FUNCIONES

Parte I

UNIDAD 3

Introducción a la Programación (11071)
Departamento de Ciencias Básicas
Universidad Nacional de Luján

ORGANIZACIÓN

En esta ocasión vamos a hablar de los siguientes puntos:

- ★ Concepto de Modularización,
- ★ Concepto de función (parámetros, resultados y operaciones),
- ★ Ámbito de las variables,
- ★ Esquema de una función y su sintaxis,
- ★ Comunicación con el programa principal,
- ★ Cómo pensar una función,
- ★ Ejemplo de funciones en Python.

MODULARIZACIÓN: FUNCIONES

A partir de ahora vamos a modularizar nuestros programas. La idea sea basa en descomponer cada problema en “subproblemas” más pequeños.

En Ciencias de la Computación se conoce a esta estrategia a partir del refrán: divide y vencerás!

Ventajas de programar de forma modular (a través de funciones):

- ★ Facilita la programación y el testing,
- ★ Permite el desarrollo incremental,
- ★ Favorece el trabajo en equipo,
- ★ Permite la reutilización del código.

FUNCIONES

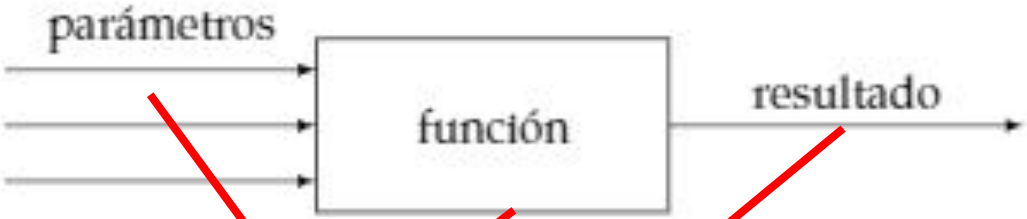
DEFINICIÓN

★ Una función es un fragmento (una porción) de programa que permite ***efectuar una operación*** determinada.

- Para efectuar las operaciones, las funciones reciben datos, a los que llamaremos ***parámetros***.
- Las funciones devuelven uno o más ***resultados***.

  (Más adelante trabajaremos este concepto en detalle)

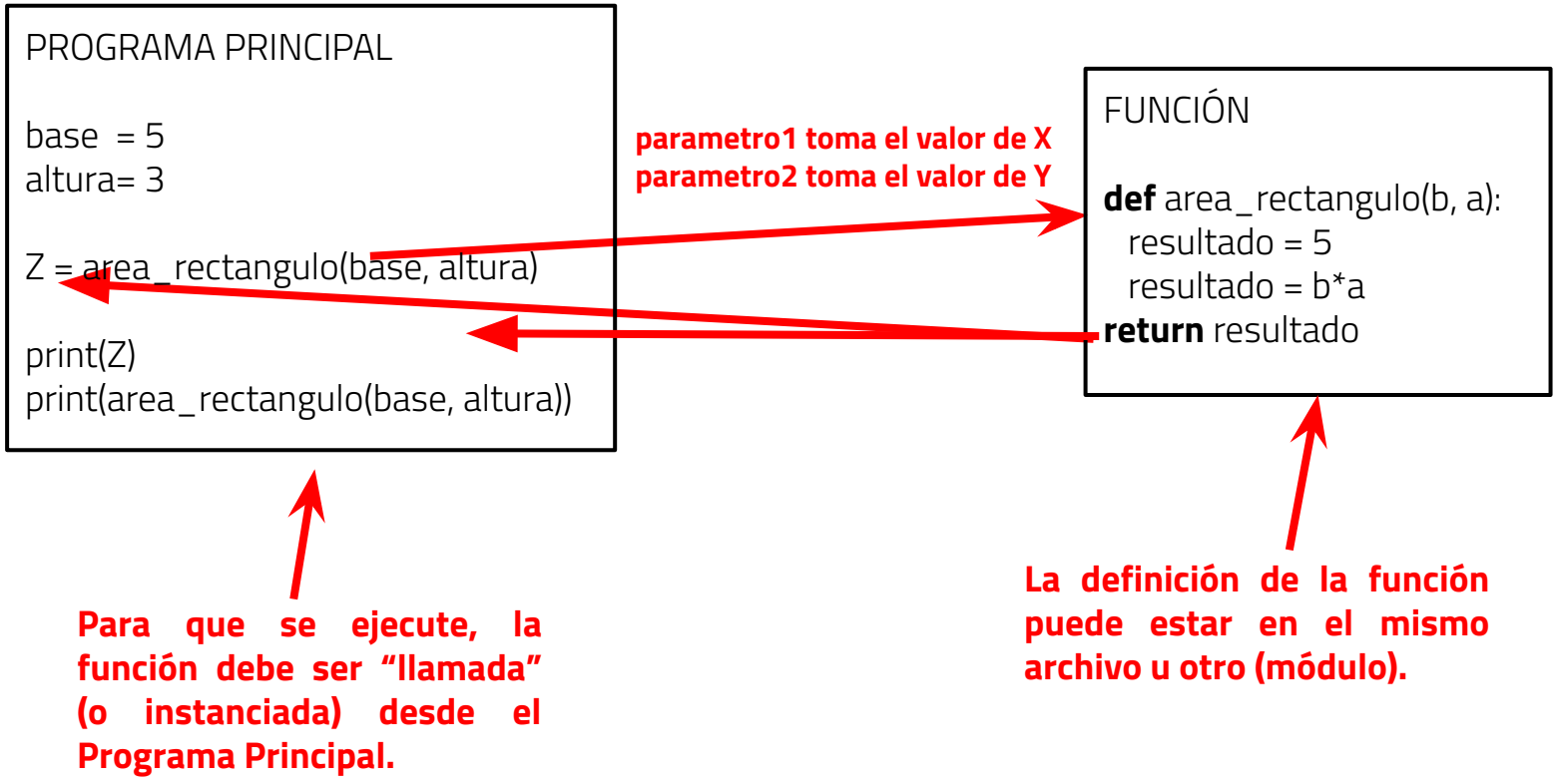
FUNCIONES: ESQUEMA Y SINTAXIS




```
def area_rectangulo(b, a):
    resultado = 0
    resultado = b*a
    return resultado
```

