

Trabajo practico N° 1

Ejercicio 1

Implementar un programa que permita generar cuatro números aleatorios y calcule el promedio.

Ejercicio 2

El lenguaje Python provee diversas formas de utilizar cadenas para mostrar resultados en la pantalla. Considerando que se tienen las variables marca = "Motorola", modelo = "G8", memoria = "3", indicar tres o más formas distintas de generar la frase "Su celular es el modelo G8 de la marca Motorola y tiene 3GB de RAM".

Ejercicio 3

En una aplicación para generar una cuenta de usuario, se aplica una serie de controles para que el usuario ingrese su nombre de usuario de manera correcta, teniendo en cuenta que debe respetar las siguientes condiciones:

- a) Debe contener al menos 6 caracteres.
- b) Debe comenzar con una letra mayúscula y todas las demás en minúsculas.
- c) No debe tener espacios al comienzo ni al final.
- d) No puede contener números.

Indicar las expresiones necesarias para realizar cada uno de los controles sobre la variable nombreUsuario. Luego implementar un programa que permita al usuario ingresar el nombre deseado e indicar si es correcto o no cumple alguna de las condiciones solicitadas. Agregar al menos dos condiciones más (a elección) e implementarlas junto a las anteriores.

Ejercicio 4

generen una lista con al menos 7 palabras y a continuación implementen un script que pueda detectar cuantas de esas palabras son palíndromos y cuál es el palíndromo con más caracteres.

Ejercicio 5

Implementar un programa que permita al usuario ingresar tres valores: valorInicial, valorFinal y salto. En base a esos tres valores, mostrar la lista de todos los números a partir de valorInicial de salto en salto hasta valorFinal. Resolver el ejercicio utilizando la estructura de control for y luego la estructura de control while. Ejemplo: si valorInicial = 10, valorFinal = 30 y salto = 5, se deberán mostrar los números "a partir de 10, de 5 en 5 hasta llegar a 20", es decir: 10, 15, 20, 25, 30.

Ejercicio 6

Implementar un programa que permita ingresar valores enteros positivos. El ingreso termina cuando el usuario no desee ingresar más valores, para lo cual se deberá definir un criterio que indique esta decisión. Luego deberá calcular y mostrar el promedio, la cantidad de valores ingresados, el mayor y menor valor.

Ejercicio 7

a) Crear una lista que contenga valores enteros desde 3 a 12 incluidos. Utilizar el identificador `listaNumeros`.

b) Indicar el resultado de las siguientes operaciones realizadas sobre la lista del inciso anterior:

- i) `x = len(listaNumeros)`
- ii) `x = listaNumeros[3]`
- iii) `x = listaNumeros[:3]`
- iv) `x = listaNumeros[3:]`
- v) `x = listaNumeros[3:6]`
- vi) `x = listaNumeros[1:8:2]`
- vii) `x = listaNumeros[-1]`
- viii) `x = listaNumeros[-6:-2]`
- ix) `x = listaNumeros[: : -1]`

Ejercicio 8

a) Implementar una función que permita extraer la última cifra de un número entero positivo (es decir, la cifra de unidad).

b) Implementar una función que permita extraer todas las cifras excepto la última de un número entero positivo (es decir, todas menos la cifra de unidad).

c) Implementar una función que devuelva la cantidad de cifras que contiene un número entero positivo.

Ejercicio 9

a) Implementar una función `contieneCifra(numero, cifra)`, que devuelva un valor booleano indicando si la cifra aparece en el número dado.

b) Implementar una función `contarCifra(numero, cifra)`, que devuelva la cantidad de veces que aparece la cifra en el número.

Ejemplos:

```
a = contieneCifra(1810, 8) # True, porque 1810 contiene al menos un 8  
b = contarCifra(1810, 1) # 2, porque 1810 contiene dos unos.
```

Ejercicio 10

a) Implementar una función que devuelva el mayor de dos números reales.

b) Utilizando la solución del inciso anterior, implementar una función que devuelva el mayor de tres números reales.

Ejercicio 11

Dada una lista de palabras, crear un diccionario para contar la frecuencia de cada palabra en la lista.

Ejemplo:

```
Si lista = ["manzana", "banana", "pera", "manzana", "pera",  
"manzana", "banana"]
```

Se debe obtener

```
frecuencia_palabras = {"manzana": 3, "banana": 2, "pera":  
2}
```

Ejercicio 12

a) Implementar una función que al pasarle dos valores (x,y) retorne un diccionario con la siguiente estructura.

p = {"x": x, "y": y}
emulando un punto P = (x,y).

b) Implementar una función que genere 10 puntos, como en el inciso anterior y lo almacene en una lista.

c) Implementar una función que encuentre el o los puntos más cercanos al origen.

AYUDAS:

Para generar valores aleatorios, será necesario importar el módulo random. Así se podrán utilizar las funciones **randint**(min,max) y **random**().

La función **divmod**(dividendo, divisor) devuelve el cociente y resto de una división entera.

Ejemplo:

div = **divmod**(14, 5) # retorna (2, 4), es decir cociente 2 y resto 4

Algunos métodos que pueden ser de utilidad en el manejo de cadenas:

len(cadena): devuelve la cantidad de caracteres que contiene la variable cadena.

cadena.**capitalize**(): convierte a mayúsculas el primer caracter de la letra de cadena, todos los demás en minúscula.

cadena.**strip**(): elimina espacios en blanco al comienzo y al final de cadena.

cadena.**endswith**(valor): devuelve True si la cadena termina con el valor indicado.

cadena.**startswith**(valor): devuelve True si la cadena empieza con el valor indicado.

cadena.**upper**(): convierte la cadena a mayúsculas.

cadena.**lower**(): convierte la cadena a minúsculas.

cadena.**isalpha**(): devuelve True si todos los caracteres de cadena están en el alfabeto.

cadena.**isnumeric**(): devuelve True si todos los caracteres de cadena son números (0-9).