**Источник:** [**https://uk.wikipedia.org/wiki/JSON**](https://uk.wikipedia.org/wiki/JSON)

**JSON** ([англ.](https://uk.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D1%96%D0%B9%D1%81%D1%8C%D0%BA%D0%B0_%D0%BC%D0%BE%D0%B2%D0%B0) *JavaScript Object Notation*, [укр.](https://uk.wikipedia.org/wiki/%D0%A3%D0%BA%D1%80%D0%B0%D1%97%D0%BD%D1%81%D1%8C%D0%BA%D0%B0_%D0%BC%D0%BE%D0%B2%D0%B0" \o "Українська мова) *запис об'єктів [JavaScript](https://uk.wikipedia.org/wiki/JavaScript" \o "JavaScript)*, вимовляється *джейсон*) — це текстовий [формат](https://uk.wikipedia.org/wiki/%D0%A4%D0%BE%D1%80%D0%BC%D0%B0%D1%82" \o "Формат) обміну [даними](https://uk.wikipedia.org/wiki/%D0%94%D0%B0%D0%BD%D1%96" \o "Дані) між [комп'ютерами](https://uk.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%27%D1%8E%D1%82%D0%B5%D1%80" \o "Комп'ютер). JSON базується на тексті, може бути прочитаним людиною. Формат дозволяє описувати [об'єкти](https://uk.wikipedia.org/wiki/%D0%9E%D0%B1%27%D1%94%D0%BA%D1%82_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F)) та інші [структури даних](https://uk.wikipedia.org/wiki/%D0%A1%D1%82%D1%80%D1%83%D0%BA%D1%82%D1%83%D1%80%D0%B0_%D0%B4%D0%B0%D0%BD%D0%B8%D1%85). Цей формат головним чином використовується для передачі структурованої інформації через [мережу](https://uk.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%27%D1%8E%D1%82%D0%B5%D1%80%D0%BD%D0%B0_%D0%BC%D0%B5%D1%80%D0%B5%D0%B6%D0%B0) (завдяки процесу, що називають [серіалізацією](https://uk.wikipedia.org/wiki/%D0%A1%D0%B5%D1%80%D1%96%D0%B0%D0%BB%D1%96%D0%B7%D0%B0%D1%86%D1%96%D1%8F)).

Розробив і популяризував формат [Дуглас Крокфорд](https://uk.wikipedia.org/wiki/%D0%94%D1%83%D0%B3%D0%BB%D0%B0%D1%81_%D0%9A%D1%80%D0%BE%D0%BA%D1%84%D0%BE%D1%80%D0%B4).

JSON знайшов своє головне призначення у написанні веб-програм, а саме при використанні технології [AJAX](https://uk.wikipedia.org/wiki/AJAX). JSON, що використовується в AJAX, виступає як заміна [XML](https://uk.wikipedia.org/wiki/XML) (використовується в [AJAX](https://uk.wikipedia.org/wiki/AJAX)) під час асинхронної передачі структурованої інформації між [клієнтом](https://uk.wikipedia.org/wiki/%D0%9A%D0%BB%D1%96%D1%94%D0%BD%D1%82_(%D1%96%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0)) та [сервером](https://uk.wikipedia.org/wiki/%D0%A1%D0%B5%D1%80%D0%B2%D0%B5%D1%80). При цьому перевагою JSON перед XML є те, що він дозволяє складні структури в атрибутах, займає менше місця і прямо інтерпретується за допомогою [JavaScript](https://uk.wikipedia.org/wiki/JavaScript) в об'єкти.

Використання

За рахунок своєї лаконічності в порівнянні з [XML](https://uk.wikipedia.org/wiki/XML), формат JSON може бути більш придатним для [серіалізації](https://uk.wikipedia.org/wiki/%D0%A1%D0%B5%D1%80%D1%96%D0%B0%D0%BB%D1%96%D0%B7%D0%B0%D1%86%D1%96%D1%8F) складних структур.

Якщо говорити про веб-застосунки, в такому ключі він доречний в задачах обміну даними як між браузером і сервером ([AJAX](https://uk.wikipedia.org/wiki/AJAX)), так і між самими серверами (програмні HTTP-інтерфейси). Формат JSON так само добре підходить для зберігання складних динамічних структур в реляційних базах даних або файловому кеші.

