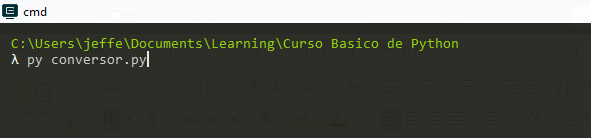
**Notas Python**

Ejecutar un archivo .py desde la consola

1. Nos ubicamos en el directorio donde se localiza el archivo
2. Escribimos en la consola py “espacio” “nombre del archivo”

****

**Funciones básicas Python conversión tipo de dato**

* **Input("")** para pedirle al usuario que introduzca datos.
* **int()** convierte string o decimal en número entero.
* **str()** para convertir números tanto decimales como enteros a strings.

**Operadores Lógicos**

* **and** para comparar si dos valores son verdaderos.
* **or** saber si al menos 1 de los dos valores es verdadero.
* **not** para invertir el valor booleano.

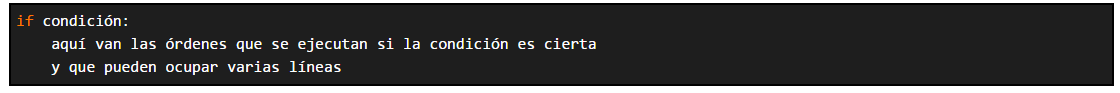
**Operadores de comparación**

* == Compara dos valores y te dice si son iguales o no.
* != Compara dos valores y te dice sin son diferentes o no.
* > Compara si es mayor que otro valor.
* < Compara si es menor que otro valor.
* >= igual o mayor que el valor a comparar.
* <= igual o menor que el valor a comparar.

**Condicionales en Python**

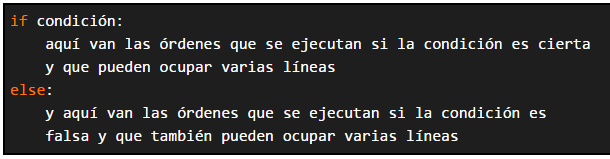
**if**

La sintaxis de la construcción if es la siguiente:



**If… else**

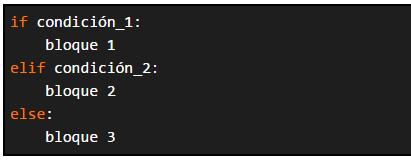
La estructura de control **if ... else ...** permite que un programa ejecute unas instrucciones cuando se cumple una condición y otras instrucciones cuando no se cumple esa condición. En inglés "if" significa "si" (condición) y "else" significa "si no". La orden en Python se escribe así:



**if ... elif…else ...**

La construcción **if ... else ...** se puede extender añadiendo la instrucción **elif:**

La estructura de control if ... elif ... else ... permite encadenar varias condiciones. elif es una contracción de else if (Se puede utilizar cuantas veces se requiera). La orden en Python se escribe así:

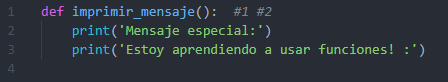


* Si se cumple la condición 1, se ejecuta el bloque 1
* Si no se cumple la condición 1 pero sí que se cumple la condición 2, se ejecuta el bloque 2
* Si no se cumplen ni la condición 1 ni la condición 2, se ejecuta el bloque 3.

Mas información condicionales <https://www.mclibre.org/consultar/python/lecciones/python-if-else.html>

**Funciones en Python**

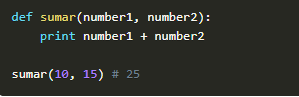
Una funció**n es una porción** o bloque de código reutilizable que se encarga de realizar una determinada tarea, se puede invocar las veces que sea necesario.



* #1 Para Declarar una funcion utilizo def nombre\_funcion
* #2 Nombre de las funciones en minuscala y palabras separadas por guion bajo

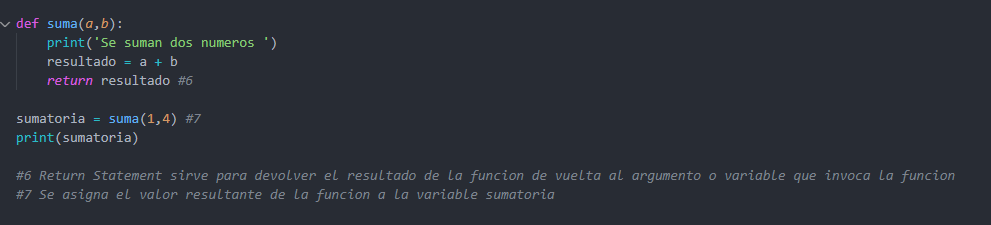
**Parámetros de una función en Python**

Un parámetro es un valor que la función espera recibir cuando sea invocada, el parámetro se declara entre los paréntesis de la función, y se le asigna valor cuando se invoca la función y dentro de esta se ponen los valores que se le va a asignar al parámetro



Mas información funciones en Python <https://devcode.la/tutoriales/funciones-en-python/>

**Return Statement** Sirve para devolver el resultado de la función de vuelta al argumento o variable que invoca la función



**Métodos Python**

Un método es una función que «pertenece a» un objeto. O es una función que es especial a un tipo de Dato en Particular

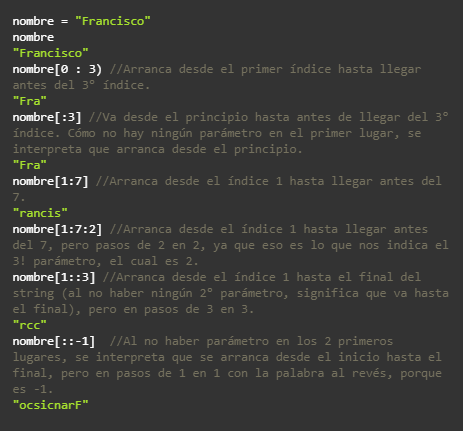
**Trabajando con Cadenas texto o Strings**

