

# JSON 解析器

**截止日期：2017 年 6 月 12 日 23:59**

## 1. 描述

本次任务是实现一个简单的 JSON 解析器，能够完整地解析 JSON 文本，并且能从中提取特定的信息。

这次会使用 MOSS 进行 **查重**，如果发现有重复率过高的，不管是谁抄的谁，都得 0 分。

## 2 要求

### 2.1 硬性要求（50 分）

- a、如果不能单独完成，可以两个人一组（最后只需交一份），不过，最好是单独完成；查重后重复率过高的，都得 0 分。
- b、在截止日期之前提交；  
未能在截止日期前完成的，每拖延一天减 10 分，直至该项为 0 分。
- c、不得使用任何第三方库；
- d、使用 C/C++、java 编写。

### 2.2 功能性要求（40 分）

所有的测试用例都是 UTF-8 编码。

```
Input 1:
{
    "id": 2333,
    "name": "compiler"
}
Output 1:
valid
```

```
Input 2:
{
    "id": 2333
    "name": "compiler"
}
Output 2:
```

line 2, position 12: expected <,>

对于下面的 abcd 要求，如果无错，输出“**valid**”；否则输出**行号**、**起始位置**以及**原因**。

- a、完成 json 基本结构（键值对、对象、数组、字符串）的解析；(10 分)
  - b、完成对 true、false、null、整数的解析；(5 分)
  - c、完成浮点数的解析；(5 分)
  - d、完成科学计数法表示的数的解析；(5 分)；
  - e、完成 json 文本的格式化。添加一个命令行参数-pretty，对于文件\*.json，如果解析成功，那么将格式化后的 json 输出到另一个文件\*.pretty.json 中。(5 分)
- test 目录下的 e 目录下，有一个乱序的 country.json 文件以及一个格式化好、较易阅读的 country.pretty.json 文件。

例如，你的程序名为 json，那么命令行的输入应该是：

```
json -pretty *.json
```

- f、对于任意的 json 文本和一个给定的查询路径，可以查找到指定的值，并能给出这个值的类型（字符串？浮点数？对象？数组？等等）。(10 分)

例如，对于提供的 country.json，假设查询路径为“/RECORDS[35]/countryname”，那么就是查找 RECORDS 字段对应的数组的第 35 个对象的 countryname 字段对应的值，也就是类型为字符串的“China”；如果不存在，则返回 null。

查询路径的表现形式没有硬性要求，可自由发挥。

JSON 本身就像是一个树形结构，所以这个部分的重点就是如何表示查询路径并且查询到需要的信息，可参考 XML 的解析（XPath），但是不必完全一样。

## 2.3 结构性要求（10 分）

- a、完整地体现一个解析器的结构，读取源文件->词法分析->语法分析->简单的语义分析->实现语义；(5 分)
- b、模块化，代码设计良好，可重用性高。(5 分)

## 3、JSON

### 3.1 BNF

```
<json> ::= <object> | <array>
<object> ::= "{" "}" | "{" <members> "}"
<members> ::= <pair> | <pair> "," <members>
<pair> ::= <string> ":" <value>
<array> ::= "[" "]" | "[" <elements> "]"
<elements> ::= <value> | <value> "," <elements>
<value> ::= <string> | <number> | <object> | <array> | "true" | "false" | "null"
<number> ::= <integer> | <float> | <scientific>
```

