

# FUNCIONES PROPIAS



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



# MATERIAL RELACIONADO

## TEORIA



**Teórica 3 (A y B) - Funciones.**



**Apunte**

**Cap.**

**3**

-

**Funciones.**

## PRÁCTICA



**TP V - Funciones propias.**

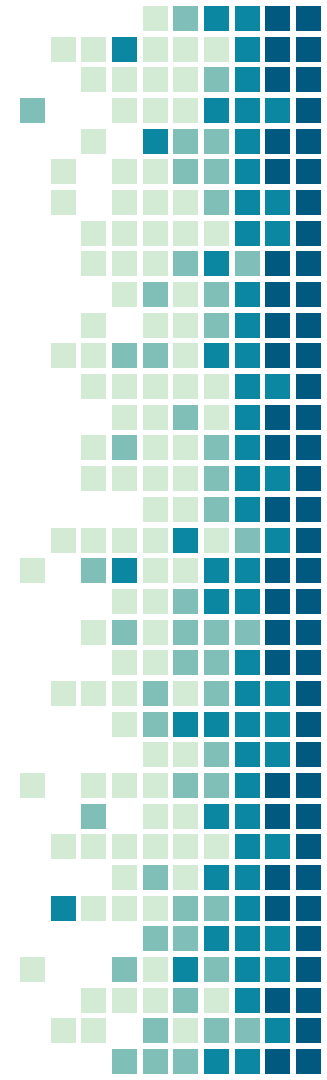


**Notebook 2 - Variables y funciones.**

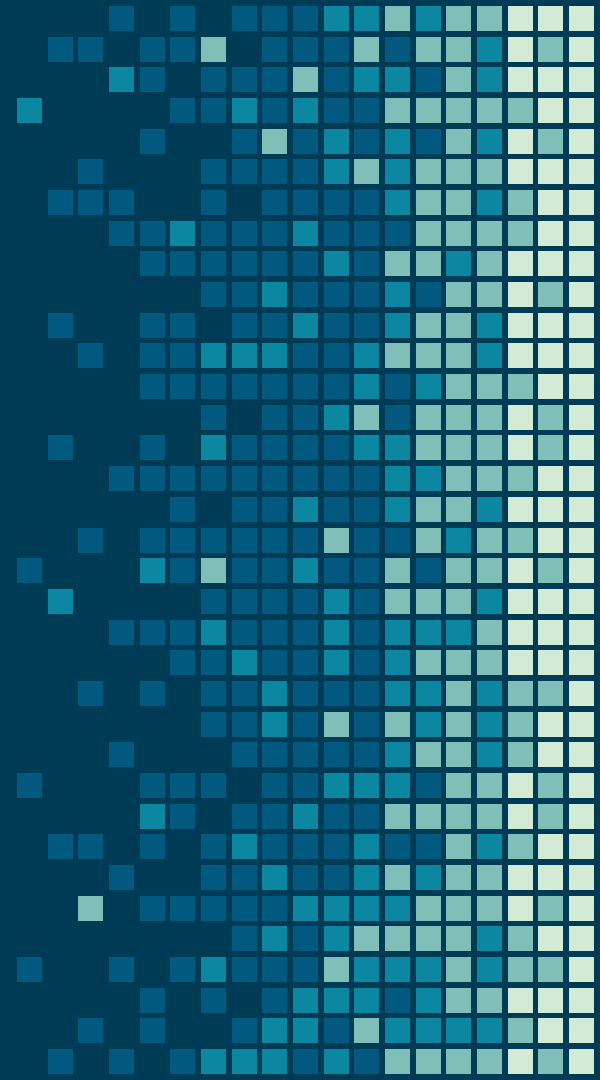
## ENTREGABLES



**HomeWork 1 - Funciones.**



# REPASO DE CONCEPTOS



# FUNCIÓN

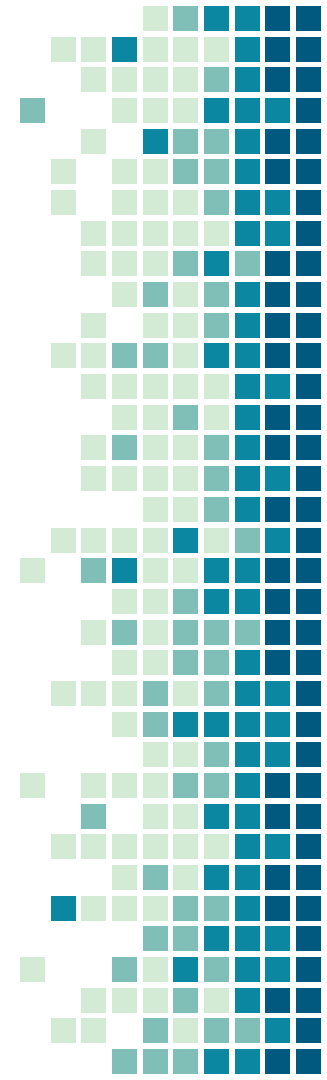
## DEFINICIÓN

Una **función** es un bloque de código al que se le da un nombre (**identificador**), y sólo se ejecuta cuando se lo invoca. Opcionalmente, una función puede recibir **parámetros** o **retornar valores**.

