

# JSON

Jeszenszky Péter

2024. október 4.

# JSON (JavaScript Object Notation)

- Könnyűsúlyú szöveges nyelvfüggetlen adatcsere formátum.
- Strukturált adatok ábrázolására szolgál.
- Ember számára is könnyen olvasható és írható formátum.
- Szoftverek által könnyen generálható és feldolgozható.
- Az ECMAScript programozási nyelvből származik.
- Webhely: <https://www.json.org/>

# ECMAScript

- A JavaScript programozási nyelv szabványosítása.
- A jelenleg aktuális a 15-es számú kiadás:
  - Ecma International, *ECMAScript 2024 Language Specification*, Standard ECMA-262, 15th ed., June 2024. <https://www.ecma-international.org/publications-and-standards/standards/ecma-262/>
- A jelenleg fejlesztés alatt álló következő verzió az ECMAScript 2025:
  - *ECMAScript 2025 Language Specification* <https://tc39.es/ecma262/>

# JavaScript

- A JavaScript kifejezést használják az ECMAScript különböző gyártók általi megvalósításaira.
  - Lásd még: *JavaScript technologies overview*  
[https://developer.mozilla.org/en-US/docs/Web/JavaScript/JavaScript\\_technologies\\_overview](https://developer.mozilla.org/en-US/docs/Web/JavaScript/JavaScript_technologies_overview)

# JavaScript motorok (1)

- SpiderMonkey (programozási nyelv: C/C++; licenc: *Mozilla Public License 2.0*) <https://spidermonkey.dev/>
  - A Mozilla projekt JavaScript motorja.
- V8 (programozási nyelv: C++; licenc: *New BSD License*)  
<https://v8.dev/> <https://github.com/v8/v8/>
  - A Chromium JavaScript motorja.
- JavaScriptCore (programozási nyelv: C++; licenc: LGPLv2)  
<https://developer.apple.com/documentation/javascriptcore> <https://github.com/WebKit/WebKit/tree/main/Source/JavaScriptCore>
  - A WebKit renderelő motorhoz fejlesztett JavaScript motor.

## JavaScript motorok (2)

- GraalVM Community Edition (CE) (programozási nyelv: Java; licenc: GPLv2) <https://www.graalvm.org/> <https://github.com/oracle/graal>
  - Linux, macOS és Windows rendszerekhez áll rendelkezésre.
  - Lásd: <https://www.graalvm.org/latest/reference-manual/js/>
- Hermes (programozási nyelv: C++; licenc: *MIT License*)  
<https://hermesengine.dev/> <https://github.com/facebook/hermes>
  - React Native alkalmazások futtatásához optimalizált JavaScript motor.

## JavaScript motorok (3)

- JerryScript (programozási nyelv: C; licenc: *Apache License 2.0*)  
<https://jerryscript.net/>  
<https://github.com/jerryscript-project/jerryscript>
  - Pehelysúlyú JavaScript motor IoT eszközökre.

# Node.js (1)

- A V8 JavaScript motorra épülő JavaScript futtató környezet, melyet skálázható hálózati alkalmazások létrehozásához terveztek.
- Webhely: <https://nodejs.org/> <https://github.com/nodejs/node>
- Licenc: *MIT License*
- Programozási nyelv: C++, JavaScript
- Platform: Linux, macOS, Windows

## Node.js (2)

- Lehetővé teszi a fejlesztők számára böngészőn kívül futó JavaScript alkalmazások létrehozását.
- Használható kliens oldali és szerver oldali alkalmazások fejlesztéséhez is.
- Csomag ökoszisztémája, az npm, a világ legnagyobb nyílt forrású könyvtár ökoszisztémája.

## Node.js (3)

“Helló, világ!” példa:

- Forrás: <https://nodejs.org/en/docs/guides/getting-started-guide/>

```
const { createServer } = require('node:http');

const hostname = '127.0.0.1';
const port = 3000;

const server = http.createServer((req, res) => {
  res.statusCode = 200;
  res.setHeader('Content-Type', 'text/plain');
  res.end('Hello, World!\n');
});

server.listen(port, hostname, () => {
  console.log(`Server running at http://${hostname}:${port}/`);
});
```

# Node.js (4)

Node.js keretrendszerek:

- Fastify (licenc: *MIT License*) <https://fastify.dev/>  
<https://github.com/fastify/fastify>
- Express (licenc: *MIT License*) <http://expressjs.com/>  
<https://github.com/expressjs/express>
- Koa Express (licenc: *MIT License*) <https://github.com/koajs/koa>  
<https://github.com/koajs/koa>
- Meteor (licenc: *MIT License*) <https://www.meteor.com/>  
<https://github.com/meteor/meteor>
- NestJS (licenc: *MIT License*) <https://nestjs.com/>  
<https://github.com/nestjs/nest>
- Sails.js (licenc: *MIT License*) <https://sailsjs.com/>  
<https://github.com/balderdashy/sails>
- ...

# Kompatibilitás

- Implementációk ECMAScript kompatibilitása:
  - *ECMAScript Compatibility Tables*  
<https://compat-table.github.io/compat-table/es6/>  
<https://github.com/compat-table/compat-table>

- Nemzetközi nonprofit szabványosító szervezet.
- Célterület: infokommunikációs technológia (ICT), fogyasztói elektronika (CE)
- Eredetileg 1961-ben alapították, jelenlegi nevén 1994 óta működik.
  - *European Computer Manufacturers Association* (ECMA)
- Webhely: <https://www.ecma-international.org/>

# Történet

- “Felfedezője” és népszerűsítője Douglas Crockford.
- “Felfedezése” 2001-ben.
- Crockford eredetileg JavaScript kliensek és Java szerverek közötti kommunikációhoz használta.
- IEEE Computer Society, *Discovering JavaScript Object Notation with Douglas Crockford*, 2012. március 28.  
<https://www.youtube.com/watch?v=kc8BAR7SHJI>
  - “I don't claim to have invented it, because it already existed in nature. I just saw it, recognized the value of it, gave it a name, and a description, and showed its benefits. But I did not invent it. I don't claim to be the first person to have discovered it.”
- Forrás: <https://inkdroid.org/2012/04/30/lessons-of-json/>

