



## Computación 2022 Verano

### Lab. 4.4. Funciones en Python. Parte II.

Enero 26, 2022

Cree una carpeta (folder) en el disco D, nómbrela con su apellido paterno seguido de su código. Ejemplo: LOPEZ12345

#### Funciones en Python

Las **funciones** son un conjunto de instrucciones que realizan una tarea determinada. Dichas instrucciones agrupadas forman una unidad (entidad, caja negra) a la cual se le da un nombre.

Podemos enumerar varias razones por la que usamos **funciones** en nuestros programas, pero por ahora nos concentraremos en dos de sus roles más importantes: la reutilización del código y el diseño descendente.

##### Maximizando la reutilización del código y minimizando la redundancia.

Cuando se escribe programas grandes de cientos de líneas de código, aparecen varias veces, a lo largo del programa, porciones de código que realizan una misma tarea determinada. También sucede que ciertas tareas ya están codificadas en otros programas. En ambos casos la porción de código que realiza una tarea determinada, es mejor encapsularlo y convertirlo en una unidad llamada **función**. Esto permite usar cuantas veces queramos y en cualquier lugar de nuestros programas, inclusive pueden ser usados por otros programas. De esta manera se logra reutilizar el código, evitar las redundancias y por consiguiente aumentar la productividad del programador y facilitar el mantenimiento.

##### Diseño descendente, descomposición estructural

Una manera de luchar con la complejidad de los problemas es usar la técnica “divide y vencerás”, es decir el diseño descendente (top-down design). La implementación de esta técnica nos lleva a dividir un programa en varias piezas llamadas **funciones**. Codificar por separado cada una de las partes de un programa, es más sencillo que codificar todo el programa a la vez.

Debido a que las funciones forman una unidad (caja negra), para poder usarlos tenemos que establecer una comunicación con ellas a través de una **interfaz**. Es decir que las funciones deben poseer una **interfaz** (medio de comunicación) con el mundo exterior. La estructura de la **interfaz** es definida por la sintaxis del lenguaje de programación, en este caso Python.

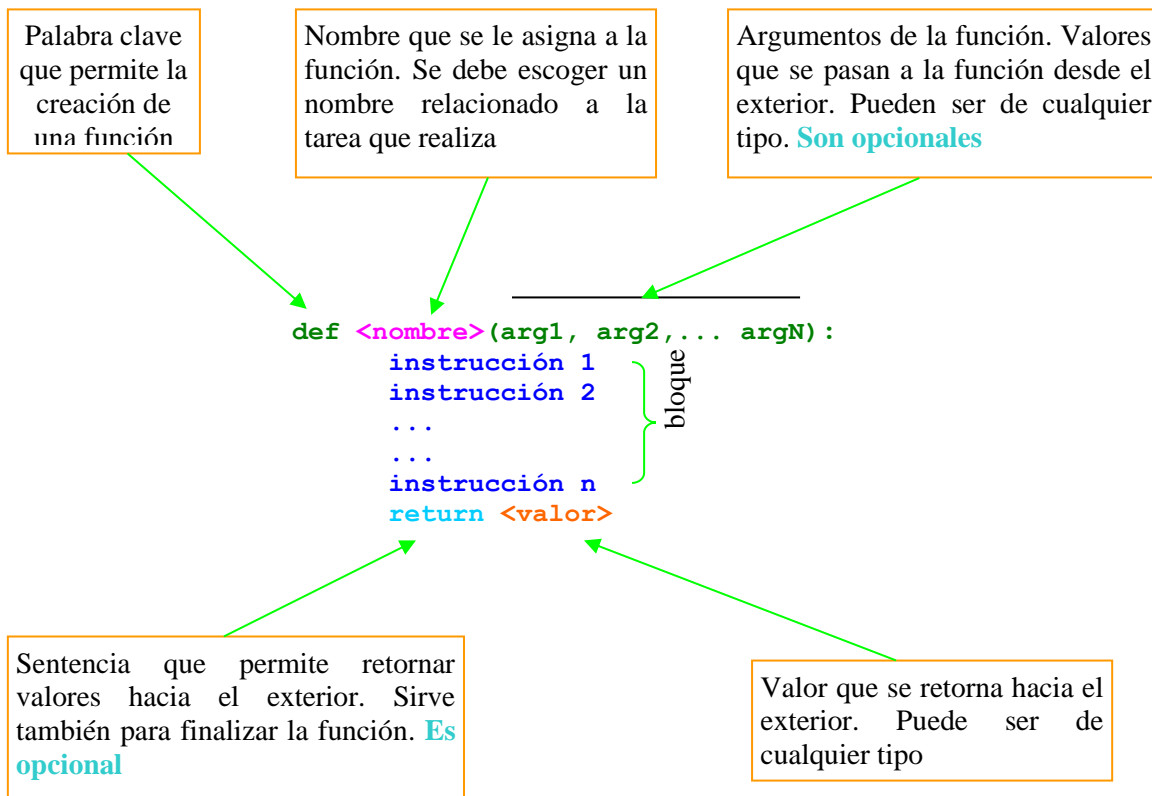
Debemos tener en cuenta además, que las instrucciones, datos y variables contenidas en el cuerpo de las funciones, no son conocidas en el exterior, ni tampoco se puede tener acceso a ellos. La única forma de comunicarnos con las funciones es a través de su interfaz.

