JSON PARSING IN PYTHON

By:
- Niva Chitrakar
- Satish Karla

JSON

- Java Script Object Notation

- Used for sending, receiving and storing the data from a same or different systems in a network

- It is a popular data format used for representing structured data.

- It enables faster accessibility, memory optimization, shorter and simpler in nature, and does not contain complicated syntax & tags.

- Python has built-in package called json

- In Python, JSON exists as a string. For example:

p='{"name":"Bob", "language":["Python","Java"]}'

Rules of writing JSON Syntax

- The data is always in key/value pairs.

- Every data object is separated by a comma.

- The objects are held within the curly braces (the flower bracket like this '{}')

- The arrays are held in square brackets.

Syntax used in JSON

- Import JSON Module:
    import json

- Parse JSON in Python
    - Function: json.loads()
    - Syntax: json.loads(json_string)

- Python read JSON file:
    - Functions: open(), json.loads()
    - Syntax: json.loads(file_object)

- Python convert to JSON string
    - Function: json.dumps()
    - Syntax: json.dumps(dict, indent)

- Writing JSON to a file
    - Function: json.dumps()
    - Syntax: json.dumps(dict, file_pointer)

Serialization and Deserialization

- Serialization:

  The process of encoding JSON is usually called serialization. This term refers to the transformation of data into a series of bytes (hence serial) to be stored or transmitted across a network.

- Deserialization:

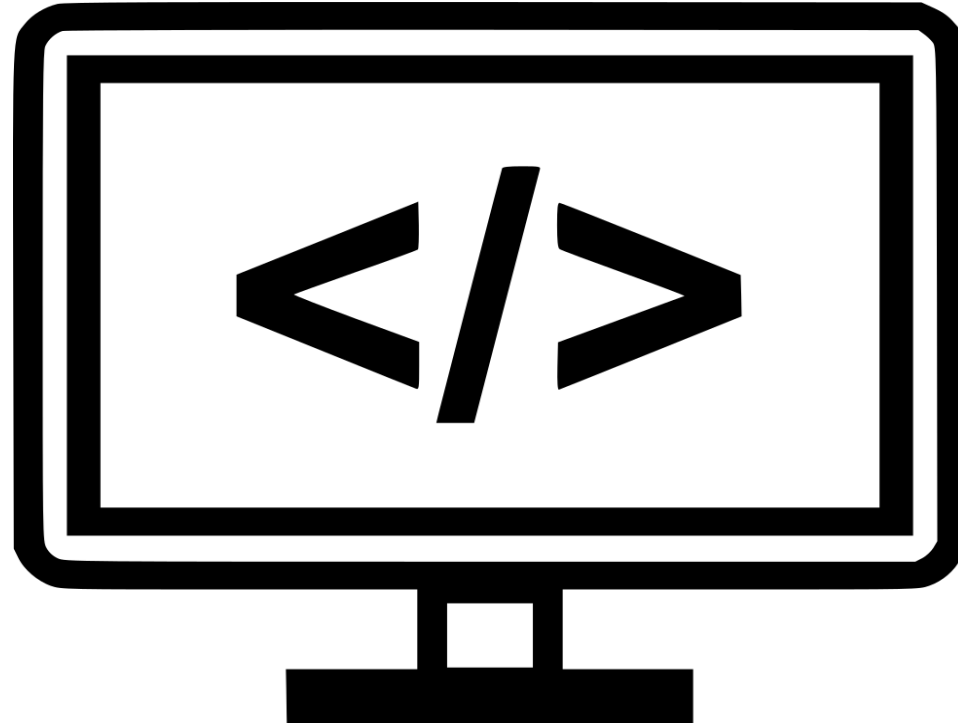 Deserialization is the reciprocal process of decoding data that has been stored or delivered in the JSON standard.

Serializing JSON

Python	JSON
dict	object
list, tuple	array
str	string
int, long, float	number
True	true
False	false
None	null

Deserializing JSON

JSON	Python
object	dict
array	list
string	str
number (int)	int
number (real)	float
true	True
false	False
null	None