JSON Cheat Sheet

rjl.io/json-tools/

Structure

```
{
  "name": "Ada",
  "age": 36,
  "admin": true,
  "tags": ["dev", "ops"],
  "address": null
}
```

A document is **one value**. Objects hold `"key": value` pairs in `{ }`; arrays hold ordered values in `[ ]`. Both nest to any depth. Standard: RFC 8259. MIME type `application/json`, encoding UTF-8.

The six types

Type	Example	Rule
object	`{"id": 7}`	Keys are double-quoted strings, should be unique
array	`[1, "a", null]`	Ordered, mixed types allowed
string	`"Ada"`	Double quotes only
number	`36 -1.5 2e3`	Decimal, no NaN or Infinity
boolean	`true false`	Lowercase, unquoted
null	`null`	Lowercase, unquoted

String escapes

`\"`	double quote	`\\`	backslash
`\/`	slash (optional)	`\n \r \t`	newline, CR, tab
`\b \f`	backspace, form feed	`\u00e9`	char by 4-digit hex

Raw characters below U+0020 (newline, tab) must be escaped. Above U+FFFF use a surrogate pair: `\ud83d\ude00`.

Number rules

Valid	Invalid	Why
`7`	`07`	no leading zeros
`0.5`	`.5` `5.`	digits both sides of the point
`-3`	`+3`	no plus sign
`1e3` `1E-2`	`0x1F`	no hexadecimal
`null`	`NaN` `Infinity`	not representable

JavaScript parses numbers as 64-bit floats: integers above 9,007,199,254,740,991 (2^53 - 1) lose precision. Send large IDs as strings.

Common errors and fixes

Error	Wrong	Right
Trailing comma	`{"a": 1,}`	`{"a": 1}`
Single quotes	`{'a': 1}`	`{"a": 1}`
Unquoted key	`{a: 1}`	`{"a": 1}`
Missing comma	`{"a": 1 "b": 2}`	`{"a": 1, "b": 2}`
Comment	`{"a": 1 // x}`	no comments in JSON
Wrong literal	`True None undefined`	`true null`
Raw newline	line break inside a string	use `\n`

Read the reported line and column, then check the character just before it.

JSON vs a JavaScript object

	JavaScript	JSON
Keys	bare, string, symbol	double-quoted only
Quotes	' " or `	" only
Trailing comma	allowed	not allowed
Comments	allowed	not allowed
Values	functions, undefined, Date	the six types only
Dates	Date object	ISO 8601 string

JSONPath basics (RFC 9535)

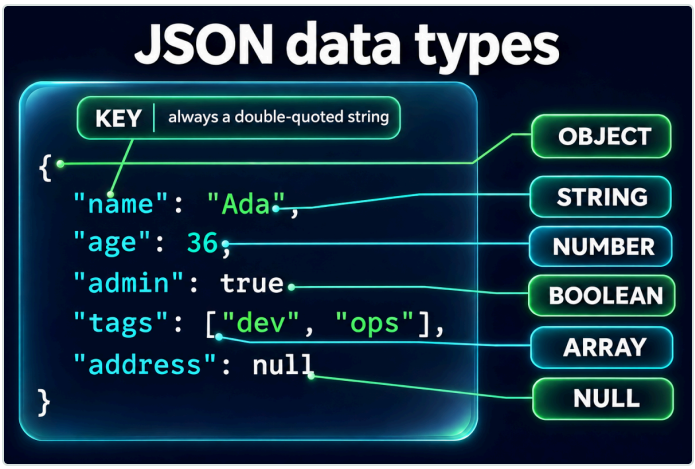
```
{"store": {"name": "Corner Books",
  "books": [{"title": "Dune", "price": 9},
            {"title": "Emma", "price": 7}]}}
```

Path	Selects
`$`	whole document
`$.store.name`	"Corner Books"
`$.store.books[0]`	first book object
`$.store.books[1].title`	"Emma"
`$..title`	every title, any depth
`$.store.books[*].price`	all prices
`$.store.books[0:1]`	slice: first book only
`$.store.books[?@.price < 8]`	filter: books under 8

The Tree View tab builds the `$`, `.key` and `[n]` forms when you click a value. `[*]`, `..`, slices and filters are standard JSONPath that the tool does not generate.

JSON Tools tabs

Format	beautify (2, 4 spaces, tabs) or minify
Validate	syntax check with line and column
Compare	added, removed, changed; optional ignore key order
Tree View	collapsible tree, search, click for JSONPath
Convert	YAML, CSV (arrays), TypeScript interface, XML



One document using all six data types.

Free tools at rjl.io