FUNCIÓN	VS	PROCEDIMIENTO
---------	----	---------------

Diferencia teórica (y algunos lenguajes utilizan estos conceptos por separado):

- Las **funciones** pueden recibir cualquier cantidad de parámetros de entrada, pero deben retornar un y sólo un valor.
- Los **procedimientos** pueden recibir cualquier cantidad de parámetros, y pueden retornar cero, uno, o más valores.



# FUNCIÓN

Una **función** es un bloque de código al que se le da un nombre (**identificador**), y sólo se ejecuta cuando se lo invoca. Opcionalmente, una función puede recibir **parámetros** o **retornar valores**.

*¡En Python sólo hay funciones!*

## FUNCIÓN VS PROCEDIMIENTO

Diferencia teórica (y algunos lenguajes utilizan estos conceptos por separado):

- Las **funciones** pueden recibir cualquier cantidad de parámetros de entrada, pero deben retornar un y sólo un valor.
- Los **procedimientos** pueden recibir cualquier cantidad de parámetros, y pueden retornar cero, uno, o más valores.

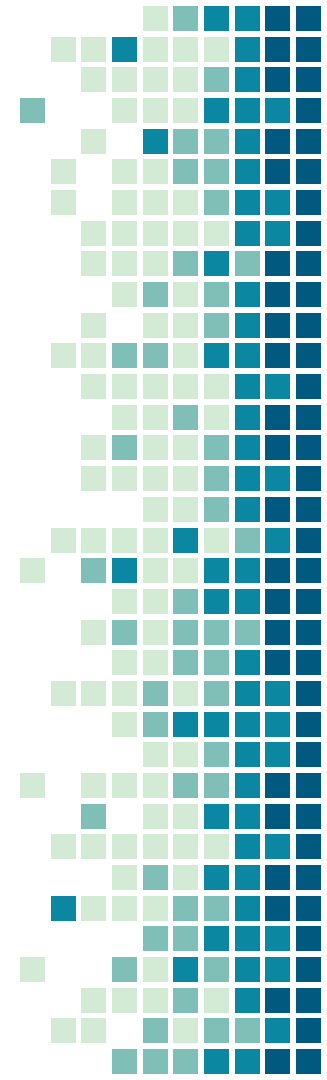
# BUILT-IN VS PROPIA

## FUNCIONES BUILT-IN

- Funciones **provistas por el lenguaje de programación**.
- Como programadores, sólo debemos invocarlas por su nombre (identificador).

## FUNCIONES PROPIAS

- Funciones **codificadas por nosotros mismos**.
- Las funciones propias deben ser **definidas** antes de poder ser **invocadas**.



# FUNCIONES PROPIAS

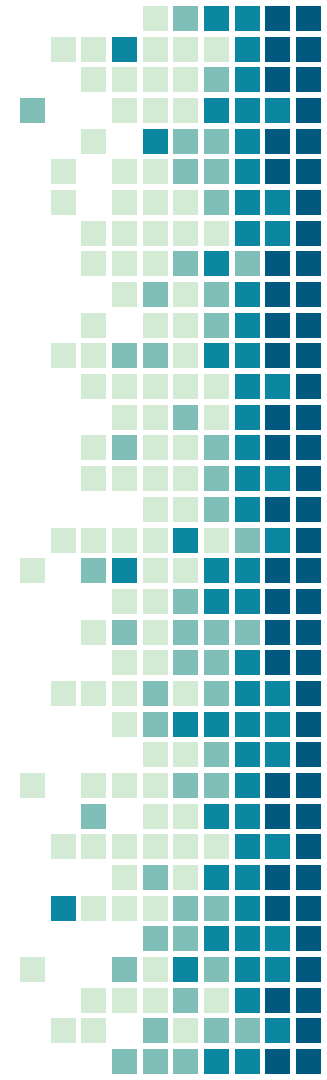
## DEFINICIÓN

La sintaxis para definir una función varía con el lenguaje, pero como mínimo se debe:

- Definir el nombre (identificador) de la función.
- Definir los parámetros que recibe la función.
- Escribir el bloque de código que conforma a la función.

## INVOCACIÓN

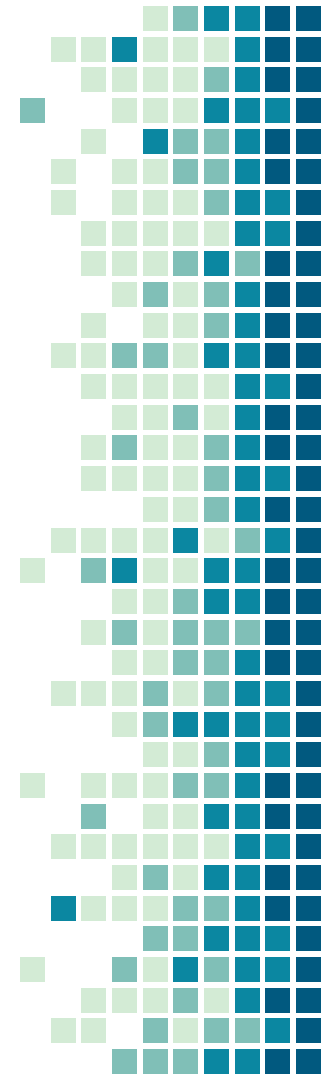
- Una vez definida la función, podremos invocarla cuantas veces lo necesitemos, simplemente utilizando su identificador, y enviándole los parámetros requeridos.
- Si la función retorna algún valor, seguramente su invocación deberá ser parte de una expresión compuesta (asignación, muestra en pantalla, cálculo, etcétera).



