

# JSON

---

JSON (Javascript Object Notation) és un format de text lleuger i humanament llegible per l'emmagatzematge i la distribució de dades estructurades. S'utilitza, principalment, per transmetre dades entre servidor i aplicacions web.



Si JSON és el format per la transmissió de dades, el format de processament de dades és l'objecte JavaScript (o **JS Object**). Ambdós són una col·lecció d'atributs que són parells clau-valor, on la clau és un string que identifica de manera única el valor associat. Aquests objectes són una característica fonamental del llenguatge de programació JavaScript i permeten representar i manipular dades complexes.

JavaScript permet la conversió entre aquests dos formats de forma eficaç i fiable.

## Sintaxi de les dades emmagatzemades en format JSON

```
{
  "cotxe": [
    {
      "matricula": "E-0198-LYX",
      "anyMatriculacio": 2022,
      "combustible": "diesel",
      "canviManual": false,
      "cv": 170,
      "companyia": {
        "marca": "opel",
        "model": "insignia"
      },
      "kilometres": null
    },
    {
      "matricula": "E-5213-DWL",
      "anyMatriculacio": 2006,
      "combustible": "gasolina",
      "canviManual": true,
      "cv": 140,
      "companyia": {
        "marca": "jeep",
        "model": "wrangler"
      },
      "kilometres": 70000
    }
  ]
}
```

Figura 1 : estructuració dades JSON

Els objectes estan guardats entre «{ }».

En aquest exemple s'hi defineix un array d'objectes cotxe amb dos elements. Indicat amb «[ ]».

Cada cotxe conté propietats: matricula, anyMatriculacio, combustible, etc. I cada cotxe conté un objecte companyia amb les seves propietats.

Cada propietat ha de tenir un valor. El valor pot ser un string, un número enter o decimal, un altre objecte, un array, un booleà o un null.

Les dades es separen amb «,». L'últim element de cada objecte no conté aquest separador.

## Com es recupera el contingut JSON?

El contingut JSON s'envia com una cadena de text. Per tal de manipular-lo, cal convertir-lo primer en objectes de Javascript, una de les maneres de fer-ho és amb la funció: `JSON.parse()`

Podem usar una funció com a segon paràmetre per tal de filtrar o alterar el contingut mentre fem la conversió de text a JSON:

```
const text = '{"name":"John", "birth":"1986-12-14", "city":"New York"}';
let obj = JSON.parse(text, function (key, value) {
  let date;
  if (key == "birth") {
    date = new Date(value);
  } else {
    date = value;
  }
  return date;
});
```

Aquí tots els camps són guardats normalment excepte el camp «birth», el contingut d'aquest camp es formateja com una data.

I com accedim al contingut que ens interessa un cop passat a objectes de JavaScript?

Per aquest exemple, utilitzarem de referència el codi JSON de la *Figura 1*.

Als elements que es defineixen a l'arrel s'hi pot accedir simplement indicant-ne el nom:

`cotxe`

En aquesta estructura l'objecte de l'arrel és un array, per accedir al primer objecte que emmagatzema l'array ho farem de la següent manera:

`cotxe[0]`

Accedim a la matrícula del segon objecte de l'array:

`cotxe[1].matricula`

Accedim a la marca del segon objecte de l'array, la marca es troba dins l'objecte companyia:

`cotxe[1].companyia.marca`

També es pot usar la sintaxi:

`cotxe[1].companyia["marca"]`

Si usem variables ho fem així:

`const nom = "marca";`

`cotxe[1].companyia[nom];`

D'aquesta manera podem obtenir i sobre escriure els valors.

## Com es creen dinàmicament els objectes JavaScript?

És necessari per automatitzar el procés de creació d'objectes JavaScript.

```
function introData (matricula, anyMatriculacio, marca) {  
    let companyia = {  
        marca  
    };  
    let cotxeIndividual = {  
        matricula,  
        anyMatriculacio,  
        companyia  
    };  
    return cotxeIndividual;  
}  
  
let cotxe = [];  
cotxe.push(introData("E-0198-LYX", 2022, "jeep"));
```

És equivalent a:

```
let cotxe = [{  
    matricula:"E-0198-LYX",  
    anyMatriculacio:2022,  
    companyia:{marca:"jeep"}  
}];
```

La sintaxi de JSON i els objectes de JavaScript són molt similars, però fixa't que les claus dels objectes no inclouen cometes.

## Com puc validar i formatejar en JSON?

Tens molts recursos online disponibles, pots provar aquesta eina de [ELMAH](#). Amb [JSON Visio](#) a més pots obtenir una representació gràfica del contingut.

