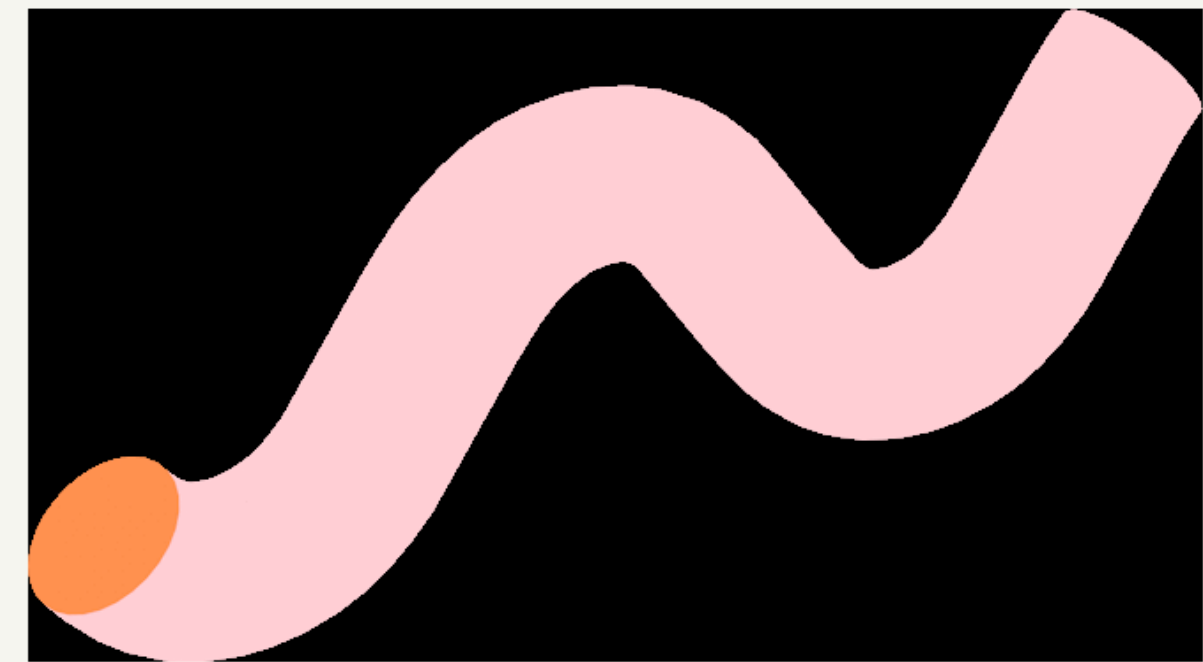


FILE I/O



El módulo para leer y escribir ficheros

**open(string,
[mode])**



file object



**Un objeto que
conlleva los
métodos read y
write, que nos
permite trabajar
con ficheros
planos**

**[https://docs.python.org/3/library/f
unctions.html#open](https://docs.python.org/3/library/functions.html#open)**

**Raw binary
files, buffer
binary y plain
texts**

with compound

- `open()`
- `close()`
- `readline()`
- `readlines()`
- `seek(index)`
- `write()`
- `writelines()`

```
with open("file.txt") as file:  
    __enter__()  
    (trabajamos con el fichero)  
    __exit__()
```

