Para aprende Python (8h 30”)

Python.org y lo descargamos

Codedex.io

https://www.youtube.com/watch?v=Kp4Mvapo5kc&list=PLNdFk2\_brsRdgQXLIlKBXQDeRf3qvXVU\_&index=1

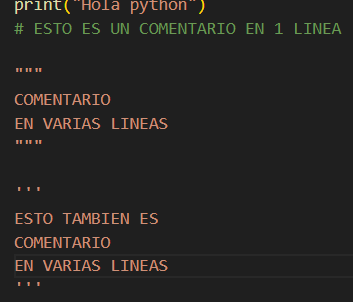
github de mour

<https://github.com/mouredev/hello-python>

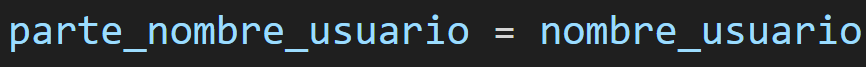
otro repo con info pero en ingles

<https://github.com/Asabeneh/30-Days-Of-Python>

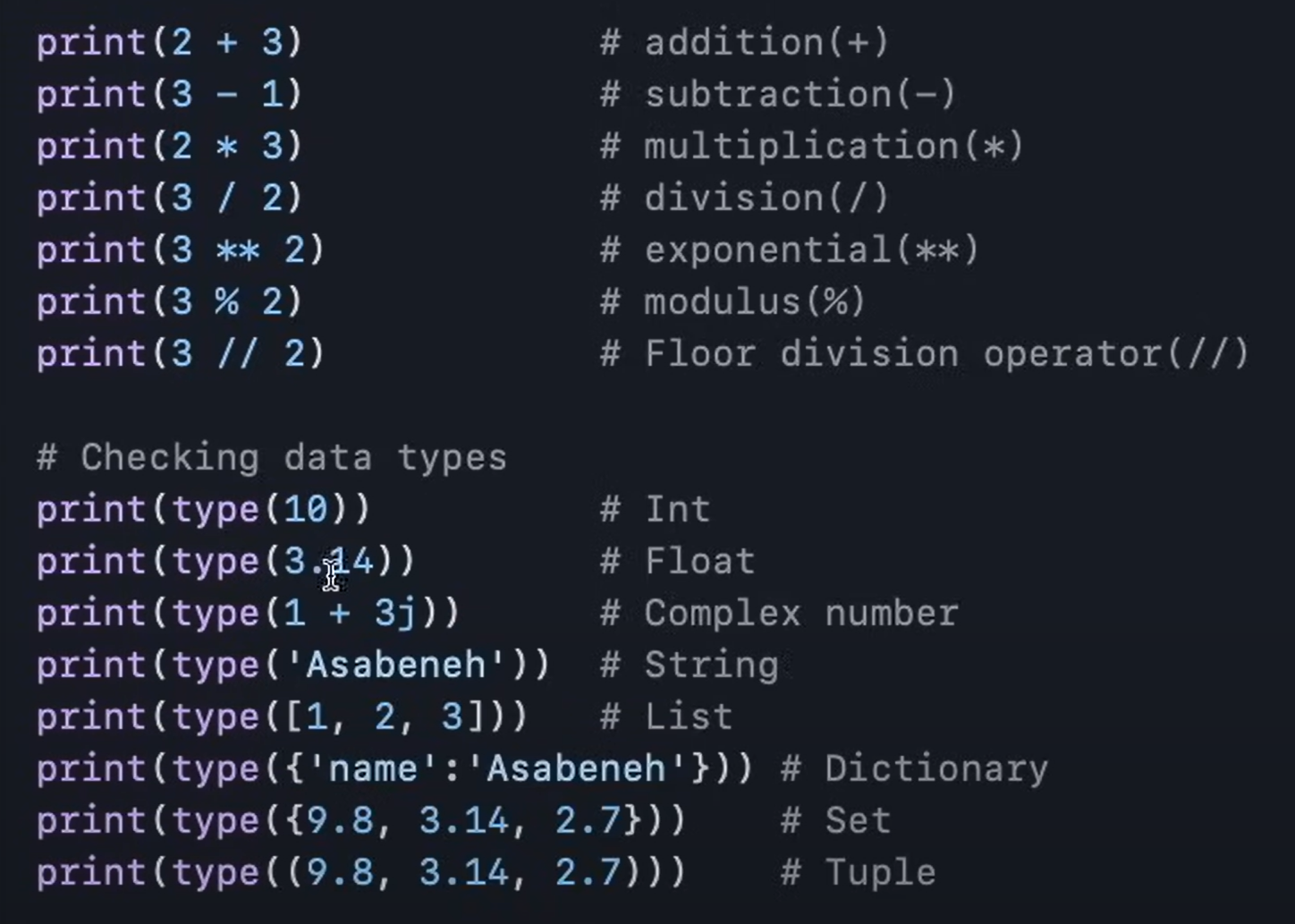
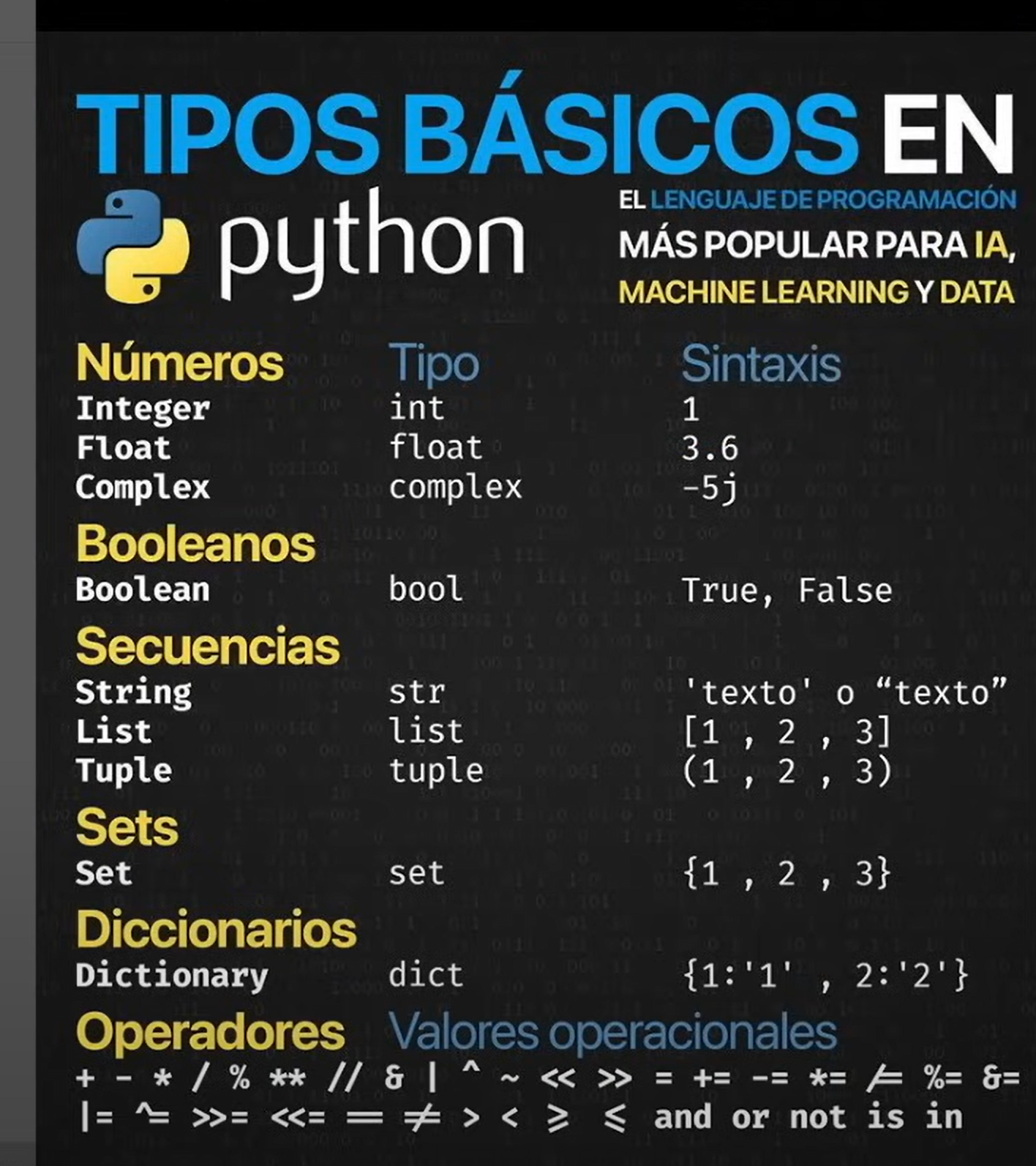
COMENTARIOS