**Métodos usuales para trabajar con texto strings**

* variable.upper() = 'todos los caracteres en MAYÚSCULAS'
* variable.capitalize() = 'solo la primera en MAYÚSCULA'
* variable.lower() = 'todos los caracteres en minúscula'
* variable.strip() = 'eliminar espacios basura del string'
* variable.replace('caractera a cambiar', 'caracter por poner') = remplazar carácter

**Slices (Rebanadas) in python**

Los slices, traducidos al español como “rebanadas”, nos permiten dividir los caracteres de un string de múltiples formas. A continuación, un ejemplo de estos:

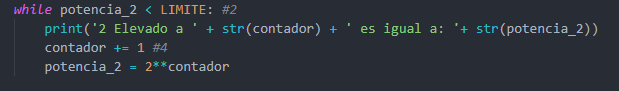


# Bucles en Python (Python Loops)

Es una función que ejecutara una serie de instrucciones x veces mientras la condición principal se cumpla

## While Loop

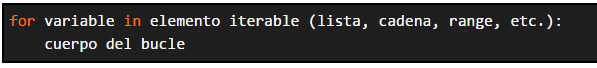
Loop While ejecutara una serie de instrucciones x veces mientras la condición principal se cumpla



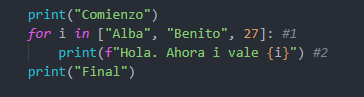
Mas informacion loops <https://www.w3schools.com/python/python_while_loops.asp>

## For Loop

En general, un bucle es una estructura de control que repite un bloque de instrucciones. Un bucle for es un bucle que repite el bloque de instrucciones un número prederminado de veces. El bloque de instrucciones que se repite se suele llamar cuerpo del bucle y cada repetición se suele llamar iteración. Sintaxis:



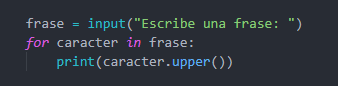
Un Ciclo FOR o FOR Loop ejecutara unas instrucciones (Cuerpo del bucle) un número determinado de veces por el elemento recorrible (Rango, Lista, Caracteres de una cadena)



Mas informacion <https://www.mclibre.org/consultar/python/lecciones/python-for.html>

### Recorrer una cadena de carecteres

Recorrer es ejecutar un ciclo por cada uno de los elemengtos de una Lista, un rango o un String

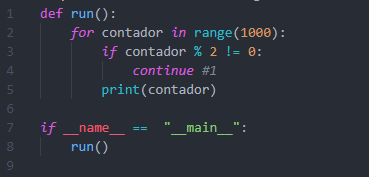


### Interrumpiendo ciclos (Continue y Breake statements)

Se puede modificar el ciclo de forma tal que ejecute ciertas vueltas y otras no las ejecute

**Continue**

Instrucción **continue** se usa para omitir el resto del código dentro de un bucle solo para la iteración actual. El bucle no termina, sino que continúa con la siguiente iteración.



**Built in functions in Python**

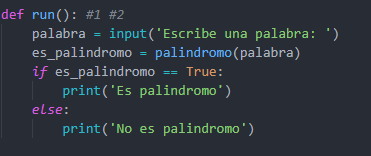
Funciones propias del lenguaje y que no tenemos que crearlas, siempre disponibles si tenemos instalado el lenguaje

* len(variable) == largo de una cadena de caracteres
* print(“Hola mundo”) == Imprimir Texto en la consola
* input(“Cuantos años tienes?”) == Solicitar un dato al user

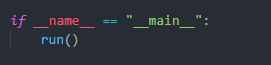
Mas info built in functions <https://docs.python.org/3/library/functions.html>

**Buenas prácticas en Python**

* [Dejar dos espacios (líneas de código) entre las funciones]
* [Crear una función principal] Usualmente se utiliza def run(): o def main():



**Entry Points (Punto de entrada)**



**# IMPORTANTE** Una FUNCION def function(): Siempre se debe crear encima del lugar en donde se va a invocar

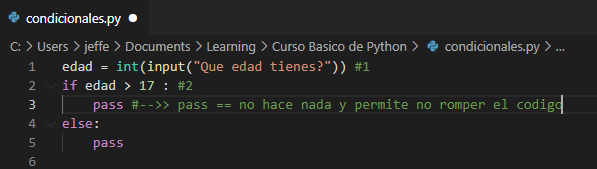
**Indentation Python (indentacion en Python**

Son los espacios (4 espacios) en el inicio de una línea de código después de una condicional o una función (Siempre después de los dos puntos **“:”** ), la indentation en Python es fundamental (obligatoria) mientras que en otros lenguajes es solo para mantener orden y la legibilidad (readability)

* Si tengo una línea de código bajo otra línea de código esta tendrá doble indentacion (y así sucesivamente con las líneas de código)

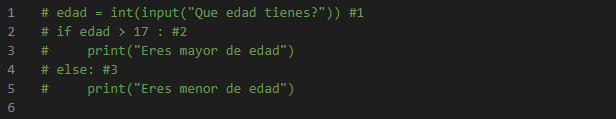
**Pass como no hacer nada en python**

Pass en Python (es un argumento que sirve para no hacer nada y poner otro código después si es necesario)



**Atajos Pytho**

* Cntr + / : Coloca un # antes de la línea o las líneas de código (sirve con una o varias líneas) y habilita o inhabilita líneas de código.



Puedo habilitar o inhabilitar un bloque de código seleccionándolo y con Ctrl + /