

Introducing JSON

JSON (JavaScript Object Notation) is a lightweight data-interchange format. It is easy for humans to read and write. It is easy for machines to parse and generate. It is based on a subset of the [JavaScript Programming Language, Standard ECMA-262 3rd Edition - December 1999](#). JSON is a text format that is completely language independent but uses conventions that are familiar to programmers of the C-family of languages, including C, C++, C#, Java, JavaScript, Perl, Python, and many others. These properties make JSON an ideal data-interchange language.

JSON is built on two structures:

- A collection of name/value pairs. In various languages, this is realized as an *object*, record, struct, dictionary, hash table, keyed list, or associative array.
- An ordered list of values. In most languages, this is realized as an *array*, vector, list, or sequence.

These are universal data structures. Virtually all modern programming languages support them in one form or another. It makes sense that a data format that is interchangeable with programming languages also be based on these structures.

In JSON, they take on these forms:

An *object* is an unordered set of name/value pairs. An object begins with { (left brace) and ends with } (right brace). Each name is followed by : (colon) and the name/value pairs are separated by , (comma).

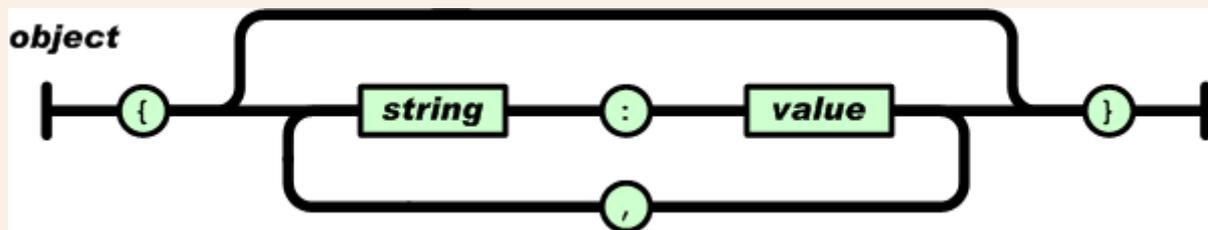
```
object
  {}
  { members }
members
  string : value
  members , string : value
array
  []
  [ elements ]
elements
  value
  elements , value
value
  string
  number
  object
  array
  true
  false
  null

string
  ""
  " chars "
```

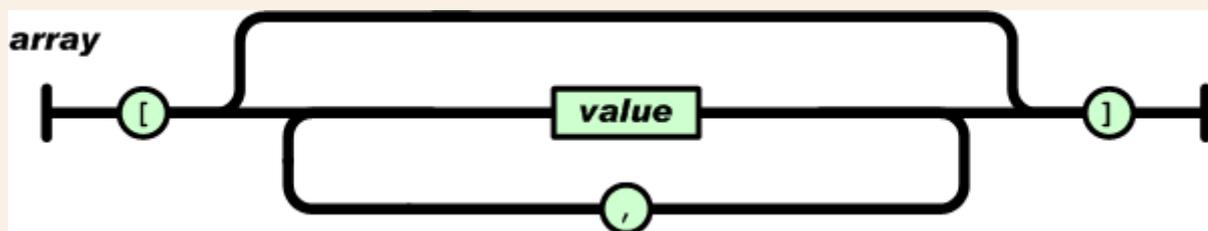
```

chars
char
chars char
char
any-Unicode-except-"-or-\-or-control
\"
\\
\/
\b
\f
\n
\r
\t
\u four-hex-digits
number
int
int frac
int exp
int frac exp
int
digit
digit1-9 digits
- digit
- digit1-9 digits
frac
. digits
exp
e digits
digits
digit
digits digit
e
e
e+
e-
E
E+
E-