La forma de nombra variables, listas, diccionarios, etc menos clases seria con forma Snake Keys (en java es Camel Keys)

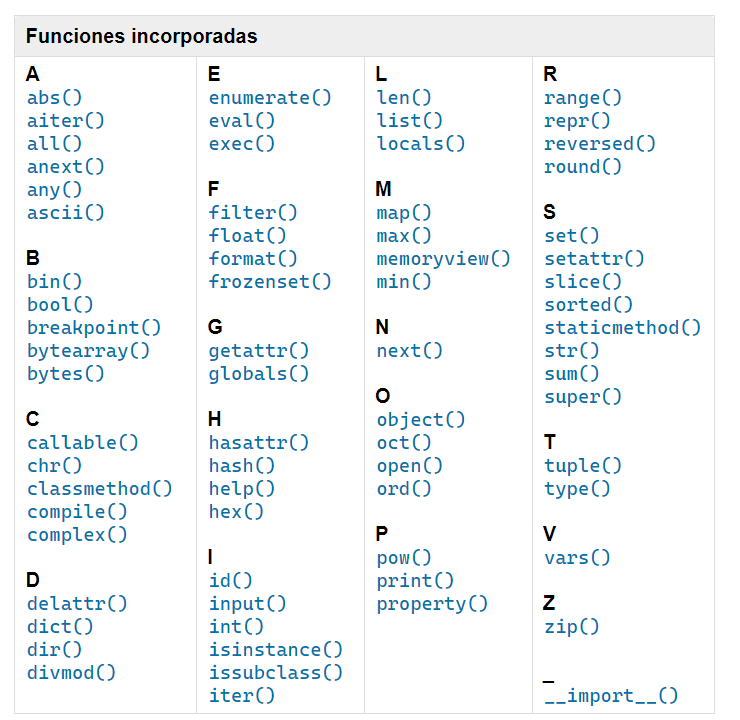


Numeros basicos y operaciones



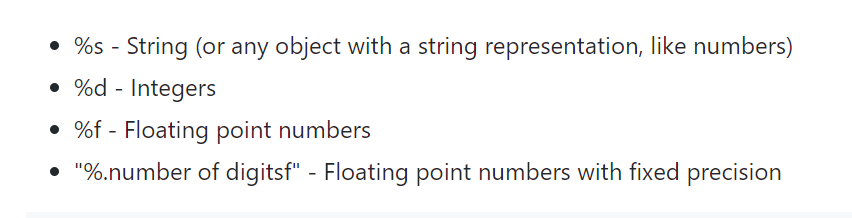
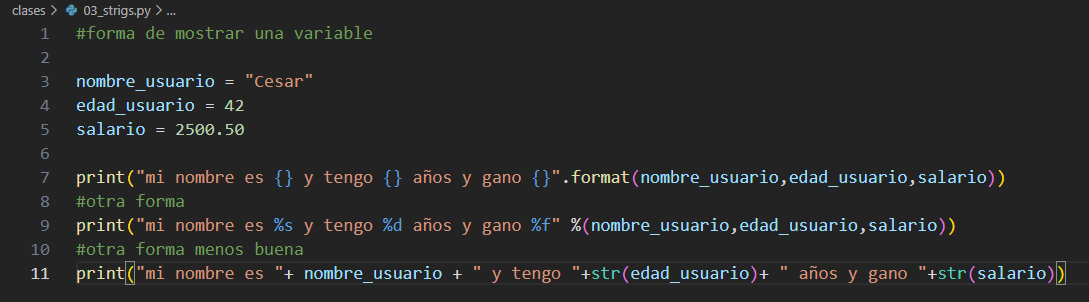
Funciones en Python que tiene incorporadas, por ejemplo, ordenación, o transformar un int en str o longitud etc. Las podemos leer en

<https://docs.python.org/es/3/library/functions.html>



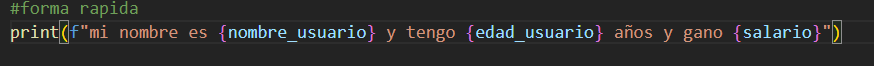
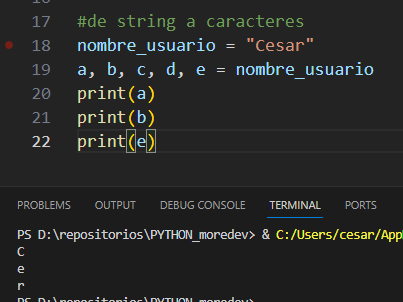
Forma de imprimir una variable o formatear

la última forma es más farragosa y menos limpia, aunque también es válida. La mejor forma seria la segunda porque así nos aseguramos que imprime el tipo de dato que realmente queremos (entero, texto, float ….)

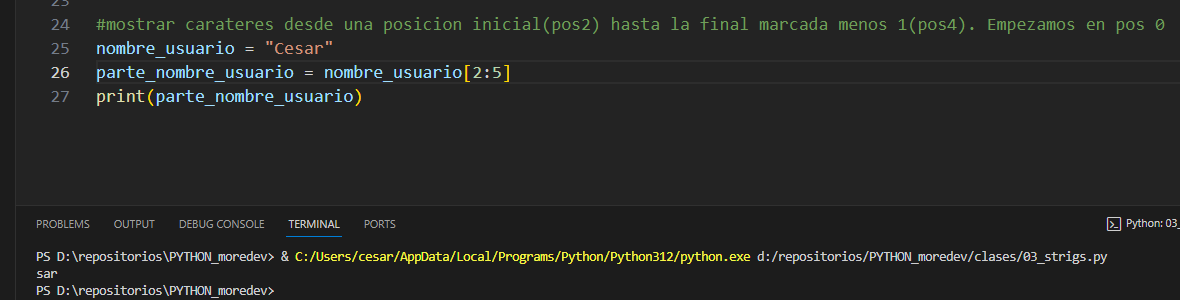


y la forma más rápida, aunque no la mejor porque como ya hemos dicho la segunda es mas correcta