## Синтаксис

JSON будується на двох структурах:

* Набір пар назва/значення. У різних мовах це реалізовано як об'єкт, запис, структура, словник, хеш-таблиця, список з ключем або асоціативним масивом.
* Впорядкований список значень. У багатьох мовах це реалізовано як масив, вектор, список, або послідовність.

Це універсальні структури даних. Теоретично всі сучасні мови програмування підтримують їх у тій чи іншій формі. Оскільки JSON використовується для обміну даними між різними мовами програмування, то є сенс будувати його на цих структурах.

У JSON використовуються такі їхні форми:

* **Об'єкт** — це послідовність пар назва/значення. Об'єкт починається з символу { і закінчується символом } . Кожне значення слідує за : і пари назва/значення відділяються комами.
* **Масив** — це послідовність значень. Масив починається символом [ і закінчується символом ]. Значення відділяються комами.
* **Значення** може бути рядком в подвійних лапках, або числом, або логічними true чи false, або null, або об'єктом, або масивом. Ці структури можуть бути вкладені одна в одну.
* **Рядок** — це послідовність з нуля або більше символів [юнікода](https://uk.wikipedia.org/wiki/%D0%AE%D0%BD%D1%96%D0%BA%D0%BE%D0%B4" \o "Юнікод), обмежена подвійними лапками, з використанням escape-послідовностей, що починаються зі зворотної косої риски \. Символи представляються простим рядком.

Тип Рядок (String) дуже схожий на String в мовах [C](https://uk.wikipedia.org/wiki/C_(%D0%BC%D0%BE%D0%B2%D0%B0_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F)) і [Java](https://uk.wikipedia.org/wiki/Java). Число теж дуже схоже на C- або Java-число, за винятком того, що вісімкові та шістнадцяткові формати не використовуються. Пропуски можуть бути вставлені між будь-якими двома лексемами.

Наступний приклад показує JSON представлення об'єкта, що описує людину. У об'єкті є рядкові поля імені і прізвища, об'єкт, що описує адресу, і масив, що містить список телефонів.

{

**"firstName"**: "Іван",

**"lastName"**: "Коваленко",

**"address"**: {

**"streetAddress"**: "вул. Грушевського 14, кв.101",

**"city"**: "Київ",

**"postalCode"**: 21000

},

**"phoneNumbers"**: [

"044 123-1234",

"050 123-4567"

]

}

## Питання безпеки[[ред.](https://uk.wikipedia.org/w/index.php?title=JSON&veaction=edit&section=6) • [ред. код](https://uk.wikipedia.org/w/index.php?title=JSON&action=edit&section=6)]

Хоча JSON призначений для передачі даних в серіалізованому вигляді, його синтаксис відповідає синтаксису [JavaScript](https://uk.wikipedia.org/wiki/JavaScript) і це створює низку проблем безпеки. Часто для обробки даних, отриманих від зовнішнього джерела у форматі JSON, до них застосовується функція eval() без якої-небудь попередньої перевірки.

### JavaScript eval()

Оскільки JSON представляється синтаксично правильним фрагментом коду JavaScript, природним способом розбору JSON-даних в JavaScript-програмі є використання вбудованої в JavaScript функції eval(), яка призначена для обчислення JavaScript-виразів. При цьому підході відпадає необхідність у використанні додаткових [парсерів](https://uk.wikipedia.org/wiki/%D0%9F%D0%B0%D1%80%D1%81%D0%B5%D1%80" \o "Парсер).