# Állomány jellemzők

- Állománynév végződés: `.json`
- IANA média típus: `application/json`

# Specifikációk

- Ecma International, *The JSON Data Interchange Format*, Second Edition, Standard ECMA-404, December 2017. <https://www.ecma-international.org/publications-and-standards/standards/ecma-404/>
- T. Bray (ed.). *RFC 8259: The JavaScript Object Notation (JSON) Data Interchange Format*. December 2017. <https://www.rfc-editor.org/rfc/rfc8259>
- ISO/IEC 21778:2017: *Information technology – The JSON data interchange syntax* <https://www.iso.org/standard/71616.html>

# JSON vs. ECMAScript

- Az ECMAScript 2019-től kezdve a JSON az ECMAScript szintaktikai részhalmaza.
- Lásd:
  - *Subsume JSON a.k.a. JSON ECMAScript*, 14 August 2019.  
<https://v8.dev/features/subsume-json>
  - Axel Rauschmayer, *Exploring ES2018 and ES2019 – JSON superset*.  
[https://exploringjs.com/es2018-es2019/ch\\_json-superset.html](https://exploringjs.com/es2018-es2019/ch_json-superset.html)

# JSON vs. XML (1)

- A JSON az XML alternatívájaként használható adatcseréhez.
- Nagyjából ugyanazokat az előnyöket kínálja, mint az XML, azonban annak hátrányai nélkül.
- Lásd: *JSON: The Fat-Free Alternative to XML*  
<https://www.json.org/xml.html>

## JSON vs. XML (2)

A JSON és az XML közös jellemzői:

- Egyszerűség (egyértelműen a JSON a nyerő)
- Az ember számára is könnyen írható és olvasható formátumok
- Szoftverek által könnyen generálható és feldolgozható formátumok (egyértelműen a JSON a nyerő)
- Interoperabilitás
- Nyíltság
- Önleíró adatábrázolás
- Univerzális adatscere formátumok

## JSON vs. XML (3)

- A fő különbség az, hogy a JSON adat-orientált, az XML pedig dokumentum-orientált.
  - Adatszerkezetek ábrázolásához a JSON tökéletes választás.
    - Előnye az XML-hez képest, hogy kevésbé bőbeszédű.
  - Dokumentum-középpontú alkalmazásokhoz az XML-t használjuk.
    - Előnye a JSON-hoz képest, hogy kiterjeszthető, és hogy kiforrottabb infrastruktúra áll hozzá rendelkezésre (XML Schema, XSLT, XQuery).

# JSON vs. XML (4)

Példa:

- XML:

```
<movie>
  <title>The Dark Knight</title>
  <year>2008</year>
  <url>https://www.imdb.com/title/tt0468569/</url>
  <standalone>>false</standalone>
</movie>
```

- JSON:

```
{
  "movie": {
    "title": "The Dark Knight",
    "year": 2008,
    "url": "https://www.imdb.com/title/tt0468569/",
    "standalone": false
  }
}
```

# JSON vs. XML (5)

- XML:

```
<properties>
  <property name="user.home">/home/jeszy</property>
  <property name="user.name">jeszy</property>
</properties>
```

- JSON:

```
{
  "properties": {
    "property": [
      {
        "@name": "user.home",
        "#text": "/home/jeszy"
      },
      {
        "@name": "user.name",
        "#text": "jeszy"
      }
    ]
  }
}
```

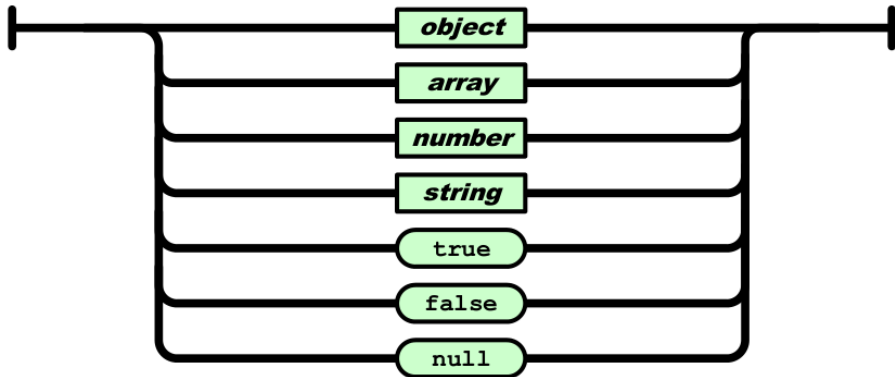
- Négy primitív adattípus ábrázolását teszi lehetővé:
  - sztringek
  - számok
  - logikai értékek
  - `null`
- Az alábbi két strukturált típus ábrázolását teszi lehetővé:
  - tömbök
  - objektumok

# Tokenek

- A JSON szöveg tokenek olyan sorozata, mely megfelel a JSON érték nyelvtani szabálynak.
- Tokenek:
  - Szerkezeti tokenek a {, }, [, ], : és , karakterek.
  - Sztringek
  - Számok
  - Literális tokenek a true, false és null karakterláncok.
- Tokenek előtt és után megengedettek *whitespace* karakterek, melyek nem lényegesek.
  - *Whitespace* karakter: HT (U+0009), LF (U+000A), CR (U+000D), szóköz (U+0020).
  - A tokenek közül csak a sztringekben megengedettek *whitespace* karakterek.

