', 'Negro', 'Rosa', 'Amarillo']

Resultado esperado: ['Verde', 'Blanco', 'Negro']

Problema 11:

Escriba un programa Python que se encargue de eliminar los elementos duplicados de la siguiente lista. Su programa debe retornar otra lista sin los duplicados.

Lista original: [1, 1, 2, 3, 4, 4, 5, 1]

Lista procesada: [1, 2, 3, 4,, 5]

Problema 12:

La mayoría de los archivos tienen extensiones de archivo, el cual es un sufijo que comienza con un punto (.) al final de su nombre. Por ejemplo, los nombres de archivo para GIF terminan en .gif y los nombres de archivo para JPEG terminan en .jpg o .jpeg. Mientras que en los sistemas operativos como Windows, el tipo de archivo le sirve al computador abrir el archivo en el formato apropiado, en la web esto es distinto. Los navegadores web, por el contrario, se basan en tipos de medios, anteriormente conocidos como tipos MIME, para determinar cómo mostrar los archivos que viven en la web.

Implemente un programa que solicite al usuario el nombre de un archivo y luego genere el tipo de archivo MIME correspondiente. Si el nombre del archivo termina en cualquiera de estos sufijos (sin importar el uso de mayúsculas y minúsculas) :

- .gif
- .jpg
- .jpeg
- .png
- .pdf
- .txt
- .zip

Si el nombre del archivo termina con algún otro sufijo que no se encuentra en la lista o no tiene ningún sufijo, en su lugar su programa deberá devolver **application/octet-stream**.

Ejemplos:

Nombre Archivo: happy.jpg **Salida Esperada:** image/jpeg

Nombre Archivo: document.pdf **Salida Esperada:** application/pdf

Para conocer los tipos MIME a utilizar, puede revisar el [siguiente enlace](#).

Nota:

- El empleo de diccionarios podría ayudar con esta tarea.
- El uso de métodos de cadena sería muy útil al momento de separar el nombre del archivo de su extensión.