

Práctico 5

Modularización - Alcance de Nombres - Archivos de Texto

Alcance de nombres

1. Crear un procedimiento llamado cuadrado que tome un número como parámetro e imprima en pantalla dicho número al cuadrado. **A fines pedagógicos, no a fines de elegancia de código, haremos un procedimiento en este caso y no una función como correspondería.*
 - a) Imprimir, desde dentro del procedimiento, las variables locales que éste posea. ¿Qué se imprime? ** utilizar locals()*
 - b) Agregar dos o tres variables extras con valores inventados, y volver a mostrar todas las variables locales. ¿Hubo cambio alguno?
2. Crear una variable llamada n, que será global, en el código del ejercicio anterior y asignarle un valor inventado. Realizar las siguientes acciones:
 - a) Mostrar el valor de n, elevado al cuadrado desde dentro del procedimiento.
 - b) Modificar el valor de n, dentro del procedimiento. ¿Qué ocurre?
 - c) Si necesitase modificar el valor de n, que es una variable global, dentro del procedimiento cuadrado, ¿Qué debería hacer?

3. ¿Qué imprimirá en pantalla el siguiente código? ¿Cuál es el alcance de la variable frase?

```
frase = "Hola"

def f():
    frase = "Es un lindo dia"
    print(frase)
```

4. ¿Qué imprimirá en pantalla el siguiente código? Determine el alcance de cada variable.

```
x = 3

def f():
    y = x + 1
    print(x)

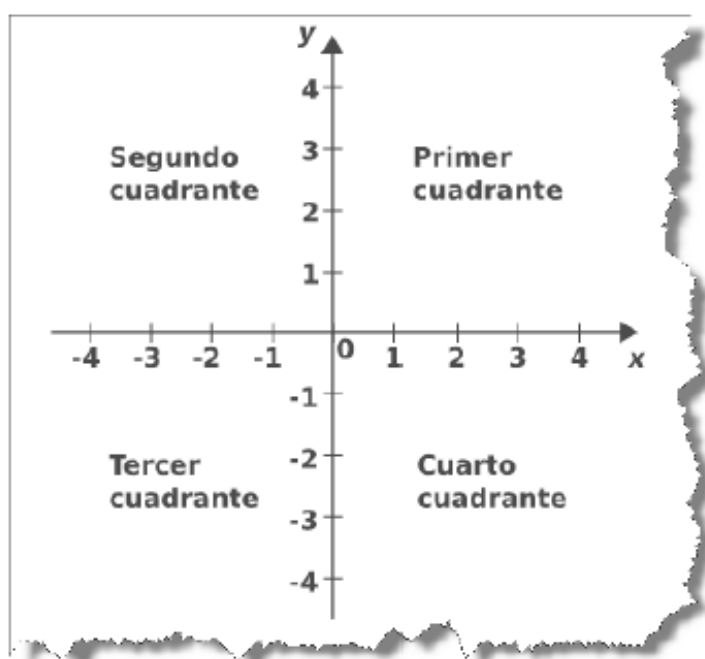
    def g():
        x = 1
        print(y)
        print(x)

    g()

f()
```

5. Crear una función que reciba un número como parámetro el cual representa el lado de un cuadrado y muestre en pantalla el perímetro y la superficie del mismo.

6. Crear una función llamada `escribir_tabla_multiplicar`, que reciba como parámetro un número entero, y escriba la tabla de multiplicar de ese número (*por ejemplo, para el 3 deberá mostrar desde $3 \times 0 = 0$ hasta $3 \times 10 = 30$*).
7. Escriba una función denominada `cuadrante(x, y)`, donde x e y son valores enteros recibidos como parámetros los cuales representan un punto, y que retorne un valor entre 1, 2, 3 o 4 de acuerdo al cuadrante que se encuentre el punto (x, y) , ingresado como parámetro, en los ejes cartesianos.



Ejemplo:

cuadrante (2,4) se encuentra en el **primer cuadrante**

cuadrante (-3,-1) se encuentra en el **tercer cuadrante**

Figura 1: Cuadrantes.

Archivos de Texto

8. Crear un archivo de texto con el bloc de notas de Windows, y guardar una palabra por línea. Crear luego un programa que muestre en pantalla, una por una, las palabras que hay guardadas en ese archivo. Mostrar además, el total de líneas que había en el archivo.
9. Crear un programa que genere un archivo de texto llamado *numeros.txt* con 10 números enteros guardados en el mismo, uno por línea.
10. Crear un programa que le pida al usuario 5 colores, y los guarde en un archivo de texto llamado *colores.txt*
11. Crear un programa que dado un archivo de números con valores entre 1 y 10 (*lo puede generar como ud. desee*) determine cuántos números iguales a 5 hay en el archivo.
12. Crear un programa que dado un archivo de texto, que contiene números (*lo puede generar como ud. desee*) determine el valor promedio y la suma de todos ellos.

