
JSON

JavaScript Object Notation

JSON

- JSON (JavaScript Object Notation). És un “**format de notació de dades**” lleuger utilitzat tant per a **guardar informació** com per a l'**intercanvi de dades** entre aplicacions.
- No es defineix com llenguatge de marques ja que no usa etiquetes com HTML o XML
- Els arxius JSON són arxius de text i usen l'extensió **.json**.
- La seua principal característica és que és un llenguatge **autodesdriptiu** i **fàcil de llegir i d'escriure** tant per les màquines com per els humans.
- És una alternativa més simple i lleugera a XML que compta amb funcions similars.
- JSON és un **estàndard** ampliament acceptat en la **comunicació** entre **clients i servidors**.
- La majoria dels llenguatges de programació tenen suport integrat per a la manipulació de JSON.
- És una eina versàtil per a l'emmagatzematge de dades i la configuració d'aplicacions.

1.- Sintaxi JSON

JSON utilitza una estructura basada en parelles **clau-valor**. (keys - values)

- Les **claus (keys)** són cadenes de caràcters (**strings**) envoltats entre cometes “ ”.
- Els **valors (values)** són un tipus de dades **JSON** vàlids.
- Les dades es separen per comes.

```
{ "key": "value", "key1": "value", "key2": "value" }
```

En JSON els valors admeten els següents **tipus de dades**:

- **Cadenes de Text (Strings)** es representen amb cometes dobles, com “**aquesta és una cadena**”. Poden contenir qualsevol caràcter **Unicode**.
- **Números: JSON** admet nombres enters i decimals, sense notació especial. **42** o **3.14**.
- **Booleans i Null: JSON** admet valors booleans, representats com **true** i **false**. Per a representar un valor nul utilitzem **null**.

```
{  
  "nom": "Aitor Tilla",  
  "edat": 26,  
  "estudiant": true,  
  "ciutat": "La Vall d'Uixó",  
  "puntuació": 5.5,  
  "comentari": "Aquest és un exemple de JSON amb diverses dades.",  
  "nulExemple": null  
}
```

En JSON les dades s'organitzen en **OBJECTES** i **ARRAYS**.

- Les claus (`{ }`) representen **objectes**.
- Els claudàtors (`[]`) representen **arrays**.

```
{ "animals": [  
  { "nom": "Dalila", "tipus": "gos" },  
  { "nom": "Monchito", "tipus": "gat" }  
]}
```

2.- Objectes JSON

Un objecte JSON és una col·lecció de parelles clau-valor, on les claus són cadenes de text i els valors poden ser de qualsevol tipus de dades suportat per JSON.

```
{ "nom": "Aitor Tilla", "edat": 26, "moto": null }
```

Objecte JSON que representa un **llibre**:

```
{  
  "títol": "El Gran Gatsby",  
  "autor": "F. Scott Fitzgerald",  
  "anyPublicació": 1925,  
  "editorial": "Scribner",  
  "disponible": true,  
  "crítiques": [4.5, 5.0, 4.2]  
}
```

- Delimitats per `{}`.
- S'organitzen en parells **clau/valor**.
- Tipus de dades
 - **Claus:** string
 - **Valors:** string, number, object, array, boolean o null.
- **Claus i valors:** Separats per dos punts
- **Parells clau/valor:** Separats per comes.

Com es mostra a l'exemple, les claus ("títol", "autor", "anyPublicació", etc.) són **cadena de text**. Els **valors** poden ser cadenes de text, nombres, booleans o fins i tot **un array** (com "crítiques") que conté valors numèrics.

3.- Arrays JSON

Un Array **JSON** és una llista ordenada de valors.

```
[ "Ford", "BMW", "Fiat" ]
```

Exemple d'Array que conté una llista de tasques:

```
{  
  "tasques": ["Comprar llet", "Anar al gimnàs", "Estudiar per l'examen"]  
}
```

- Delimitats per [].
- Molt semblants als arrays en Java o Javascript
- Elements separats per comes.
- Tipus de dades dels elements: **string, number, object, array, boolean o null.**

En este exemple l'**array** està contingut **dins de l'objecte com el valor associat a la clau "tasques"**. L'array conté tres cadenes de text que representen tasques diferents.

4.- Combinacions d'Objectes i Arrays

Les combinacions de dades en **JSON** són molt variades i flexibles, ja que podem anidar objectes dins d'objectes, arrays dins d'arrays i combinar-los segons les nostres necessitats.

