CHECKPOINT 4 - PREGUNTAS TEÓRICAS

1. ¿Cuál es la diferencia entre una lista y una tupla en Python?

La diferencia principal esta en que las listas son mutables y las tuplas son inmutables, las listas se cierran con corchetes [], las tuplas se cierran con paréntesis (). Dependiendo con los datos que queramos trabajar, si son modificadas o no, elegiremos una u otra. Si los datos van a ser mutables obviamente utilizaremos lista y si no queremos que sean modificadas, utilizaremos las tuplas.

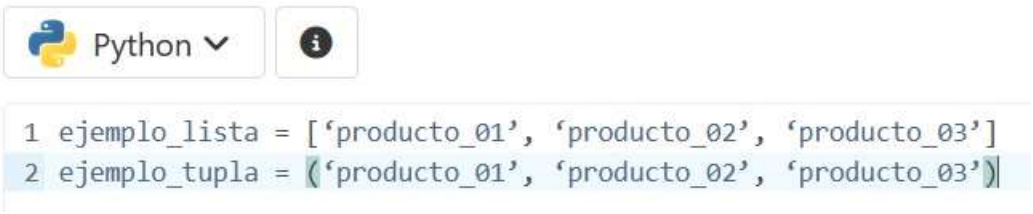
La sintaxis de una lista es la siguiente:

ejemplo_lista = [‘producto_01’, ‘producto_02’, ‘producto_03’]

La sintaxis de una tupla es la siguiente:

ejemplo_tupla = (‘producto_01’, ‘producto_02’, ‘producto_03’)

Captura:



2. ¿Cuál es el orden de las operaciones?

En Python, el orden de las operaciones se aplica igual que en las matemáticas convencionales de toda la vida, aunque no tenga la menor idea de cual es su verdadero orden de memoria, siempre podemos consultar en su documentación para asegurarnos que la operación funcione correctamente.

Orden de prioridades de arriba abajo según la documentación:

Operador	Descripción
`(expressions...)`, `[expressions...]`, `{key: value...}`, `{expressions...}`	Expresión de enlace o entre paréntesis, despliegues de lista, diccionario y conjunto
`x[index]`, `x[index:index]`, `x(arguments...)`, `x.attribute`	Subscripción, segmentación, invocación, referencia de atributo
`await x`	Expresión await
`**`	Exponenciación [5]
`+x`, `-x`, `~x`	NOT positivo, negativo, bit a bit
`*`, `@`, `/`, `//`, `%`	Multiplicación, multiplicación de matrices, división, división entera a la baja, resto (módulo) [6]
`+`, `-`	Adición y sustracción

3. ¿Qué es un diccionario Python?

Un diccionario en Python es un cajón donde almacenamos elementos como clave y su valor. Podemos encontrarnos con diccionarios con la sintaxis de una lista para asegurarnos de que no se repitan claves, pero no es lo más común.

Sintaxis de un diccionario común:

```
diccionario_comun = {
        'producto_01': 'precio_01',
        'producto_02': 'precio_02',
        'producto_03': 'precio_03',
}
```

Sintaxis de un diccionario como lista:

```
diccionario_lista = {'pepito', 'gorgorito', 'pepito', 'san pepito'}
```

Captura:

4. ¿Cuál es la diferencia entre el método ordenado y la función de ordenación?

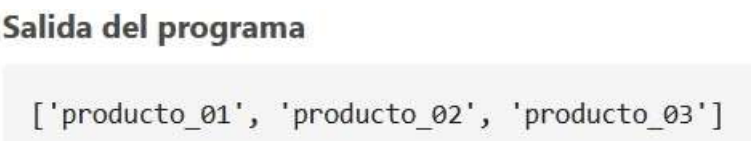
La diferencia principal esta en que el método de ordenado (sort) ordena la lista en la misma variable y la función de ordenación (sorted) ordena en una nueva variable creando una copia de la lista.

sort:

```
lista = ['producto_01', 'producto_02', 'producto_03']

lista.sort()

print(lista)
```



sorted:

```
lista = ['producto_01', 'producto_02', 'producto_03']

lista_copia = lista.sorted()
```

5. ¿Qué es un operador de reasignación?

Un operador de reasignación actualiza el valor anterior por el actual reasignando el valor con un segundo operador delante del operador de asignación.

Podríamos entender como operadores de reasignación los siguientes de la lista (operadores compuestos):

+=

-=

*=

/=

%=

//=

**=

Etc...