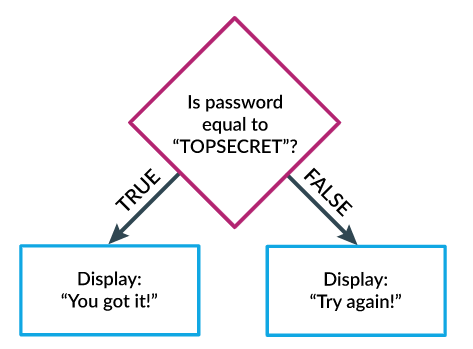
VALENTINA SANCHEZ VILLAMIZAR

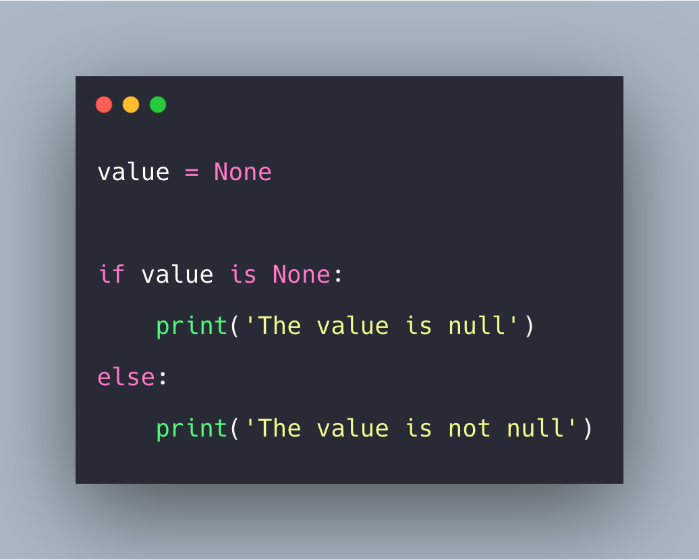
FALSE Y TRUE

Una variable booleana es una variable que sólo puede tomar dos posibles valores: True (verdadero) o False (falso). En general los elementos nulos o vacíos se consideran False y el resto se consideran True.



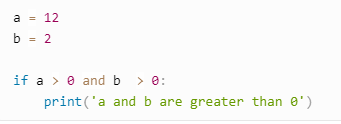
NONE

None de Python, denota falta de valor. Este objeto no tiene métodos. Debe tratarse como cualquier otro objeto con respecto a los recuentos de referencia. Podríamos decir que su traducción sería nulo y significa que el valor es desconocido, por ejemplo podemos declarar una variable con valor = Null y sería algo así como que el valor de esa variable es desconocido.



AND

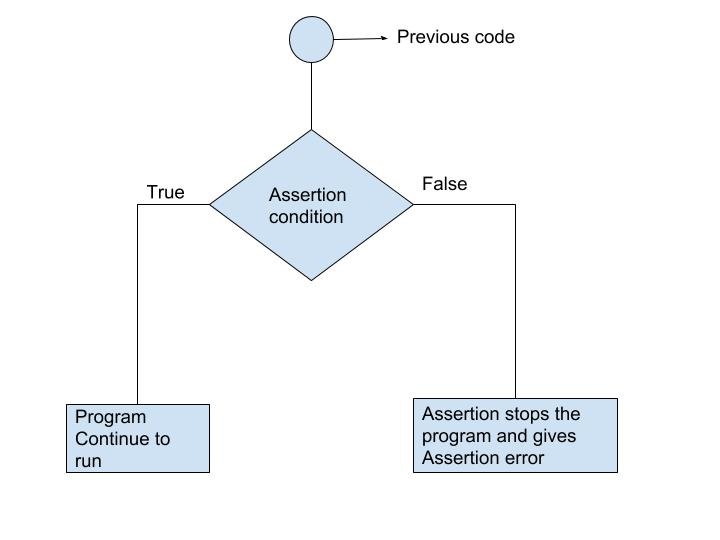
El operador AND lógico ( && ) devuelve true si ambos operandos son true y devuelven false de lo contrario. Los operandos se convierten implícitamente al tipo bool antes de la evaluación y el resultado es de tipo bool .



