



Funciones






◇ ¿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** o **subrutina**.



Ventajas del uso de funciones

Programación modular

Ayuda a mantener un orden en el programa

Es posible reutilizar código entre programas

Es más fácil realizar correcciones al código

Permite que varios programadores colaboren





2 Tipos de Funciones



Predefinidas / Built-in.



UDF – Funciones
Definidas por el Usuario.



Funciones Predefinidas



`input()`



`print()`



`len()`



O cualquier otra function
que **importemos** desde
una libreria, como:


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



Funciones Definidas por el Usuario

UDF





¿Cómo se declara una función?

Documentación de la función.

```
def nombreFuncion(parametro1, parametro2):
```

```
    # Cuerpo de la función.
```

```
    return result # Valor de retorno opcional.
```

Uso de la función.

```
nombreFuncion() # si no hay return, es una subrutina.
```

```
miVariable = nombreFuncion(2,3) # esta sí es una función.
```





Función de Ejemplo

Suma de 2 números.

```
def sumaBinaria(x, y):  
    misuma = x + y  
    return misuma
```

#Uso de la función.

```
resultado = sumaBinaria(2,3)
```





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.

- Una variable dentro de una función es invisible para otras funciones.

Los parámetros de una función se consideran variables locales.



Project

io_using_file.py

function_return.py

function_local.py

function_global.py

> Desktop

```
1  x = 50
2
3  def func(x):
4      print('x is', x)
5      x = 2
6      print('Changed local x to', x)
7
8  func(x)
9  print('x is still', x)
10
```



Variables Globales

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

Dos o más funciones pueden modificar la misma variable global.

- Tenemos que ser cuidadosos con esta funcionalidad, la lógica del programa debe considerarlo.
- 

Project

io_using_file.py

function_return.py

function_local.py

function_global.py


> Desktop

```
1  x = 50
2
3  def func():
4      global x
5      print('x is', x)
6      x = 2
7      print('Changed global x to', x)
8
9  func()
10 print('Value of x is', x)
11
```



Tarea en Canvas

Generar dos programas que incluyan tres funciones cada uno.

- ◇ *Escriba un programa que contenga las siguientes funciones:*
 - *Función que calcule el área de un triángulo.*
 - *Función que calcule el área de un círculo.*
 - *Función que calcule el área de un cuadrado.*
 - ◇ *Escriba un programa que contenga las siguientes funciones:*
 - *Función que calcule la suma de dos números.*
 - *Función que calcule la división de dos números.*
 - *Función que calcule el residuo de una división.*
- 



Otras fuentes para consultar

INGLÉS

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

ESPAÑOL

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

