

# TC1027. Programación para negocios

## Funciones

Definidas por el usuario y predefinidas



# ¿Qué es una función?

**Las funciones son pedazos de código reutilizables.**

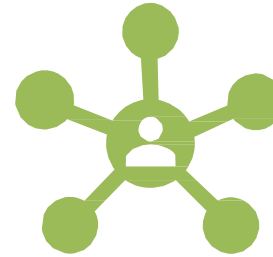
Una función es un pedazo de código que tiene un nombre asociado, que realiza una serie de tareas y que regresa un valor.

Si no regresa ningún valor, entonces se le conoce como **procedimiento**.

# Tipos de Funciones



**Predefinidas**



**Definidas por el usuario**

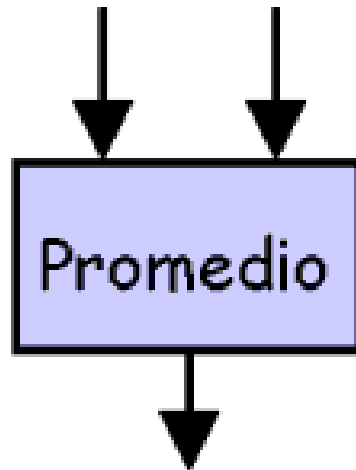


# Funciones definidas por el usuario

# ¿Qué es una función?

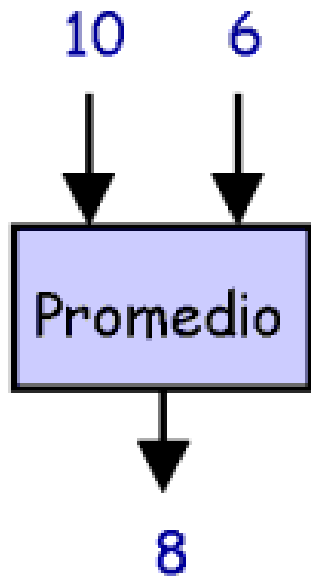
Una función la podemos ver como un proceso que recibe valores de entrada y a partir de ellos produce un valor de salida.

Por ejemplo:

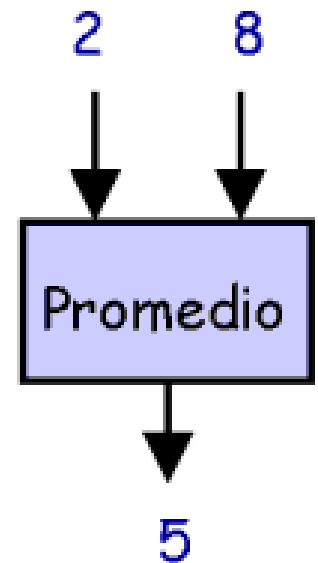


# Función

- La función **Promedio** recibe dos valores (**10** y **6**) dos valores al azar, y lleva a cabo el cálculo para obtener el promedio de estos dos números y regresar el valor resultante, en nuestro ejemplo 8.



Una vez definida una función, esta puede ser aplicada a diferentes valores y obtener en cada caso el resultado correspondiente.



# ¿Cómo se define una función?

- En Python, la definición de la función promedio sería la siguiente:

```
def promedio (a, b):  
    return (a+b)/2
```

- La definición de una función está compuesta de dos partes:
  - **Encabezado** de la función.
  - **Cuerpo** de la función.

# Partes de una función

Encabezado

```
def promedio (a, b):
```

Cuerpo

```
    return (a+b)/2
```

En el **encabezado** se define:

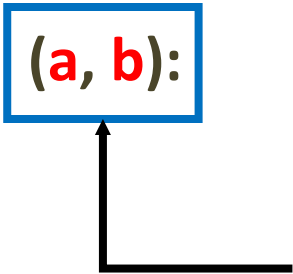
- El **nombre** de la función
- y los **parámetros** de la función





# Encabezado de una función

`def Promedio (a, b):`



**Parámetros:** valores de entrada

Los parámetros de la función son una lista de variables que nos permiten identificar los valores de entrada de la función. En este caso, el primer valor de entrada va a ser identificado por la variable **a** y el segundo valor de entrada va a ser identificado por la letra **b**.

