



Argentina
programa
4.0

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

Trabajo Práctico VIII

Estructuras repetitivas condicionales

Existen situaciones en las que no está claro, o no es posible conocer con antelación, cuántas veces vamos a repetir un bloque de código. En estos casos debemos utilizar una estructura que nos permita repetir código "hasta que se cumpla determinada condición". En Python, esa estructura es el **while**, que tiene la siguiente forma:

```
while [condición]:  
    [código a repetir]
```

En donde:

- **while** es una palabra reservada que utilizamos para indicar que vamos a repetir un bloque de código de manera condicional.
- La **condición** es una expresión booleana (verdadera o falsa), que se evaluará antes de cada iteración, de manera que si es verdadero, el bloque se ejecutará una vez más, y si es falsa, el bucle finaliza, y el programa continúa su ejecución.
- El **código a repetir** es el bloque de instrucciones que se ejecutará en cada iteración del bucle (siempre que la condición sea verdadera).

A las expresiones o variables booleanas que se utilizan como condición de un **while** se las suele llamar **bandera**, **flag**, o **guarda**, entre otros sinónimos.

Las iteraciones condicionales se pueden utilizar para mejorar la forma en que **validamos la entrada del usuario**, ya que podemos, por ejemplo, repetir el código para leer y validar el dato a leer, hasta que el mismo sea válido.

1. Cree un script que le pida al usuario ingresar palabras, una a una, hasta que el usuario ingrese la palabra "parar". A medida que se van ingresando las palabras, el programa simplemente debe mostrarlas en pantalla. Al detectar la palabra para detenerse, debe mostrar el mensaje "--- TERMINADO ---".
2. Cree un script que le solicite al usuario ingresar notas de parciales por teclado, hasta que el usuario ingrese el valor -1, indicando que ya no hay más notas para cargar. Una vez ingresadas las notas, el programa debe informar la nota promedio (tenga cuidado de no incluir al -1 dentro del promedio).
3. Cree un script que le solicite al usuario leer un número entero entre 1 y 100. El programa debe ser capaz de solicitarle al usuario que reingrese el número cuantas veces sea necesario, hasta que el usuario provea un dato válido. Cada vez que detecte un error de validación, informe al usuario cuál fue el error, con los mensajes "El dato ingresado no es numérico.", o "El número ingresado está fuera del rango permitido.". Finalmente, cuando el usuario ingrese un dato válido, muestre el mensaje "[NÚMERO] es válido. Gracias!".



Argentina
programa
4.0

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

4. Vuelva a construir el menú del ejercicio 6 del Trabajo Práctico VI, pero esta vez agregue una nueva opción llamada “Salir”. El programa deberá mostrar el menú y reaccionar a la opción seleccionada por el usuario, pero ahora, al ejecutar una opción, en vez de terminar el programa, se mostrará nuevamente el menú, hasta que el usuario seleccione la opción 4. *Salir*.

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

Tip 1: cree al menos una función propia, que se encargue exclusivamente de mostrar el menú de opciones en pantalla.

Tip 2: puede utilizar la instrucción `os.system('cls')` para limpiar la pantalla antes de volver a mostrar el menú. De esta manera su programa será más legible en la terminal. Para poder utilizar dicha instrucción deberá importar, al principio del script, la librería `os` de la siguiente manera:

```
import os  
# su código
```

Tip 3: antes de limpiar la pantalla y volver a mostrar el menú, puede esperar a que el usuario confirme que ha terminado de leer la información presentada, simplemente utilizando la función `input('¡PRESIONE ENTER PARA CONTINUAR!')`.

5. Si bien el **while** es útil cuando desconocemos la cantidad de veces que repetiremos un bloque de instrucciones, también puede ser utilizado en los mismos casos que es utilizado el **for** (aunque la inversa no es verdadera). **Rehaga todos los ejercicios del Trabajo Práctico VIII utilizando un *while* en lugar de un *for*.**