Funciones

13. Escriba una función que calcule la n -ésima potencia de un número, recibiendo como parámetro un número real base y otro entero llamado exponente.
La definición de la función es: $y = x^n$
donde x representa la base y n representa el exponente.
Nota: tener en cuenta que n puede ser un número negativo.
Ejemplo: $2^3 = 8$ y $2^{-3} = 0.125$
14. Crear una función que reciba una cadena de caracteres y una letra como parámetros, y devuelva la cantidad de veces que dicha letra aparece en la cadena.
Por ejemplo, si la cadena es 'Barcelona' y la letra es 'a', debería devolver 2 (*aparece 2 veces*).
15. Con la función creada en el ejercicio anterior, elabore un programa en donde se ingresa un carácter y 10 palabras; y muestre la cantidad total de veces que apareció el carácter en las 10 palabras.
16. Crear una función lógica (*función que retorna un valor lógico*) que determine si un número entero es par o impar.
17. Crear una función que reciba un carácter y un número como parámetros e imprima en pantalla un triángulo formado por ese carácter que tenga como ancho inicial el número recibido como parámetro.
Por ejemplo, si el carácter es * y el ancho es 4, debería escribir:
-
- ```

**
*
```
- 
18. Crear una función *es\_primo*, que reciba un número entero como parámetro y devuelva verdadero si es un número primo o falso en caso contrario.
19. Crear una función que dados dos valores distintos, ingresados por parámetro, devuelva el mayor de ellos.
20. Desarrollar una función que dados cinco números, recibidos por parámetro, devuelva el promedio de ellos. Se puede generalizar para  $n$  parámetros devolviendo el promedio de los mismos.
21. Dado un número entero formado sólo por los dígitos 0 (cero) y 1 (uno), diseñe una función que compruebe si el número tiene o no la misma cantidad de ceros que de unos.
22. Desarrollar una función que retorne la posición de un carácter (*la primera vez que aparece*) dentro de la cadena de  $N$  caracteres de longitud, donde se reciben como parámetro la cadena y el carácter respectivamente.

## Procedimientos

23. Crear un procedimiento que se encargue de crear un archivo de texto, con el nombre que se le dé como argumento. Y que lo llene con 250 números al azar entre 1 y 100.
24. Crear una segunda versión del procedimiento anterior, que ahora tome dos parámetros extras,  $a$  y  $b$  para poder indicarle el intervalo de valores que se desean para los números al azar. O sea, ahora el procedimiento generará un archivo de texto, del nombre que se le dé, con valores al azar en  $[a, b]$ .
25. Realizar un procedimiento que tome como parámetro una longitud e imprima en pantalla un rectángulo de numerales, hueco por dentro. Por ejemplo, si se ingresó 4, se verá en pantalla: *Tip: Puede ser útil pensarlo por línea horizontal*  
Generalizarlo, luego, en una versión 2, para un parámetro extra: el carácter que se usará para dibujar el rectángulo, en vez de usar siempre un numeral.

```


####
```

## Variado

26. Diseñar un algoritmo que permita aplicar un descuento del 10 % al monto total de una compra si la forma de pago empleada es mediante débito, 13 % si la compra la realiza mediante pago contado-efectivo o aumente en un 4 % si es un solo pago y se realiza en pago con tarjeta. El usuario deberá ingresar el monto de la compra realizada y la forma de pago utilizada. Si es débito o efectivo, deberá aplicar el descuento, sino realizar el recargo correspondiente.
27. Realizar modificaciones para el ejercicio anterior para que permita variantes en las tarjetas de crédito ingresadas. Visa, Master, American Express con recargos en 5 %, 7 % ó 9 % respectivamente sobre el valor de la compra.
28. Diseñar un algoritmo que ayude al personal de ventas a realizar el cálculo de los intereses sobre un producto, cuando un cliente intenta pagar en cuotas con alguna tarjeta que acepte el comercio. Las tarjetas aceptadas son 3 y con estas se pueden abonar en 3, 6 y 12 cuotas. Se aclaran los recargos para cada una de las opciones. Para la primer tarjeta 3 pagos con 4 % mensual, 6 pagos 4,5 % mensual y 12 pagos 5 % mensual. Para la segunda tarjeta 3 pagos con 3 % mensual, 6 pagos 4 % mensual y 12 pagos 5 % mensual. Para la tercer tarjeta 3 pagos con 3,8 % mensual, 6 pagos 5 % mensual y 12 pagos 5,3 % mensual. Para el caso seleccionado, deberá mostrar la tasa de financiación mensual, la tasa anual (*tasa mensual  $\times$  12*) y el valor del producto aplicado los intereses correspondientes.
29. Integrar en lo posible la solución entre ambos ejercicios anteriores.
30. Crear un programa que permita al usuario ingresar un número en base 10 y lo devuelva en base binaria.

31. Crear un programa que permita al usuario ingresar un número en base 10 y la base a la cual desea convertirlo. Mostrar el resultado de la conversión. Siempre que la base sea menor a 10.
32. Crear un programa que permita ingresar un número en base 2, y lo convierta a base 10.