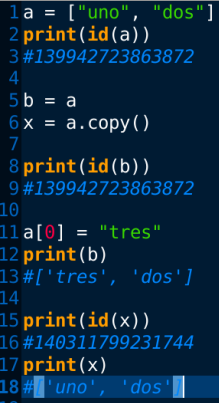
**Funciones Python**

**Principales conceptos:**

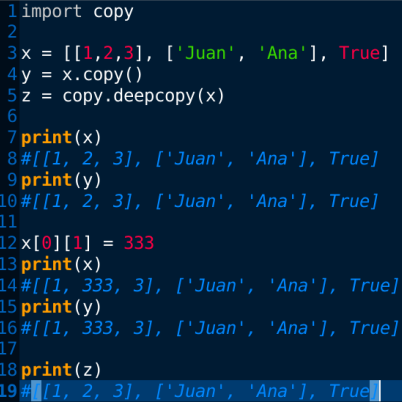
* **Mutabilidad:** define si un dato puede ser cambiado tras ser inicializado o si siempre mantiene el mismo valor, tanto para la variable original como para las demás referencias que tenga valor.



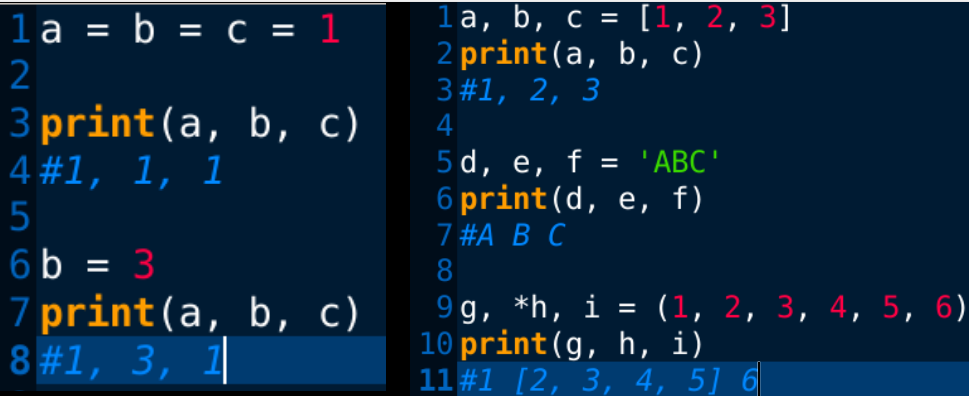
* **Puntero**: (dirección de memoria, apunta a un lugar de la memoria) variable cuyo contenido es la dirección de memoria de otra variable.

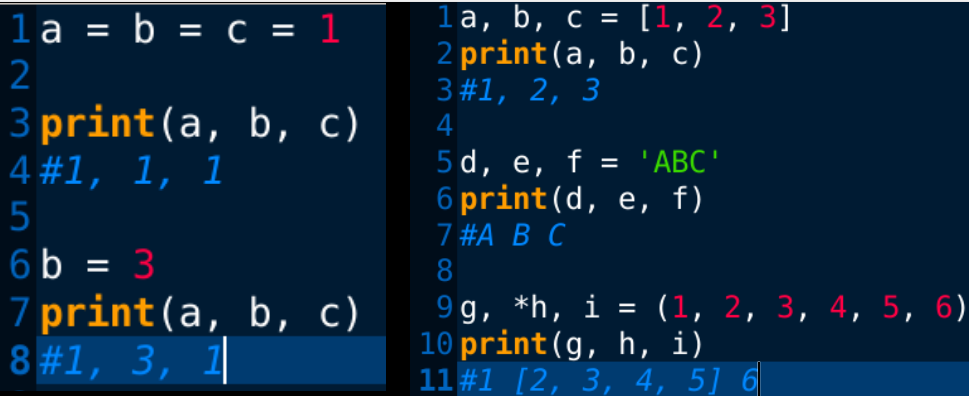


* **Copy():** clona un objeto en otro, pero es una copia superficial.

