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

Última versión: marzo 2021.

Trabajo Práctico VI

Estructura alternativa y operadores lógicos

*Para tomar decisiones dentro de un algoritmo utilizaremos la sentencia condicional **if**:*

- ***if** evalúa el resultado de una condición booleana, y ejecuta una porción de código contenido en ella si el resultado es "verdadero" (true).*
- *La condición de la sentencia **if** puede estar compuesta por varias subcondiciones; cada condición booleana se relaciona con los operadores lógicos como **conjunción** (and) o la **disyunción** (or).*
- *Se puede especificar un código a ejecutar si la condición evaluada es "falsa" (false), utilizando la sentencia **else**.*
- *Es posible anidar sentencias **if** y **else** de manera infinita.*
- *En varias ocasiones será útil incluir un nuevo **if** en la sentencia **else**. Python nos permite abreviar esto escribiendo **elif (condición):**.*
- *Recordar que en Python **la indentación es parte de la sintaxis**; esto es muy importante para saber qué instrucciones están dentro del bloque de un **if**, **else**, o **elif**.*

---

1. Cree un script que le solicite al usuario ingresar un número por teclado, y luego le informe en pantalla si su número es mayor que 10; antes de finalizar e independientemente de lo que haya sucedido antes, el script mostrará un mensaje de despedida. Ejemplos de cómo debería verse la salida del script:

Número mayor que 10:

```
Tu número (N) es mayor que 10!

Saludos!
```

Número menor o igual que 10:

```
Saludos!
```

2. Modifique el script anterior para que mantenga el funcionamiento, pero ahora, cuando el número no es mayor que 10, también se muestre un mensaje "*Tu número (N) es menor o igual que 10!*".

3. Cree un script que le solicite al usuario ingresar dos números por teclado, y luego indique por pantalla cuál de ellos es el mayor. Contemple la posibilidad de que los números sean iguales, y muestre un mensaje acorde.

4. Cree un script que le solicite al usuario ingresar un número por teclado, y le informe con un mensaje si su número es positivo, negativo, o 0.

5. Cree un script que, dado un número de día de la semana ingresado por teclado, muestre el nombre de ese día en lenguaje natural. La relación entre números y días de la semana es la siguiente:

```
1 = Domingo.
2 = Lunes.
3 = Martes.
4 = Miércoles.
5 = Jueves.
6 = Viernes.
7 = Sábado.
```

6. Cree un script que le solicite a un alumno de la asignatura *Introducción a la Programación* que ingrese las notas de sus dos parciales. Como resultado, se le debe informar al alumno su *situación*, junto con la nota promedio. Las reglas para saber la situación de un alumno son las siguientes:

   - Para estar **promovido** (es decir, cursada aprobada y no requiere rendir final), el alumno debe haber aprobado ambos parciales y tener un promedio mayor o igual a 8.

   - Para estar **regular** (cursada aprobada, pero debe rendir final), el alumno debe haber aprobado ambos parciales (nota mayor o igual a 4).

   - Si el alumno no ha aprobado ambos parciales (es decir, tiene nota menor que 4 en alguno de ellos), entonces queda en condición de **libre** (es decir, puede rendir un final extendido o recursar).

7. Cree un script que determine si un triángulo es *isósceles, equilátero,* o *escaleno*. Para determinar esto, se le solicitará al usuario ingresar tres números, correspondientes a los tres lados del triángulo.

```
equilátero    = todos los lados iguales
isósceles     = dos lados iguales
escaleno      = todos los lados diferentes
```

8. **Las estructuras alternativas pueden utilizarse para *validar datos***. Por ejemplo, si se intenta crear una función que tome dos enteros como parámetro y muestre el resultado de su división, puede ocurrir un error si el denominador es cero. Utilice la estructura alternativa para validar que el denominador no sea cero; en caso de serlo, la función deberá mostrar el mensaje “No se puede dividir por 0!” en lugar de intentar mostrar el resultado.

9. Escriba una función que reciba un entero positivo como parámetro, y muestre en pantalla el resultado de su factorial. El factorial de un entero positivo *n* es:

```
n! = 1 * 2 * 3 * 4 * … * (n-1) * n
```

Si la función no recibe un número válido, deberá mostrar el mensaje "Parámetro inválido, sólo se aceptan números enteros positivos".

10. Python ofrece algunas funciones built-in para facilitar la implementación de validaciones. A continuación se listan algunas de las más comunes:

valor`.isdigit()`
Retorna *True* si todos los caracteres de *valor* son numéricos, *False* en caso contrario.

valor`.isalpha()`
Retorna *True* si todos los caracteres de *valor* son alfabéticos (no numéricos), *False* en caso contrario.

valor`.isalphanum()`
Retorna *True* si *valor* es una combinación alfanumérica (caracteres y números), *False* en caso contrario.

Codifique una función que reciba un parámetro cualquiera (que puede contener letras, números, o una combinación de ambas), e indique si el mismo es un número, una palabra, o un valor alfanumérico. Compruebe que su función resuelve el problema enviándole valores correspondientes a las tres posibilidades.

11. Modifique el script del ejercicio anterior para que el valor evaluado provenga del usuario de programa.