Техніка використання eval() робить систему вразливою, якщо джерело JSON-даних, що використовуються, не відноситься до надійних. Такими даними може виступати шкідливий JavaScript-код для атак за допомогою ін'єкції коду. За допомогою цієї вразливості можливо здійснювати крадіжку даних, підробку [автентифікації](https://uk.wikipedia.org/wiki/%D0%90%D0%B2%D1%82%D0%B5%D0%BD%D1%82%D0%B8%D1%84%D1%96%D0%BA%D0%B0%D1%86%D1%96%D1%8F). Проте, вразливість можна усунути за рахунок використання додаткових засобів перевірки даних на коректність. Наприклад, до виконання eval() отримані від зовнішнього джерела дані можуть перевірятися за допомогою [регулярних виразів](https://uk.wikipedia.org/wiki/%D0%A0%D0%B5%D0%B3%D1%83%D0%BB%D1%8F%D1%80%D0%BD%D0%B8%D0%B9_%D0%B2%D0%B8%D1%80%D0%B0%D0%B7). У RFC, що визначає JSON[[1]](https://uk.wikipedia.org/wiki/JSON" \l "cite_note-1) пропонується використовувати такий код для перевірки його відповідності формату JSON

**var** my\_JSON\_object = !(/[^,:{}\[\]0-9.\-+Eaeflnr-u \n\r\t]/.test(

text.replace(/"(\\.|[^"\\])\*"/g, ''))) &&

eval('(' + testedData + ')');

Як безпечніша альтернатива eval() була запропонована нова функція parseJSON(), здатна обробляти тільки JSON-дані. Вона була представлена в четвертій версії стандарту [ECMAScript](https://uk.wikipedia.org/wiki/ECMAScript) і описана в статті [JSON: Знежирена альтернатива XML](http://www.webcitation.org/65OHt7Ehp). Архів [оригіналу](http://www.json.org/fatfree.html) за 2012-02-12.. В наш час[[*Коли?*](https://uk.wikipedia.org/wiki/%D0%92%D1%96%D0%BA%D1%96%D0%BF%D0%B5%D0%B4%D1%96%D1%8F:%D0%A1%D1%82%D0%B0%D1%82%D1%82%D1%96,_%D1%89%D0%BE_%D0%BD%D0%B5%D0%BE%D0%B1%D1%85%D1%96%D0%B4%D0%BD%D0%BE_%D0%BF%D0%BE%D0%BB%D1%96%D0%BF%D1%88%D0%B8%D1%82%D0%B8)] вона доступна як бібліотека JavaScript [http://www.JSON.org/json2.js](http://www.json.org/json2.js) і буде включена в п'яту редакцію ECMAScript.

### Вбудований JSON

Останні версії [веб-браузерів](https://uk.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D0%B1%D1%80%D0%B0%D1%83%D0%B7%D0%B5%D1%80) мають вбудовану підтримку JSON і здатні його обробляти без виклику функції eval(), що призводить до описаної проблеми. Обробка JSON у такому разі зазвичай здійснюється швидше. Так у червні 2009 року вбудовану підтримку JSON мали такі браузери:

* [Mozilla Firefox](https://uk.wikipedia.org/wiki/Mozilla_Firefox) 3.5+, [SeaMonkey](https://uk.wikipedia.org/wiki/SeaMonkey" \o "SeaMonkey) 2, та [Thunderbird](https://uk.wikipedia.org/wiki/Thunderbird) 3[[2]](https://uk.wikipedia.org/wiki/JSON#cite_note-2)
* [Microsoft Internet Explorer](https://uk.wikipedia.org/wiki/Microsoft_Internet_Explorer) 8[Native JSON in IE8](http://www.webcitation.org/65OHu3fNT). Архів [оригіналу](http://blogs.msdn.com/ie/archive/2008/09/10/native-json-in-ie8.aspx) за 2012-02-12. (англ.)
* [Opera 10.5](https://uk.wikipedia.org/wiki/Opera)+[[3]](https://uk.wikipedia.org/wiki/JSON#cite_note-3)
* Браузери, засновані на [WebKit](https://uk.wikipedia.org/wiki/WebKit" \o "WebKit) (наприклад, [Google Chrome](https://uk.wikipedia.org/wiki/Google_Chrome), [Apple Safari](https://uk.wikipedia.org/wiki/Apple_Safari))[[4]](https://uk.wikipedia.org/wiki/JSON#cite_note-4)

Принаймні дві популярні бібліотеки JavaScript використовують вбудований JSON у разі його доступності:

* [jQuery](https://uk.wikipedia.org/wiki/JQuery)[[5]](https://uk.wikipedia.org/wiki/JSON#cite_note-5)
* [Dojo](https://uk.wikipedia.org/wiki/Dojo)[[6]](https://uk.wikipedia.org/wiki/JSON#cite_note-6)

