

FUNCIONES

Se utilizan para **escribir código que se va a utilizar repetidamente**.

Sintaxis:

```
def1 nombre_funcion2 (parámetros/argumentos)3 :4  
    código5  
    return/print6
```

1. **def**: palabra clave para escribir la función

2. **nombre de la función**: nombre elegido, mejor sencillo e intuitivo

3. **paréntesis**: obligatorios

- **parámetros**: opcionales, variables que se definen en la función y reciben valores al aplicar la función.

Ej: def multiplicar(num1, num2):	multiplicar(2, 3)
return num1 * num2	<i>output: 6</i>

- **argumentos**: opcionales, valores que se dan a la función, números, strings, listas, tuplas, diccionarios...

4. **:** los dos puntos son obligatorios

5. **código** que se va a repetir al aplicar esa función

6. **return**: devuelve un valor que puede ser utilizado más adelante en el programa dentro o fuera de esta función. Return para el código.

print: muestra información en la pantalla. Imprime una variable o una cadena de texto, no un valor.

Variables locales: definidas dentro de una función y solo se aplican en esta, en el resto del archivo no existe esa variable.

Variables globales: están definidas para todo el archivo, las que ya hemos visto.

FUNCIONES LAMBDA

Una forma de definir funciones con solo un código y que solo van a usarse una vez.

No se definen con “def”, sino con “lambda”.

Se usan definiendo una variable

Sintaxis un parámetro:

Variable = lambda argumento: código

variablex = lambda x: x * 2

variablex(4) → output: 8

Sintaxis más de un parámetro:

Variable = lambda argumentos: código

variabley = lambda x, y: (x + y) * 2

variabley(1, 5) → output: 12

Random: genera números aleatorios o una muestra aleatoria en una secuencia.

Sintaxis:

```
import random
```

```
random.sample(population, k)
```

- population: población de donde sacar la muestra: cualquier secuencia iterable (lista, tupla, strings...)
- k: cantidad de la muestra

```
import random
```

```
variablez = random.sample(range(1,51), 20) → saca 20 números aleatorios de un rango del 1 al 50 en la variable llamada variablez
```