



11071: Introducción a la Programación

Departamento de Ciencias Básicas

Universidad Nacional de Luján

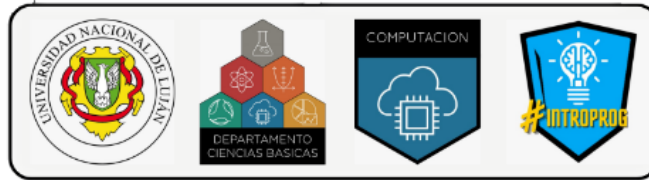
Trabajo Práctico IX

Validación de entrada

Hasta el momento asumimos que el usuario, cuando le solicitamos ingresar algún dato por teclado, siempre nos provee un valor válido, o en el peor de los casos, podemos validar una única vez su entrada. Cuando el valor era inválido, nos limitábamos a terminar el programa. Sin embargo, en un contexto real, debemos contemplar la posibilidad de que el usuario ingrese datos inválidos más de una vez, y el programa debe seguir funcionando. En forma general, **validar un dato** significa:

- Verificar que es del tipo correcto. Esto es importante ya que si, por ejemplo, intentamos realizar una operación numérica sobre un valor alfanumérico, se producirá un **error en tiempo de ejecución** por incompatibilidad de tipos.
- Verificar si el valor está dentro del rango permitido. Una vez que sabemos que el dato es del tipo correcto, debemos verificar que el valor esté dentro del rango que aceptaremos como válido. Por ejemplo, una nota de parcial tiene que ser un número entero entre 1 y 10 (el valor 12 sería inválido); o un día de la semana debería ser un string y coincidir con el literal "lunes", "martes", "miércoles", "jueves", "viernes", "sábado", o "domingo" (el valor "osvaldo" sería inválido).
- Reaccionar a una entrada inválida. Al detectar que el usuario ha ingresado un valor inválido, debemos decidir qué hacer. Para el contexto de este trabajo práctico, nos valdremos de las estructuras repetitivas para permitir el reingreso de los datos.

1. Cree un script que le solicite al usuario ingresar una temperatura en grados Celsius, y valide que la entrada es correcta, teniendo en cuenta que la temperatura debe ser un valor numérico, y el rango válido está entre -18 y 50. El programa debe permitir reingresar el dato cuantas veces sea necesario, hasta que el usuario provea un dato válido. Procure informar al usuario cuando su dato es inválido, y cuáles son los valores aceptados.
2. Cree un script que le solicite al usuario ingresar su edad. Verifique que el dato ingresado sea válido, teniendo en cuenta que la edad es un número entero, y el rango válido para este programa es de 18 a 60 años. El programa debe solicitar el reingreso de manera indefinida, hasta que el dato sea correcto.
3. Modifique todos los ejercicios anteriores para que en lugar de permitir el reintento de manera ilimitada, el programa permita sólo 10 reintentos. Si el usuario supera el límite de reintentos, el programa debe terminar con el mensaje "Usted está jugando conmigo, yo me retiro".

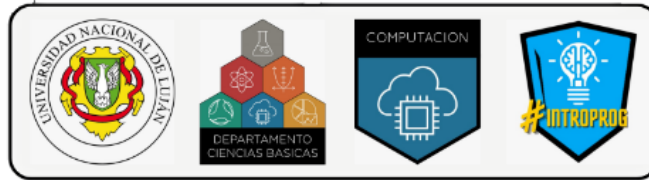


11071: Introducción a la Programación
Departamento de Ciencias Básicas
Universidad Nacional de Luján

4. La técnica de validación para un conjunto específico de valores se puede utilizar para construir menús de opciones. Construya un menú que le muestre al usuario lo siguiente:

```
***** MI PROGRAMA *****  
1. Saludar.  
2. Informar temperatura.  
3. Mostrar nombre de materia.  
4. Salir.  
  
Seleccione una opción [1-4]:
```