# Parámetros de una función



- La cantidad de variables que se definan en la **lista de parámetros** dependerá de los valores de entrada que necesite la función. Si requiere **5 datos de entrada**, tendrá **5 variables**. Si no requiere de valores de entrada, entonces la lista será vacía, (paréntesis vacíos).
- Las variables definidas en la lista de parámetros **sólo podrán ser utilizadas en el cuerpo de la función**. Una vez que la función haya terminado de hacer sus cálculos y haya regresado el resultado, **las variables de los parámetros son desechadas**.

# Cuerpo de la función

- El **cuerpo** de la función es una lista de instrucciones de Python. Esta lista de instrucciones define el proceso que la función debe llevar a cabo.
- Para indicar el resultado de la función se utiliza la palabra **return**, seguida de la expresión que calcula el valor resultante.



# ¿Cómo se utiliza una función?

- Cuando una función es llamada es necesario que nos hagamos cargo del valor resultante, asignando la función a una variable o formando parte de una expresión mayor.
- Por ejemplo:

**x = promedio(4, 8)**

**y = promedio(9, 3) + promedio(7, 1)**

**z = promedio(11, promedio(12, 18))**

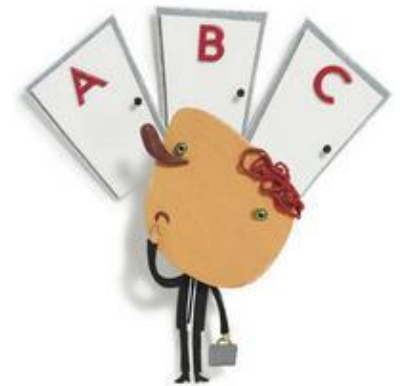


# ¿Cómo se utiliza una función?

- La manera de llamar a una función consiste en escribir el nombre de la función y después entre paréntesis la lista de los valores de entrada.

Por ejemplo:

**nombre( valor1, valor2, ..., valorn)**



- Cuando la función es llamada, cada uno de los valores de entrada son asignados a las variables de la **lista de parámetros** de la función, **en el mismo orden como fueron definidas.**

# ¿Cómo se utiliza una función?

- Por ejemplo, cuando se hace la llamada:

**promedio( 4, 8 )**

- la variable **a** de la definición de la función toma el valor de **4** y la variable **b** toma el valor de **8**.

**x=promedio(4,8)**

```
def promedio ( a, b ):
    return (a+b)/2
```

**6**



# Actividad grupal

Definir el algoritmo y el programa en Python  
que **eleve al cuadrado un numero real.**



# Algoritmo: Función cuadrado

```
función cuadrado (valor)  
    regresar (valor*valor)
```



**Solución**



# Actividad grupal

```
def cuadrado (x):  
    return x*x
```



//Encabezado de la función cuadrado

//Cuerpo de la función cuadrado

```
num = float(input("Introduce un numero: "))
```

```
resultado=cuadrado(num);
```

//Llamada a la función cuadrado

```
print(num, " elevado al cuadrado es: ", resultado)
```

## Solución





## Funciones predefinidas

# Funciones predefinidas



`input()`



`print()`



`len()`



O cualquier otra función que **importemos** desde una librería, como:

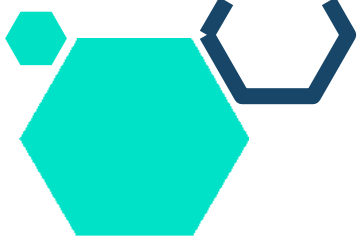
`random.randint()`  
`random.random()`

# Funciones de cadenas

Función	Utilidad	Ejemplo	Resultado
<b>print()</b>	Imprime en pantalla el argumento.	<code>print ("Hola")</code>	Hola
<b>len()</b>	Determina la longitud en caracteres de una cadena.	<code>len("Hola Python")</code>	11
<b>replace()</b>	Reemplaza una cadena por otra	<code>texto = "Manuel es mi amigo" print (texto.replace ("es", "era"))</code>	Manuel era mi amigo
<b>upper()</b>	Convierte una cadena en Mayúsculas	<code>texto = "Manuel es mi amigo" print (texto.upper())</code>	MANUEL ES MI AMIGO
<b>lower()</b>	Convierte una cadena en Minúsculas	<code>texto = "MaNueL eS mI AmIgo" print (texto.lower())</code>	manuel es mi amigo



# Funciones numéricas



Función	Utilidad	Ejemplo	Resultado
<b>str()</b>	Convierte un valor numérico a texto	<b>str(22)</b>	'22'
<b>int()</b>	Convierte a valor entero	<b>int('22')</b>	22
<b>float()</b>	Convierte un valor a decimal	<b>float('2.22')</b>	2.22