Paso de string a caracteres o desempaquetado de caracteres

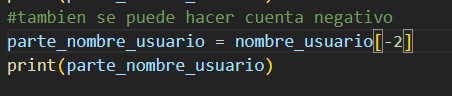


muestra caracteres en posición inicial y final del string que deseamos

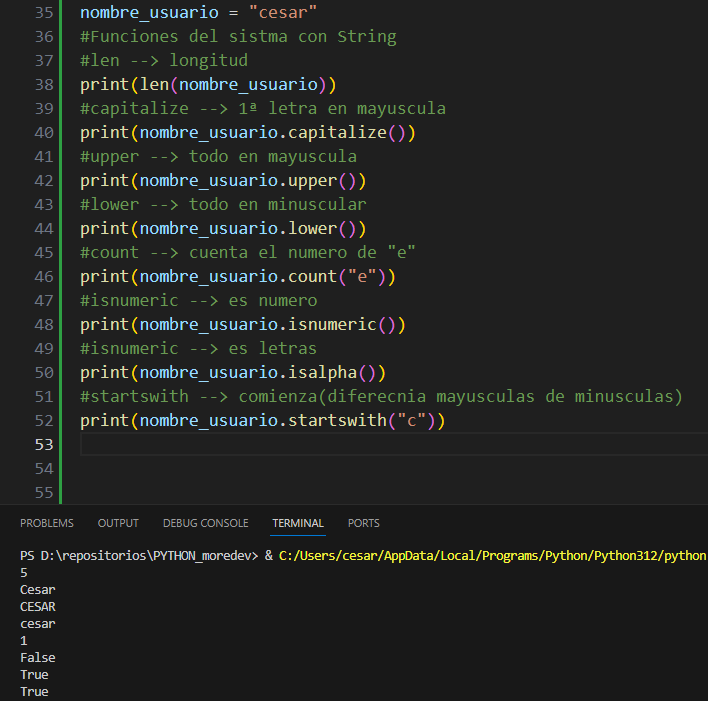
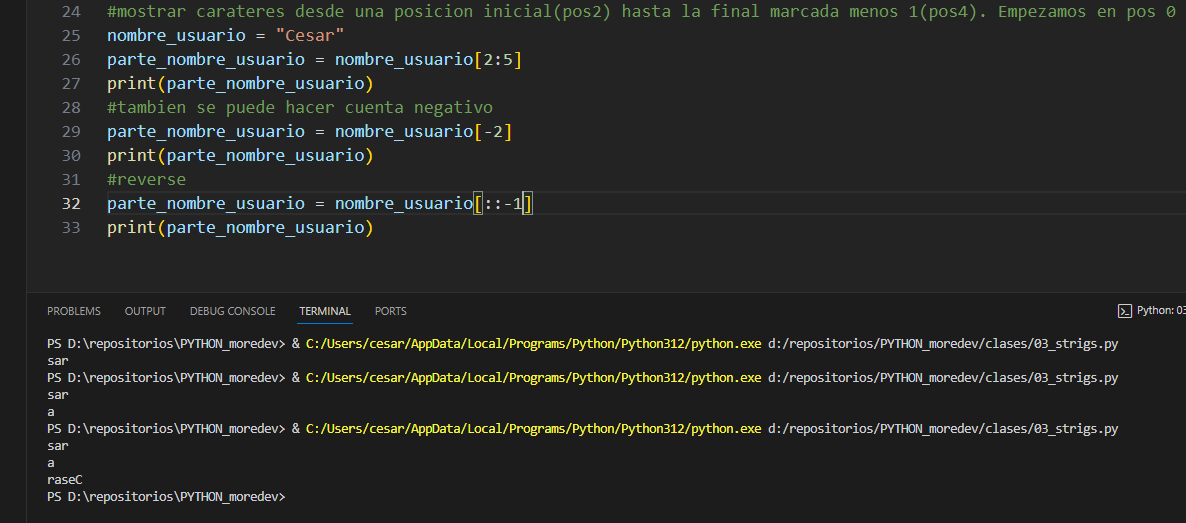


también podemos hacer cuenta negativa o empezando desde el final, que el ejemplo nos muestra el carácter 'a'

y si queremos darla la vuelta seria ::-1



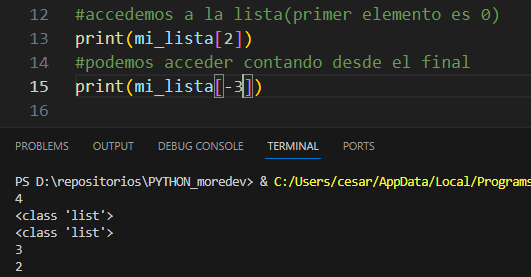
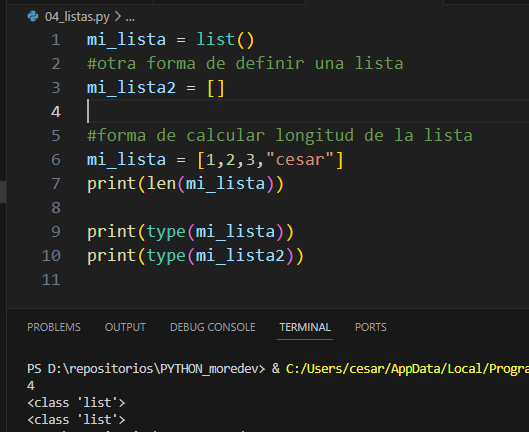
Funciones del sistema con string



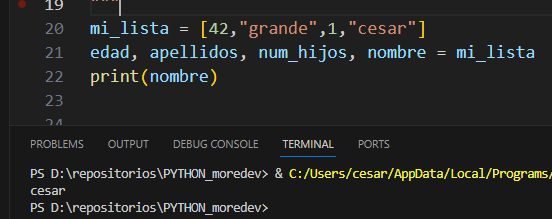
Listas

Las listas no es lo mismo que los arrays(las listas tienes funciones como ordenación, reverse, ….. y los arrays no) y se definen entre [ ]. Permite diferentes tipos de datos en la lista, de forma descolocada, repetir valores

se puede acceder contando desde el final de la lista poniendo -1 o -2 etc



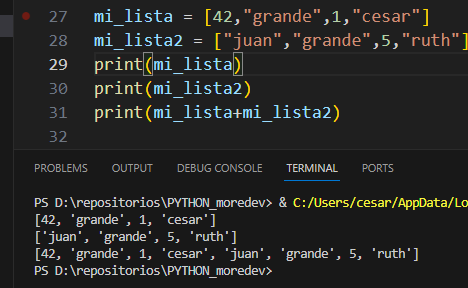
Empaquetar listas



forma de asignar a la lista de forma manual los valores



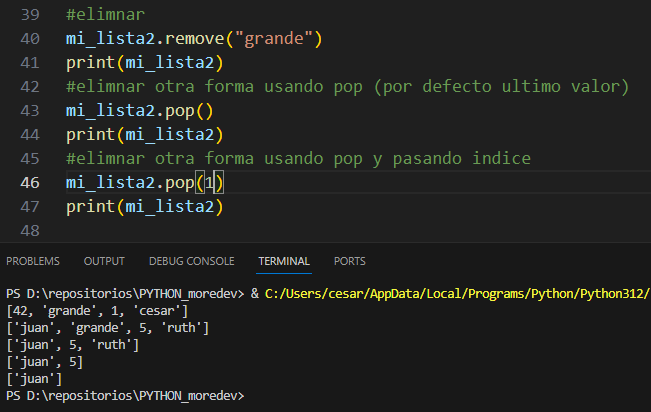
se puede sumar listas, pero no restar, ni multiplicar, dividir … solo sumar