AS

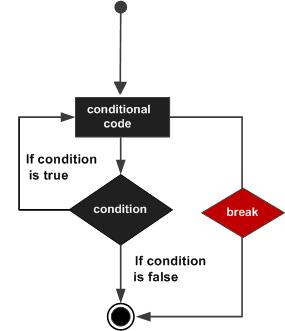
ASSERT

Los AssertionErrors son errores que pueden levantarse al cumplirse una condición. Se diferencian del raise Exception porque no es necesario escribir explícitamente la condición, ya que el assert valida la condición y, si no se cumple, levanta el error.



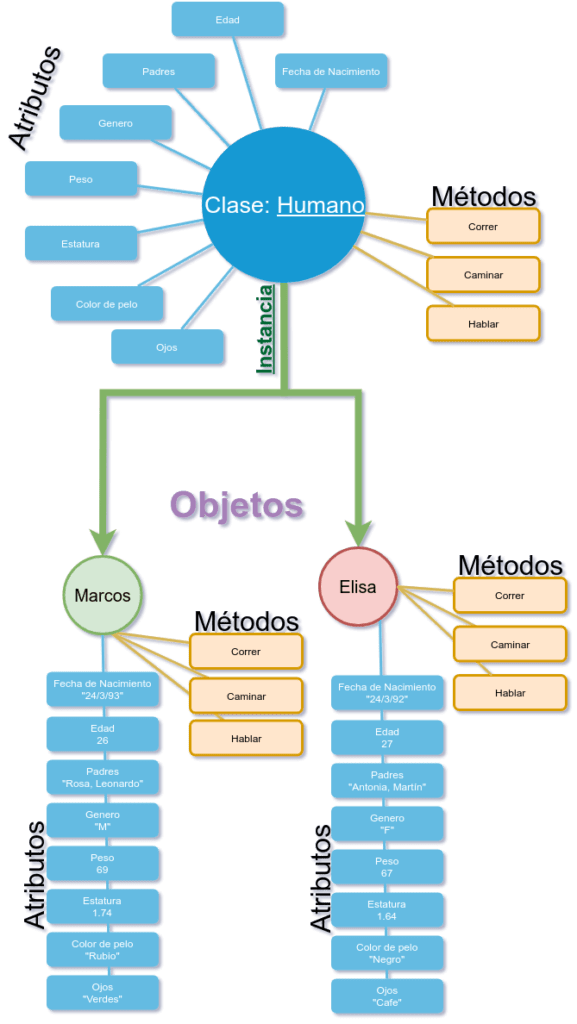
BREAK

La instrucción break le proporciona la oportunidad de cerrar un bucle cuando se activa una condición externa. Debe poner la instrucción break dentro del bloque de código bajo la instrucción de su bucle, generalmente después de una instrucción if condicional.



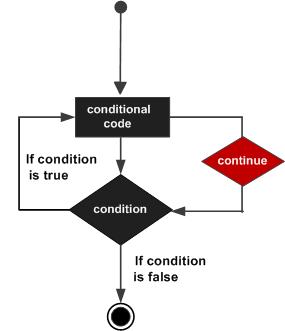
CLASS

Las clases proveen una forma de empaquetar datos y funcionalidad juntos. Al crear una nueva clase, se crea un nuevo tipo de objeto, permitiendo crear nuevas instancias de ese tipo. Cada instancia de clase puede tener atributos adjuntos para mantener su estado.