Algunes combinacions son:

- **Objectes simples:**

Un objecte JSON amb múltiples parelles clau-valor.

```
{  
  "nom": "Jaume",  
  "edat": 18,  
  "ciutat": "La Vall d'Uixó"  
}
```

- **Arrays simples:**

Un array JSON amb una llista d'elements de qualsevol tipus.

```
[ "blau", "verd", "groc" ]
```

- **Objectes dins d'objectes:**

Nidificació d'objectes, on un objecte conté d'altres objectes com valors.

```
{
  "persona": {
    "nom": "Veronica",
    "edat": 28
  },
  "adreça": {
    "carrer": "Carrer Gran, 123",
    "ciutat": "Fodeguilla"
  }
}
```

- **Arrays dins d'arrays:**

Nidificació d'arrays, on un array conté altres arrays com elements.

```
[
  [1, 2, 3],
  ["a", "b", "c"]
]
```

- **Objectes amb arrays:**

Un objecte que conté un array com un dels seus valors.

```
{
  "noms": ["Gerard", "Jordi"],
  "punts": [8.5, 7.0]
}
```

- **Arrays d'objectes:**

Un array que conté una llista d'objectes com elements.

```
[
  { "nom": "Vicent", "edat": 25 },
  { "nom": "Carles", "edat": 29 }
]
```

- **Arrays amb tipus de dades mixtes:**

Un array que conté elements de diferents tipus (nombres, cadenes, objectes, booleans, etc.).

```
[42, "groc", true, {"objecte": "aninat"}]
```

- **Objectes amb tipus de dades mixtes:**

Un objecte que combina diferents tipus de valors en les seves parelles clau-valor.

```
{
  "nom": "MariCarmen",
  "edat": 30,
  "actiu": true,
  "aficions": ["lectura", "caminar"]
}
```

- **Arrays dins d'objectes i viceversa:**

Objectes que contenen arrays, i arrays que contenen objectes.

```
{
  "persones": [
    { "nom": "Andrei", "edat": 35 },
    { "nom": "Esther", "edat": 28 }
  ],
  "colors": ["verd", "blau"]
}
```

- **Objectes buits i arrays buits:**

Objectes sense cap parella clau-valor i arrays sense elements.

```
{
  "informacio": {},
  "interessos": [],
  "historial_compres": []
}
```

Nota: Els objectes arrays buits són **placeholders (marcador de posició)**, s'utilitza per reservar un espai o ubicació futures de dades o continguts.

- **Valors nuls i valors booleans:**

La combinació de valors nuls i valors booleans dins d'objectes o arrays.

```
{
  "valor1": null,
  "valor2": true,
  "valor3": false
}
```

- **Arrays amb elements anidats:**

Elements de l'array que són sí mateixos objectes o arrays.

```
[  
  [1, 2, 3], ["a", "b", "c", [true, false]]  
]
```

o també

```
[  
  "Texto",  
  42,  
  ["a", "b", "c", [1, 2, 3]],  
  {  
    "nombre": "Lorena",  
    "edad": 30  
  }  
]
```

- **Objectes amb objectes i arrays amb arrays anidats:**

Nidificació de múltiples nivells de dades.

```
{  
  "dades": {  
    "noms": ["Nereida", "Juan"],  
    "punts": [9.0, 8.5],  
    "metadades": {  
      "creador": "admin",  
      "dataCreació": "1812-03-19"  
    },  
    "valors": [  
      [1, 2],  
      [3, 4]  
    ]  
  }  
}
```

5.- Consideracions i Errors comuns en JSON

- **Case-Sensitive**

Distingeix entre majúscules i minúscules

```
{  
  "nom": "Fina",  
  "Nom": "Pepe"  
}
```

- **Format per un únic element (Array o Objecte)**

Incorrecte

Incorrecte	Correcte
<code>[1, 2, 3], ["a", "b", "c"]</code>	<code>[[1, 2, 3], ["a", "b", "c"]]</code>
<code>"nom": "Sergi", "edat": 21</code>	<code>{ "nom": "Sergi", "edat": 21 }</code>

- **En Arrays i Objectes l'últim element no pot anar seguit de coma (,)**

```
{ nom: "Sergi", "edat": 21, }  
  
0  
["blau", "verd", "groc", ]
```

- **No permet Claus Duplicades**

No permet claus duplicades en un objecte

```
{  
  "color": "vermell",  
  "color": "blau"  
}
```

Tampoc en objectes amb objectes i claus repetides

```
{  
  "usuari": {  
    "nom": "Jaume",  
    "email": "j.aragovalles@edu.gva.es"  
  },  
  "usuari": {  
    "nom": "MariCarmen",  
    "email": "m.gomezcana@edu.gva.es"  
  }  
}
```

