



**Persistencia de datos**



# Manipulación de documentos JSON

---



## Índice

- El módulo json
- Obtener un JSON a partir de una cadena
- Leer un documento JSON de un fichero
- Transformación de JSON a texto
- Envío de un documento JSON a un fichero



## El módulo json

- Deberá ser importado al principio del programa
- Proporciona métodos para manipular documentos JSON desde una aplicación Python, sin tener que acceder como fichero de texto
- Entre sus principales métodos están:
  - `loads(documento)`. Carga un documento JSON que se encuentra en memoria
  - `load(fichero)`. Carga un documento JSON a partir de un fichero
  - `dumps(json)`. Convierte un documento JSON en una cadena
  - `dump(documento,fichero)`. Graba el documento JSON en un fichero



## Obtener un JSON a partir de una cadena

- Dada una cadena de caracteres con un documento JSON, el método `loads` permite cargar en memoria un documento a partir de dicha cadena.
- A partir de ahí, es tratado como un diccionario (objeto JSON) o lista de diccionarios (array de objetos JSON)
- El siguiente ejemplo muestra todos los valores de la propiedad "nombre" de los objetos JSON del array:

```
myjson='[{"nombre":"lucas","telefono":33333}, {"nombre":"gema","telefono":1111}]'
conver=json.loads(myjson)
for n in conver:
    print(n["nombre"])
```



## Leer documento JSON de un fichero

- Si el documento JSON está en un fichero, se puede recuperar directamente como array u objeto JSON mediante el método `load()`
- El siguiente ejemplo muestra todos los títulos de los libros almacenados en un archivo `datos.json`:

```
with open('datos.json') as file:
    datos = json.load(file)
    for l in datos:
        print(l["titulo"])
```

- El documento `datos.json` sería:

```
[{"titulo": "Java 10", "autor": "nadie", "precio": 20}, {"titulo": "python", "autor": "Martin", "precio": 30}]
```



## Transformación de JSON a texto

- Utilizando el método `dumps()` de `json` podemos transformar un documento JSON a una cadena de texto:

```
with open('datos.json') as file:
    datos = json.load(file)
cad=json.dumps(datos) #El documento leído es transformado en un texto
print(cad)
```

- Mediate el atributo opcional *separators*, podemos indicar que separador de clave/valor utilizar, así como separador de parejas:

```
myjson=' [{"nombre":"lucas","telefono":33333}, {"nombre":"gema","telefono":1111}] '
conver=json.loads(myjson)
print(json.dumps(conver,separators=("-",".")))
# [{"nombre"."lucas"-"telefono".33333}-{"nombre"."gema"-"telefono".1111}]
```



## Envío de JSON a fichero

- Mediante el método `dump()` de `json` se puede enviar el contenido de un documento JSON a un fichero:

```
#lectura de JSON desde memoria
myjson='[{"nombre":"lucas","telefono":33333}, {"nombre":"gema","telefono":1111}]'
conver=json.loads(myjson)
#envía documento a fichero
with open("personas.json", "w") as write_file:
    json.dump(conver, write_file)
```