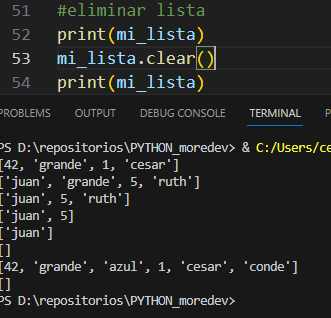
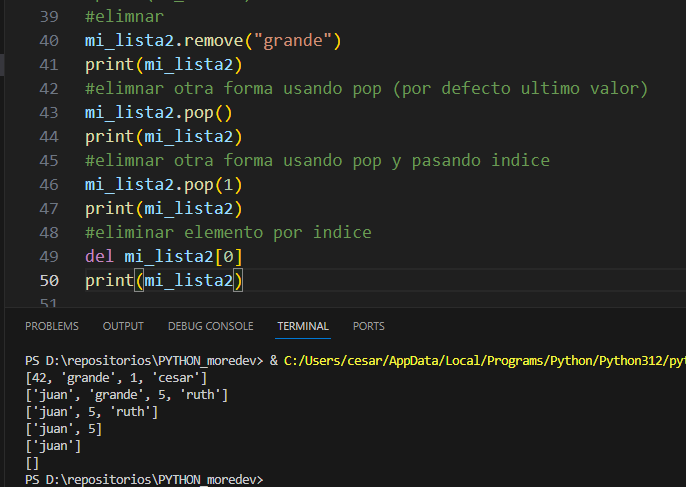
añadir elementos a la lista append y también insert diciendo indice y objeto a insertar. También podemos eliminar un objeto con remove, si ya tenemos elementos con el mismo valor, solo elimina el primero que encuentra. Si queremos eliminar el ultimo valor usaremos pop o pop e indice pero la diferencia es que con el pop me interesa guardar ese elemento que hemos borrado de la lista en otra variable por ejemplo y con el remove le elimino directamente.



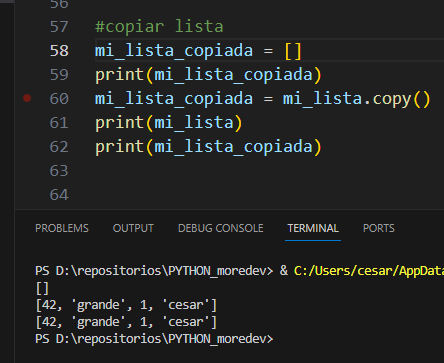
Eliminar elemento por indice del mi\_lista[indice]



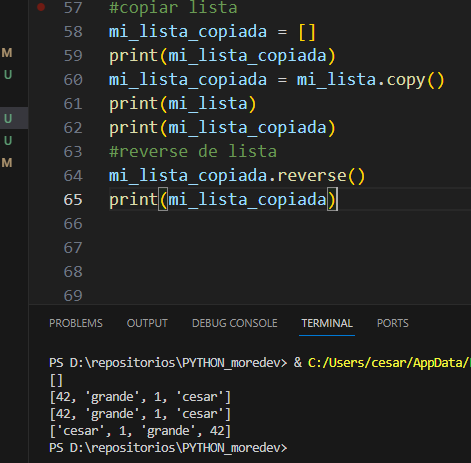
limpiar la lista con clear



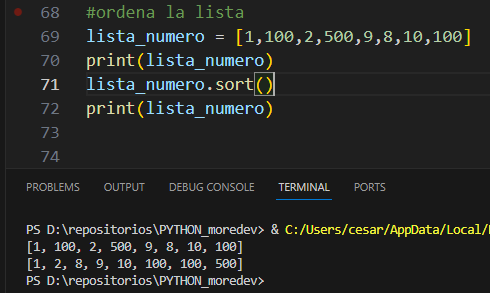
copiar lista con copy



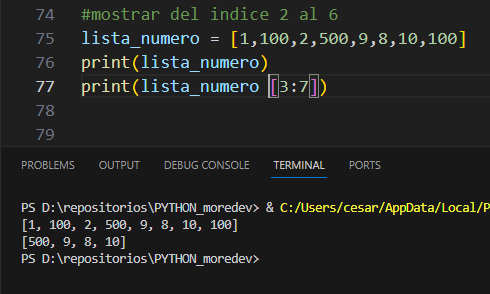
dar la vuelta a la lista con reverse



si queremos ordenar la lista podemos hacerlo alfabeticos con sort

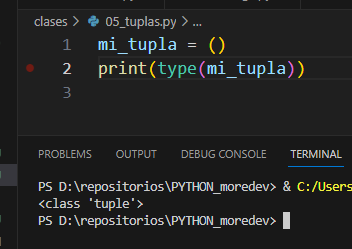
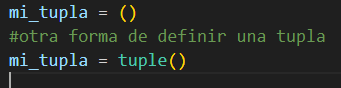


mostrar rango de la lista que le indiquemos con los indices tambien llamado slices (sublista)

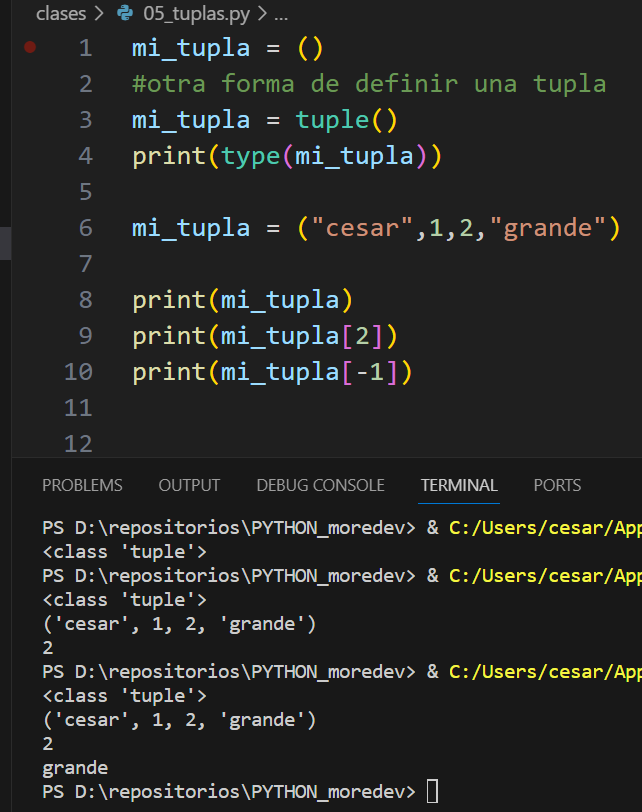


TUPLAS

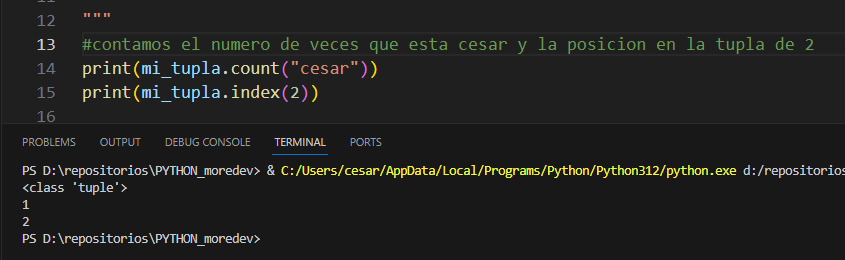
Una tupla se diferencia de la lista es que inmutable, una vez que se inicialize la tupla no deja ni cambiar valores ni insertar mas valores



