**Features of JSON package**

JSON python module allows us encoding and decoding our data structures in convenient format.

**Basic Usage**:

* **json.dump**(obj, fp, skipkeys=False, ensure\_ascii=True, check\_circular=True, allow\_nan=True, cls=None, indent=None, separators=None, default=None, sort\_keys=False, \*\*kw) - Serialize obj as a JSON formatted stream to fp (a .write()-supporting [file-like object](https://docs.python.org/3.5/glossary.html#term-file-like-object)) using this conversion table.
* **json.dumps**(obj, skipkeys=False, ensure\_ascii=True, check\_circular=True, allow\_nan=True, cls=None, indent=None, separators=None, default=None, sort\_keys=False, \*\*kw) - Serialize obj to a JSON formatted [str](https://docs.python.org/3.5/library/stdtypes.html#str) using this [conversion table](https://docs.python.org/3.5/library/json.html#py-to-json-table). The arguments have the same meaning as in [dump()](https://docs.python.org/3.5/library/json.html#json.dump).
* **json.load**(*fp*, *cls=None*, *object\_hook=None*, *parse\_float=None*, *parse\_int=None*, *parse\_constant=None*, *object\_pairs\_hook=None*, *\*\*kw*) - Deserialize *fp* (a .read()-seupporting [file-like object](https://docs.python.org/3.5/glossary.html#term-file-like-object) containing a JSON document) to a Python object using this [conversion table](https://docs.python.org/3.5/library/json.html#json-to-py-table).
* **json.loads**(*s*, *encoding=None*, *cls=None*, *object\_hook=None*, *parse\_float=None*, *parse\_int=None*, *parse\_constant=None*, *object\_pairs\_hook=None*, *\*\*kw*) - Deserialize *s* (a [str](https://docs.python.org/3.5/library/stdtypes.html#str) instance containing a JSON document) to a Python object using this [conversion table](https://docs.python.org/3.5/library/json.html#json-to-py-table).

**Encoding and Decoding:**

* ***class*json.JSONDecoder**(*object\_hook=None*, *parse\_float=None*, *parse\_int=None*, *parse\_constant=None*, *strict=True*, *object\_pairs\_hook=None*) - Simple JSON decoder.

Performs the following translations in decoding by default:

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

* ***class*json.JSONEncoder**(*skipkeys=False*, *ensure\_ascii=True*, *check\_circular=True*, *allow\_nan=True*, *sort\_keys=False*, *indent=None*, *separators=None*, *default=None*) - Extensible JSON encoder for Python data structures.

Supports the following objects and types by default:

| **Python** | **JSON** |
| --- | --- |
| dict | object |
| list, tuple | array |
| str | string |
| int, float, int- & float-derived Enums | number |
| True | true |
| False | false |
| None | null |

**Exceptions:**

* ***exception*json.JSONDecodeError**(*msg*, *doc*, *pos*, *end=None*) - Subclass of [ValueError](https://docs.python.org/3.5/library/exceptions.html" \l "ValueError" \o "ValueError) with the following additional attributes:

**msg -** The unformatted error message.

**doc -** The JSON document being parsed.

**pos -** The start index of *doc* where parsing failed.

**lineno -** The line corresponding to *pos*.

**colno -** The column corresponding to *pos*.