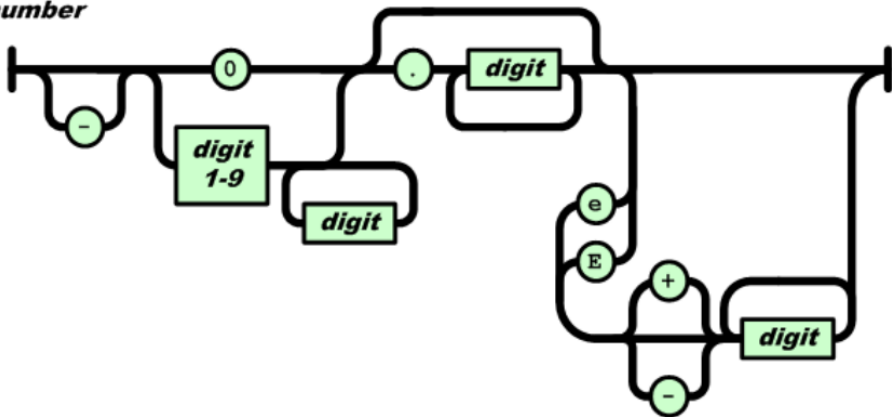
# JSON érték

*value*



# Számok (1)

*number*



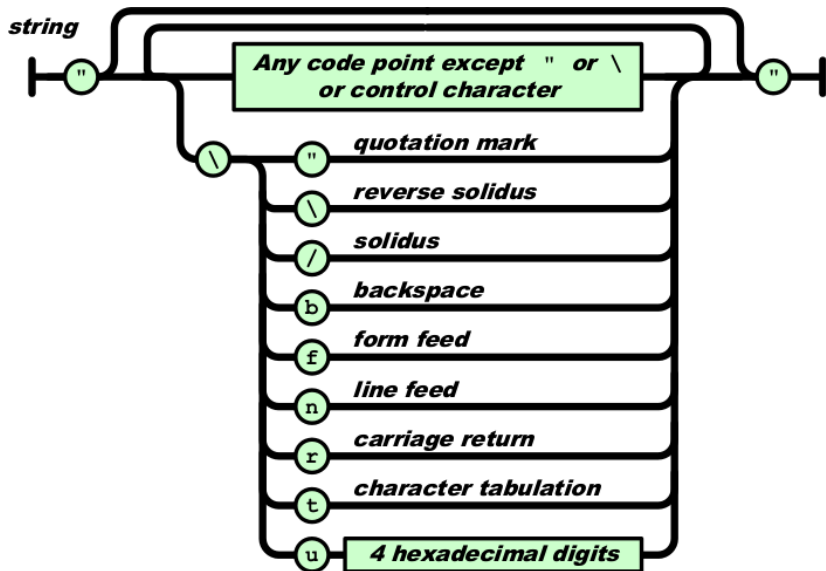
## Számok (2)

- Nincs korlátozás a számok tartományára és pontosságára.
- A gyakorlatban célszerű szem előtt tartani az interoperabilitást.
  - Például dupla pontosságú lebegőpontos számok használata javasolt.
- Példák:
  - 0
  - -1.0
  - 2.718282
  - 1E-12

# Sztringek (1)

- Unicode karakterek sorozatai, melyeket idézőjelek (U+0022) határolnak.
- Bármely karaktert tartalmazhatják, azonban az alábbiakat csak levédve:
  - idézőjel (U+0022), *backslash* (U+005C), vezérlő karakterek (U+0000–U+001F).
- Speciális karakterek megadásához rendelkezésre állnak a szokásos escape szekvenciák, mint például `\`, `\\`, `\t`, `\n`, `\r`.
- A BMP-hez tartozó Unicode karakterek megadhatóak `\unnnn` módon, ahol `nnnn` a karakterkód négy hexadecimális számjeggyel ábrázolva.
- Példák: `"`, `"\"Hello, World!\n\""`, `"\u263A"`, `"\u263a"`

## Sztringek (2)



# Tömbök (1)

- Tetszőleges számú érték rendezett sorozata (lehet üres).
- Az elemek tetszőleges típusúak lehetnek (akár tömbök is).

## Tömbök (2)

Példák:

- `["Athos", "Porthos", "Aramis", "d'Artagnan"]`
- `[9, 14, 19, 25, 26, 28]`
- `["Pi", 3.141593, null, true]`
- `[[45.7370889, 16.1133866], [48.5852340, 22.8981217]]`

# Objektumok (1)

- Tetszőleges számú név-érték párból állnak.
- A név tetszőleges sztring, az érték tetszőleges JSON érték.
- A név-érték párokra a tag (*member*) elnevezést is használjuk.

## Objektumok (2)

RFC 8259:

- Olyan objektum interoperábilis, melynél a tagok nevei egyediek.
- Eltérően viselkedhetnek az alkalmazások nem egyedi nevek esetén.
- Nem minden JSON elemző esetén állapítható meg a név-érték párok sorrendje.

## Objektumok (3)

Példa:

```
{  
  "title": "Alien",  
  "year": 1979,  
  "rating": 8.5,  
  "votes": 981765,  
  "genres": ["horror", "sci-fi"],  
  "url": "https://www.imdb.com/title/tt0078748/"  
}
```

# Objektumok (4)

Példa:

```
{  
  "artist": "Porcupine Tree",  
  "title": "Fear of a Blank Planet",  
  "year": 2007,  
  "tracks": [  
    {  
      "title": "Fear of a Blank Planet",  
      "length": 448  
    },  
    {  
      "title": "My Ashes",  
      "length": 307  
    },  
    {  
      "title": "Anesthetize",  
      "length": 1062  
    }  
  ]  
}
```

# Objektumok (5)

Példák:

- Frankfurter <https://www.frankfurter.app/>
  - Lásd például: <https://api.frankfurter.app/latest?base=HUF>
- Chuck Norris Jokes API <https://api.chucknorris.io/>
  - Lásd például: <https://api.chucknorris.io/jokes/random>
- Nominatim <https://nominatim.openstreetmap.org/>  
<https://nominatim.org/>
  - Lásd például: <https://nominatim.openstreetmap.org/search?q=debrecen&format=json>

# Karakterkódolás

RFC 8259:

- JSON szöveg különböző rendszerek közötti átvitelekor az UTF-8 karakterkódolást kell használni.