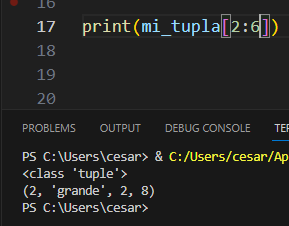
Forma de acceder a los valores



La tuplas permite count e index



Te permite tambien mostrar lo que queramos con los slices

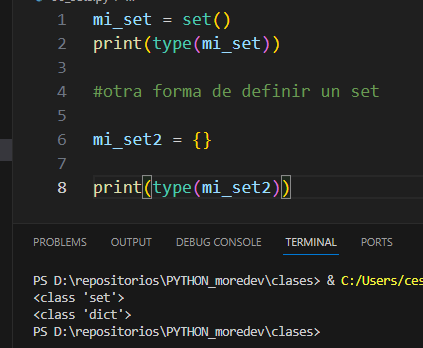
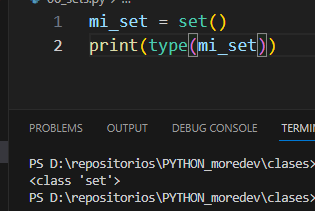


Podemos borrar la tupla, no es vaciar la tupla es borrar la variable

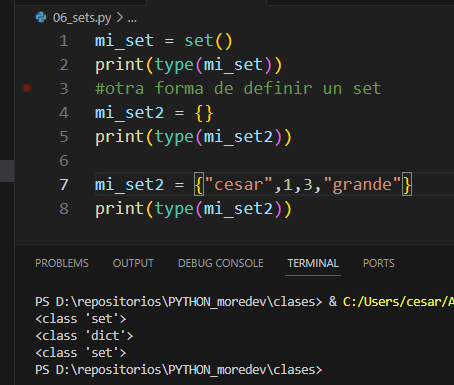


Sets o conjuntos

Dentro del conjuto de datos no se pueden repetir valores



Como vemos en el ejemplo si lo inicializamos mi\_set= {} nos indica que es un diccionario, como el tipado de python es débil en el momento que rellenemos el set no indicara que es un set en vez de un diccionario. Ver ejemplo

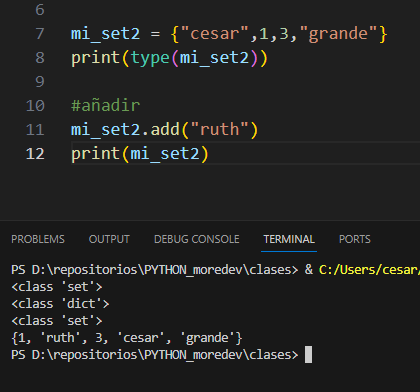


En un sets no existe un orden establecido para sus elementos, por lo tanto no podemos

acceder a un elemento en concreto.

De este hecho se deriva igualmente que no podemos modificar un elemento existente,

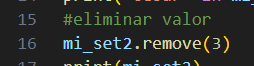
ya que ni siquiera tenemos acceso al mismo.



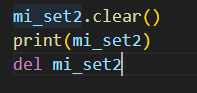
Busqueda de elementos en los sets, lo cual devuelve true o false



También podemos eliminar datos



Podemos eliminar todos los datos del set con clear y eliminar con del



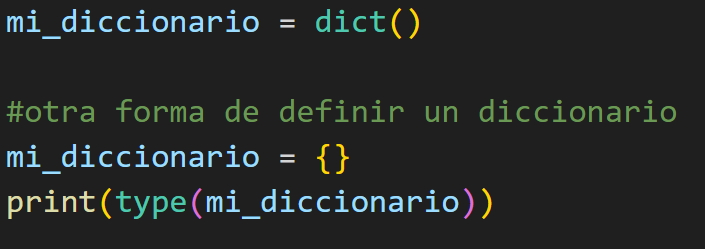
Se puede unir dos sets con union



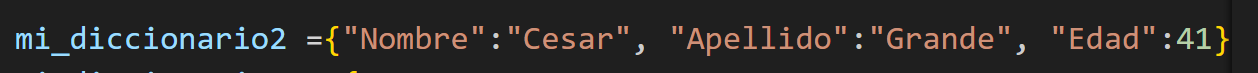
Y se puede hacer la diferencia



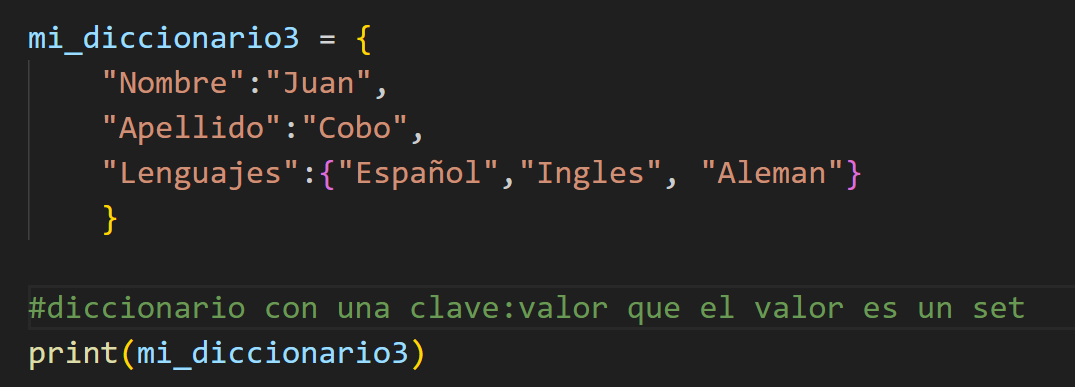
**Diccionarios**



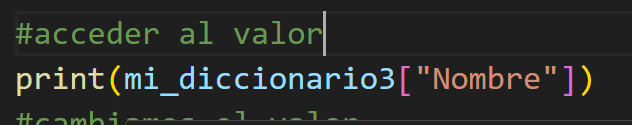
La forma de entender esto es clave:valor



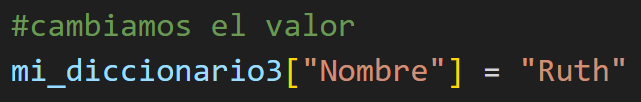
Posibilidad de mezclarde conceptos, dentro de un diccionario podemos meter un set o lo que necesitamos



Para acceder al valor en este caso imprime Juan



Cambiamos el valor que tiene una clave



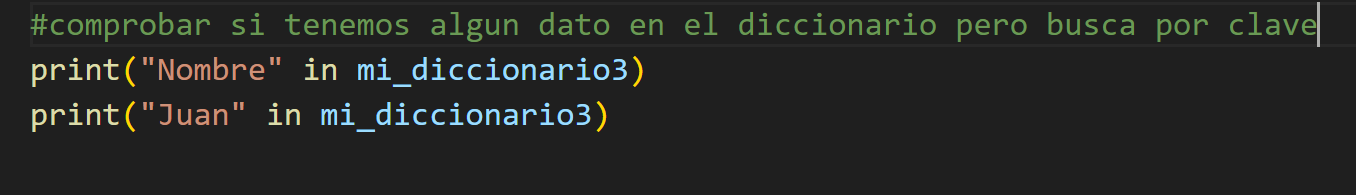
Añadimos datos al diccionario



Borrar elemento



Si buscamos por la clave nos devuleve true en el caso de nombre y false en el caso de Juan