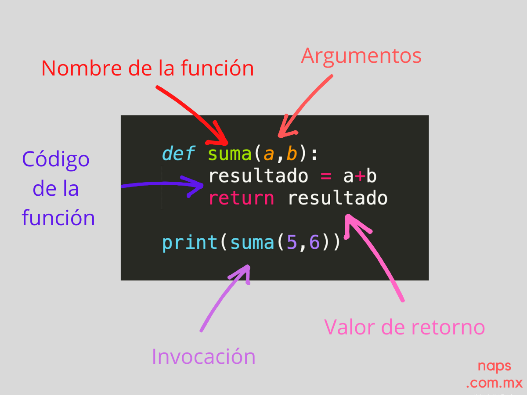
CONTINUE

La instrucción continue da la opción de omitir la parte de un bucle en la que se activa una condición externa, pero continuar para completar el resto del bucle. Es decir, la iteración actual del bucle se interrumpirá, pero el programa volverá a la parte superior del bucle.



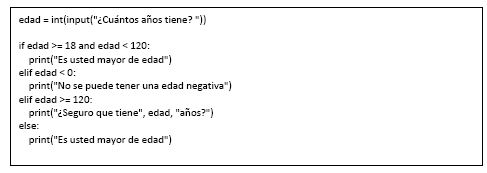
DEF

def es una palabra reservada que indica a Python que una nueva función está siendo definida. Luego viene una función con un nombre válido de tu elección. Los nombres válidos empiezan con una letra o un guion bajo, pero pueden incluir números.



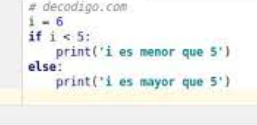
ELIF

Elif podría traducirse como Else If y se utiliza para enlazar varios else if sin aumentar las tabulaciones, vendría a ser como un “si es”



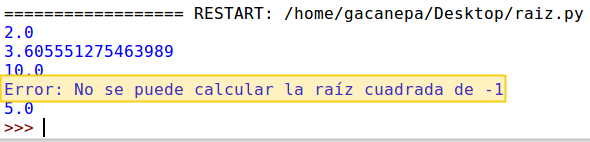
ELSE

Una sentencia if..else en Python significa: "Cuando la expresión if se evalúa como True, entonces ejecuta el código que le sigue. Pero si se evalúa como False, entonces ejecuta el código que sigue después de la sentencia else ."