- Cuando el usuario ingrese la opción 1, se mostrará el mensaje *"Hola, bienvenido a mi programa interactivo!"*.
 - Cuando el usuario ingrese la opción 2, se mostrará el mensaje *"Hay una sensación térmica de 20 grados Celsius."*.
 - Cuando el usuario ingrese la opción 3, se mostrará el mensaje *"Estás en la materia Introducción a la Programación!"*.
 - Cuando el usuario ingrese la opción 4, el programa debe terminar, mostrando el mensaje *"Hasta la próxima!"*.
 - Si el usuario ingresa una opción inválida, se muestra el mensaje *"Opción inválida."*.
5. Cree un script que le solicite al usuario ingresar cuál es su color favorito, limitando las opciones a *rojo*, *verde*, y *azul*. Aclaraciones:
- Puede asumir que el usuario ingresará los strings en minúsculas. Opcionalmente, puede investigar el uso de las funciones **upper()** y **lower()** para transformar la entrada a mayúsculas o minúsculas y evitar así posibles errores de validación por este detalle.
 - Al validar entre un conjunto de opciones prefijadas (en lugar de hacerlo en un rango), es posible que no sea necesario validar el tipo del dato ingresado por teclado.
 - Al detectar un dato inválido, el programa debe darle las siguientes opciones al usuario:



11071: Introducción a la Programación
Departamento de Ciencias Básicas
Universidad Nacional de Luján

```
** DATO INVÁLIDO **  
1. Reintentar.  
2. Salir.
```

- La opción **1. Reintentar** le permite al usuario reingresar el dato de manera indefinida, siempre mostrando las opciones ante cada intento fallido.
- La opción **2. Salir** finaliza el programa.

6. Al implementar programas interactivos, es normal que la pantalla se llene de texto, en detrimento de la experiencia de usuario. Para solucionar este inconveniente, Python nos provee una función para *limpiar* la pantalla (notar que esto no tendrá ningún efecto sobre la lógica del programa, simplemente dejará la terminal de comandos en blanco). El uso de esta función depende del sistema operativo que esté utilizando, pero para empezar deberá importar el módulo **os**, escribiendo la siguiente línea al comienzo de su script:

```
import os
```

Una vez importado el módulo **os**, ya podrá utilizar la función para limpiar la pantalla en cualquier punto de su programa.

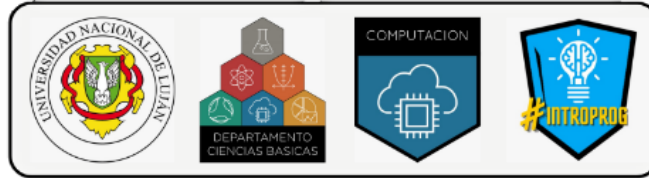
- En **Windows**:

```
os.system('cls')
```

- En **Linux**:

```
os.system('clear')
```

Modifique todos los ejercicios de este TP para que cada vez que se reintente una entrada o se muestre un menú de opciones, la pantalla esté limpia.



11071: Introducción a la Programación
Departamento de Ciencias Básicas
Universidad Nacional de Luján

¡A testear!



En los siguientes ejercicios aprenderemos a crear nuestros propios tests unitarios, y los vamos a definir con la siguiente estructura:

```
def test<NombreDeLaFuncion> ():  
  
    assert funcion1(parametros) == <resultadoEsperado>  
    assert funcion1(parametros) == <resultadoEsperado>  
    assert funcion1(parametros) == <resultadoEsperado>  
  
    print("Paso")
```

1. Rehaga el **ejercicio 5 (color favorito)** considerando los siguientes puntos:
 - Cree la función `color_favorito`, que deberá recibir como parámetro un string `s` y validar si ese string coincide con alguna de las opciones. Deberá retornar **True** si es valido y **False** en caso contrario
 - Omita el ingreso por teclado, sino que el valor deberá entrar como parámetro de la función.
2. Cree un test unitario para el **ejercicio 5**.