```

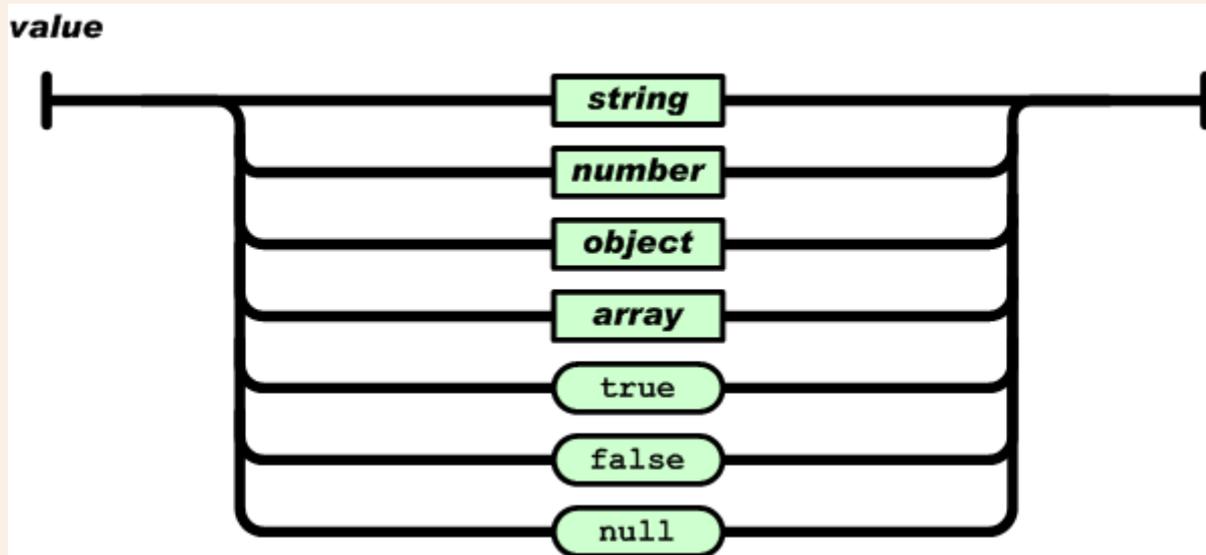


An *array* is an ordered collection of values. An array begins with [(left bracket) and ends with] (right bracket). Values are separated by , (comma).

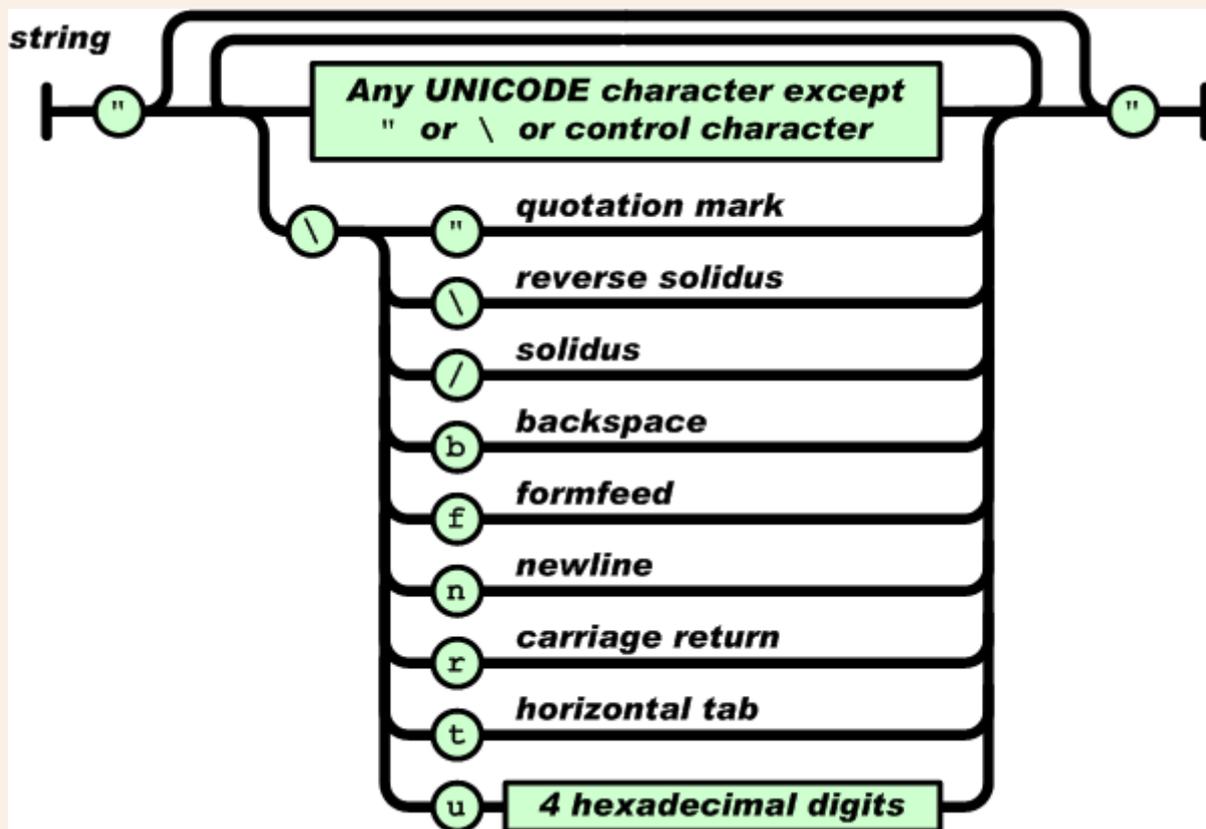


A *value* can be a *string* in double quotes, or a *number*, or true or false or null, or an *object* or an *array*.

These structures can be nested.



A *string* is a collection of zero or more Unicode characters, wrapped in double quotes, using backslash escapes. A character is represented as a single character string. A *string* is very much like a C or Java string.



A *number* is very much like a C or Java number, except that the octal and hexadecimal formats are not used.

- [JSON in PHP:](#)
 - [Zend_JSON](#)
 - [JSON-PHP.](#)
 - [PHP-JSON.](#)
 - [Services_JSON.](#)
 - [JSON PHP.](#)
- [JSON in Python.](#)
- [JSON in Rebol.](#)
- [JSON in Ruby.](#)
- [JSON in Squeak.](#)