## Cómo definir e invocar funciones simples

Para usar una función, primero tenemos que definirla y luego invocarla.

### Definición de una función

La definición se realiza usando la palabra clave **def**. Esta palabra clave permite crear una función, sin embargo, exige que la asignemos un nombre y proporcionemos los argumentos. El formato general es la siguiente:



**Ejemplo 1:** El siguiente programa contiene una función que devuelve el producto de dos números. Corra el programa y analice cómo se define una función y la forma de invocarla (llamarla).

```
# Ejemplo4.3.1.py
# Creación y uso de una función definida por el usuario

# Definición de la función:
def square( x,y ):
    z=x*y
    return z
# ***** Fin de la función *****

a=int(input("ingrese un valor"))
b=int(input("ingrese un valor"))

p=square( a,b )

print( p )
```

**Invocación de la función.** Se invoca colocando el nombre de la función y pasándole los parámetros si son necesarios. El valor devuelto se guarda en la variable **p**.

**¿Qué sucede si colocamos `print(z)` después de `print(p)`?, ¿por qué?**

**Ejemplo 2:** El siguiente programa modifica el ejemplo 1. El valor retornado no se guarda en otra variable, se imprime directamente.

```
# Ejemplo4.3.2.py
# Creación y uso de una función definida por el usuario

# ***** Definición de la función: *****
def square( x,y ):
    z=x*y
    return z
# ***** Fin de la function *****
a=int(input("ingrese un numero para a:"))
b=int(input("ingrese un numero para b:"))

print(square( a,b ))
```

Aquí se invoca la función y el valor devuelto se imprime directamente.

**Ejemplo 3:** El siguiente programa modifica el ejemplo 1 para imprimir el cuadrado de los números del 1 al 10 inclusive.

```
# Ejemplo4.3.3.py
# Creación y uso de una función definida por el usuario

# Definición de la función:
def square( x ):
    z=x*x
    return z

for c in range( 1, 11 ):
    p = square( c )
    print(p)
```

Antes de iniciar con los ejercicios, responder:

- ¿En qué lugar se define una función, antes de la invocación, después de la invocación o en cualquier lugar?
- ¿Es necesario la instrucción `return`? ¿qué usos tiene?
- ¿Una función siempre tiene argumentos?
- ¿Se puede definir más de una función en un programa?
- Las variables que están dentro de una función, ¿pueden ser usados fuera de la función?
- ¿Cuántas veces se puede invocar una función?