Operaciones con diccionarios

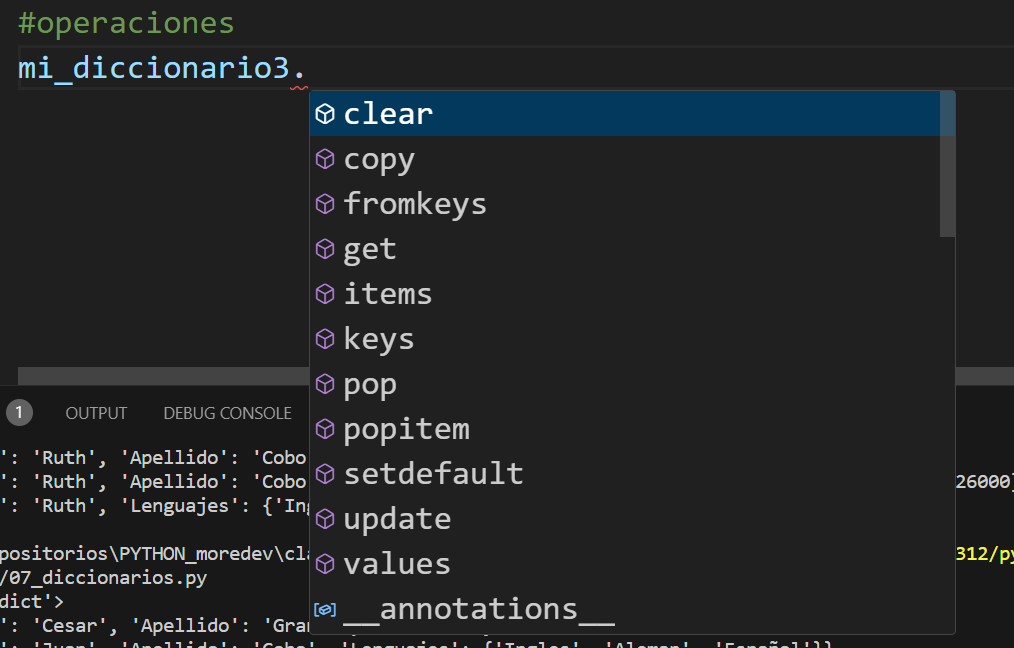
Items --> devuelvo solo los valores

Keys --> devuelvo solo las claves

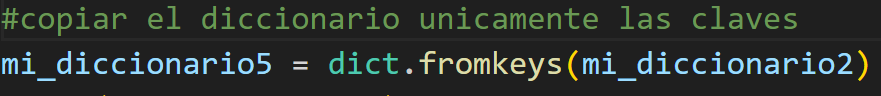
Clear --> limpiar

Copy --> copiar diccionario

etc



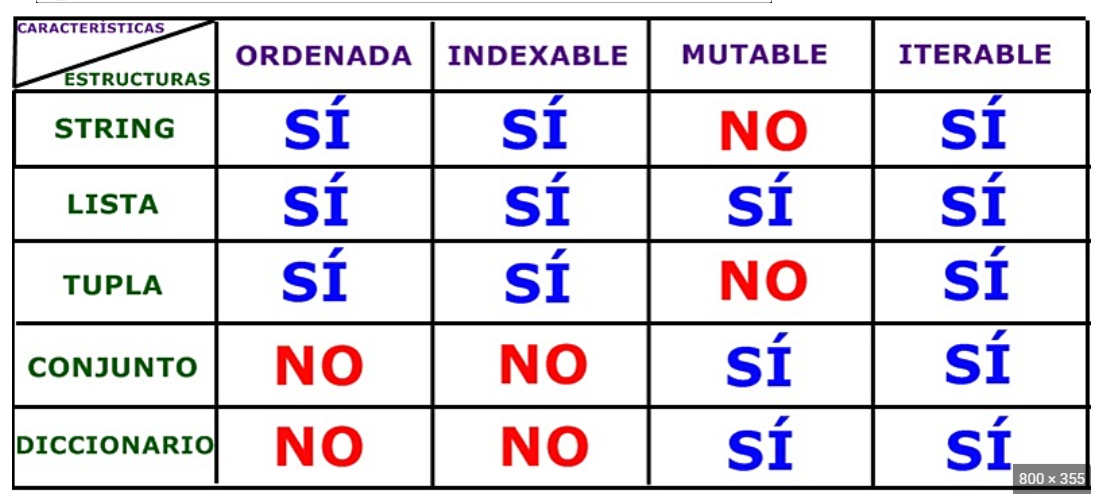
copiar el diccionario unicamente las claves



Obtenemos lo siguiente si hacemos un print al mi\_diccionario5



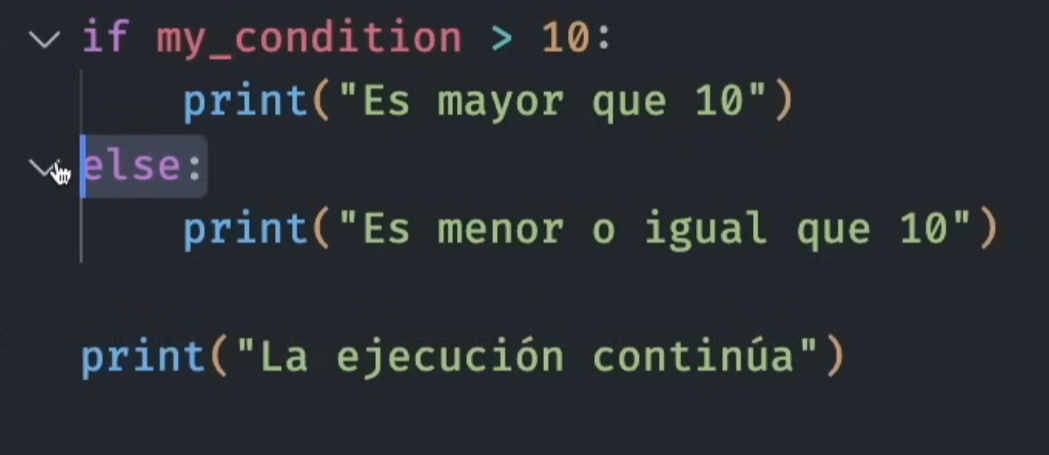
**RESUMEN DE LAS CARACTERISTCAS DE LISTA, TUPLA, SET(O LLAMADO CONJUNTO) Y DICCIONARIO**



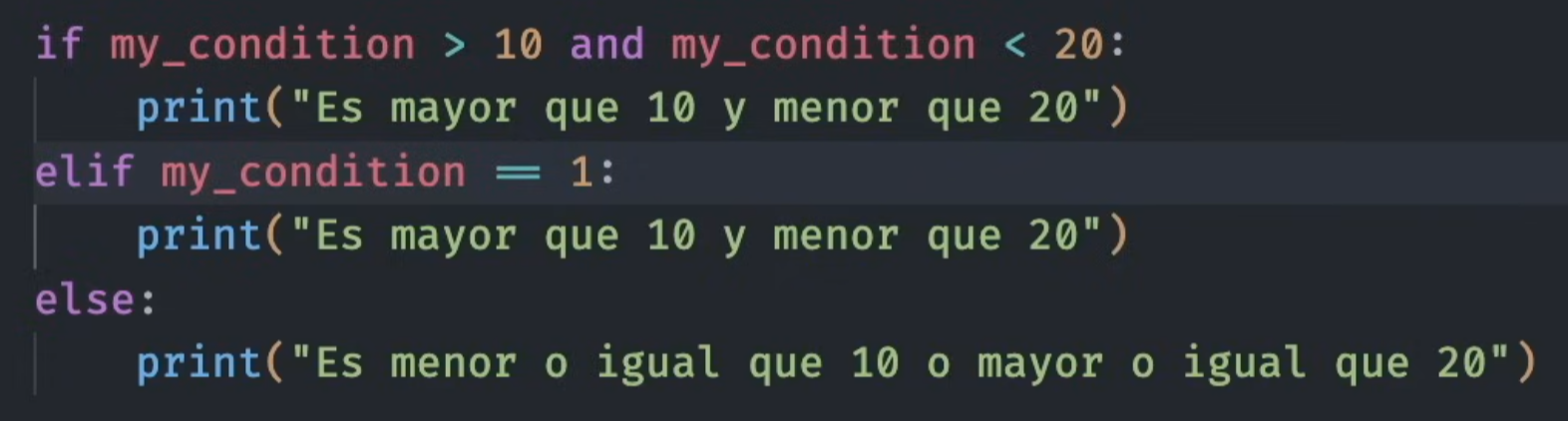
**Condicionales**

If condicion:

