

## Como Cortar un ciclo (Definido e Indefinido)

**break:** Interrumpe la ejecución del ciclo actual. Puede utilizarse tanto para ciclos definidos como indefinidos.

**continue:** “Saltea” la ejecución del ciclo a la siguiente iteración. Puede utilizarse tanto para ciclos definidos como indefinidos.

**return [valor]:** Finaliza la ejecución de una función, y además corta la ejecución del ciclo actual, en caso de estar dentro del cuerpo del ciclo

## Funciones de Cadenas:

**cadena[i:j]:** Permite obtener un segmento de la cadena, desde la posición i inclusive, hasta la posición j exclusive.

**cadena.format(valores):** Al igual que f"...", genera una cadena interpolada, pero a partir de una cadena de formato separada de los valores a interpolar.

**cadena.isdigit():** Devuelve True si todos los caracteres de la cadena son dígitos, False en caso contrario.

**cadena.isalpha():** Devuelve True si todos los caracteres de la cadena son alfabéticos, False en caso contrario.

**cadena.isalnum():** Devuelve True si todos los caracteres de la cadena son alfanuméricos, False en caso contrario.

**cadena.capitalize():** Devuelve True si todos los caracteres de la cadena son alfanuméricos, False en caso contrario.

**isdecimal(string.isdecimal()):** El método isdecimal() devuelve True si todos los los caracteres son dígitos decimales (0-9).

**Zip():** La función toma iterables (pueden ser cero o más), los agrega en una tupla y loDevuelve.

**cadena.upper():** Devuelve una copia de la cadena convertida a mayúsculas.

**cadena.lower():** Devuelve una copia de la cadena convertida a minúsculas.

**replace(“caracter A reemplazar”, “CARACTER PARA REEMPLAZAR”):** reemplaza un caracter por otro

**split():** serpara la cadena en palabras

## Funciones de Listas :

**pop(list.pop(i)):** El método elimina el elemento en el índice dado de la lista y devuelve el elemento eliminado.

**append(list.append(elemento)):** Agrega un objeto (un número, una cadena o un otra lista) al final de mi\_lista

**remove(list.remove(elemento)):** Elimina el primer elemento coincidente (que se pasa como argumento) de la lista.

`insert(lista.insertar(i, elem))`: El método inserta un elemento en la lista en el índice especificado.

`reverse(list.reverse())`: El método invierte los elementos de la lista.

`extend(list1.extend(i))`: agrega todos los elementos de un iterable (lista, tupla, cadena, etc.) al final de la lista.

`copy(new_list = list.copy())`: Método devuelve una copia superficial de la lista.

`clear(lista.borrar())`: El método elimina todos los elementos de la lista.

`max()`: La función devuelve el elemento más grande en un iterable.

`min()`: La función devuelve el elemento más pequeño en un iterable.

`reversed(reversed(sequence_object))`: El método calcula el reverso de un objeto de secuencia dado y lo devuelve en forma de lista.

`map(mapa (función, iterable, ...))`: function aplica una función determinada a cada elemento de un iterable (lista, tupla, etc.) y devuelve un iterador.

`slice(slice(start, stop, step))`: La función devuelve un objeto de división que se usa para dividir cualquier secuencia (cadena, tupla, lista, rango o bytes).

`sorted(sorted(iterable, key=None, reverse=False))`: La función ordena los elementos de un iterable dado en un orden específico (ascendente o descendente) y lo devuelve como una lista(hace una nueva)

`sort(sorted(list, key=..., reverse=...))`: El método ordena los elementos de una lista en orden ascendente o descendente.(modifica la lista)

`filter(filter(function, iterable))`: La función extrae elementos de un iterable (lista,tupla, etc.) para los cuales una función devuelve True.

lambda: Función anónima

`index(list.index(element, start, end))`: Método devuelve el índice del elemento especificado en la lista.

## Diccionario:

`get(k, [])`: que recibe una clave *k* y un valor por omisión *v*, y devuelve el valor asociado a la clave *k*, en caso de existir, o el valor *v* en caso contrario.

for clave in diccionario:

Permite recorrer una a una todas las claves almacenadas en el diccionario.

`diccionario.keys()`: Devuelve una secuencia con todas las claves que se hayan ingresado al diccionario.

`diccionario.values()`: Devuelve una secuencia con todos los valores que se hayan ingresado al diccionario.

`diccionario.items()`: Devuelve una secuencia con tuplas de dos elementos, en las que el primer elemento es la clave y el segundo el valor.

`diccionario.pop(clave)`: Quita del diccionario la clave y su valor asociado, y devuelve el valor.

## Manejo de Archivos:

`with open("archivo.txt") as archivo:`

`#`

`# procesar el archivo`

`#`

`readline()`: lee una línea del archivo

`write()`: se utiliza para escribir datos en un archivo.

`rstrip("algo")`: Remueve el carácter asignado, elimina únicamente del final de la cadena.

`strip("algo")`: se usa para eliminar caracteres especificados (por defecto, espacios en blanco).

`csv.writer()`: se utiliza para escribir datos en un archivo CSV.

`Csv.reader()`: se utiliza para leer datos de un archivo CSV.

`archivo.write(cadena)`: Escribe sobre un archivo una cadena

`archivo.writelines(listas_de_cadenas)`: Escribe sobre un archivo una lista de cadenas

`os.path.isfile(ruta)`: Indica si la ruta existe y es un archivo.

`os.path.isdir(ruta)`: Indica si la ruta existe y es un directorio