**JSON is a Standard**

Internet Engineering Task Force (IETF) and European Computer Manufactures Association (ECMA) have recognized JSON as a standard, Douglas Crockford were the originally created JSON and in March 2014, Tim Bray an updated version of Douglas.

JSONLint is used for validate the JSON array in a text area, for example:

{ “thisIs”: “My first JSON document”}

Factors have popularized JSON:

• The explosive growth of RESTful APIs based on JSON

• The simplicity of JSON’s basic data structures

• The increasing popularity of JavaScript

JSON is replacing XML because is smaller, has less overhead and is more compact.

**Core JSON: t**his format includes JSON Data, Value Types, versions, comments, and File/MIME types.

JSON has the following core Data Types:

Name (or Key)/value pair: consists of a key (a data attribute) and a value.

Object: an unordered collection of name/value pairs.

Array: a collection of ordered values.

Name/value pairs have the following characteristics:

* Each name is on the left side of the colon (:), a String, and must be surrounded by double quotes
* The value is to the right of the colon.

Objects consist of name/value pairs, and following characteristic:

* Are enclosed within a beginning left curly brace ({) and an ending right curly brace (})
* Consist of comma-separated, unordered, name/value pairs
* Can be empty, { }

Arrays: containing nested Objects and Arrays and following characteristics:

* Are enclosed within a beginning left brace ([) and an ending right brace (])
* Consist of comma-separated, ordered values (see the next section)
* Can be empty, [ ]

**JSON Value Types:** object, array, string, number, Boolean, and null.

**Valid JSON String properties**

* Strings consist of zero or more Unicode characters enclosed in quotation marks (""). Please see the following list for additional valid characters.
* Strings wrapped in single quotes (') are not valid

And JSON String can contain backslash-escaped characters, for example: \n, \r, \u, and others.

**Valid JSON Number properties**

* Numbers are always in base 10 (only digits 0–9 are allowed) with no leading zeros.
* Numbers can have a fractional part that starts with a decimal pont (.).
* Numbers can have an exponent of 10, which is represented with the e or E nota‐
* Unlike JavaScript, numbers can’t have a value of NaN or Infinity.

**Valid JSON Boolean properties**

* Only true or false.
* The true or false value on the righthand side of the colon (:) is not surrounded by “”.

**Valid JSON Null properties**

* Are not not surrounded by quotes
* Indicate that a key/property has no value
* Act as a placeholder

In conclusion JSON is a lightweight data exchange format. Reading and writing it is simple for humans, while for machines it is simple to interpret and generate it. It is based on a subset of the JavaScript Programming Language, also it is completely language independent but uses conventions that are widely known to programmers in the C family of languages, including C, C ++, C #, Java, JavaScript, Perl, Python, and others.

**Beautiful Code Chapter 3**

The most beautiful code I have written gives us a series of tips to make our code more understandable both for the creator of the code and for other software developers.

The quicksort is an elegant and easy-to-understand algorithm, but a few years ago, ordering an array of n elements took a certain time, since it required a number of comparisons for each element of the array to be ordered correctly, One way to analyze your runtime with a recurrence ratio similar to that of Quicksort is the Knuth's publication.

From Alan Perlis's phrase, who said what? Is it possible that the software is not like anything else, that it is destined to be discarded: that the objective is to see it as a soap bubble ?, concluding that the most beautiful code and its functions are those that are not created or do not exist remembering that a code should have fewer and fewer lines of code.