

Clase N° 9b

JSON

© Lic. Ricardo Thompson

Concepto

- **JSON** (*JavaScript Object Notation*) es un formato basado en texto destinado al almacenamiento e intercambio de información entre sistemas heterogéneos, similar al estándar XML.

© Lic. Ricardo Thompson

Concepto

- En Python suelen estar asociados a diccionarios, aunque también pueden utilizarse otros tipos de dato.
- Todas las funciones para su manipulación están disponibles en el módulo ***json***.

© Lic. Ricardo Thompson

Concepto

- El proceso de convertir datos a formato JSON se denomina ***serialización***, y consiste en transformar un objeto almacenado en memoria a una secuencia de bytes que puedan ser transferidos a otros dispositivos o enviados a través de una red.

© Lic. Ricardo Thompson

Concepto

- El proceso contrario se denomina ***deserialización*** y consiste en decodificar un flujo JSON para convertirlo en datos utilizables desde Python.

© Lic. Ricardo Thompson

Codificación

- Para convertir un objeto Python a formato JSON se utiliza la función `dumps()`:

<variable> = json.dumps(<objeto>)

© Lic. Ricardo Thompson

Función dumps()

```
import json

alumnos = { }
alumnos[1000000] = {
    "nombre": "Martin Fernandez",
    "promedio": 7.65
}
alumnos[1000001] = {
    "nombre": "Melina Gallo",
    "promedio": 8.91
}

datos = json.dumps(alumnos)
print(datos)
```

© Lic. Ricardo Thompson

Función dumps()

- Esto convierte el diccionario a formato JSON y lo almacena en una variable de tipo string, la que luego podrá ser resguardada en un archivo o enviada por cualquier medio disponible:

```
{"1000000": {"nombre": "Martin Fernandez",
"promedio": 7.65}, "1000001": {"nombre": "Melina Gallo", "promedio": 8.91}}
```

© Lic. Ricardo Thompson

Función dump()

- También existe la función **dump(<objeto>, <archivo>)**, que es similar a dumps() pero no devuelve un string sino que graba el objeto JSON en un archivo.

© Lic. Ricardo Thompson

Función dump()

```
[...]
```

```
try:
```

```
    arch = open("archivo.json", "wt")
```

```
    json.dump(alumnos, arch)
```

```
except (FileNotFoundError, OSError) as error:
```

```
    print("ERROR DE GRABACION: ", error)
```

```
finally:
```

```
    try:
```

```
        arch.close()
```

```
    except NameError:
```

```
        pass
```

© Lic. Ricardo Thompson

Importante

- **Al convertir un diccionario a formato JSON las claves del mismo son automáticamente transformadas en cadenas de caracteres.**

© Lic. Ricardo Thompson

Decodificación

- **La conversión de un objeto JSON (almacenado en un string) a una variable Python se realiza con la función loads():**

<variable> = json.loads(<objeto>)

alumnos = json.loads(datos)

© Lic. Ricardo Thompson

Función load()

- Para convertir un archivo JSON a una variable Python existe la función load():

<variable> = json.load(<archivo>)

© Lic. Ricardo Thompson

Función load()

import json

try:

arch = open("archivo.json", "rt")

alumnos = json.load(arch)

except (FileNotFoundError, OSError) as error:

print("ERROR DE LECTURA: ", error)

finally:

try:

arch.close()

except NameError:

pass

print(alumnos)

© Lic. Ricardo Thompson

¡A practicar!

© Lic. Ricardo Thompson