

Ejercicio 1:

Crear un programa que imprima el mensaje “hello world!” por pantalla.

Ejercicio 2:

Crear un programa que almacene el mensaje “hello world!” en una variable y luego lo imprima por pantalla.

Ejercicio 3:

Crear un programa que pida al usuario nombre y edad e imprima dichos datos en renglones distintos.

Ejercicio 4:

Crear un programa que pregunte el nombre del usuario y un número entero e imprima por pantalla en líneas distintas el nombre del usuario tantas veces como el número introducido.

Ejercicio 5:

Crear un programa que pregunte el nombre del usuario y después de que el usuario lo introduzca muestre por pantalla <NOMBRE> tiene <n> letras.

Ejercicio 6:

Crear un programa que realice la siguiente operación aritmética $[(3+2)/2*5]^2$. Mostrar el resultado por pantalla.

Ejercicio 7:

Crear un programa que pida al usuario dos números enteros y muestre por pantalla la <n> entre <m> da un cociente <c> y un resto <r> donde <n> y <m> son los números introducidos por el usuario, y <c> y <r> son el cociente y el resto de la división entera respectivamente.

Ejercicio 8:

Crear un programa que pregunte al usuario una cantidad a invertir, el interés anual y el número de años, y muestre por pantalla el capital obtenido en la inversión.

Ejercicio 9:

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 el 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. Crear 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. Mostrar el resultado por pantalla.

Ejercicio 10:

Crear un programa que pida al usuario su edad y muestre por pantalla si es mayor de edad o no, siendo 18 años la mayoría de edad.

Ejercicio 11:

Crear un programa que almacene la cadena de caracteres *contraseña* en una variable, pregunte al usuario por la contraseña e imprima por pantalla si la contraseña introducida por el usuario coincide con la guardada en la variable.

Ejercicio 12:

Crear un programa que pida al usuario dos números y muestre por pantalla su división. Si el divisor es cero, el programa debe devolver un mensaje indicando que no se puede dividir por 0.

Ejercicio 13:

Crear un programa que pida al usuario un número entero y muestre por pantalla si es par o impar.

Ejercicio 14:

En una determinada empresa, sus empleados son evaluados al final de cada año. Los puntos que pueden obtener en la evaluación comienzan en 0.0 y pueden ir aumentando, traduciéndose en mejores beneficios. Los puntos que pueden conseguir los empleados pueden ser 0.0, 0.4, 0.6 o más, pero no valores intermedios entre las cifras mencionadas. A continuación se muestra una tabla con los niveles correspondientes a cada puntuación. La cantidad de dinero conseguida en cada nivel es de 100000 multiplicada por la puntuación del nivel.

Nivel	Puntuación
Inaceptable	0.0
Aceptable	0.4
Meritorio	0.6 o más

Crear un programa que lea la puntuación del usuario e indique su nivel de rendimiento, así como la cantidad de dinero que recibirá el usuario.

Ejercicio 15:

Crear 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 500 y si es mayor de 18 años, 1000.

Ejercicio 16:

Crear un programa que pregunte al usuario su edad y muestre por pantalla todos los años que ha cumplido (desde 1 hasta su edad).

Ejercicio 17:

Crear un programa que muestre por pantalla la tabla de multiplicar del 1 al 10.

Ejercicio 18:

Crear un programa que pida al usuario una palabra y luego muestre por pantalla una a una las letras de la palabra introducida empezando por la última.

Ejercicio 19:

Crear un programa en el que se pregunte al usuario por una frase y una letra, y muestre por pantalla el número de veces que aparece la letra en la frase.

Ejercicio 20:

Crear un programa que muestre el eco de todo lo que el usuario introduzca hasta que el usuario escriba “salir” que terminará.

Ejercicio 21:

Crear un programa que almacene las asignaturas de un curso (por ejemplo Matemáticas, Física, Química, Historia y Lengua) en una lista y muestre por pantalla el mensaje “Yo estudio <asignatura>”, donde <asignatura> es cada una de las asignaturas de la lista.

Ejercicio 22:

Crear un programa que almacene los vectores (1,2,3) y (-1,0,2) en dos listas y muestre por pantalla su producto escalar.

Ejercicio 23:

Crear un programa que guarde en un diccionario los precios de las frutas de la tabla, pregunte al usuario por una fruta, un número de kilos y muestre por pantalla el precio

de ese número de kilos de fruta. Si la fruta no está en el diccionario debe mostrar un mensaje informando de ello.

Fruta	Precio (kg)
Banana	80
Manzana	100
Pera	50
Naranja	70

Ejercicio 24:

Crear un programa que almacene el diccionario con los créditos de las asignaturas de un curso {'Matemáticas': 6, 'Física': 4, 'Química': 5} y después muestre por pantalla los créditos de cada asignatura en el formato <asignatura> tiene <créditos> créditos, donde <asignatura> es cada una de las asignaturas del curso, y <créditos> son sus créditos. Al final debe mostrar también el número total de créditos del curso.

Ejercicio 25:

Crear un programa que, a partir de una lista con todas las letras del abecedario, haga una copia y borre de esta última todas las vocales. Luego imprimir por pantalla ambas listas, la completa y la sin vocales.

Para crear la lista del abecedario se puede importar la variable `ascii_lowercase` del módulo `string`:

```
from string import ascii_lowercase

lista_abc = list(ascii_lowercase)
print(lista_abc)
```

```
['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r', 's', 't', 'u', 'v', 'w', 'x', 'y', 'z']
```