

# 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)
<b>"key": value</b>	Key phải là chuỗi có dấu ngoặc kép
<b>No trailing comma</b>	Phần tử cuối không được có dấu phẩy
<b>No comments</b>	JSON không cho phép comments

### Ví Dụ Tối Giản

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

## Kiểu Dữ Liệu

### Sáu Kiểu Giá Trị

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

### Ký Tự Escape Trong Chuỗi

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

## Objects

### Cú Pháp Object

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

### Quy Tắc

<b>Keys</b>	Phải là chuỗi duy nhất có dấu ngoặc kép
<b>Values</b>	Bất kỳ kiểu JSON hợp lệ
<b>Order</b>	Thứ tự key không được đảm bảo
<b>Nesting</b>	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

<b>Ordered</b>	Phần tử giữ thứ tự chèn
<b>Mixed types</b>	Phần tử mảng có thể khác kiểu
<b>Indexing</b>	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

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

## Xác Thức Schema

### Ví Dụ JSON Schema

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

## Từ Khóa Schema

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

## jq Cơ Bản

### Bộ Lọc Phổ Biến

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

### Ví Dụ jq

```
echo '{"a":1}' | jq 'a'
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

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