# Librería Math

Importar la librería math

**import math**

Función	Utilidad	Ejemplo	Resultado
<b>math.sqrt()</b>	Calcula la raíz cuadrada de un número	<code>print(math.sqrt(25))</code>	8.0
<b>math.pow(x,y)</b> $x^y$	Eleva x a la potencia y	<code>print(math.pow(2,3))</code>	10.0
<b>math.pi</b>	Da el valor de PI	<code>print(math.pi)</code>	3.141592653 589793

Funciones de la librería Math: <https://docs.python.org/3/library/math.html>

# Variables y su contexto

- ◇ Local
- ◇ Global





# Variables Locales



Una variable local es creada dentro de una función, y no puede ser accesada desde fuera de la función.

Funciones diferentes pueden tener variables que se llamen exactamente igual, sin tener ningún conflicto de nombres.



Los parámetros de una función se consideran variables locales.



# Ejemplo: Variables Locales

```
1 def fun(x):  
2     print("x es:", x)  
3     x = 2  
4     print("Cambió x de manera local a", x)  
5  
6 x = 50  
7 fun(x)  
8 print("x continua siendo", x)
```

Shell ×

```
>>> %Run variablesLocales.py
```

```
x es: 50  
Cambió x de manera local a 2  
x continua siendo 50
```



# Variables Globales

Una variable global puede ser accesada desde cualquier parte del programa.

Dos o más funciones pueden modificar la misma variable global.



# Ejemplo: Variables Globales

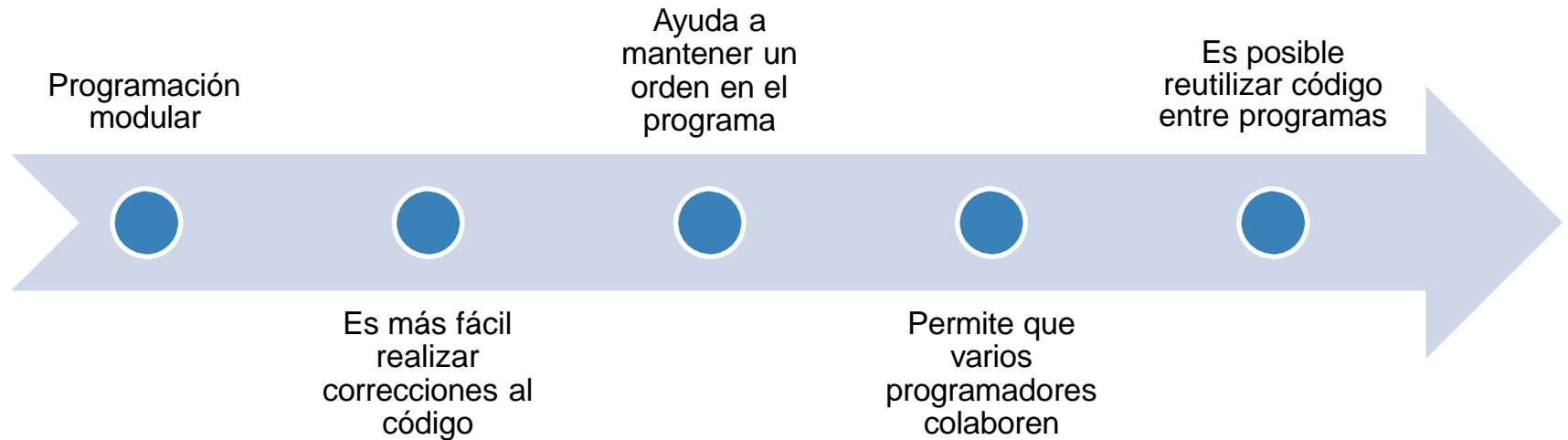
```
1 def fun():
2     global x
3     print("x es:", x)
4     x = 2
5     print("Cambió x de manera global a", x)
6
7 x = 50
8 fun()
9 print("El valor de x es", x)
```

Shell ×

```
>>> %Run variablesGlobales.py
```

```
x es: 50
Cambió x de manera global a 2
El valor de x es 2
```

# Ventajas del uso de funciones



# Fuentes para consultar

<https://python.swaroopch.com/functions.html>

<http://www.mclibre.org/consultar/python/lecciones/python-funciones-1.html>