其中：

<string>: 字符串, 包含转义符以及 Unicode

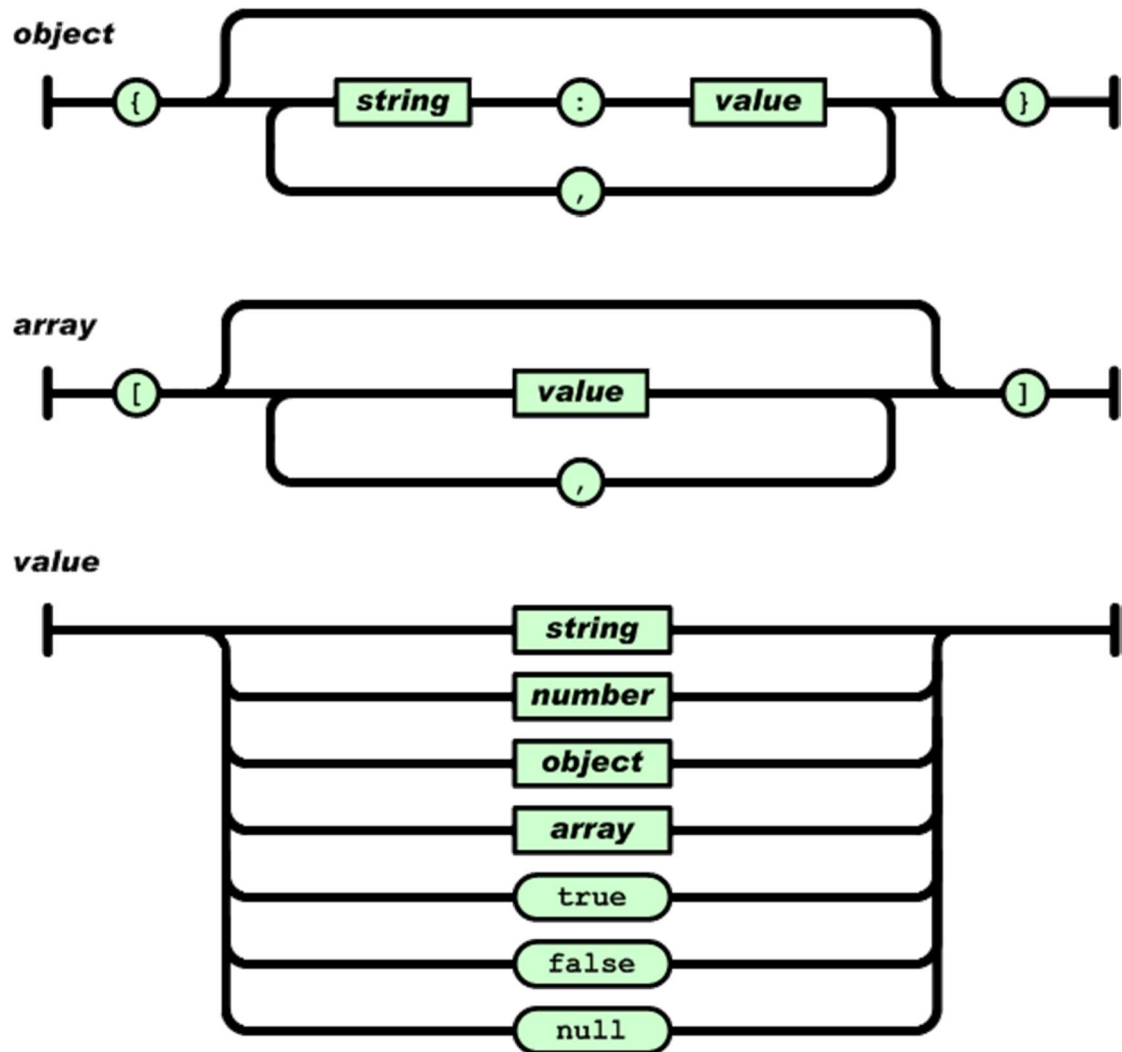
<integer>: 整数

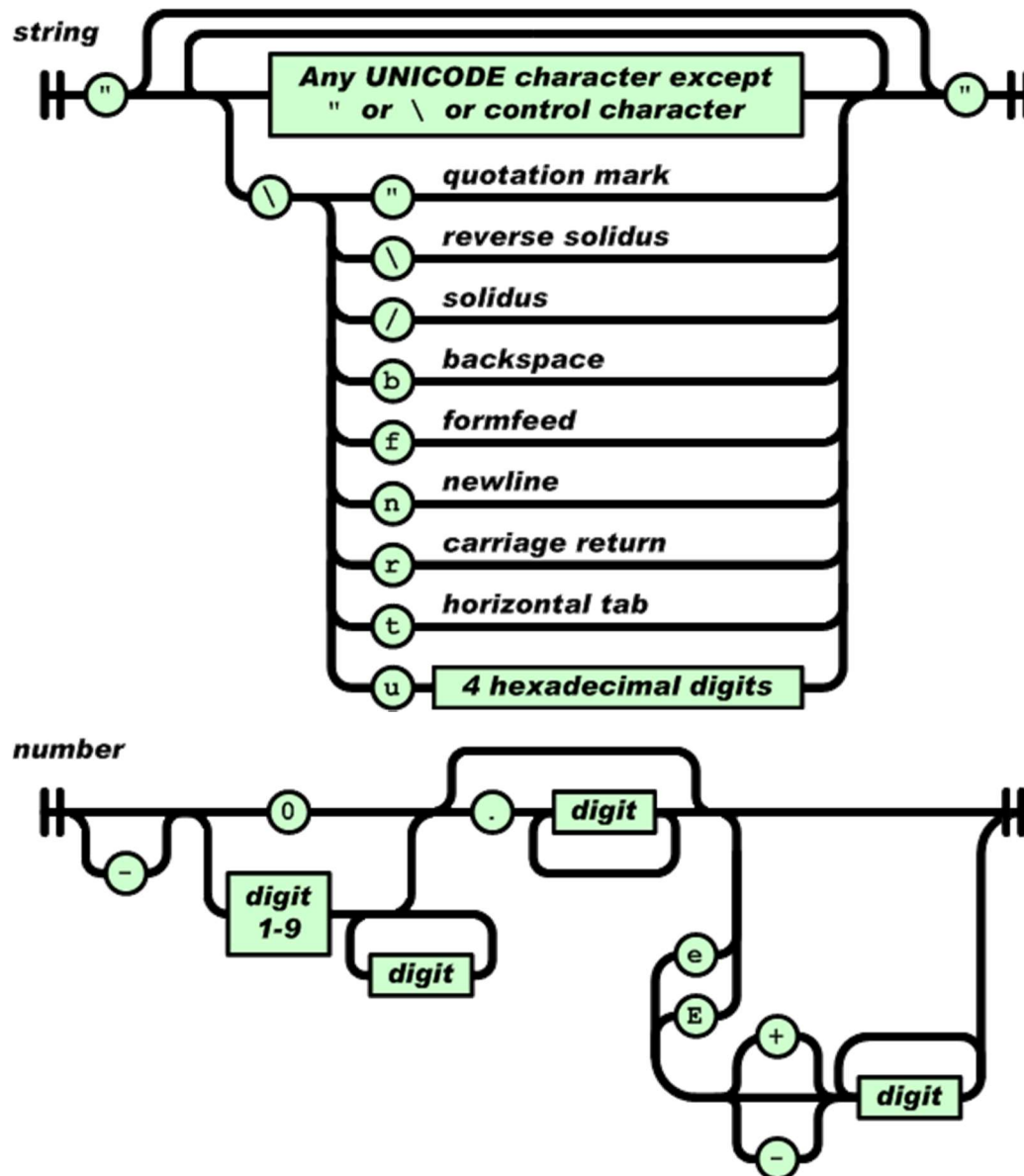
<float>: 浮点数

<scientific>: 科学计数法表示的数

每个 token 之间, 可以有任意多个空白符。

## 3.2 DFA





参考资料: <http://json.org>

## 4、提交须知

- 1) 将程序导出（编译）为可执行文件.jar 或者.exe;
- 2) 程序不需要 GUI，接受一个 json 文件路径为命令行参数;
- 3) 程序执行无错请立即退出，不要在代码里写 `cin.get()`或者 `system("pause")`之类的来等待输入退出;
- 4) 编写一个描述你的思路和实现的说明文档，必须有运行结果截图；如果是两个人组队，那么两个人都要写，并且描述一下自己做了哪些工作；
- 5) 源码放在 src 文件夹下，可执行文件、文档、测试用例与 src 同级；
- 6) 将所有的文件、文件夹放在.zip 压缩文件中，文件名格式：学号-班级-姓名.zip；如果是组队的，那么格式为：学号 1&学号 2-班级-姓名 1&姓名 2.zip
- 7) 提交邮箱: [compiler2017@126.com](mailto:compiler2017@126.com)，收到作业会有自动回复。