**JSON** (or **JavaScript Object Notation**) is a text-based format for storing and transmitting structured data. Although it originates in the JavaScript language, it is still considered to be language-independent: it works with almost any programming language. JSON's lightweight syntax allows you to easily store and send everything from numbers and strings to arrays and objects to other apps. You can also create more complex data structures by linking arrays to each other.

**Basic syntax and structure**

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 :. Here is an example for you:

{

"first\_name": "Sophie",

"last\_name": "Goodwin",

"age": 34

}

Here, you can see some user data in JSON format.

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"]

Most often, an array will include similar elements.

JSON does not support comments.

**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"

}

]

}

But in practice, such objects often look similar.

**camelCase vs. snake\_case**

If you have read the JSON objects examples carefully, you might have a lingering question: what style of compound word writing should be used for JSON?

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.

Withsnake\_casestyle, 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. Many of the websites you visit likely use JSON as well.

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

**Conclusion**

Now, you have seen that JSON is easy to understand and use, and it's fantastic since it's a handy tool for transferring data between applications. In working practice, you probably won't have to create JSON files yourself; you will get them from other sources, but if you want to save the code on your computer, save the files to the *.json* extension.

Read more on this in [What is data scraping and how to do it right](https://hyperskill.org/blog/post/what-is-data-scraping-and-how-to-do-it-right) on Hyperskill Blog.

Create a JSON object that has only a single field named country. This field stores the name of a country.

{"country":"Canada"}

Which parentheses are used to enclose key-value pairs of a JSON structure?{}

Create a JSON object with song and musician keys. Specify your favorite song in the object.

{

"song":"USA",

"muician":"james"

}

Select a valid JSON from the list below.

{ "first\_name": "Sophie", "last\_name": "Goodwin" }

What syntax is used to separate the key from its value in JSON?

{"key" ? value}

<code>:</code>

Hey, stranger, let's get acquainted! Write a JSON object with the keys name and age; specify your data in the values.

P.S. If you do not want to use your data, you can use random values.

{"name":"james","age":34}

=============================================================

Create a JSON object that represents a library.

A library has only a single books field, but each book has exactly two fields: title and author. Assign information about books you've read to these fields.

Fill the library with at least three books and enter the resulting JSON.

You should end up with nested objects. Look at section 2 in theory to remember their syntax.  
You can check if the JSON you wrote is valid at https://jsonlint.com/

{

"books": [

{

"title": "book1",

"author": "james"

},

{

"title": "book2",

"author": "james"

},

{

"title": "book3",

"author": "james"

}

]

}

What does JSON stand for? JavaScript Object Notation

Choose a single-line JSON-structure which represents a pet. It must have two keys: name and type (a cat, a dog, or something exotic).

{ "name": "Yasha", "type": "parrot" }