



```
def my_function(param1, param2, ...):  
    pass
```

Funciones y Módulos

► Introducción a funciones

- Una función es un grupo de declaraciones que están en un programa con el propósito de realizar una tarea específica.



- La mayoría de los programas son lo suficientemente grandes para ser divididos en pequeñas sub-tareas.

Sentencia	def funcion1():
Sentencia	Sentencia
Sentencia	Sentencia
Sentencia	
Sentencia	def funcion2():
Sentencia	Sentencia
Sentencia	Sentencia
Sentencia	
Sentencia	def funcion3():
Sentencia	Sentencia
Sentencia	Sentencia

- En lugar de escribir el programa como una larga secuencia de sentencias, podemos escribirlo como unas pocas y pequeñas funciones.

► Ventajas de la programación modular

1. **Código simple:** Pequeñas funciones son mas sencillas de leer que muchas sentencias.
2. **Reúso de código:** Si una tarea especifica se ejecuta varias veces, el uso de funciones reduce el duplicado de código
3. **Testing eficiente:** Encontrar errores es mas sencillo. Se puede hacer testing de cada función
4. **Desarrollo rápido:** Ligado al reúso de código. Una función puede ser utilizada para cualquier otro programa





Funciones Vacías

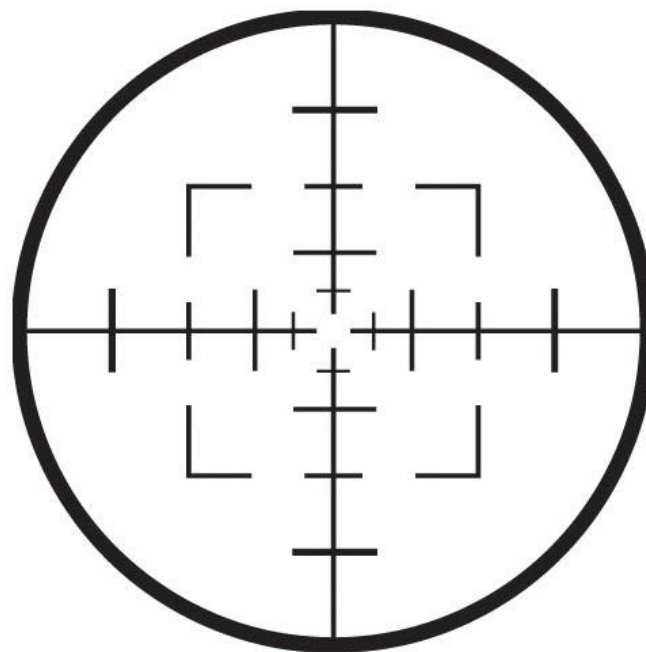
- Son funciones que simplemente ejecutan las sentencias que contienen y después terminan.

```
def nombreDeFuncion():  
    Sentencia  
    Sentencia  
    etc.
```

- Se hace notar la palabra reservada `def` y la indentación.

► Alcance de variables

- El alcance de las variables nos permite saber hasta donde pueden ser *vistas* las variables por todo el programa (*Las funciones que lo componen*).



► Variables Locales

- Si una variable local es creada dentro de una función su valor no puede ser accedido por otra función externa. Funciones diferentes pueden tener variables con el mismo nombre.