# JSON megjelenítés böngészőkben

- Firefox: tartalmaz beépített JSON megjelenítőt.
  - Lásd:  
[https://firefox-source-docs.mozilla.org/devtools-user/json\\_viewer/](https://firefox-source-docs.mozilla.org/devtools-user/json_viewer/)
- Chromium, Google Chrome:
  - Ajánlott kiterjesztések:
    - JSON Formatter <https://chromewebstore.google.com/detail/json-formatter/bcjindcccaagfpajjmafapmmgkkggoa>  
<https://github.com/callumlocke/json-formatter>
    - JSON Viewer Pro <https://chromewebstore.google.com/detail/json-viewer-pro/eifflpmocdbdmebjapkkhbfmdgijcc>  
<https://github.com/rbrahul/Awesome-JSON-Viewer>

# Szerkesztők (1)

Szabad és nyílt forrású szoftverek:

- JSON Editor (programozási nyelv: JavaScript; licenc: *Apache License 2.0*) <https://github.com/josdejong/jsoneditor/>
  - Online: <http://jsoneditoronline.org/>
- Visual Studio Code (platform: Linux, macOS, Windows; licenc: *MIT License*) <https://code.visualstudio.com/>  
<https://github.com/Microsoft/vscode>
  - Lásd: *Editing JSON with Visual Studio Code*  
<https://code.visualstudio.com/docs/languages/json>

## Szerkesztők (2)

Nem szabad szoftverek:

- <oXygen/> XML Editor (platform: Linux, macOS, Windows)  
<https://www.oxygenxml.com/>
  - Lásd: [https://www.oxygenxml.com/xml\\_editor/json\\_editor.html](https://www.oxygenxml.com/xml_editor/json_editor.html)
- IntelliJ IDEA (platform: Linux, macOS, Windows)  
<https://www.jetbrains.com/idea/>
  - Lásd: <https://www.jetbrains.com/help/idea/json.html>

# JSON formázás

- JSON szöveg formázása a parancssorban:

```
python -m json.tool <file>
```

```
cat <file> | python -m json.tool
```

```
curl <url> | python -m json.tool
```

- Használat:

```
python -m json.tool --help
```

- Lásd: <https://docs.python.org/3/library/json.html#module-json.tool>

# Programkönyvtárak

Lásd: <https://www.json.org/>

- C++:
  - nlohmann/json (licenc: *MIT License*) <https://json.nlohmann.me/>  
<https://github.com/nlohmann/json>
  - RapidJSON (licenc: *MIT License*) <http://rapidjson.org/>  
<https://github.com/Tencent/rapidjson>
- Java:
  - Gson (licenc: *Apache License 2.0*) <https://github.com/google/gson>
  - Jackson (licenc: *Apache License 2.0*)  
<https://github.com/FasterXML/jackson>
- Python: a standard könyvtár része a json modul  
<https://docs.python.org/3/library/json.html>

# JSON küldése HTTP kérésekben: HTTPie (1)

Parancssori HTTP kliens.

- Webhely: <https://httpie.io/>
- Tároló: <https://github.com/httpie/cli>
- Programozási nyelv: Python
- Platform: Linux, macOS, Windows
- Licenc: *New BSD License*

JSON támogatás: <https://httpie.io/docs/cli/json>

## JSON küldése HTTP kérésekben: HTTPie (2)

Példa: [GitHub Gist](#) létrehozása a parancssorban

```
http https://api.github.com/gists \  
  public:=true \  
  description="Hello, World!" \  
  files:='{ "hello.txt": { "content": "Hello, World!" } }' \  
  -a <username> \  
  -v
```

Lásd: <https://docs.github.com/en/rest/gists>

# JSON kiterjesztések

- JSON5 <https://json5.org/> <https://github.com/json5/json5>
  - A JSON egy javasolt kiterjesztése, mely azt az emberek számára könnyebben olvashatóvá és írhatóvá teszi.
    - Például megjegyzések használata, tagok neveinél elhagyhatóak a határoló karakterek.
  - A JSON superhalmaza, azonban nem vezet be új adattípusokat.
  - Specifikáció: *The JSON5 Data Interchange Format*  
<https://spec.json5.org/>
- YAML: YAML Ain't Markup Language <https://yaml.org/>
  - A JSON superhalmaza.
  - Jobban olvasható az emberi szem számára, mint a JSON, és számos újdotságot is bevezet.
  - Specifikáció: *YAML Ain't Markup Language (YAML) Version 1.2*  
<https://yaml.org/spec/1.2.2/>

# BSON (“Binary JSON”)

- Bináris adatsere formátum.
- A MongoDB NoSQL adatbázis-kezelő rendszer használja:  
<https://www.mongodb.org/>
  - Adattároláshoz és hálózati adatátvitelhez is alkalmazásra kerül.
- Specifikáció: <https://bsonspec.org/>
- Kiterjeszti a JSON adattípusait.
  - Például időbélyeg, reguláris kifejezés.
  - Nincs azonban `number` adattípus, helyette az `int32`, `int64` és `double` adattípusok használata.

# JSON Lines (1)

- Sor-orientált formátum soronként egy JSON érték tárolásához.
- Újsor-elválasztott JSON (*newline-delimited JSON*) néven is ismert.
- Webhely: <https://jsonlines.org/>
- Állománynév végződés: `.jsonl`
- Lásd még: [https://en.wikipedia.org/wiki/JSON\\_streaming](https://en.wikipedia.org/wiki/JSON_streaming)

# JSON Lines (2)

## Példák:

- ```
["title", "year", "standalone"]  
["Batman Begins", 2005, false]  
["The Dark Knight", 2008, false]  
["The Dark Knight Rises", 2012, false]  
["Interstellar", 2014, true]
```
- ```
{ "title": "Batman Begins", "year": 2005, "standalone": false }  
{ "title": "The Dark Knight", "year": 2008, "standalone": false }  
{ "title": "The Dark Knight Rises", "year": 2012, "standalone": false }  
{ "title": "Interstellar", "year": 2014, "standalone": true }
```

## JSON Lines (3)

Használat: [https://jsonlines.org/on\\_the\\_web/](https://jsonlines.org/on_the_web/)