## Com puc iterar els objectes JavaScript?

Igual que amb els arrays, podem usar el bucle for ... in.

```
let cotxe = {
  matricula: "E-0198-LYX",
  anyMatriculacio: 2022,
  companyia: { marca: "jeep" }
};

for (let propietat in cotxe) {
  if (typeof cotxe[propietat] === 'object') {
    // Si la propietat és un objecte (a l'exemple, 'companyia'), iterem sobre les seves propietats
    for (let subPropietat in cotxe[propietat]) {
      console.log(`${subPropietat} -> ${cotxe[propietat][subPropietat]}`);
    }
  } else {
    console.log(`${propietat}: ${cotxe[propietat]}`);
  }
}
```

```
matricula: E-0198-LYX
anyMatriculacio: 2022
marca -> jeep
```

Per visualitzar, des de la terminal, un gran volum de dades pots usar: `console.table()`

## Com s'envia el contingut JSON?

Tal com hem vist, el contingut JSON s'envia en format text. Per fer la inversa i convertir JavaScript Objects en text cal usar la funció: `JSON.stringify()`

Per exemple:

```
const person = {
  name: "Pitu",
  age: 31,
  city: "Olot"
};

const jsonPerson = JSON.stringify(person);
console.log(jsonPerson);
```

## Emmagatzematge de dades en el costat client

En el següent exemple es crea un objecte JavaScript, es guarda com a JSON i es recupera en un altra pàgina usant `localStorage`:

<https://github.com/xbaubes/DesenvolupamentWeb/tree/main/Frontend/JSON/localStorage>

## ACTIVITATS

1. Crea un array d'objecte JavaScript sobre el que vulguis. Converteix-lo a JSON i mostra'l. Converteix-lo novament a objectes de Javascript i mostra'l.

Enllaç a Github des d'on descarregar el fitxer **JSON** i els fitxers que permeten interactuar-hi:  
<https://github.com/xbaubes/DesenvolupamentWeb/tree/main/Frontend/JSON/exercici1>

cotxes.json : Conté dades de cotxes en format JSON.

### **Des de entorn client:**

json.js : Aquí escriuràs el teu codi per treballar amb les dades JSON, però primer les hauràs de convertir a JavaScript Object.

json.html : Crida el fitxer cotxes.json i json.js.

### **Des de entorn servidor:**

jsonNode.js : Aquí escriuràs el teu codi per treballar amb les dades JSON, però primer les hauràs de convertir a JavaScript Object.

2. Afegeix un objecte cotxe en format JSON al fitxer cotxes.json. Afegeix un segon objecte cotxe un cop passat el contingut JSON de text a objectes. Procura que et serveixin per testejar tots els casos de l'activitat 3.

Per exemple,  
pots utilitzar els següents models:

```
Matrícula: E-5297-JCM
Any de matriculació: 2015
Marca: jeep
Model: grand cherokee

Matrícula: GI-6781-BX
Any de matriculació: 1998
Marca: opel
Model: astra

Matrícula: E-1625-EAC
Any de matriculació: 2008
Marca: jeep
Model: wrangler
```

3. Escriu una funció que retorni un array d'objectes JavaScript que contingui la matrícula i l'any de matriculació dels cotxes amb any de matriculació posterior a 2010.
4. Escriu una funció que copii l'array d'objectes JavaScript i en retorni un on tingui modificat el kilometratge de tots els cotxes per deixar-los a 0.  
Hem de fer una còpia profunda, és a dir que copii tots els elements de l'objecte i no deixi referències. Per copiar un JS Object podem convertir-lo primer a JSON i després novament a JS Object.

5. Escriu una funció que obtingui totes les marques i les agrupi amb els seus respectius models. Tingues en compte que cada marca pot tenir varis models. Crea un objecte JavaScript exclusivament amb aquest contingut: Una marca i els models que pertanyen a la marca. Crea un objecte JavaScript per cada nova marca que trobis. No repeteixis ni marques ni models dins la mateixa marca. Afegeix-los a un array.

Incloent els models proposats a l'exercici 1 el resultat seria:

```
▼ (2) [{...}, {...}] ⓘ  
  ▼ 0:  
    marca: "opel"  
    ▶ model: (2) ['insignia', 'astra']  
    ▶ [[Prototype]]: Object  
  ▼ 1:  
    marca: "jeep"  
    ▶ model: (2) ['wrangler', 'grand cherokee']  
    ▶ [[Prototype]]: Object
```

Fes que retorni el resultat amb un format diferent: Un JS Object on les claus són les marques i els valors són arrays de models de la marca. Aquest format permet accedir més fàcilment a la marca desitjada.

```
> {opel: Array(2), jeep: Array(2)}  
Marca: opel  
Models: insignia,astra  
Marca: jeep  
Models: wrangler,grand cherokee
```

Mostra els resultats.

