Cheat Sheet Python

Listas

Definir una lista:
mi_lista = [1, 2, 3, "hola", True, 3.14]

Solicitar datos a la lista:
mi_lista[2]
Devuelve el número 3.

Modificar una lista:
mi_lista[2]=9
Esto hace que el tercer elemento sea un 9.

Recorrer una lista:
```
for elemento in mi_lista:
        print(elemento)
```

Métodos relevantes:

append(): Agrega un elemento al final de la lista.
mi_lista = [1, 2, 3]
mi_lista.append(4)
Resultado: [1, 2, 3, 4]

extend(): Agrega múltiples elementos de otra lista.
mi_lista.extend([5, 6])
Resultado: [1, 2, 3, 4, 5, 6]

insert(): Inserta un elemento en una posición específica. En este ejemplo el 1 es la posición
mi_lista.insert(1, "nuevo")
Resultado: [1, "nuevo", 2, 3, 4, 5, 6]

remove(): Elimina el primer elemento coincidente en la lista.
mi_lista.remove("nuevo")
Resultado: [1, 2, 3, 4, 5, 6]

pop(): Elimina y devuelve el elemento en una posición específica (por defecto, el último).
elemento = mi_lista.pop()
Resultado: elemento = 6; mi_lista = [1, 2, 3, 4, 5]

sort(): Ordena los elementos de la lista (funciona solo si son comparables).
mi_lista.sort()
Si `mi_lista` es [3, 1, 4, 2], el resultado será [1, 2, 3, 4]
Hay varios **tipos** de **sort**():

Orden descendente:
mi_lista.sort(reverse=True)
print(mi_lista)
Resultado: [9, 6, 5, 5, 2, 1]

Mediante una clave:
mi_lista = ['perro', 'gato', 'elefante', 'ratón']
mi_lista.sort(key=len) # Ordenar por longitud de las palabras print(mi_lista)
Resultado: ['gato', 'perro', 'ratón', 'elefante']

count(): Cuenta cuántas veces aparece un elemento específico en la tupla.
mi_tupla = (1, 2, 2, 3, 4)
veces = mi_tupla.count(2)
Resultado: veces = 2

index(): Devuelve el índice de la primera aparición de un valor en la tupla.
indice = mi_tupla.index(3)
Resultado: indice = 3

Tuplas

Definir una tupla
```
mi_tupla = (1, 2, 3, "hola", True, 3.14)
mi_tupla[0]
```
Devuelve el valor 1.
Característica: Inmutabilidad (No modificable)

Recorrer una tupla:
```
for elemento in mi_tupla:
        print(elemento)
```

Ejemplo:

```
mi_tupla = (1, 2, 3)
mi_lista = list(mi_tupla) # Convertir a lista
mi_tupla = tuple(mi_lista) # Convertir de nuevo a tupla
```

Diccionarios

Estructura:

```
persona= {
        "nombre": "Alice",
        "edad": 30,
         "ciudad": "Madrid",
        "es_estudiante": False
}
```

Acceso a datos del diccionario:
```
persona["nombre"]
```
Devuelve Alice

Modificar un diccionario
```
mi_diccionario["edad"] = 31
```

Recorrer un diccionario:

Solo las claves
```
for clave in persona:
        print(clave)
```

Solo los valores:
```
for valor in persona.values():
        print(valor)
```

Recorrer claves y valores
```
for clave, valor in persona.items():  #Items devuelve los pares de clave - valor del diccionario
        print(f"{clave}: {valor}")
```

Métodos relevantes:

get(): **Obtiene** el **valor asociado** con una **clave**, o un valor por defecto si la clave no existe.
```
mi_diccionario = {"nombre": "Alice", "edad": 30}
edad = mi_diccionario.get("edad")
```
Resultado: edad = 30

keys(): **Devuelve** una **lista** de todas las claves en el diccionario.
```
claves = mi_diccionario.keys()
```
Resultado: claves = dict_keys(["nombre", "edad"])

values(): **Devuelve** una **lista** de todos los valores en el diccionario.
```
valores = mi_diccionario.values()
```
Resultado: valores = dict_values(["Alice", 30])

items(): **Devuelve** una **lista** de **pares clave-valor** en forma de tuplas.
```
elementos = mi_diccionario.items()
```
Resultado: elementos = dict_items([("nombre", "Alice"), ("edad", 30)])

update(): Actualiza el diccionario con otro diccionario o con pares clave-valor.
```
mi_diccionario.update({"ciudad": "Madrid", "edad": 31})
```
Resultado: mi_diccionario = {"nombre": "Alice", "edad": 31, "ciudad": "Madrid"}

pop(): Elimina un elemento con una clave específica y devuelve su valor.
```
nombre = mi_diccionario.pop("nombre",)
```
Resultado: nombre = "Alice"; mi_diccionario = {"edad": 31, "ciudad": "Madrid"}

clear(): Elimina todos los elementos del diccionario.
```
mi_diccionario.clear()
```
Resultado: mi_diccionario = {}

Lectura y escritura de ficheros

Leer Ficheros:

Acceso al fichero → open(“archivo.txt” , **“**modo”**)**

Modos → “r” Lectura “w” Escritura(si ya existe el archivo lo reescribe) “a” Anexa el texto que escribes

1. **Lee el contenido de un archivo y lo muestra en pantalla.**

```
fichero = open(“estudiantes.txt”,”r”)
for linea in fichero:
        nombre, edad = linea.strip().split(“-”)
        estudiantes[nombre] = edad
fichero.close()
```

2. **Escribe una frase en un archivo.**

```
fichero = open(“estudiantes.txt”,”w”)

for estudiante in estudiantes:
        fichero.write(estudiante + “ - “ + str(estudiantes[estudiante]))
fichero.close()
```

strip() → quita los espacios en blanco al principio y al final
split(“parámetro”) →Divide por el parámetro designado

Modo de apertura	Descripción	Acción
`'w'`	Escritura	Si el fichero no existe lo crea. Si existe, borra su contenido
`'r'`	Lectura	Si existe fichero: lo abre. Si no existe: excepción `FileNotFoundError`
`'a'`	Añadir	Si fichero no existe, lo crea para escritura. Si existe, añade al final
`'w+'`	Actualizar	Escritura/ lectura. Si el fichero no existe lo crea. Si existe: borra
`'r+'`	Actualizar	Lectura/Escritura. Si no existe: excepción `FileNotFoundError`
`'a+'`	Añadir	Escritura/lectura. Si existe, añade al final.
`'b'`	Binario	Abre en binario. Combinadas con otras banderas: establece modo
`'x'`	Creación	Abre exclusivamente para crear fichero. Si ya existe, falla