

Tham Khảo Nhanh JSON

Cú pháp, kiểu dữ liệu, objects, arrays, jq

Cú Pháp

Quy Tắc

<code>{ }</code>	Object (cặp key-value không thứ tự)
<code>[]</code>	Mảng (danh sách giá trị có thứ tự)
"key": value	Key phải là chuỗi có dấu ngoặc kép
No trailing comma	Phần tử cuối không được có dấu phẩy
No comments	JSON không cho phép comments

Ví Dụ Tối Giản

```
{
  "name": "Alice",
  "age": 30,
  "active": true
}
```

Kiểu Dữ Liệu

Sáu Kiểu Giá Trị

"string"	Văn bản UTF-8 có dấu ngoặc kép
42 / 3.14	Số (số nguyên hoặc số thực)
true / false	Boolean
null	Null (không có giá trị)
<code>{ }</code>	Object
<code>[]</code>	Mảng

Ký Tự Escape Trong Chuỗi

<code>\"</code>	Dấu ngoặc kép
<code>\\</code>	Dấu gạch chéo ngược
<code>\n \t</code>	Xuống dòng, tab
<code>\uXXXX</code>	Unicode escape (hex)

Objects

Cú Pháp Object

```
{
  "id": 1,
  "name": "Widget",
  "tags": ["new", "sale"]
}
```

Quy Tắc

Keys	Phải là chuỗi duy nhất có dấu ngoặc kép
Values	Bất kỳ kiểu JSON hợp lệ
Order	Thứ tự key không được đảm bảo
Nesting	Object có thể chứa object

Arrays

Cú Pháp Mảng

```
[1, "two", true, null, {"key": "val"}]
```

Mảng Kiểu Hỗn Hợp

```
{
  "matrix": [[1, 2], [3, 4]],
  "empty": []
}
```

Quy Tắc

Ordered	Phần tử giữ thứ tự chèn
Mixed types	Phần tử mảng có thể khác kiểu
Indexing	Bắt đầu từ 0 (trong hầu hết ngôn ngữ)

Lồng Nhau

Cấu Trúc Lồng

```
{
  "user": {
    "name": "Alice",
    "address": { "city": "Boston" },
    "scores": [95, 88, 72]
  }
}
```

Mẫu Truy Cập

obj.user.name	Ký hiệu chấm (JavaScript)
obj["user"]["name"]	Ký hiệu ngoặc vuông
obj.user.scores[0]	Chỉ số mảng trong object lồng

Xác Thực Schema

Ví Dụ JSON Schema

```
{
  "type": "object",
  "properties": {
    "name": { "type": "string" },
    "age": { "type": "integer", "minimum": 0 }
  },
  "required": ["name"]
}
```

Từ Khóa Schema

type	string, number, integer, boolean, object, array, null
required	Mảng tên thuộc tính bắt buộc
properties	Định nghĩa thuộc tính object mong đợi
enum	Hạn chế tập giá trị cố định
minLength / maxLength	Ràng buộc độ dài chuỗi
minimum / maximum	Ràng buộc phạm vi số

jq Cơ Bản

Bộ Lọc Phổ Biến

.	Identity — truyền input qua
.key	Truy cập key object
.key.nested	Truy cập key lồng nhau
.[0]	Phần tử mảng đầu tiên
.[]	Lặp qua tất cả phần tử mảng
select(.age > 20)	Lọc theo điều kiện
map(.name)	Biến đổi từng phần tử
length	Độ dài mảng hoặc chuỗi
keys	Các key object dưới dạng mảng

Ví Dụ jq

```
echo '{"a":1}' | jq '.a' # 1
echo '[1,2,3]' | jq 'map(. * 2)' # [2,4,6]
cat data.json | jq '.users[].name'
cat data.json | jq '.[] | select(.active)'
```

Mẫu Phổ Biến

API Response

```
{
  "status": 200,
  "data": [{"id": 1, "name": "Alice"}],
  "meta": {"total": 42, "page": 1}
}
```

File Cấu Hình

```
{
  "host": "localhost",
  "port": 8080,
  "debug": false,
  "features": ["auth", "logging"]
}
```

Mẹo

Validate	Dùng jsonlint hoặc python -m json.tool
Pretty print	jq . file.json hoặc python -m json.tool
JSONL	Một JSON object mỗi dòng (phân cách newline)
JSON5 / JSONC	Mở rộng cho phép comments và dấu phẩy cuối