

Laboratorio

Práctica 3. Programación con Python

Unidad 3. Introducción a Python y a la eficiencia de los programas

1. Objetivos

- Desarrollar programas que involucren elementos básicos de programación: expresiones con operadores aritméticos y relacionales, entrada y salida de datos, entre otros.
- Desarrollar programas de baja a mediana complejidad con estructuras algorítmicas secuenciales, condicionales e iterativas.
- Aprender a utilizar un IDE (Integrated Development Environment) para el desarrollo de programas.
- Aprender a utilizar un depurador.

2. Marco teórico

En esta práctica se pondrán en uso los conceptos fundamentales que forman la base de la programación en cualquier lenguaje. Más precisamente, la práctica tratará los siguientes aspectos:

- Cómo capturar datos que el usuario ingrese: función `input()`
- Cómo mostrarle datos al usuario en la consola: función `print()`
- Cómo hacer operaciones matemáticas simples con los datos: operadores `+`, `-`, `*`, `/`, `//` y `%`
- Cómo escribir instrucciones que condicionen la ejecución del programa: instrucciones `if` y `else`
- Cómo escribir segmentos de código que se repitan: instrucciones `while` y `for`

3. Tareas a realizar

Utilice el lenguaje Python para implementar todos los algoritmos que desarrolló en la Práctica 2.

Adicionalmente, desarrolle un algoritmo y su correspondiente programa en Python para los ejercicios extra que se listan abajo. Envíe en un archivo comprimido con todos los programas desarrollados, usando el recurso tarea en el aula virtual.

Siga los siguientes pasos para resolver cada uno de los siguientes problemas:

- a. Haga un análisis del problema y determine datos de entrada y de salida.
- b. Plantee el algoritmo en un diagrama de flujo.
- c. Hágle una prueba de escritorio.
- d. Implemente el algoritmo diseñado en un programa en Python.
- e. Hágle pruebas para verificar su funcionamiento.
- f. Utilice el depurador para encontrar las causas de los posibles fallos.

3.1. Escriba un programa que tome un carácter (es decir, un string de longitud 1) y determine si el carácter es vocal o consonante.

Test

A continuación, se muestran algunos valores de test para verificar el programa:

Carácter	Salida
'a'	Vocal
'z'	Consonante
'b'	Consonante
'U'	Vocal
'#'	

Ayuda

- Recuerde considerar mayúsculas y minúsculas.
- Para el test puede hacer uso del operador `in`.

3.2. Escriba un programa que le pida al usuario una palabra o frase y una letra. El programa deberá imprimir la misma frase o palabra ingresada, pero ocultando la letra que ingresó el usuario con un asterisco.

Test

A continuación, se muestran algunos valores de test para el problema:

Palabra o frase	Letra	Salida
3 caballos tienen 12 patas	a	3 c*b*llos tienen 12 p*t*s
abstraer	r	abst*ae*
racionalizar	p	racionalizar

Ayuda

- Recuerde los conceptos de indexación y concatenación de strings.
- Puede construirse un nuevo string (al inicio vacío) para mostrarlo al final o mostrar en cada iteración una letra.

- 3.3. Haga un programa que determine si una palabra ingresada por el usuario es palíndroma o no. Utilice la instrucción while.
- 3.4. Reciba una palabra del usuario y un número entero menor que 26. El programa debe cambiar cada letra por la que le corresponda al dar saltos en el alfabeto de acuerdo al número especificado por el usuario. La nueva palabra codificada deberá ser mostrada en pantalla.

Test

A continuación, se muestran algunos valores de test para verificar el programa:

Palabra	Número	Salida
compilador	1	dpnqjmbeps
carro	25	bzqqn
binario	3	elqdulr

Ayuda

- Cree una cadena con las 26 letras del alfabeto y utilice index() para averiguar la posición de cada letra que necesite.

4. Evaluación

La evaluación se basará en la sustentación de los programas desarrollados, donde también se demuestre el manejo del depurador. Además, se hará un quiz individual sobre los aspectos del lenguaje Python cubiertos en la práctica.