JSON

JSON (or **JavaScript Object Notation**) is a text-based format for storing and transmitting structured data.

JSON's lightweight syntax allows you to easily store and send everything from numbers and strings to arrays and objects to other apps.

Although it originates in the JavaScript language, it is still considered to be language-independent.

JSON text can be built on one of two structures:

- a collection of the **key:value** pairs (associative array);
- an orderly set of values (array or list).

JSON objects are written in curly braces {}, and their key:value pairs are separated by a comma ,. The key and the value in the pair are separated by a colon :

```
{
   "first_name": "Sophie",
   "last_name": "Goodwin",
   "age": 34
}
```

Keys in an object are always strings, but values can be any of seven types of values, including another object or array.

Note that you should not put a comma (,) after the last key:value pair.

Again, the value in the array can be of any type, including another array or object. Here is an example of an array:
["night", "street", false, [345, 23, 8, "juice"], "fruit"]

JSON does not support comments.

Most often, an array will include similar elements.

Nested objects

JSON is a highly flexible format. You can nest objects inside other objects as properties:

```
{
  "persons": [
    {
      "firstName": "Whitney",
      "lastName": "Byrd",
```

```
    "age": 20
   },
   {
    "firstName": "Eugene",
    "lastName": "Lang",
    "age": 26
   },
   {
    "firstName": "Sophie",
    "lastName": "Goodwin",
    "age": 34
   }
 ]
}
```

The data has a tree-like structure if objects and arrays contain other objects or arrays.
The nested objects are fully independent and may have different properties:

```
{
 "persons": [
   {
    "firstName": "Whitney",
    "age": 20
   },
   {
    "firstName": "Eugene",
    "lastName": "Lang"
   }
 ]
}
```

camelCase vs. snake_case

CamelCase is a style where compound words are together without spaces, but each word inside the phrase starts with a capital letter. The style is called camelCase because the capital letters inside the word resemble camel's humps.

With snake_case style, compound words are written with the underscore.

The right choice of JSON naming convention depends directly on your programming language and libraries. You can use both camelCase and snake_case; any choice will be valid but do not mix them in one JSON.

JSON advantages

JSON is a widespread format for data exchange on the Internet because of its strong advantages:

- compactness;
- flexibility;
- high readability, even for people far from programming;
- most programming languages have functions and libraries for reading and creating JSON structures.

JSON is a popular format for transmitting structured data over a network. When you serialize data to JSON, you can easily deserialize it back without losing any information.

One of the main advantages of JSON over plain text is that it allows you to describe relationships between objects through nesting and key-value pairs.

Other popular applications of JSON are data storage and configuration files for other programs.