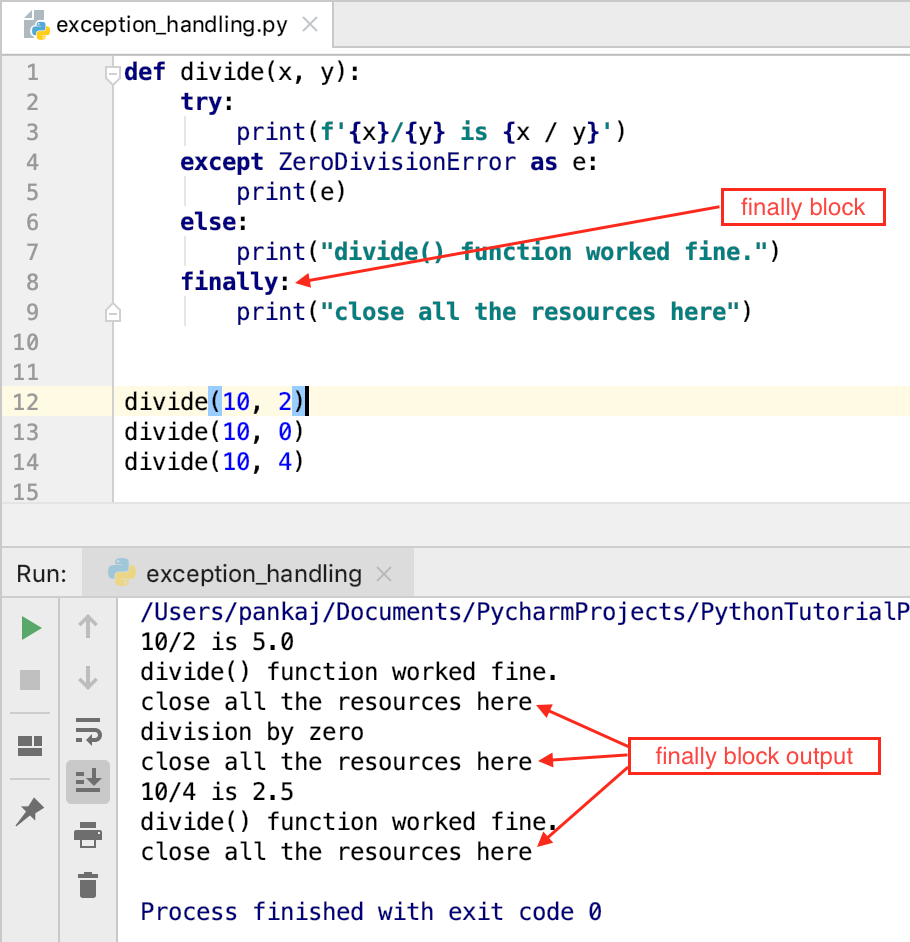
EXCEPT

El bloque except se ejecutará cuando el bloque try falle debido a un error. Este bloque contiene sentencias que generalmente nos dan un contexto de lo que salió mal en el bloque try.



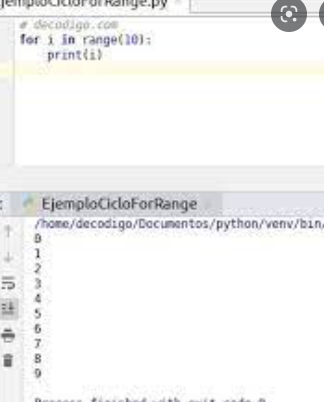
FINALLY

Puede ubicarse un bloque finally donde se escriben las sentencias de finalización, que son típicamente acciones de limpieza. La particularidad del bloque finally es que se ejecuta siempre, haya surgido una excepción o no.



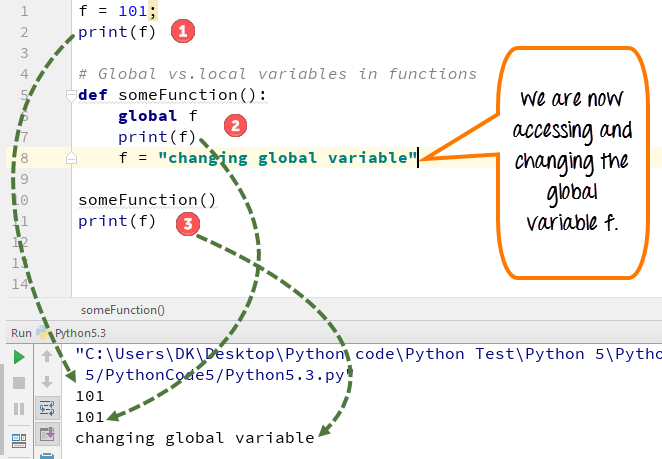
FOR

Un bucle for establece la variable iteradora en cada valor de una lista, arreglo o cadena proporcionada y repite el código en el cuerpo del bucle for para cada valor de la variable iteradora.



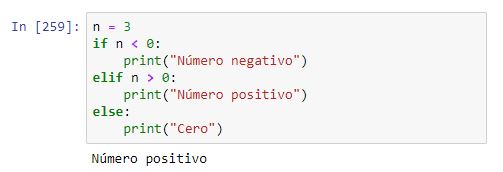
GLOBAL

Son aquellas definidas en el cuerpo principal del programa fuera de cualquier función. Son accesibles desde cualquier punto del programa, incluso desde dentro de funciones. También se puede acceder a las variables globales de otros programas o módulos importados.