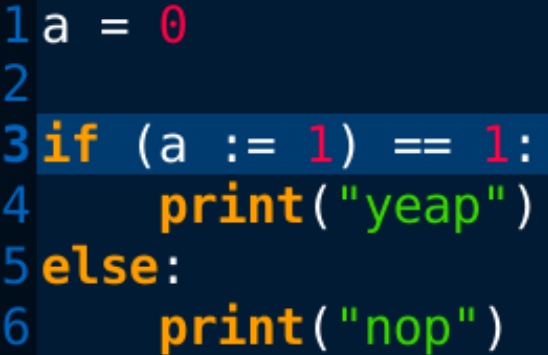


* **Asignaciones simples (e**jemplo: x = a**) y múltiples:** una de las operaciones más usadas en algoritmos. Permite la inicialización o redefinición del valor de una variable

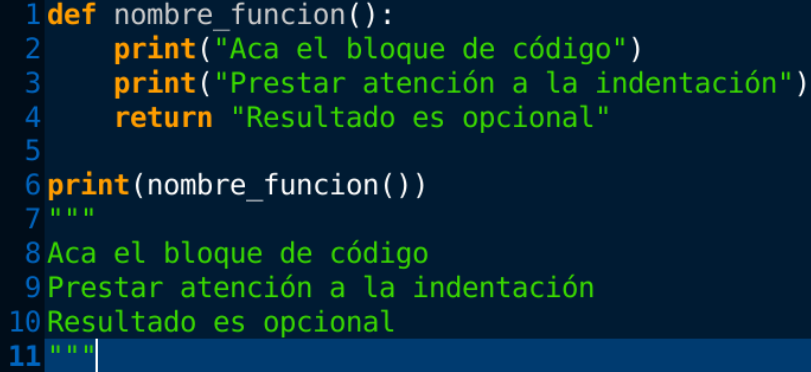




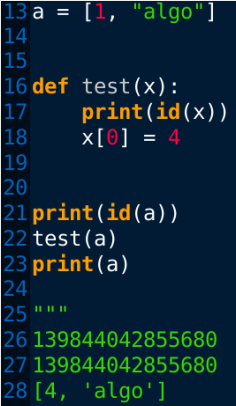
* **Operador Walrus (:=)**: desde Python 3.8 es la forma de asignar cuando se usa dentro de una expresión. Mejora la visibilidad.



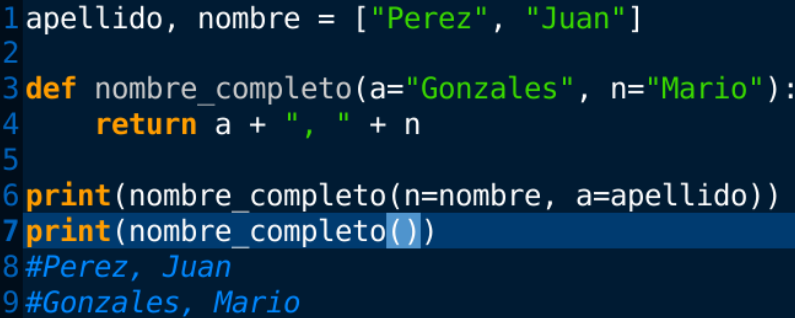
* **Funciones**: Subprograma o subrutina que forma parte de un algoritmo principal (procedimientos VS funciones).
* Ayuda a organizar mejor el código.
* Permite la reutilización de porciones de código y evita repetirse.
* Facilita el mantenimiento y testeo (divide y conquistarás)



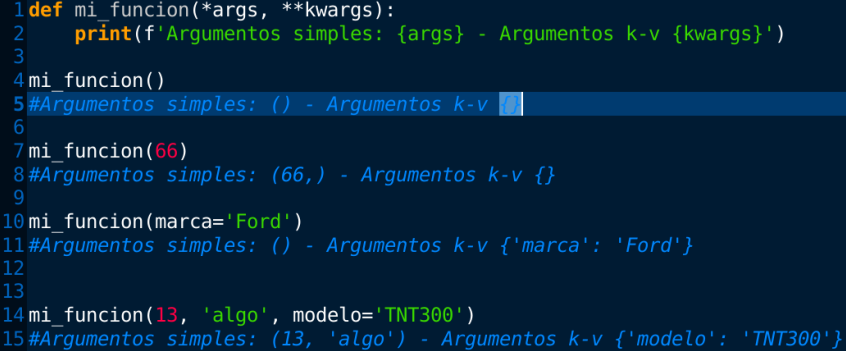
* **Parámetros y argumentos:** es el mecanismo para la introducción de variables o valores del exterior de la función.
* Los nombres de variables utilizados en la definición de la función son los parámetros.
* Las variables que portal los valores desde el exterior al utilizar la función son los argumentos.
* Todos los argumentos son pasados a la función por referencia.



