**List Comprehension**

[doc pandas](https://docs.python.org/3/tutorial/datastructures.html#list-comprehensions)

[doc python](https://chrisalbon.com/code/python/data_wrangling/pandas_list_comprehension/)

**%% time** → nos dice el tiempo que tarda en ejecutarse la celda

DEFINICIÓN:

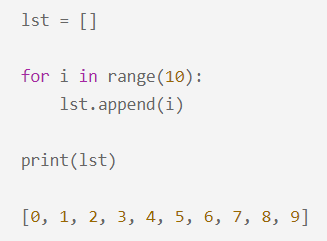
Es una forma de **simplificar la lógica** en **UNA sola línea** de código Python.

Queremos aprender a COMBINAR **almacenamiento** con **iteración**, con el objetivo de reducir tiempos de ejecución y almacenamiento.

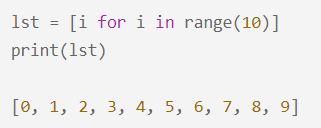
Además, al ser todo en una línea **te permite meterlo en una función** (en el return).

FUNCIONAMIENTO: 1 BUCLE FOR

1º elemento a insertar (.append(elemento a insertar)



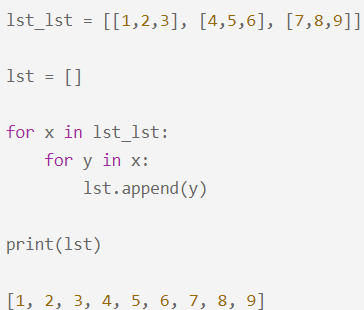
2º luego los bucles



→ Los **[]** hace que los resultados **se devuelvan en una lista**.

MÚLTIPLES BUCLES FOR (ejemplo: listas anidadas)

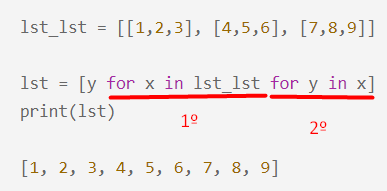
→ ponemos los **bucles en orden**



1º elemento a insertar (.append(elemento a insertar)

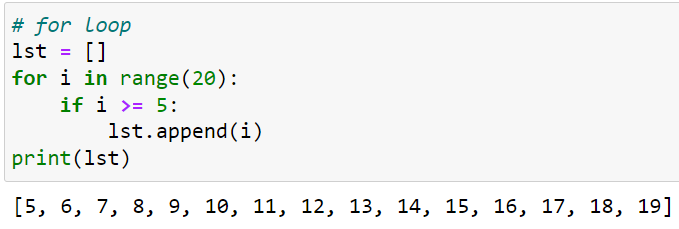
2º bucle 1

3º bucle 2



AGREGAR DECLARACIONES CONDICIONALES:

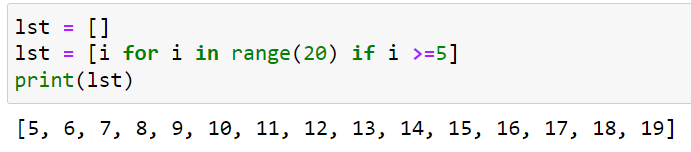
→primero **bucles** después **IFS**



1º elemento a insertar (.append(elemento a insertar)

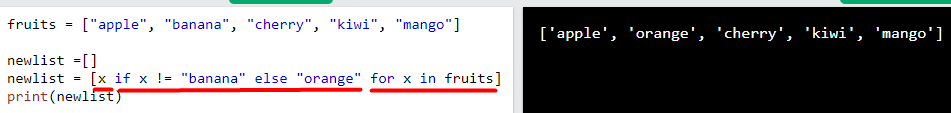
2º los bucles

**3º los condicionales**



\* Si aparte del If también hay un else o un elif:

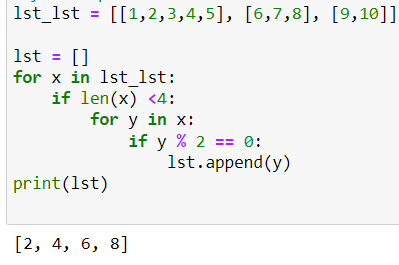
→ pondremos **primero IF y, ELSE y ELIF,** después **bucles**



