

# JSON 快速参考

语法、数据类型、对象、数组、jq

## 语法

### 规则

`{ }` 对象 (无序键值对)  
`[ ]` 数组 (有序值列表)  
**"key": value** 键必须是双引号字符串  
**No trailing comma** 最后一项不得有逗号  
**No comments** JSON 不允许注释

### 最小示例

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

## 数据类型

### 六种值类型

**"string"** 双引号 UTF-8 文本  
**42 / 3.14** 数值 (整数或浮点数)  
**true / false** 布尔值  
**null** 空值 (值缺失)  
`{ }` 对象  
`[ ]` 数组

### 字符串转义序列

`\"` 双引号  
`\\` 反斜杠  
`\n \t` 换行符、制表符  
`\uXXXX` Unicode 转义 (十六进制)

## 对象

### 对象语法

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

### 规则

**Keys** 必须是唯一的引号字符串  
**Values** 任意合法 JSON 类型  
**Order** 键的顺序无保证  
**Nesting** 对象可以嵌套对象

## 数组

### 数组语法

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

### 混合类型数组

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

### 规则

**Ordered** 元素保持插入顺序  
**Mixed types** 数组项可以是不同类型  
**Indexing** 从零开始 (大多数语言)

## 嵌套

### 嵌套结构

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

### 访问模式

`obj.user.name` 点号访问 (JavaScript)  
`obj["user"]["name"]` 方括号访问  
`obj.user.scores[0]` 嵌套对象中的数组索引

## Schema 验证

### JSON Schema 示例

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

### Schema 关键字

**type** string、number、integer、boolean、object、array、null  
**required** 必填属性名数组  
**properties** 定义对象的预期属性  
**enum** 限制为一组固定值  
**minLength / maxLength** 字符串长度限制  
**minimum / maximum** 数值范围限制

## jq 基础

### 常用过滤器

`.` 恒等——原样传递输入  
`.key` 访问对象键  
`.key.nested` 访问嵌套键  
`[0]` 数组第一个元素  
`[ ]` 遍历所有数组元素  
`select(.age > 20)` 按条件过滤  
`map(.name)` 对每个元素进行转换  
`length` 数组长度或字符串长度  
**keys** 对象的键作为数组

## jq 示例

```
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)'
```

## 常见模式

### API 响应

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

### 配置文件

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

### 使用提示

**Validate** 使用 jsonlint 或 python -m json.tool  
**Pretty print** jq -f file.json 或 python -m json.tool  
**JSONL** 每行一个 JSON 对象 (换行符分隔)  
**JSON5 / JSONC** 允许注释和尾随逗号的扩展格式