CSV

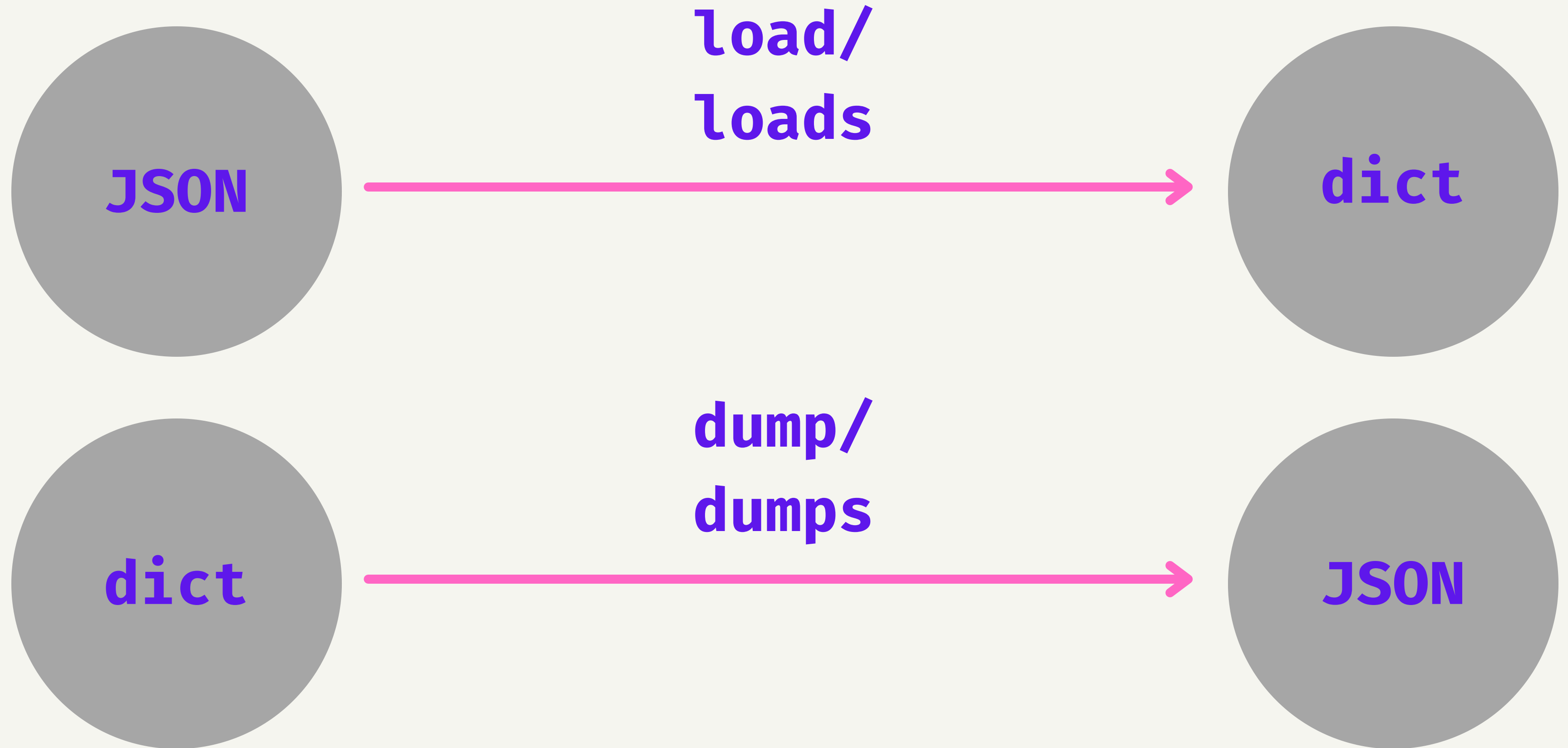
```
1.import csv
2.with open("*.csv", [mode], [...]) as file:
3.    csv_reader/writer = csv.reader(file, [...])
4.    csv_reader. 1.writerow(...)
                  2.writerows(...)
```

1.encoding=?
2.errors=?
3.newline=?

1.delimiter = ?
2.dialect = ?

writerow(...) → Aceptara listas salvo que
el dialecto especificado
haya sido dict_writer

Excel,
unix_dialect,
dict_reader ...



¿Cuándo debemos agregar la "s"?

**Cuando estemos trabajando con strings y
NO con archivos**


```
1.import json
2.with open("*.json", [mode], [...]) as file:
3.    json.dump(validJSON, file)
4.    json.load(validJSON)
```

```
1.import json
2.dict_ej_1 = {"a": False, "b": None}
3.json.dumps(dict_ej_1) # {"a": false, "b": null}
4.dict_ej_2 = {"a": String, "b": Array}
5.json.loads(dict_ej_2) # {"a": str, "b": list}
```