MÚLTIPLES BUCLES Y CONDICIONANTES (múltiples iteracciones y condicionales)

→ ponemos los **bucles** y los **IFS en orden**

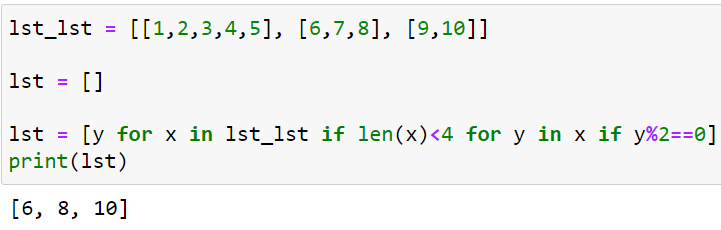
*Ejemplo: queremos de un listado anidado obtener los pares siempre que los elementos del primer listado tengan menos de 4 elementos*



1º elemento a insertar (.append(elemento a insertar)

2º bucle 1 + if

3º bucle 2 + if



**USOS PRÁCTICOS**

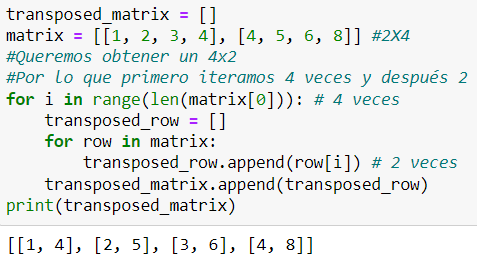
1. INVERTIR UNA MATRIZ: - (múltiples bucles con múltiples append)

**TENEMOS**:

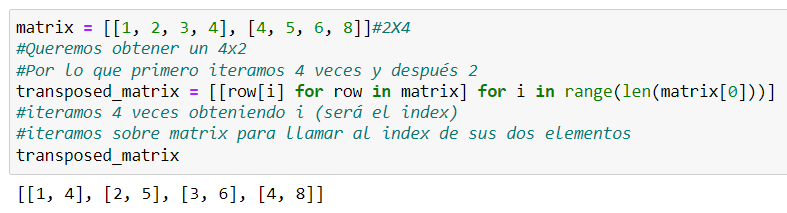
* matrix = [[1, 2, 3, 4], [4, 5, 6, 8]]
* queremos INVERTIRLA

**HAREMOS**:

* Ejemplo **tradicional:**



* Ejemplo **Lists Comprehension: Como queremo invertir, los bucle for van al revés**

****

1. LEER VARIOS ARCHIVOS(múltiples iteracciones y condicionales)

**TENEMOS**:

* una carpeta con múltiples archivos (algunos csv y algunos word)
* los csv tienen el mismo formato de columnas

**HAREMOS**:

1. Obtener todos los ficheros csv

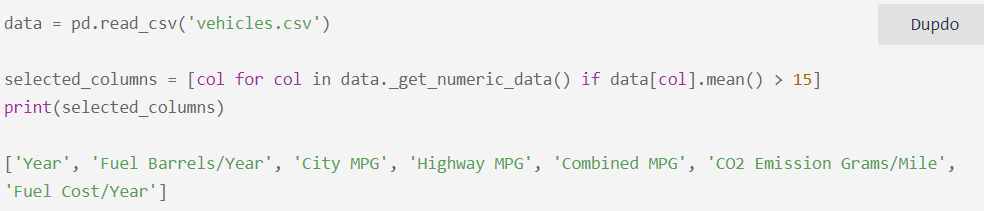


1. Que Pandas combine todos los archivos solo en uno.



1. SELECCIONAR COLUMNAS DE UN CSV QUE CUMPLAN UN CRITERIO

Otro caso de uso sería seleccionar columnas de marcos de datos en función de una condición que tengan en común. El siguiente ejemplo lee nuestro archivo de datos de vehículos, usa el \_get\_numeric\_data método para recuperar todas las columnas numéricas y luego usa una lista de comprensión para devolver solo las columnas que tienen una media mayor que 15.



**OTRO CASO:**