### Підробка крос-доменного запиту

Непродумане використання JSON робить сайти вразливими до підробки міжсайтових запитів (CSRF або XSRF)[[7]](https://uk.wikipedia.org/wiki/JSON#cite_note-7). Оскільки тег <script> допускає використання джерела, що не належить до того ж домену, що і використовуваний ресурс, це дозволяє виконувати код даних, представлених у форматі JSON, в контексті довільної сторінки, що робить можливою компрометацію паролів або іншої конфіденційної інформації користувачів, що пройшли авторизацію на іншому сайті.

Це є проблемою тільки у разі вмісту в JSON-даних конфіденційної інформації, яка може бути компрометована третьою стороною і якщо сервер розраховує на політику одного джерела, блокуючи доступ до даних при виявленні зовнішнього запиту. Це не є проблемою, якщо сервер визначає допустимість запиту, надаючи дані тільки у разі його коректності. HTTP [cookie](https://uk.wikipedia.org/wiki/Cookie) не можна використовувати для визначення цього. Виключне використання HTTP cookie використовується підробкою міжсайтових запитів.

## Розширення

### JSONP

JSONP або «JSON з підкладкою» є розширенням JSON, коли назва функції зворотного виклику вказується як вхідний аргумент. Спочатку ідея була запропонована в блозі MacPython в 2005 році[[8]](https://uk.wikipedia.org/wiki/JSON" \l "cite_note-8), і в наш час[*[Коли?](https://uk.wikipedia.org/wiki/%D0%92%D1%96%D0%BA%D1%96%D0%BF%D0%B5%D0%B4%D1%96%D1%8F:%D0%A1%D1%82%D0%B0%D1%82%D1%82%D1%96,_%D1%89%D0%BE_%D0%BD%D0%B5%D0%BE%D0%B1%D1%85%D1%96%D0%B4%D0%BD%D0%BE_%D0%BF%D0%BE%D0%BB%D1%96%D0%BF%D1%88%D0%B8%D1%82%D0%B8" \o "Вікіпедія:Статті, що необхідно поліпшити)*] використовується багатьма [Web 2.0](https://uk.wikipedia.org/wiki/Web_2.0) [застосунками](https://uk.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D0%B7%D0%B0%D1%81%D1%82%D0%BE%D1%81%D1%83%D0%BD%D0%BE%D0%BA" \o "Веб-застосунок), такими, як Dojo Toolkit Applications, Google Toolkit Applications[[9]](https://uk.wikipedia.org/wiki/JSON#cite_note-9) і zanox Web Services. Подальші розширення цього протоколу були запропоновані з урахуванням введення додаткових аргументів, як, наприклад, у разі JSONPP[[10]](https://uk.wikipedia.org/wiki/JSON#cite_note-10) за підтримки [S3DB](https://uk.wikipedia.org/w/index.php?title=S3DB&action=edit&redlink=1) вебсервісів.

Оскільки JSONP використовує скрипт-теги, виклики по суті відкриті світові. З цієї причини JSONP може бути недоречними для зберігання конфіденційних даних[[11]](https://uk.wikipedia.org/wiki/JSON" \l "cite_note-RIAspot-11).

Включення скриптових тегів від віддалених сайтів дозволяє їм передати **будь-який** контент на сайті. Якщо віддалений сайт має [вразливості](https://uk.wikipedia.org/wiki/%D0%9C%D1%96%D0%B6%D1%81%D0%B0%D0%B9%D1%82%D0%BE%D0%B2%D0%B8%D0%B9_%D1%81%D0%BA%D1%80%D0%B8%D0%BF%D1%82%D1%96%D0%BD%D0%B3), які дозволяють виконати ін'єкції JavaScript, то початковий сайт також може бути ними зачеплений.

### BSON

*Докладніше:*[*BSON*](https://uk.wikipedia.org/wiki/BSON)

BSON — це бінарна форма представлення простих структур даних і асоціативних масивів (які називають об'єктами або документами). Назва «BSON» заснована на визначенні JSON і неофіційно означає «Binary JSON» (бінарний JSON).