# JSON Lines (4)

## Implementációk:

- Java:

- Gson (licenc: *Apache License 2.0*) <https://github.com/google/gson>
  - A `com.google.gson.JsonStreamParser` osztály használható JSON Lines formátumú adatok beolvasásához.
- Jackson jr (licenc: *Apache License 2.0*)  
<https://github.com/FasterXML/jackson-jr>

- Python:

- jsonlines (licenc: *New BSD License*) <https://jsonlines.readthedocs.io/>  
<https://github.com/wbolster/jsonlines>

# Alkalmazások (1)

- Ajax (*Asynchronous JavaScript and XML*)
  - Lásd: <https://developer.mozilla.org/en-US/docs/Web/Guide/AJAX>

## Alkalmazások (2)

Adatcsere és -tárolás:

- Firefox: könyvjelzők exportálása és importálása
  - Lásd: *Restore bookmarks from backup or move them to another computer* <https://support.mozilla.org/en-US/kb/restore-bookmarks-from-backup-or-move-them>

# Alkalmazások (3)

## Adatcsere és -tárolás (folytatás):

- GeoJSON <https://geojson.org/>
  - Formátum földrajzi adatszerkezetek ábrázolásához.
  - Specifikáció: Howard Butler, Martin Daly, Allan Doyle, Stefan Hagen, Tim Schaub. *RFC 7946: The GeoJSON Format*. August 2016.  
<https://www.rfc-editor.org/rfc/rfc7946>
- JSON-LD <https://json-ld.org/>
  - Pehelysúlyú szintaxis kapcsolt adatok (*Linked Data*) JSON-ba történő sorosításához.
  - Specifikáció: *JSON-LD 1.1: A JSON-based Serialization for Linked Data* (W3C Recommendation, 16 July 2020)  
<https://www.w3.org/TR/json-ld/>

# Alkalmazások (4)

## Konfigurációs adatok tárolása:

- `package.json`: <http://package.json.is/>  
<https://docs.npmjs.com/files/package.json>
  - npm <https://www.npmjs.com/> <https://github.com/npm/cli>
  - Grunt <https://gruntjs.com/> <https://github.com/gruntjs/grunt>
    - Lásd: <https://gruntjs.com/getting-started#package.json>
  - Visual Studio Code <https://code.visualstudio.com/>  
<https://github.com/microsoft/vscode>
    - Lásd: <https://code.visualstudio.com/api/references/extension-manifest>

# Alkalmazások (5)

Konfigurációs adatok tárolása (folytatás):

- Visual Studio Code (`settings.json`)
  - Lásd: <https://code.visualstudio.com/docs/getstarted/settings>
- WebExtensions (`manifest.json`) <https://developer.mozilla.org/en-US/docs/Mozilla/Add-ons/WebExtensions>
  - Lásd: <https://developer.mozilla.org/en-US/docs/Mozilla/Add-ons/WebExtensions/manifest.json>

# Alkalmazások (6)

## Webszolgáltatások:

- Meta Developer Documentation  
<https://developers.facebook.com/docs/>
- Flickr API <https://www.flickr.com/services/api/>
- GitHub REST API <https://docs.github.com/en/rest/>
- Nominatim API  
<https://nominatim.org/release-docs/develop/api/Overview/>

## Alkalmazások (7)

NoSQL adatbázisok: számos dokumentum-orientált adatbázis használja adattárolásra a JSON-t.

- Apache CouchDB (programozási nyelv: Erlang; platform: Linux, macOS, Windows; licenc: *Apache License 2.0*)  
<https://couchdb.apache.org/> <https://github.com/apache/couchdb>
- EJDB2 (programozási nyelv: C; platform: Android, iOS, Linux, macOS, Windows; licenc: *MIT License*) <https://ejdb.org/>  
<https://github.com/Softmotions/ejdb>
- RethinkDB (programozási nyelv: C++; platform: Linux, macOS; licenc: *Apache License 2.0*) <https://rethinkdb.com/>  
<https://github.com/rethinkdb/rethinkdb>
- UnQLite (programozási nyelv: C; platform: Linux, macOS, Windows; licenc: *Simplified BSD License*) <https://unqlite.symisc.net/>  
<https://github.com/symisc/unqlite>

# JSON Schema (1)

- JSON-alapú sémanyelv JSON dokumentumok érvényesítéséhez.
- Webhely: <https://json-schema.org/>
- Aktuális verzió: 2020-12
- A legszélesebb körben támogatott verzió a 2018-ban kiadott draft-07 verzió.

## JSON Schema (2)

### Specifikációk:

- Austin Wright (ed.), Henry Andrews (ed.), Ben Hutton (ed.), Greg Dennis. *JSON Schema: A Media Type for Describing JSON Documents*. June 16, 2022.  
<https://json-schema.org/draft/2020-12/json-schema-core>
- Austin Wright (ed.), Henry Andrews (ed.), Ben Hutton (ed.). *JSON Schema Validation: A Vocabulary for Structural Validation of JSON*. June 16, 2022.  
<https://json-schema.org/draft/2020-12/json-schema-validation>
- Geraint Luff, Henry Andrews (ed.), Ben Hutton (ed.). *Relative JSON Pointers*. January 28, 2020.  
<https://datatracker.ietf.org/doc/html/draft-bhutton-relative-json-pointer-00>

# JSON Schema (3)

Kapcsolódó specifikáció:

- Paul C. Bryan (ed.), Kris Zyp, Mark Nottingham (ed.), *RFC 6901: JavaScript Object Notation (JSON) Pointer*. April 2013.  
<https://www.rfc-editor.org/rfc/rfc6901>
  - Egy szintaxist határoz meg JSON dokumentumokon belüli értékek azonosításához.
  - Példák:
    - `/country`
    - `/places/0`
    - `/places/0/longitude`