# FUNCIONES EN PYTHON

```
# definición
def identificador(parámetros):
    # bloque de código

# invocación
identificador(parámetros)
```





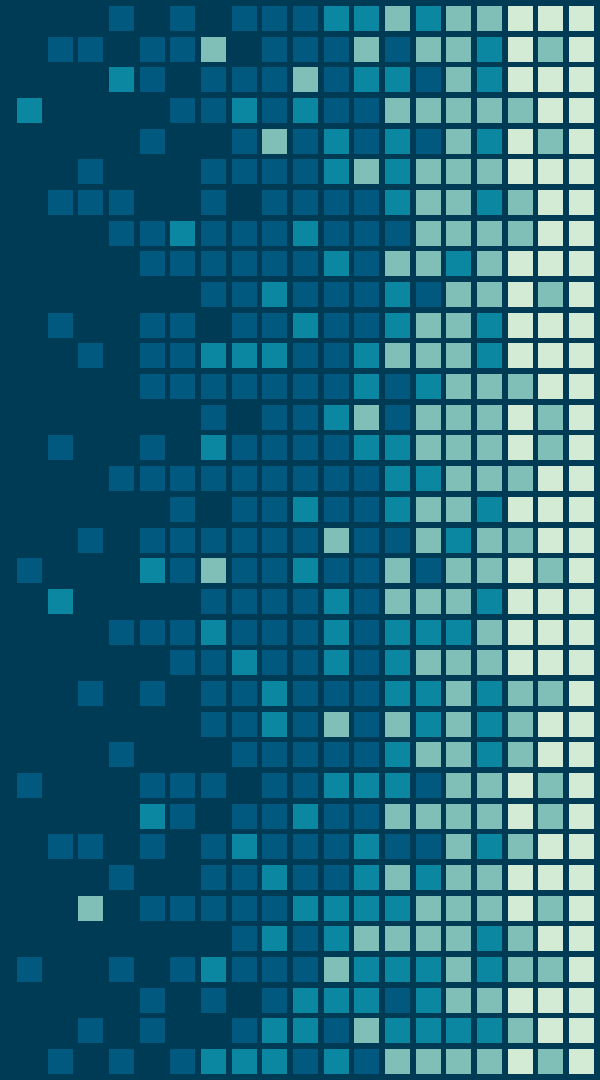
# EJEMPLO

```
# definición
def que_gusto_tiene_la_sal():
    print('Salado!')

# invocación
que_gusto_tiene_la_sal()
```

# EJERCICIOS GUIADOS

*Para resolución de TP V.*



# EJERCICIO MODELO (TP V)

## OBJETIVO DEL EJERCICIO

Comprender y ejercitar el concepto de **función propia**.

## ENUNCIADO

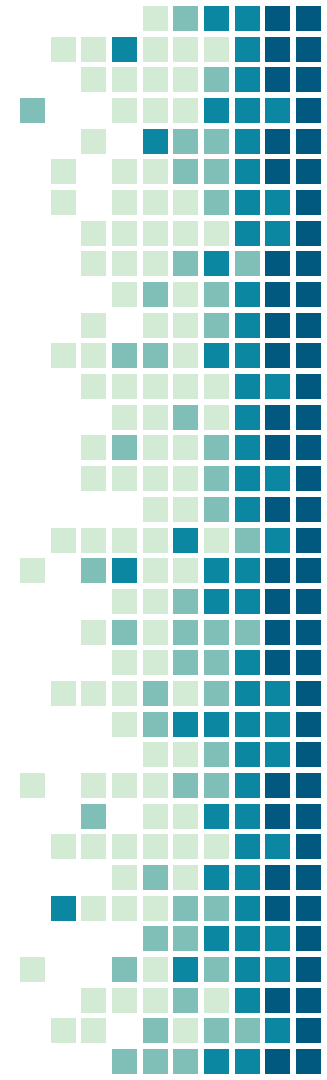
Escriba una función que reciba dos enteros como parámetro, y luego retorne el resultado de la suma.

## CONTINUACIÓN

Modifique la función para que en lugar de retornar el resultado, **lo muestre en pantalla**.

## CONTINUACIÓN (2)

Pruebe invocar a la función utilizando dos valores leídos desde teclado.



# IMPORTANTE!

RETORNAR  $\neq$  MOSTRAR