**Para cada uno de los siguientes ejercicios implemente el programa Python correspondiente. Guarde vuestros programas en vuestra carpeta de trabajo.**

1. Una frase en el idioma español puede contener vocales acentuadas, la letra ñ y la u con diéresis. Implemente una función que reciba una frase en español y devuelva la frase con las vocales acentuadas reemplazadas por las mismas vocales sin acentuar, la ñ por n, y la u con diéresis por u.

Ingrese una frase: En España no hay pingüinos, en Bélgica tampoco

La frase modificada: En Espana no hay pinguinos, en Belgica tampoco

2. Implemente una función que reciba una frase y devuelva la frase con las letras mayúsculas reemplazadas por las letras minúsculas correspondientes.

Ingrese una frase: El ESPAÑOL es un idioma procedente del LATÍN

La frase modificada: el español es un idioma procedente del latín

3. Implementar una función que determine si un carácter es o no es una letra. La función debe de devolver `True` si el carácter es letra o `False` en caso contrario. Ejemplo:

Ingrese un carácter: b

¿El carácter ingresado es letra?: True

Ingrese un carácter: Z

¿El carácter ingresado es letra?: True

Ingrese un carácter: #

¿El carácter ingresado es letra?: False

Ingrese un carácter: aaa

¿El carácter ingresado es letra?: False

4. Implementar una función que determine si un carácter es o no es un dígito. La función debe de devolver `True` si el carácter es dígito o `False` en caso contrario. Ejemplo:

Ingrese un carácter: 5

¿El carácter ingresado es dígito?: True

Ingrese un carácter: G

¿El carácter ingresado es dígito?: False

Ingrese un carácter: 358

¿El carácter ingresado es dígito?: False

5. Las palabras de una frase (texto), pueden estar separadas por uno o más espacios o por comas. La frase puede contener signos de interrogación, admiración, mayúsculas, minúsculas, dígitos y tildes. Implemente una función que reciba una frase y devuelva la frase compactada, es decir, eliminando cualquier carácter que no sea letra o dígito. Sugerencia: use las funciones de los ejercicios 3 y 4.

Ingrese una frase: Ser o no ser, esa es la pregunta

La frase modificada es: Seronoseresaeslapregunta

Ingrese una frase: Necesito un 14(catorce), para cómputo

La frase modificada es: Necesitoun14catorceparacómputo

6. Implementar una función que reciba un texto y devuelva el mismo texto invertido. Escriba un programa para probar la función.

Ejemplo:

Ingrese la cadena de caracteres: Universidad

El texto invertido es: dadisrevinU

7. Implemente una función que determine si una palabra es palíndromo.  
Sugerencia: Use la función del ejercicio nº 6.
8. Una frase es palíndromo si se lee igual de derecha a izquierda y de izquierda a derecha, pero obviando los espacios en blanco, signos de puntuación y tildes. Implemente una función que reciba una frase en español o inglés e indique si es o no es un palíndromo. Escriba un programa para probar la función.  
Sugerencia: use las funciones previamente implementadas según corresponda.

```
Ingrese una frase: Átale, demoníaco Caín, o me delata
La frase es palíndroma: True

Ingrese una frase: Adán no cede con Eva y Yavé no cede con nada
La frase es palíndroma: True

Ingrese una frase: Nota épica: ¡nací peatón!
La frase es palíndroma: True

Ingrese una frase: Madam, in Eden I'm Adam
La frase es palíndroma: True

Ingrese una frase: A Santa lived as a devil at NASA
La frase es palíndroma: True

Ingrese una frase: Was it a car or a cat I saw?
La frase es palíndroma: True
```

Guarde todos vuestros programas y vuestra hoja de respuestas en una carpeta con el nombre su **Apellido** paterno seguido de vuestro **DNI**, luego comprima esta carpeta. Envíe este archivo a [victor.melchor.e@upch.pe](mailto:victor.melchor.e@upch.pe) , especificando como asunto **Lab4.4**.