## JSON Schema (4)

- **JSON dokumentum:** az `application/json` média típus által leírt információ erőforrás, azaz egy JSON érték.
- **Példány (*instance*):** egy olyan JSON dokumentum, melyre egy séma vonatkozik.
- **JSON séma:** egy olyan JSON dokumentum, mely példányokat ír le.
  - Egy objektum vagy egy logikai érték.
  - Sémák egymásba ágyazhatók.
    - A legkülső sémát **gyökér sémának** (*root schema*) nevezzük, a többi **alsémának** (*subschema*).
  - Média típus: `application/schema+json`

## JSON Schema (5)

- **Tulajdonság** (*property*): egy objektum példány egy tagja.
- **Kulcsszó** (*keyword*): egy séma objektum egy példányra vonatkozó tulajdonsága.
  - A kulcsszavak JSON példányokra vonatkozó megszorításokat fejeznek ki vagy további információkkal annotálják a példányokat.
  - Például: "properties", "type", "\$ref"

## JSON Schema (6)

- **Szótár (*vocabulary*)**: adott célra szolgáló kulcsszavak egy halmaza a szintaxisukkal és jelentésükkel együtt.
  - A szótárat alkotó kulcsszavaknak jól meghatározott szintaxisa és jelentése van.
  - Az alábbi kapcsolódó specifikációk egy-egy szótárat definiálnak:
    - Austin Wright (ed.), Henry Andrews (ed.), Ben Hutton (ed.), Greg Dennis. *JSON Schema: A Media Type for Describing JSON Documents*. June 16, 2022.  
<https://json-schema.org/draft/2020-12/json-schema-core>
    - Austin Wright (ed.), Henry Andrews (ed.), Ben Hutton (ed.). *JSON Schema Validation: A Vocabulary for Structural Validation of JSON*. June 16, 2022.  
<https://json-schema.org/draft/2020-12/json-schema-validation>

# JSON Schema (7)

- **Meta-séma:** egy sémát leíró séma.
  - Példa: JSON Schema meta-séma:
    - <https://json-schema.org/draft/2020-12/schema>
    - <http://json-schema.org/draft-07/schema>

# JSON Schema (8)

Eszközök:

- Szabad és nyílt forrású szoftverek:
  - Visual Studio Code (platform: Linux, macOS, Windows; licenc: *MIT License*) <https://code.visualstudio.com/>  
<https://github.com/Microsoft/vscode>
    - Lásd: *Editing JSON with Visual Studio Code*  
<https://code.visualstudio.com/docs/languages/json>
- Nem szabad szoftverek:
  - Oxygen XML Editor (platform: Linux, macOS, Windows)  
<https://www.oxygenxml.com/>
    - Lásd: *Editing JSON Schema Documents*  
<https://www.oxygenxml.com/doc/versions/26.1/ug-editor/topics/editing-JSON-schema.html>
  - IntelliJ IDEA (platform: Linux, macOS, Windows)  
<https://www.jetbrains.com/idea/>
    - Lásd: <https://www.jetbrains.com/help/idea/json.html>

# JSON Schema (9)

Implementációk: <https://json-schema.org/tools>

- C++:
  - json-schema-validator (licenc: *MIT License*)  
<https://github.com/pboettch/json-schema-validator>
- Java:
  - json-schema (licenc: *Apache License 2.0*)  
<https://github.com/everit-org/json-schema>
- JavaScript:
  - Ajv (licenc: *MIT License*) <https://ajv.js.org/>  
<https://github.com/ajv-validator/ajv>

# JSON Schema (10)

Implementációk: <https://json-schema.org/tools>

- .NET:

- Json.NET Schema (licenc: AGPLv3)

- <http://www.newtonsoft.com/jsonschema>

- <https://github.com/JamesNK/Newtonsoft.Json.Schema>

- Webes interfész: <https://www.jsonschemavalidator.net/>

- Python:

- jschon (licenc: *MIT License*) <https://jschon.readthedocs.io/en/latest/>

- <https://github.com/marksparkza/jschon>

- Webes interfész: <https://jschon.dev/>

- jsonschema (licenc: *MIT License*)

- <https://python-jsonschema.readthedocs.io/>

- <https://github.com/python-jsonschema/jsonschema>

# JSON Schema (11)

JSON Schema Store <https://www.schemastore.org/json/>  
<https://github.com/SchemaStore/schemastore>

- Közismert JSON állományformátumok sémáinak gyűjteménye.
- Szerkesztő támogatás: IntelliJ IDEA, Microsoft Visual Studio, ...

# JSON Schema (12)

JSON sémák társítása JSON dokumentumokhoz:

- A JSON séma specifikációk nem biztosítanak erre szolgáló dokumentum szintű lehetőséget.
- Implementáció-specifikus megoldások:
  - <code>oXygen</code> XML Editor:  
<https://www.oxygenxml.com/doc/versions/26.1/ug-editor/topics/json-associating-schema-directly-in-doc.html>
  - Visual Studio Code:  
[https://code.visualstudio.com/docs/languages/json#\\_json-schemas-and-settings](https://code.visualstudio.com/docs/languages/json#_json-schemas-and-settings)

# JSON Schema (13)

További ajánlott irodalom:

- Michael Droettboom, *Understanding JSON Schema*.  
<https://json-schema.org/understanding-json-schema/>  
<https://github.com/json-schema-org/understanding-json-schema>

# JSON Schema példák (1)

- Séma:
    - `true`
    - `{}`
  - Érvényes példányok: bármely JSON érték
  - Nem érvényes példányok: nincsenek
- Séma:
    - `false`
    - `{ "not": {} }`
  - Érvényes példányok: nincsenek
  - Nem érvényes példányok: bármely JSON érték

## JSON Schema példák (2)

Séma:

```
{  
  "type": "string"  
}
```

Érvényes példányok:

- ""
- "Hello, World!\n"

## JSON Schema példák (3)

Séma:

```
{  
  "type": ["string", "null"]  
}
```

Érvényes példányok:

- ""
- "Hello, World!\n"
- null

## JSON Schema példák (4)

Séma:

```
{  
  "type": "array",  
  "items": { "type": "string" }  
}
```

Érvényes példányok:

- []
- ["Hello, World!\n"]
- ["Sun", "Mon", "Tue", "Wed", "Thu", "Fri", "Sat"]

## JSON Schema példák (5)

Séma:

```
{  
  "type": "array",  
  "items": { "type": "string" },  
  "minItems": 1,  
  "uniqueItems": true  
}
```

Érvényes példányok:

- ["sci-fi"]
- ["crime", "drama"]

Nem érvényes példányok:

- []
- ["sci-fi", "sci-fi"]

## JSON Schema példák (6)

Séma:

```
{  
  "$schema": "https://json-schema.org/draft/2020-12/schema",  
  "type": "object",  
  "properties": {  
    "name": { "type": "string" },  
    "age": { "type": "integer" },  
    "email": { "type": "string" },  
    "webpage": { "type": "string" }  
  },  
  "required": ["name", "age", "email"]  
}
```

## JSON Schema példák (7)

Érvényes példány (folytatás):

```
{  
  "name": "Douglas Crockford",  
  "age": 69,  
  "email": "douglas@crockford.com",  
  "webpage": "https://www.crockford.com/"  
}
```

## JSON Schema példák (8)

Séma (folytatás):

```
{
  "$schema": "https://json-schema.org/draft/2020-12/schema",
  "type": "object",
  "properties": {
    "name": { "type": "string" },
    "age": { "type": "integer", "minimum": 0 },
    "email": { "type": "string", "format": "email" },
    "webpage": { "type": "string", "format": "uri",
      "pattern": "^http(s)?://.*"
    }
  },
  "required": ["name", "age", "email"]
}
```

# JSON Schema példák (9)

Séma:

```
{
  "$schema": "https://json-schema.org/draft/2020-12/schema",
  "type": "object",
  "properties": {
    "title": { "type": "string" },
    "year": { "type": "integer" },
    "rating": { "type": "number", "minimum": 0, "maximum": 10 },
    "votes": { "type": "integer", "minimum": 0 },
    "genres": {
      "type": "array",
      "items": { "type": "string" },
      "minItems": 1,
      "uniqueItems": true
    },
    "url": { "type": "string", "format": "uri" }
  },
  "required": ["title", "year", "rating", "votes", "genres", "url"],
  "additionalProperties": false
}
```

# JSON Schema példák (10)

## Séma:

```
{
  "$schema": "https://json-schema.org/draft/2020-12/schema",
  "type": "object",
  "properties": {
    "artist": { "type": "string" },
    "title": { "type": "string" },
    "year": { "type": "integer" },
    "tracks": {
      "type": "array",
      "items": {
        "type": "object",
        "properties": {
          "title": { "type": "string" },
          "length": { "type": "integer", "minimum": 0 }
        },
        "required": ["title", "length"],
        "additionalProperties": false
      },
      "minItems": 1
    }
  },
  "required": ["artist", "title", "year", "tracks"],
  "additionalProperties": false
}
```

# JSON Schema példák (11)

Séma:

```
{
  "$schema": "http://json-schema.org/2020-12/schema",
  "type": "object",
  "patternProperties": {
    "^[a-z]{2}$": { "type": "string" }
  },
  "additionalProperties": false,
  "minProperties": 2
}
```

Érvényes példány:

```
{
  "en": "Lord of the Rings",
  "de": "Der Herr der Ringe",
  "hu": "A gyűrűk ura"
}
```

# JSON Schema példák (12)

Séma:

```
{
  "$schema": "https://json-schema.org/draft/2020-12/schema",
  "type": "object",
  "properties": {
    "name": { "type": "string" },
    "born": { "$ref": "#/$defs/event" },
    "died": { "$ref": "#/$defs/event" },
    "gender": { "enum": ["female", "male"] }
  },
  "required": ["name", "born", "gender"],
  "$defs": {
    "event": {
      "type": "object",
      "properties": {
        "date": { "type": "string" },
        "place": { "type": "string" }
      }
    }
  }
}
```

## JSON Schema példa (13)

Érvényes példány (folytatás):

```
{
  "name": "Edgar Allan Poe",
  "born": {
    "date": "1809-01-19",
    "place": "Boston, Massachusetts, United States"
  },
  "died": {
    "date": "1849-10-07",
    "place": "Baltimore, Maryland, United States"
  },
  "gender": "male"
}
```

# JSON Schema példák (14)

Séma:

```
{
  "$schema": "https://json-schema.org/draft/2020-12/schema",
  "type": "object",
  "properties": {
    "name": { "type": "string" },
    "born": { "$ref": "#/$defs/event" },
    "died": { "$ref": "#/$defs/event" },
    "gender": { "enum": ["female", "male"] }
  },
  "required": ["name", "born", "gender"],
  "$defs": {
    "event": {
      "type": "object",
      "properties": {
        "date": { "type": "string" },
        "place": { "type": "string" }
      },
      "anyOf": [
        { "required": ["date"] },
        { "required": ["place"] }
      ]
    }
  }
}
```

# JSON Schema példák (15)

```
{
  "$schema": "https://json-schema.org/draft/2020-12/schema",
  "type": "object",
  "properties": {
    "name": { "type": "string" },
    "born": { "$ref": "#/$defs/event" },
    "died": {
      "allOf": [
        { "$ref": "#/$defs/event" },
        {
          "properties": {
            "cause": { "type": "string" }
          },
          "required": ["cause"]
        }
      ]
    },
    "gender": { "enum": ["female", "male"] }
  },
  "required": ["name", "born", "gender"],
  "$defs": {
    "event": {
      "type": "object",
      "properties": {
        "date": { "type": "string" },
        "place": { "type": "string" }
      }
    }
  }
}
```

## JSON Schema példák (16)

Érvényes példány (folytatás):

```
{
  "name": "John F. Kennedy",
  "born": {
    "date": "1917-05-29",
    "place": "Brookline, Massachusetts, United States"
  },
  "died": {
    "date": "1963-11-22",
    "place": "Dallas, Texas, United States",
    "cause": "assassination"
  },
  "gender": "male"
}
```