Podremos definir variables propias de nuestras funciones

FUNCIONES: COMUNICACIÓN CON EL PROGRAMA PRINCIPAL



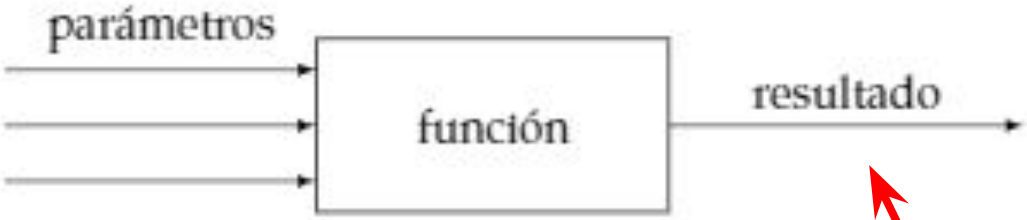
ÁMBITO DE LAS VARIABLES

Como vimos antes, podemos definir variables dentro del cuerpo de las funciones.

Las variables que se declaran dentro de una función no existen fuera de ella y por eso se las denomina ***variables locales***.

Fuera de la función, se puede acceder únicamente a los valores que se devuelven mediante la palabra reservada ***return***.

FUNCIONES: ¿Cómo pensarlas?



Segundo: Qué datos necesita mi función para obtener el/los resultados esperados?

Tercero: Cómo transformar los datos de entrada en el/los resultado/s mediante un proceso algorítmico?

Primero: Qué debe hacer mi función? Devuelve resultado o resultados? Cuales?

FUNCIONES: EJEMPLO

CONSIGNA

Construir una función que permita calcular el perímetro de un rectángulo.

ANÁLISIS

perímetro (un valor numérico)

base (valor numérico)
altura (valor numérico)

base*2 + altura*2

Veamos cómo hacerlo en Python!


31
HTML
C++
CSS
JAVA

FUNCIONES

Parte II

UNIDAD 3

Introducción a la Programación (11071)
Departamento de Ciencias Básicas
Universidad Nacional de Luján

ORGANIZACIÓN

Ahora, vamos a trabajar los siguientes puntos relacionados con funciones:

- ★ Documentación de funciones
- ★ Devolver vs Imprimir
- ★ Devolver múltiples resultados
- ★ Cómo utilizar una función en un programa (Módulos)

DOCUMENTACIÓN DE FUNCIONES

Para que sea reutilizable, es muy importante documentar en nuestras funciones:

- ★ cuál es la tarea que realiza,
- ★ cuáles son los parámetros que recibe y,
- ★ qué es lo que devuelve.

La documentación de una función se coloca en la primera línea del cuerpo de la misma.

```
def area_rectangulo(b, a):
    """
    Esta funcion recibe la base y la altura de un rectangulo y devuelve el area
    """
    resultado = 0
    resultado = b*a
    return resultado
```

FUNCIONES: IMPRIMIR vs. DEVOLVER

Cuando vayamos a diseñar nuestras funciones vamos a tomar decisiones, entre ellas:

★ ¿Realizo el ingreso de datos necesarios para la función dentro de ella?

En general, una función es más reutilizable si recibe parámetros en lugar de leer datos mediante la función input.

★ ¿Imprimo los resultados de mi función o los devuelvo al programa principal?

De la misma manera, una función es más reutilizable si devuelve un resultado (utilizando return) en lugar de imprimirlo (utilizando print).

FUNCIONES: DEVOLVER +1 RESULTADOS

Cuando nuestros problemas lo requieran, es posible devolver múltiples resultados, para ello tendremos que tener en cuenta dos cuestiones:

DEFINICIÓN DE LA FUNCIÓN

Vamos a poner los valores a devolver uno a uno separados por "," en el **return.** Por ejemplo: **return x, y, z**

LLAMADA A LA FUNCIÓN

Vamos a invocar a la función asignando la misma a varias variables (una por cada valor a retornar separados por ",". Por ejemplo: **a, b, c = nombre_funcion(m)**

UTILIZAR UNA FUNCIÓN EN UN PROGRAMA

Existen dos formas básicamente:

- ★ La más sencilla es definir la función en el mismo archivo en el que está mi programa.
- ★ La otra opción, **más interesante**, es crear varios módulos.

Un módulo es un archivo **Python** (extensión .py) donde iremos definiendo nuestras funciones.

Luego, en nuestro módulo principal **importamos** el módulo con la instrucción ***import*** y el nombre del archivo.


31
HTML
C++
CSS
JAVA