



**CARRERA DE:** INGENIERÍA INFORMÁTICA  
**MATERIA:** PARADIGMA DE PROGRAMACIÓN  
**FORMACIÓN TEÓRICA:** GUÍA DE EJERCICIOS

## ***GUÍA DE EJERCICIOS***

### **TEMA: EJERCICIOS GENERALES EN PYTHON**

**OBJETIVO:** Familiarizarse con el lenguaje utilizando las estructuras mas comunes del lenguaje. Mostrar su utilidad y comprobar su funcionamiento.

1. Escribir una función que reciba como parámetros el inicio y fin (inclusive) de un rango numérico. La función debe:
  - a. Imprimir en pantalla todos aquellos números que sean divisibles por 7 pero no sean divisibles por 5.
  - b. Imprimir el mismo resultado anterior, pero separados por coma.

Resultado Esperado: Por ejemplo, si se invoca con los parámetros 1 y 100 (puntos a y b)

7  
14  
21  
28  
42  
49  
56  
63  
77  
84  
91  
98

7, 14, 21, 28, 42, 49, 56, 63, 77, 84, 91, 98

2. Escribir la función factorial, que reciba como parámetro el numero inicial y compute su resultado.
  - a. Ejemplo factorial(8) =  $8*7*6*5*4*3*2*1 = 40320$ . Recuerde que factorial de 0 por definición es 1.
  - b. Hacer la implementación inversa (si lo hizo recursivo, hacerlo iterativo o viceversa)
3. Escribir una función que genere y retorne un diccionario ASCII. Para ello, las claves deben ser las letras a partir de la 'a' y el valor debe ser el número ASCII (a -> 97, b -> 98, ...).

Tips: se puede utilizar la función **chr** para convertir un número en su correspondiente letra o **ord** para la situación inversa (conocer el valor ASCII de una letra). También recordar que se puede crear un diccionario vacío con **dict()**.

Resultado Esperado:

```
{'a': 97, 'b': 98, 'c': 99, 'd': 100, 'e': 101, 'f': 102, 'g': 103, 'h': 104, 'i': 105, 'j': 106, 'k': 107, 'l': 108, 'm': 109, 'n': 110, 'o': 111, 'p': 112, 'q': 113, 'r': 114, 's': 115, 't': 116, 'u': 117, 'v': 118, 'w': 119, 'x': 120, 'y': 121, 'z': 122}
```



4. Escribir un programa que reciba como parámetro un string de elementos separados por coma y retorne una lista conteniendo cada elemento.

Resultado Esperado: Si se utiliza como parámetro el string '14, Juana Perez, True' se espera que la función retorne la lista ['14', 'Juana Perez', 'True']

- b. Modificar la función anterior, pero obteniendo correctamente cada dato según su tipo (numero entero, string o boolean). Se considera que el string siempre tiene la forma de primer campo un entero, luego un texto y por ultimo un boolean.

Resultado Esperado: Si se envía '14, Juana Perez, True' se espera que la función retorne la lista [14, 'Juana Perez', True]

- c. Utilizando la función realizada en el punto 4b, hace una nueva función que reciba una lista de strings y retorna otra lista, como listas con los campos formateados (mantener el mismo formato de string que el punto b, es decir, entero, texto y boolean).

Resultado Esperado: Si se envía la siguiente lista como parámetro de la función:

```
[  
    '14, Juana Perez, True',  
    '16, Raul Dell, False',  
    '18, Mariana Castillo, True',  
    '176, Pedro Rodríguez, False'  
]
```

Se espera obtener la lista:

```
[[14, 'Juana Perez', True], [16, 'Raul Dell', True], [18, 'Mariana Castillo',  
True], [176, 'Pedro Rodríguez', True]]
```

5. Escriba una función que reciba como parámetro el radio de un círculo y devuelva una tupla conteniendo en el primer elemento el perímetro y en el segundo el área del mismo.

TIPs: Recuerde que la fórmula del perímetro es  $(2 * \pi * r)$  y el área se define como  $(\pi * r^2)$ . Se puede utilizar la constante pi definida en el módulo math (`import math`)

Resultado Esperado: Si se utiliza la función con el valor de  $r=5$ , entonces debe devolver la tupla (31.415, 78.539)

6. Se necesita un programa que reciba por línea de comando una serie de palabras, hasta que reciba la palabra "exit". Una vez recibida dicha instrucción, debe mostrar por salida standard las mismas palabras ingresadas, almacenadas en una lista, pero ordenadas alfabéticamente y cada una debe estar capitalizada.

Resultado Esperado: El programa le solicita al usuario que ingrese y este escribe:

```
Ingrese palabra: hola  
Ingrese palabra: QUE  
Ingrese palabra: tal  
Ingrese palabra: como
```



**FIE****FACULTAD DE INGENIERIA DEL EJERCITO**  
**Universidad de la Defensa Nacional**

9. Se necesita un programa que reciba por parámetro un texto, y que devuelva una tupla conteniendo en el primer lugar, la cantidad de letras (mayúsculas o minúsculas), en el segundo lugar la cantidad de dígitos numéricos y en el tercer lugar, otros símbolos.

Resultado Esperado (tupla): Por ejemplo, si se utiliza como parámetro el texto: "Esta es una mañana LLuviosa!! 25 días más serán así??" se debería obtener como resultado la tupla (38, 2, 13)

TIPs: Python cuenta con las funciones `isdigit()` y `isalpha()` utilizadas sobre strings (o "chars") que pueden ser útiles para identificar tipos de caracteres. Por ejemplo, `'9'.isdigit()` retorna `True` y `'!'.isalpha()` retorna `False`. Buscar en la documentación mayor detalle.