# JSON Schema példák (17)

## Séma:

```
{
  "$schema": "https://json-schema.org/draft/2020-12/schema",
  "type": "object",
  "properties": {
    "type": { "enum": ["book", "serial"] },
    "title": { "type": "string" },
    "publisher": { "type": "string" }
  },
  "required": ["type", "title", "publisher"],
  "if": {
    "properties": { "type": { "const": "book" } }
  },
  "then": {
    "properties": {
      "isbn": { "type": "string", "pattern": "^([0-9]{13})$" }
    },
    "required": ["isbn"]
  },
  "else": {
    "properties": {
      "issn": { "type": "string", "pattern": "^([0-9]{4}-[0-9]{3}[0-9X])$" }
    },
    "required": ["issn"]
  }
}
```

## JSON Schema példák (18)

Érvényes példányok:

- ```
{  
  "type": "book",  
  "title": "The Hound of the Baskervilles",  
  "publisher": "Penguin Books",  
  "isbn": "9780241952870"  
}
```
- ```
{  
  "type": "serial",  
  "title": "IEEE Internet of Things Journal",  
  "publisher": "IEEE",  
  "issn": "2327-4662"  
}
```

# JSON Schema példák (19)

Séma: *tuple validation*

```
{
  "$schema": "https://json-schema.org/draft/2020-12/schema",
  "type": "array",
  "prefixItems": [
    {
      "type": "number",
      "minimum": -90,
      "maximum": 90
    },
    {
      "type": "number",
      "minimum": -180,
      "maximum": 180
    }
  ],
  "items": false
}
```

# JSON Schema példák (20)

Valós példák:

- GeoJSON: <https://json.schemastore.org/geojson>
- `manifest.json` (WebExtensions):  
<https://json.schemastore.org/webextension>
- `manifest.json` (Chromium, Google Chrome):  
<https://json.schemastore.org/chrome-manifest>
- `package.json` (npm): <https://json.schemastore.org/package>
- SWAPI – The Star Wars API <https://swapi.dev/>
  - <https://swapi.dev/api/planets/schema>
  - <https://swapi.dev/api/species/schema>
  - <https://swapi.dev/api/starships/schema>

# JSON Schema alkalmazások

Webes felhasználói felületek generálása JSON sémából:

- Alpaca (programozási nyelv: JavaScript; licenc: *Apache License 2.0*)  
<http://www.alpaca.js.org/> <https://github.com/gitana/alpaca>
- jsonforms (programozási nyelv: TypeScript; licenc: *MIT License*)  
<https://jsonforms.io/> <https://github.com/eclipsesource/jsonforms>
- react-jsonschema-form (programozási nyelv: JavaScript; licenc: *Apache License 2.0*)  
<https://rjsf-team.github.io/react-jsonschema-form/>  
<https://github.com/rjsf-team/react-jsonschema-form>

Lásd: <https://json-schema.org/tools?query=&sortBy=name&sortOrder=ascending&groupBy=toolingTypes&licenses=&languages=&drafts=&toolingTypes=schema-to-web-ui>

# XML-JSON konverzió (1)

- <oxygen/> XML Editor <https://www.oxygenxml.com/>
  - Tools >> JSON to XML...
  - Tools >> XML to JSON...
- Visual Studio Code
  - Bővítmény: XML to JSON (licenc: *MIT License*) <https://marketplace.visualstudio.com/items?itemName=buianhthang.xml2json>  
<https://github.com/anhthang/vscode-xml2json>

## XML-JSON konverzió (2)

- JSON-java (programozási nyelv: Java; licenc: *JSON License*)  
<https://github.com/stleary/JSON-java>
  - Az XML osztály `JSONObject()` metódusa egy XML dokumentumot egy ekvivalens JSON objektummá alakít.
- xml-js (programozási nyelv: JavaScript; licenc: *MIT License*)  
<https://github.com/nashwaan/xml-js>

## XML-JSON konverzió (3)

- JsonML (JSON Markup Language) <http://www.jsonml.org/>  
<https://github.com/mckamey/jsonml>
  - Célja XML dokumentumok veszteségmentes ábrázolása JSON szintaxissal.
  - Szintaxis: <http://www.jsonml.org/syntax/>

# Lekérdező nyelvek (1)

- XPath, XQuery: JSON támogatás a legutóbbi, 3.1 számú verzióban jelent meg.
  - További információk: <https://www.w3.org/XML/Query/>

## Lekérdező nyelvek (2)

Számos egyedi megoldás létezik, számomra ezek tűnnek ígéretesnek:

- JSONiq <https://www.jsoniq.org/>
  - Az XQuery-n alapuló deklaratív funkcionális nyelv (JSON lekérdezés és feldolgozás).
  - Implementációk:
    - RumbleDB (programozási nyelv: Java; licenc: *Apache License 2.0*)  
<https://rumbledb.org/> <https://github.com/RumbleDB/rumble>
- JSONata (licenc: *MIT License*) <https://jsonata.org/>  
<https://github.com/jsonata-js/jsonata>
  - Az XPath 3.1 elérési útvonalak szemantikája által inspirált lekérdező és transzformációs nyelv.
  - JavaScript-ben írt referencia implementáció.

## Lekérdező nyelvek (3)

Számos egyedi megoldás létezik, számomra ezek tűnnek ígéretesnek:  
(folytatás)

- jq (programozási nyelv: C; platform: Linux, macOS, Windows; licenc: *Expat License*) <https://jqlang.github.io/jq/>  
<https://github.com/jqlang/jq>
  - Parancssori JSON feldolgozó.
- JMESPath <https://jmespath.org/> <https://github.com/jmespath>
  - Implementációk: Go, Java, JavaScript, Lua, .NET, PHP, Python, Ruby, Rust (licenc: *MIT License*)
- ObjectPath (programozási nyelv: Python; licenc: *MIT License*)  
<http://objectpath.org/> <https://github.com/adriank/ObjectPath>

# Köszönetnyilvánítás

Köszönet Szathmáry Lászlónak a hasznos észrevételekért és a jq-ért.