

## 14. Errores y excepciones

1. Localiza el error en el siguiente bloque de código. Crea una excepción para evitar que el programa se bloquee y que explique al usuario la causa y/o solución:

2. resultado = 10/0

4. No es posible dividir entre cero, debes introducir un número distinto.

2. Partimos de la siguiente lista:

6. lista=['gato', 'perro', 'ratón', 'pato', 'elefante']

7. Crea un programa que solicite el índice de un elemento de la lista y que a continuación muestre dicho elemento por pantalla.

8. ['gato', 'perro', 'ratón', 'pato', 'elefante']

9. Introduce el índice del elemento de la lista que quieres mostrar por pantalla:3

10. pato

11. En caso de que el índice sea un número entero demasiado mayor (>4) se producirá un error. Crea una excepción para evitar que el programa se bloquee y que explique al usuario la causa y solución.

13. ['gato', 'perro', 'ratón', 'pato', 'elefante']

14. Introduce el índice del elemento de la lista que quieres mostrar por pantalla:5

15. El índice se encuentra fuera del rango.

16. Debes utilizar un número mayor o igual que cero y menor que la longitud de la lista.

3. (Necesario haber visto diccionarios). Partimos del siguiente diccionario:

18. colores = { 'rojo':'red', 'verde':'green', 'negro':'black' }

19. Crea un programa que solicite un término del diccionario para mostrar su valor asociado por pantalla:

20. Introduce el nombre de un color en español: azul

21. azul en inglés se dice blue

22. En caso de que el término no esté incluido en el diccionario se producirá un error. Crea una excepción para evitar que el programa se bloquee y que explique al usuario la causa y solución.

23. Introduce el nombre de un color en español: amarillo

24. Error: El término amarillo no se encuentra en este diccionario, debes probar con otro que sí exista.

4. Crea un programa que solicite dos números enteros por pantalla y muestre el resultado de su suma. En caso de que uno de los valores introducidos sea una cadena de caracteres el programa debe gestionar el error informando de la causa y solución.

26. Introduce el primer valor: 5

27. Introduce el segundo valor: 4

28. 5 + 4 = 9

```
30. Introduce el primer valor: 3
31. Introduce el segundo valor: 2.5
32. El valor introducido no es un número entero
```

5. Partiremos de la siguiente lista:

```
34. elementos = [1, 'pie', -2]
```

Realiza una función llamada `agregar_una_vez(lista, elemento)` que reciba como parámetros una lista y un elemento. La función debe añadir el elemento al final de la lista con la condición de no repetir ningún elemento. Además, si este elemento ya se encuentra en la lista se debe invocar un error de tipo `ValueError` que debes capturar y mostrar este mensaje en su lugar: `Error: Imposible añadir elementos duplicados => [elemento]`.

Cuando tengas la función crea un programa que permita añadir elementos a la lista de forma indefinida. Intenta añadir los siguientes valores a la lista: 10, "pie", "Hola" y luego muestra su contenido.

Sugerencia: Puedes utilizar la sintaxis "elemento in lista", si el elemento está en la lista genera un valor `True`, si no lo está genera un valor `False`

```
35. La lista actual es: [1, 'pie', -2]
36. Valor que quieres añadir a la lista: 10
37. La lista actual es: [1, 'pie', -2, '10']
38. Valor que quieres añadir a la lista: pie
39. Error: Imposible añadir elementos duplicados => pie
40. La lista actual es: [1, 'pie', -2, '10']
41. Valor que quieres añadir a la lista: Hola
42. La lista actual es: [1, 'pie', -2, '10', 'Hola']
```

6. Escribe el código de una función que ha de ser capaz de aceptar números enteros positivos. La función ha de chequear que el número introducido cumple la condición dada. La función ha de ser capaz de gestionar los valores que no sean enteros o que sean enteros negativos o si se introduce un texto o si pulsamos "enter" sin haber introducido un valor, informando del error.

7. Mejora el programa anterior consiguiendo que en caso de excepción la función solicite un nuevo valor hasta que este tenga el formato correcto.

```
45. Introduce un número entero positivo: -5
46. El valor es número entero negativo
47. Introduce un número entero positivo: 5.2
48. El número es decimal
49. Introduce un número entero positivo: hola
50. El valor introducido no es un número
51. Introduce un número entero positivo: 4
52. 4
```

8. Ejecuta el programa que desarrollaste en el ejercicio número 4 de la hoja 13. Módulo `random` e introduce como cantidad de dinero a apostar un valor negativo ¿qué ocurre? Prueba ahora a introducir una cadena de texto ¿algún problema? Gestiona estas dos excepciones de tal forma que el programa funcione de forma correcta. Es decir, cuando

introducamos un valor no apropiado, el programa en lugar de interrumpirse debe indicarnos cual es el error y solicitar un valor adecuado. Este proceso se debe repetir hasta que el usuario utilice el programa de forma correcta.