IF

En Python, la sentencia if se utiliza para ejecutar un bloque de código si, y solo si, se cumple una determinada condición. Por tanto, if es usado para la toma de decisiones. Es decir, solo si condición se evalúa a True, se ejecutarán las sentencias que forman parte de bloque de código.



IMPORT

Al usar la forma from ... import ... , le indicamos a Python específicamente cuáles objetos queremos importar de un módulo en particular. Luego, los objetos importados se incorporan a nuestro archivo como cualquier otro objeto definido dentro de él. Por esta razón, no es necesario (ni posible) usar el prefijo mimodulo

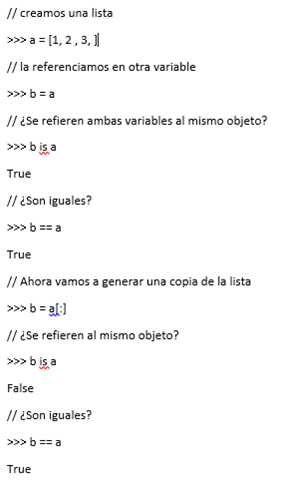


IN

In devuelve True si el valor especificado se encuentra en la secuencia. En caso contrario devuelve False.

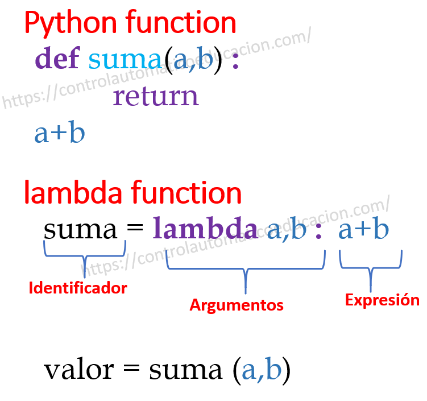
IS

El operador ìs se utiliza para comparar la identidad de dos objetos. Esto es, si los objetos a los que las variables hacen referencia son el mismo o no. Lo indica que esta comparación es más estricta que la realizada por el operador ==.



LAMBDA

Las expresiones lambda en Python son una forma corta de declarar funciones pequeñas y anónimas (no es necesario proporcionar un nombre para las funciones lambda). Las funciones Lambda se comportan como funciones normales declaradas con la palabra clave def.

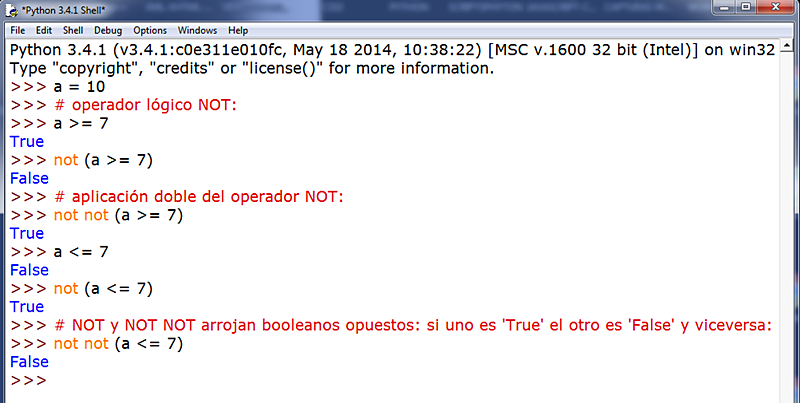


NON LOCAL

La función nonlocal indica que la variable dada sobre la que se trabaja no pertenece localmente a la función anidada dada. Sin embargo, esto no significa que la variable especificada con la palabra clave nonlocal sea una variable global.