Print(“holas”)

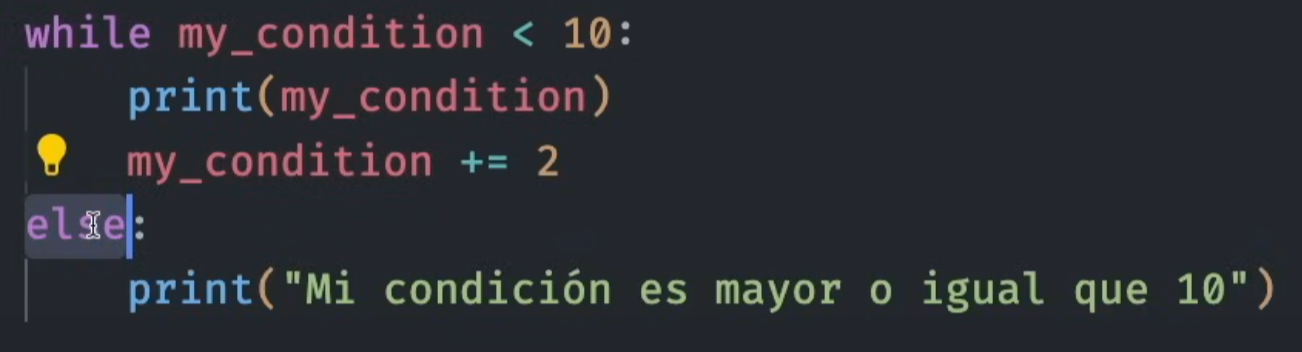


En python la diferencia de java es que el if elseif que los junta en **elif**

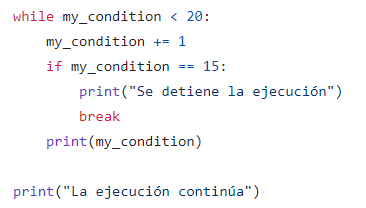


Loops o bucles o ciclos

While es igual que en java pero en python al while se le puede meter un else



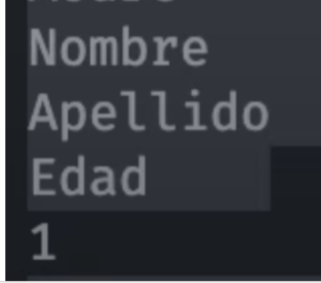
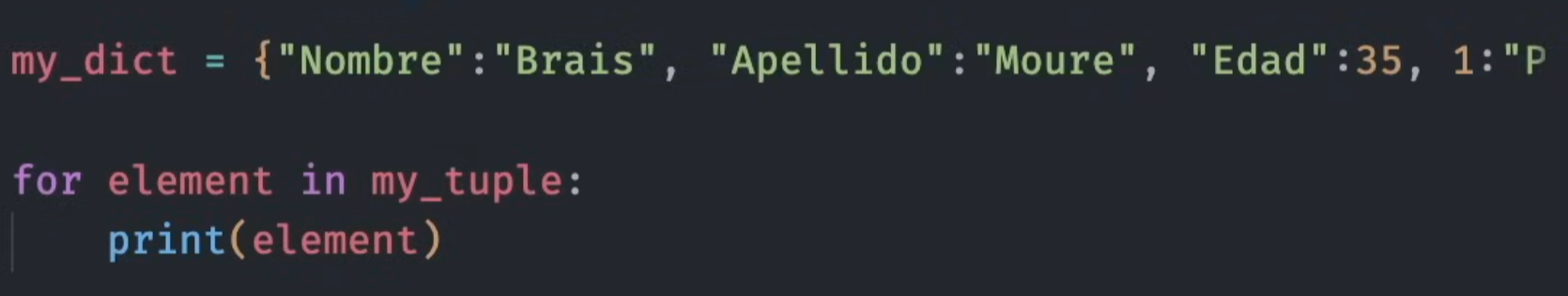
Se puede detener un while con break



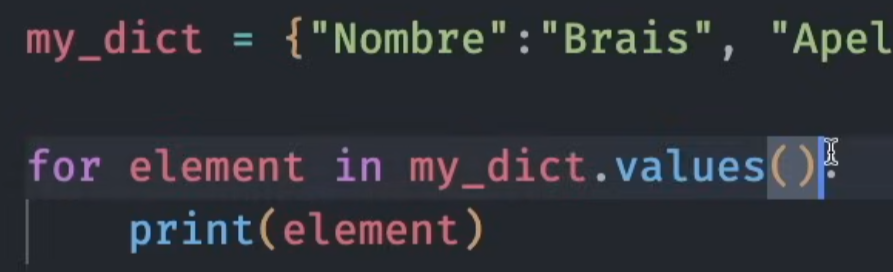
For



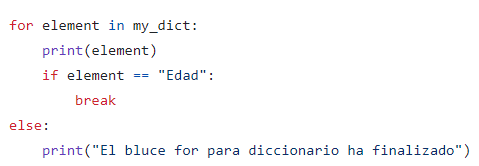
En los diccionarios cuando hacemos un for se imprime la clave y no los valores, el resto como la lista o la tupla se imprime los valores que estén dentro.



Si queremos imprimir los valores debemos pasar los valores del diccionario a una lista y entonces ya se imprimen dichos valores