Enllaç a Github on pots veure un JS Object amb funcions:

<https://github.com/xbaubes/DesenvolupamentWeb/tree/main/Frontend/JSON/exerciciJSObject>

Els JS Objects poden incloure funcions per donar funcionalitats noves a l'objecte i fer-lo més interactiu, a més de mantenir el codi més ordenat. Aquesta propietat es perd si es converteix a JSON.

6. Crea un JS Object de nom «estudiant» amb els següents atributs:
- nom → string
  - edat → number
  - notes → array
  - calcularMitjanaNotes → function : retorna la nota mitjana de l'array notes.
- Emplena'l amb valors adequats.

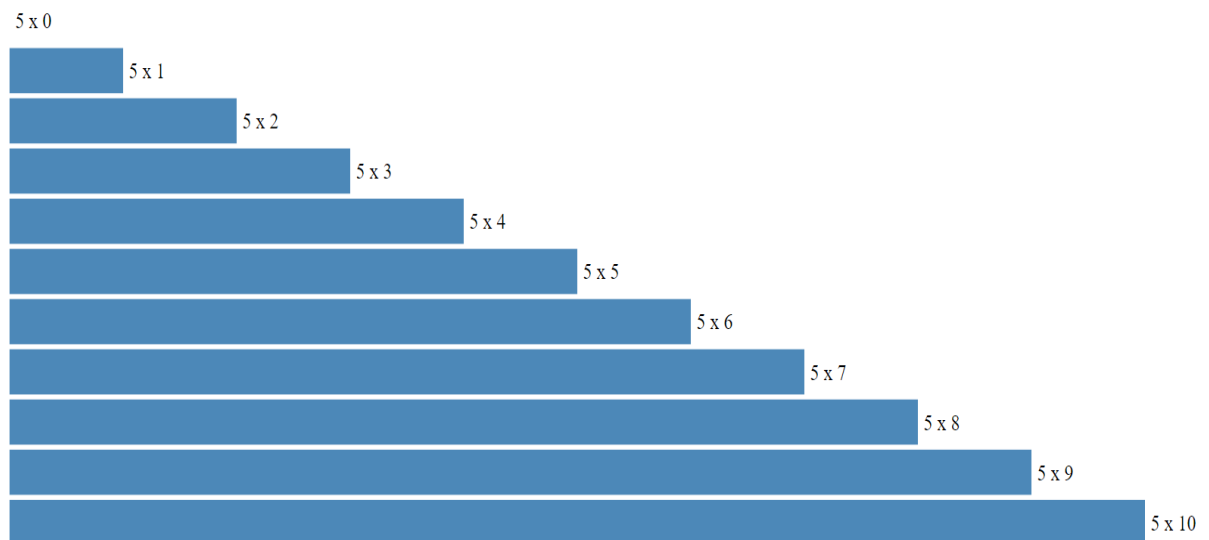
7. **[Frontend]** JS Object és un format molt usat per entregar dades a llibreries JavaScript com D3.js.

<https://github.com/xbaubes/DesenvolupamentWeb/tree/main/Frontend/JSON/D3.js>

Completa la funció «generarTaulaMultiplicar» del fitxer «dataGenerator.js». A aquesta funció se li passi un nombre i genera un JS Object de la taula de multiplicar amb el següent format (per exemple per la taula del 5):

```
const taulaDel5 = [  
  {nom: "5 x 0", valor: 0},  
  {nom: "5 x 1", valor: 5},  
  ...  
  {nom: "5 x 10", valor: 50}  
];
```

Quan li passis les dades correctament es generarà el gràfic i podràs validar visualment si has realitzat l'activitat correctament:



## BIBLIOGRAFIA I WEBGRAFIA

«ChatGPT». <https://chat.openai.com/chat>

«ECMA International».

[https://www.ecma-international.org/wp-content/uploads/ECMA-404\\_2nd\\_edition\\_december\\_2017.pdf](https://www.ecma-international.org/wp-content/uploads/ECMA-404_2nd_edition_december_2017.pdf)

«JSON Schema». <https://json-schema.org/learn/getting-started-step-by-step>

«W3 Schools». [https://www.w3schools.com/js/js\\_json\\_intro.asp](https://www.w3schools.com/js/js_json_intro.asp)



Autor: Xavier Baubés Parramon

Aquest document es llicència sota Creative Commons versió 4.0.  
Es permet compartir i adaptar el material però reconeixent-ne l'autor original.