NOT

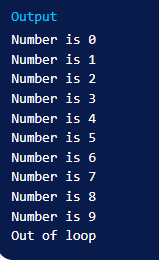
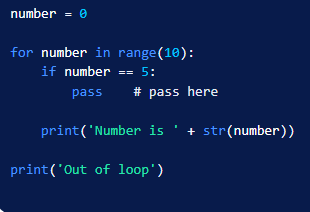
Si el valor es de tipo booleano, «NOT» actúa como operador de negación. Si el valor es False, «not valor» será «True», y la(s) declaración(es) del bloque «if» se ejecutará(n)



OR

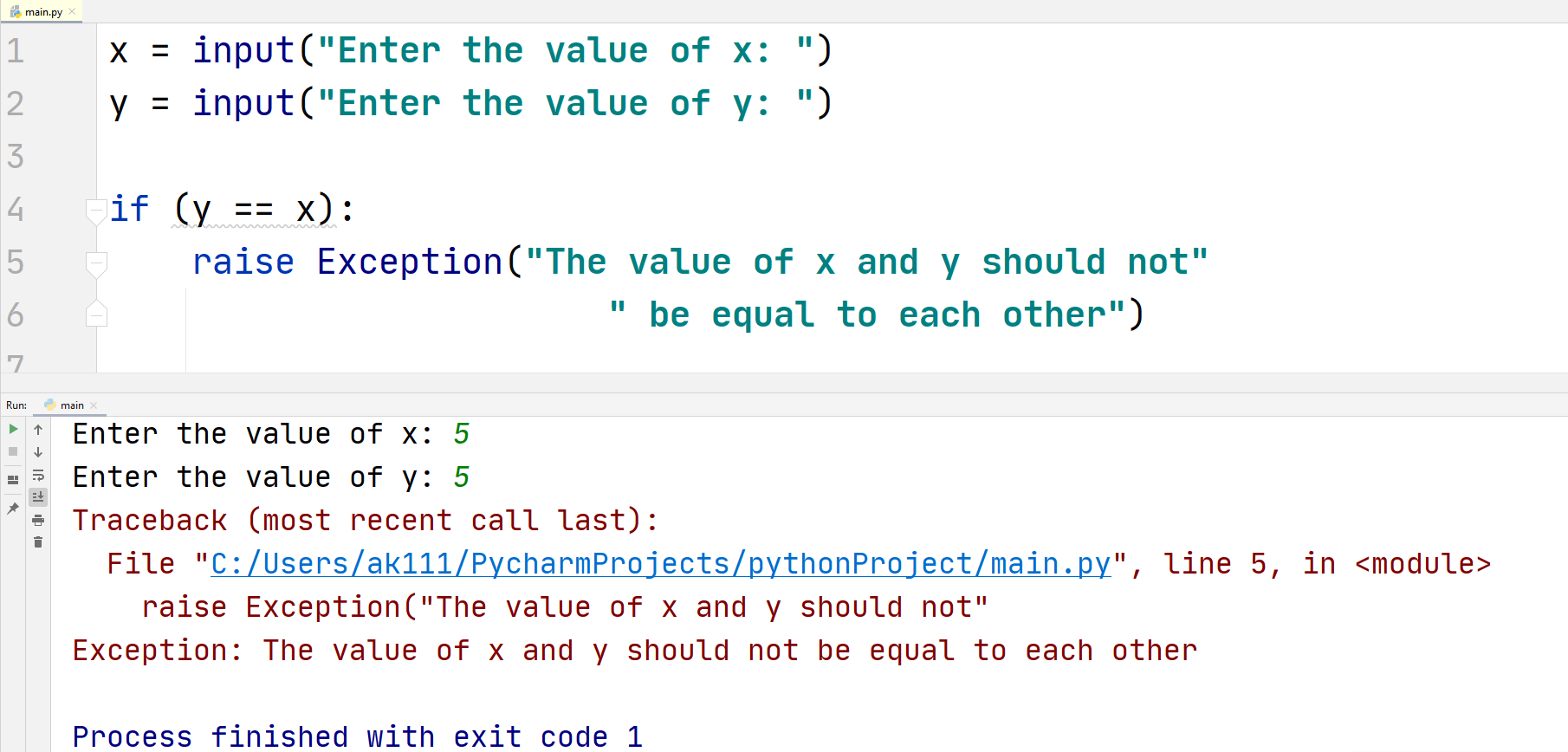
PASS

La instrucción pass permite manejar la condición sin que el bucle se vea afectado de ninguna manera; todo el código continuará leyéndose a menos que se produzca la instrucción break u otra instrucción. Igual que con las demás instrucciones, la instrucción pass se encuentra dentro del bloque de código abajo de la instrucción del bucle, normalmente después de una instrucción if condicional.



