

Práctica 3. Programación con Python

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`
- Cómo acceder a los elementos de un `string`

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) Hágale una prueba de escritorio.
- d) Implemente el algoritmo diseñado en un programa en Python.

- e) Hágale 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, consonante o un dígito. Además, para las vocales y consonantes determine si es mayúscula o minúscula.

Test:

Entrada	Salida
a	Vocal - Minúscula
X	Consonante - Mayúscula
8	Dígito
\$	-

Puede hacer uso del operador **in**

3.2. Realice un programa que permita cambiar los espacios " " en una oración por un guión "-"

Nota: No se permite el uso de la instrucción **replace()**, solo instrucciones básicas.

Test:

Entrada	Salida
Solucionario de los ejercicios	Solucionario-de-los-ejercicios
Universidad	Universidad

3.3. Se necesita verificar la seguridad de una contraseña siguiendo las siguientes condiciones:

- Su longitud debe estar entre 8 y 20 caracteres.
- Debe incluir letras, números.
- Debe incluir al menos uno de los siguientes caracteres especiales:
!, ", #, \$, %, &, /, (,), =, ?, +, *
- No debe ser palíndroma.

Finalmente debe indicar en pantalla si ha cumplido los requisitos de forma individual.

Ejemplo 1:

```
>>Ingrese contraseña: c4ratxmu!

Longitud correcta: ok
Incluye letras y números: ok
Incluye caracteres especiales: ok
No es palíndroma: ok

La contraseña se ha creado satisfactoriamente!!!
```

Ejemplo 2:

```
>>Ingrese contraseña: a0r?ene?r0a

Longitud correcta: ok
Incluye letras y números: ok
Incluye caracteres especiales: ok
No es palíndroma: error

La contraseña no cumple todos los requisitos!!!
```

3.4. Crea un algoritmo para ocultar mensajes. El programa debe recibir un mensaje y reorganizar los caracteres de tal forma que las letras con índices pares queden al inicio y las impares después. Por ejemplo la palabra “**secreto**” quedará como “**sceoert**” debido a los caracteres **s,c,e,o** corresponden a los índices **0,2,4,6**, y los caracteres **e,o,r,t** corresponden a los índices **1,3,5**. Además si el caracter es una vocal, debe cambiarla por su posición dentro de las vocales, es decir **a = 1, e = 2, i = 3, o = 4, u = 5**.

Ejemplo:

```
Ingresa mensaje: No traduzcas esto manualmente

Mensaje encriptado: N4 t15csrdz1 2ts4 mn1mn21512t
```

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.