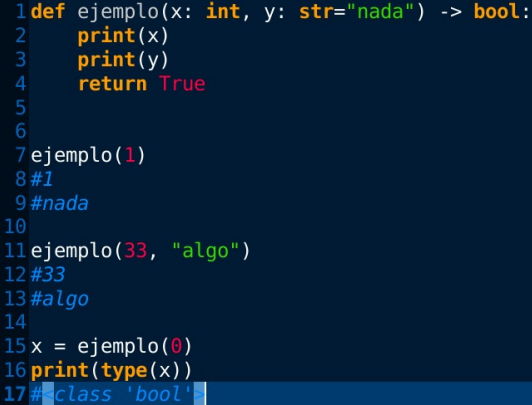
* **Argumentos**: pasados en el mismo orden que fueron definidos los parámetros o utilizando los nombres definidos - internamente.
* **Parámetros:** pueden tener valor por defecto si al invocar la función no se le pasan argumentos.



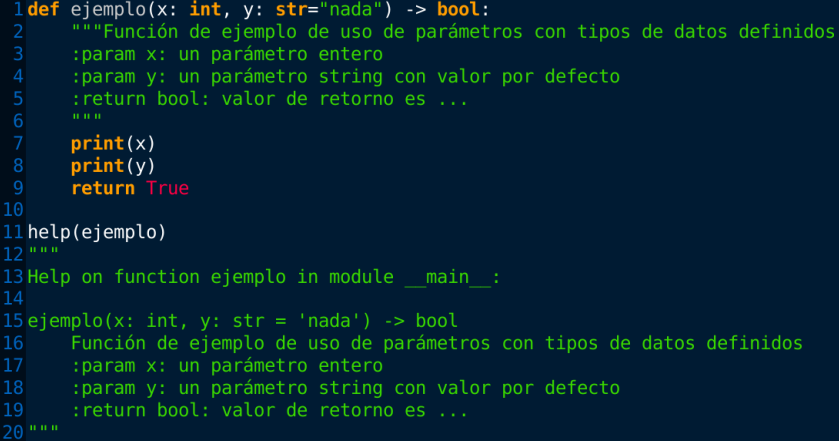
* **args y kwargs:**
* Mecanismos para pasar una cantidad dinámica de argumentos a una fución.
* Si son solo argumentos posicionales (en forma de lista simple) se usa \*
* Si son argumentos del tipo clave-valor se usa \*\*

****

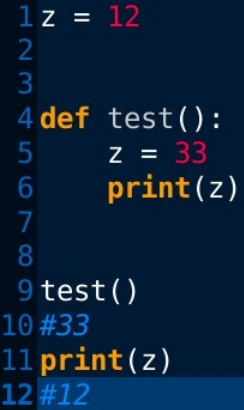
* **Tipado**: como python es tipado dinámico, se definen los tipos de dato de los parámetros al momento de la asignación en tiempo de ejecución. Si queremos forzar o restringir el tipo de dato a aceptar en la función se debe especificar.

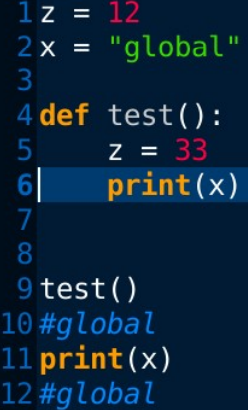
****

* **Documentación**: Sistema para documentar funciones. Consiste en utilizar un comentario de bloque inmediatamente despues de la definición y dentro de la misma.

****

* **Ámbito o contexto de una variable:** una variable es global cuando puede ser accedida desde cualquier parte del programa. Será local cuando solo pueda ser accedida en el contexto donde fue definida.
* Dependiendo dónde fueron declaradas o inicializadas serán variables globales o locales.







* **Excepciones**: gestión de casos anómalos o problemas inesperados. Por ejemplo, división entre 0 o acceso a variable no definida (null pointer) Se producen de forma automática o manual.