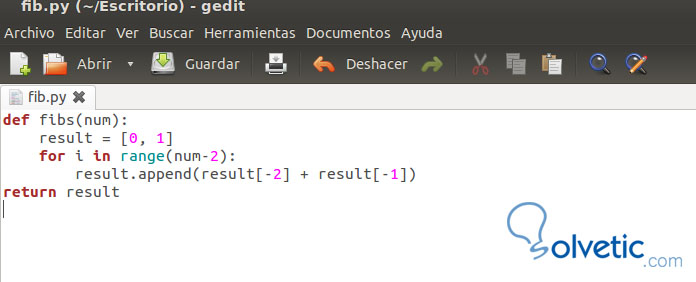
RAISE

Para levantar una excepción de cualquier tipo, utilizamos también la sentencia raise, pero indicándole el tipo de excepción que deseamos lanzar y pasando a la excepción los parámetros con información adicional que queramos brindar.



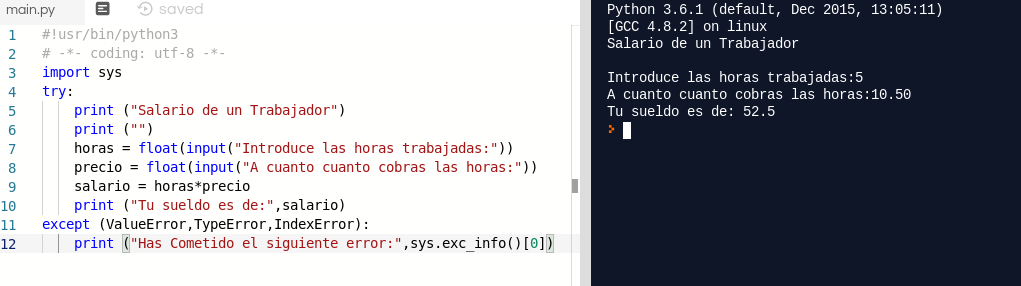
RETURN

Una función siempre devuelve un valor. La función utiliza la palabra clave return para devolver un valor; si no desea devolver ningún valor, se devolverá el valor predeterminado None . El nombre de la función se usa para llamar a la función, pasando los parámetros necesarios entre paréntesis.



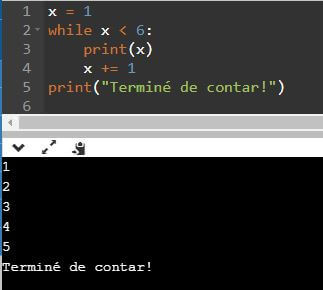
TRY

EL bloque try es el bloque con las sentencias que quieres ejecutar. Sin embargo, podrían llegar a haber errores de ejecución y el bloque se dejará de ejecutarse. El bloque except se ejecutará cuando el bloque try falle debido a un error.



WHILE

Python utiliza el bucle while de forma similar a otros lenguajes populares. El bucle while evalúa una condición y luego ejecuta un bloque de código si la condición es verdadera. El bloque de código se ejecuta repetidamente hasta que la condición llega ser o es falsa



WITH

La sentencia with se usa para ajustar la ejecución de un bloque con métodos definidos por un administrador de contexto

YIELD

Se le llama así a un objeto que es capaz de retornar sus miembros uno a la vez. Entre los iterables se encuentran estructuras de secuencia como son: strings