- **Json no admet comentaris**

6.- Utilització caràcters especials i de Control

En JSON es poden usar caràcters especials, de control i caràcters Unicode escapats adequadament. Per exemple:

```
{
  "missatge": "Aquest és un exemple amb caràcters especials: \"cometes dobles\". També
conté línies noves:\nLínia 1\nLínia 2",
  "barra_invertida": "Això és una barra invertida escapada: \\",
  "tabulacio": "Això és una tabulació: \tTabulació 1\tTabulació 2",
  "simbol": "Símbol de copyright © (c) representat com Unicode: \\u00A9",
  "caracter_control": "Això és un caràcter de control representat com Unicode: \\u20AC"
}
```

El caràcter `\\u20AC` representa el símbol €

Nota: Poden trobar molts validadors de JSON online, com https://jsonlint.com/
--

Exemple JSON

```
{
  "estudiant": {
    "nom": "Armando Bronca Segura",
    "edat": 20,
    "matriculat": true,
    "assignatures": ["Llenguatge de Marques", "Accés a Dades", "Programació Web"],
    "notes": {
      "llenguatge_de_marques": {
        "parcial_1": 9.5,
        "parcial_2": 8.7
      },
      "accés_a_dades": {
        "parcial_1": 8.0,
        "parcial_2": 9.2
      },
      "programacio_web": {
        "parcial_1": 7.8,
        "parcial_2": 8.9
      }
    }
  },
  "professor": {
    "nom": "Jaume Aragó",
    "especialitzacio": "Accés a Dades",
    "contacte": {
      "correu": "j.aragovalls@edu.gva.es",
      "telefon": "+34 555 555 555"
    }
  },
  "institut": "IES Benigasló",
  "curs_academic": "1936-1937",
  "comentaris": [
    {
      "data": "1936-10-15",
      "text": "Armando ha fet bons progressos en el curs fins ara."
    },
    {
      "data": "1936-11-30",
      "text": "És important seguir treballant l'accés a dades."
    }
  ]
}
```

En este exemple de JSON trobem:

- **Objectes:**

→ **Objecte "estudiant":** Conté informació sobre l'estudiant, com el nom, l'edat, l'estat de matrícula, les assignatures i les notes en les assignatures. És l'objecte més gran i conté els següents atributs:

- "nom"
- "edat"
- "matriculat"
- "assignatures"
- "notes"

→ **Objecte "notes":** Objecte que es troba anidat dins de l'objecte "estudiant" i conté les notes per a cada assignatura. Conté els següents atributs:

- "llenguatge_de_marques"
- "bases_de_dades"
- "programacio_web"

→ **Objecte "professor":** Conté informació sobre el professor, com el nom, l'especialització i la informació de contacte. Atributs:

- "nom"
- "especialitzacio"
- "contacte"

→ **Objecte "contacte":** Este objecte el trobem anidat dins de l'objecte "professor" i conté la informació de contacte del professor. Conté els següents atributs:

- "correu"
- "telefon"

→ **Objecte "comentaris":** És un array d'objectes que conté comentaris sobre l'estudiant. Cada objecte de comentari conté la data i el text del comentari. Aquest objecte és l'últim dins del document JSON.

- **Arrays:**

→ **Array "assignatures":** El trobem dins de l'objecte "estudiant" i conté una llista de les assignatures que l'estudiant està cursant.

→ **Array "comentaris":** Aquest array està dins del document principal i conté una llista de comentaris, on cada element de l'array és un objecte que conté la data i el text d'un comentari.

En este exemple, trobem una jerarquia d'objectes i la presència d'arrays que permeten representar dades estructurades relacionades amb un estudiant, un professor, l'institut, el curs acadèmic i els comentaris sobre el rendiment acadèmic de l'estudiant.

Esta estructura proporciona una organització eficient de la informació en format JSON.

7.- JSON vs XML

JSON	XML
<pre>{ "empleats": [{ "nom": "Jaume", "cognoms": "Aragó" }, { "nom": "Isabel", "cognoms": "Sanahuja" }, { "nom": "Sergi", "cognoms": "Alemany" }]}</pre>	<pre><empleats> <empleat> <nom>Jaume</nom> <cognoms>Aragó</cognoms> </empleat> <empleat> <nom>Isabel</nom> <cognoms>Sanahuja</cognoms> </empleat> <empleat> <nom>Sergi</nom> <cognoms>Alemany</cognoms> </empleat> </empleats></pre>

→ **Similituds:**

- Son fàcils d'entendre per humans
- Són jeràrquics
 - Valors dins de valors
- Poden ser parsejats i utilitzats per molts **llenguatges de programació**.

→ **Diferències:**

- JSON és més curt
- JSON és més fàcil de llegir i escriure
- JSON Pot utilitzar arrays.
- XML suporta atributs
- Parseig:
 - XML és molt complex de parsejar
 - JSON pot ser parsejat de forma molt fàcil