

WHAT IS JSON?

JSON (JavaScript Object Notation) is a lightweight, text-based, language-independent data interchange format. Standardized as RFC 8259 (IETF) and ECMA-404. Created by Douglas Crockford in 2002.

STANDARDS & SPECS

- RFC 8259** Current IETF internet standard
- ECMA-404** Grammar-only specification
- ISO/IEC 21778** International standard (2017)
- MIME type** application/json
- File ext.** .json
- Encoding** UTF-8 (required for network)

QUICK EXAMPLE

```

{
  "model": "gpt-4o",
  "version": 2024,
  "active": true,
  "temp": 0.7,
  "prompt": null,
  "caps": ["text", "vision"]
}

```

THE 6 JSON DATA TYPES

String Unicode text in double quotes
"Hello, World"

Number Integer or float — one type
42 / -7.5 / 1e10

Boolean Must be lowercase
true / false

Null Absence of value — lowercase
null

Object Unordered key-value pairs
{ "key": value }

Array Ordered sequence of values
[1, 2, 3]

WHAT JSON DOES NOT SUPPORT

- No comments** // or /* */ are invalid
- No trailing commas** {"a":1,} is invalid
- No single quotes** Keys and strings need double quotes
- No undefined** Only null represents absence
- No NaN/Infinity** Use null or a string instead
- No Date type** Use ISO 8601 string: "2025-03-21"
- No binary data** Base64-encode binary as a string

Keys must be double-quoted strings

```
"key": value
```

No trailing commas anywhere

```
{*a*:1} not {"a":1,}
```

Strings: double quotes only

```
"text" not 'text'
```

Numbers: no leading zeros

```
42 not 042
```

No NaN, Infinity allowed

```
Use null instead
```

Whitespace ignored between tokens

```
Indent freely
```

UTF-8 required for network JSON

```
Per RFC 8259 §8.1
```

PYTHON QUICK REFERENCE

```

import json

# Parse string → dict
data = json.loads(s)

# Parse file → dict
data = json.load(f)

# Dict → string
s = json.dumps(data, indent=2)

# Dict → file
json.dump(data, f, indent=2)

```

JSON IN AI & LLMs

API Requests/Responses

All LLM APIs (OpenAI, Anthropic, Google)

Model Config

config.json in every HuggingFace model

JSON Mode

Forces structured JSON output from GPT/Claude

Pydantic Validation

Validate AI output against a schema

Tool / Function Calls

LLM agent tool definitions in JSON Schema