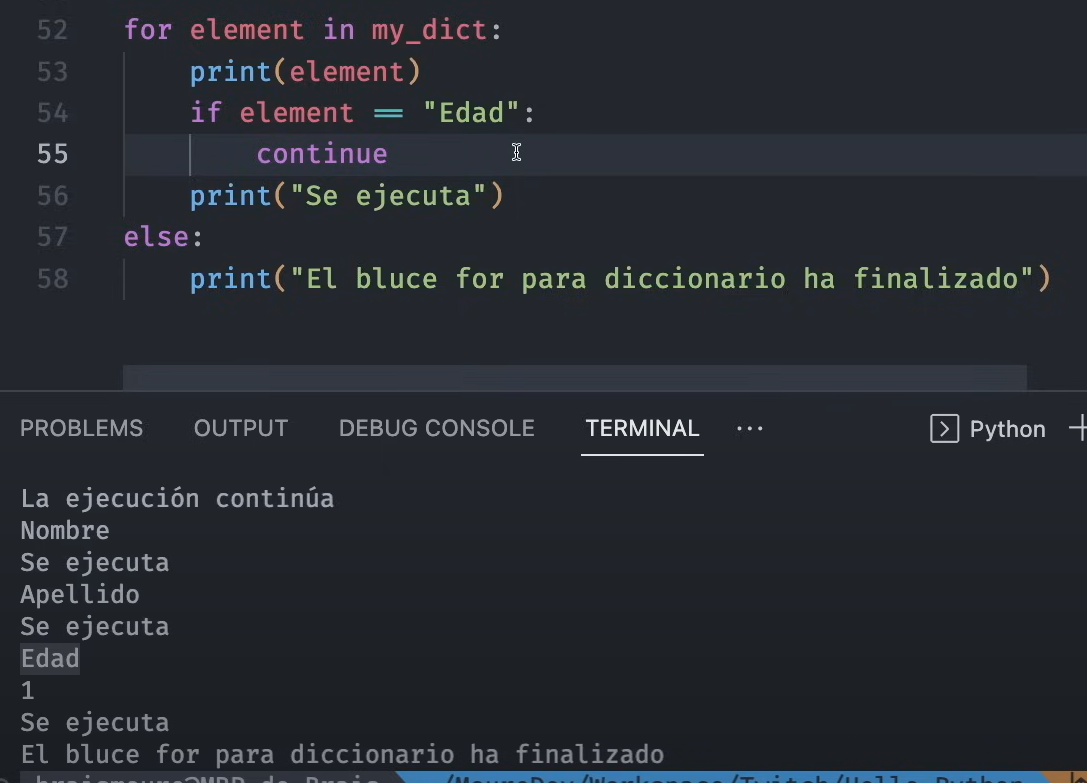


Podemos parar el for con break



Cuando llegue a Edad se sale del For y no imprime el else

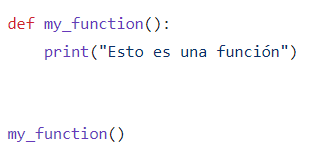
Lo siguiente es continue que lo que hace una vez que se cumpla la condición, es volver al for sin ejecutar lo que esta debajo, salta hasta la siguiente iteracion



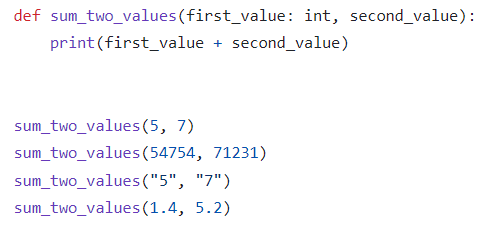
Y si usamos PASS en el for no haria nada

FUNCIONES

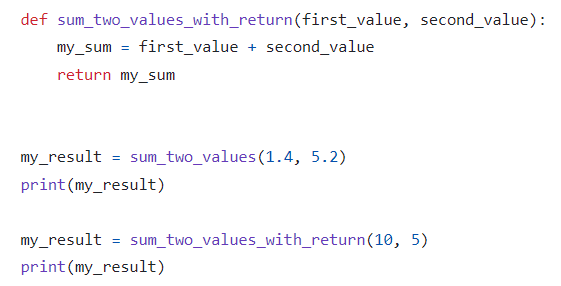
Llamada básica de función



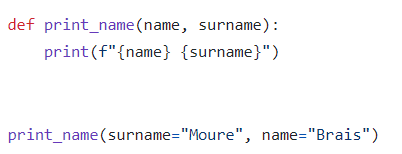
Función con parámetros de entrada/argumentos



Función con parámetros de entrada/argumentos y retorno

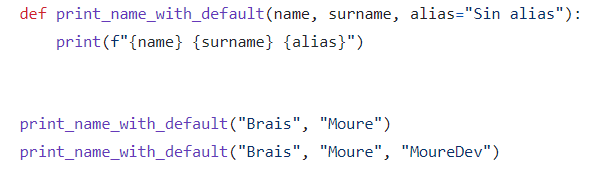


Función con parámetros de entrada/argumentos por clave



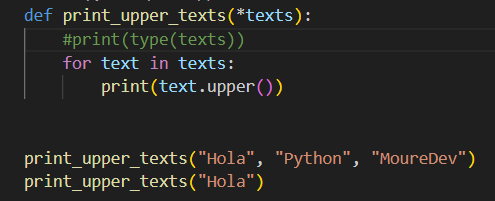
Función con parámetros de entrada/argumentos por defecto

En este caso sino pasamos el argumento alias en la llamada, por fecto nos pone el valor que definimos en la función, en nuestro caso “Sin alias”

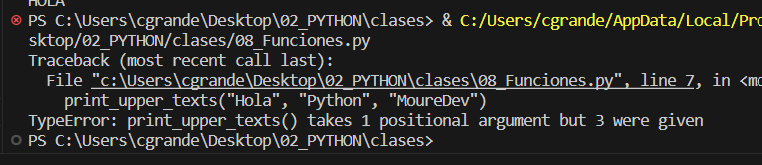


Función con parámetros de entrada/argumentos arbitrarios,

Con esta funcion le paso valores sin definir el numero exacto de parametros ( asi no da error), en el ejemplo le paso 3 paramentros y en la segunda llamada 1



Si quito el \* sale error de parametros

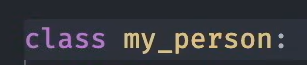


CLASES

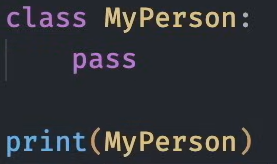
La forma de nombrar las clases es (forma Camel Keys)



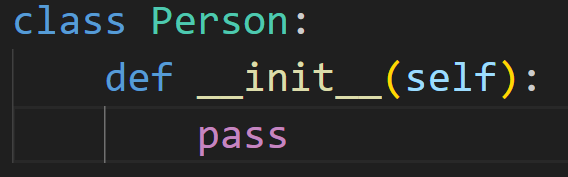
Esta forma no seria correcta en buenas prácticas en Python(forma Snake Keys)



Forma de crear clase sin argumentos seria con el pass pero no es lógico crear una clase vacía, sino usas el pass da error porque toda clase debería tener argumentos

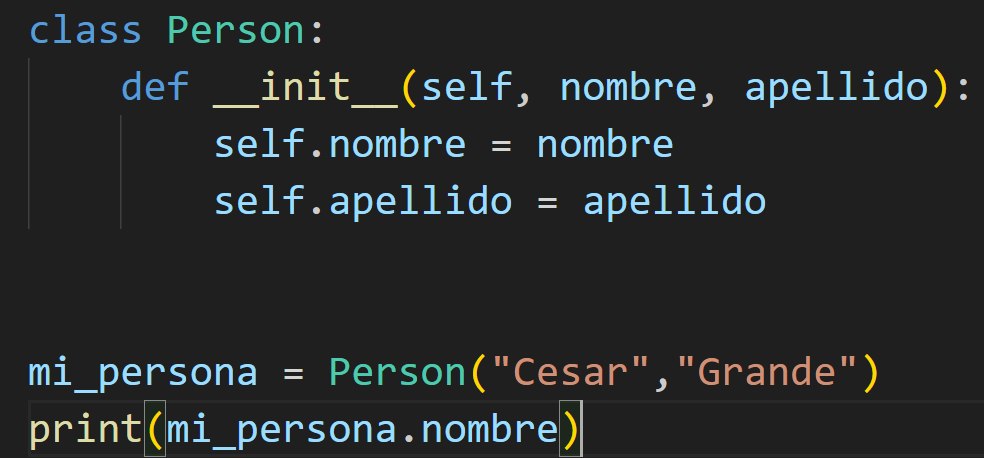


Constructor de clase



Ante de empezar, la forma correcta de trabajar seria crear un fichero por cada clase y despues con import llamamos a cada clase que necesitemos, no tener un unico fichero con todas las clase, etc

Creamos una clase con argumentos pasados en la llamada y accedemos a ellos



Otra forma de mostrar

