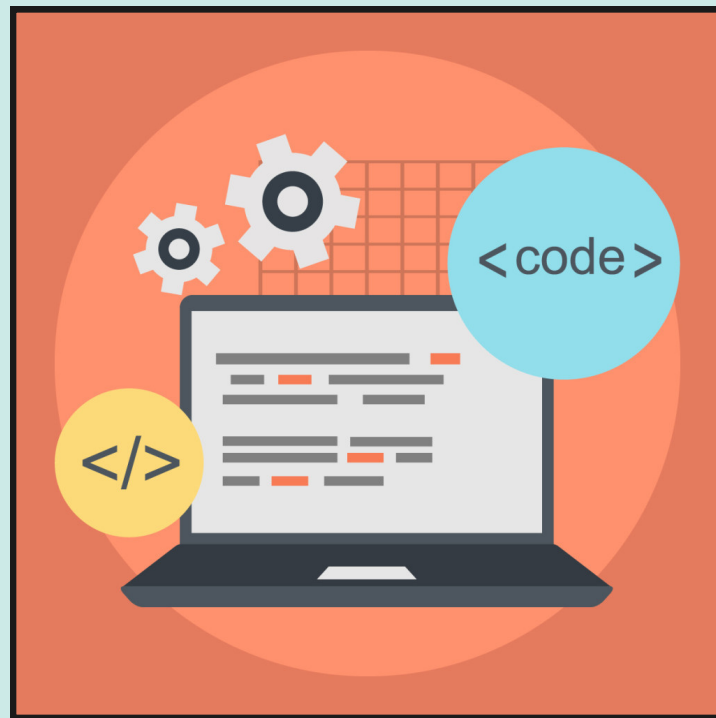


# Anàlisi i interpretació de fragments de codi.

Jaume Joan Pericàs Oliver  
poj@paucasesnovescifp.cat



# CONTINGUTS

**1** Les dades

**2** L'emmagatzematge  
de dades

**3** Introducció als  
llenguatges de marques

**4** Classificació dels  
llenguatges de marques

# Les dades

Les dades són **representacions d'aspectes del món real** que es recullen per fer càlculs, mostrar-les, organitzar-les, etc. L'objectiu és que posteriorment algú pugui fer alguna cosa amb elles, com prendre decisions o generar noves dades

## Exemple

Una base de dades pot contenir dades sobre un vehicle, com ara la marca, el model, l'any de fabricació, el consum de combustible, el quilometratge i l'historial de manteniments. Aquestes dades poden ser analitzades per optimitzar els serveis d'una empresa de lloguer de vehicles.



# Les dades

## Característiques de les dades

A qui van dirigides (Destinatari):



**Humans:** Quan les dades estan destinades a éssers humans, sovint necessiten una estructura específica i comprensible. Això inclou formats llegibles com taules, gràfics o informes, que ajuden a organitzar i interpretar la informació fàcilment. Per exemple, una taula de resultats financers ha de ser clara i fàcil d'entendre per a qui la llegeix.



**Programes o màquines:** En canvi, quan les dades estan destinades a ser processades per programes informàtics, cal que estiguin estructurades de manera que siguin fàcilment processables i identificables. Això inclou l'ús de formats estàndard com el JSON, XML o CSV, que permeten que els sistemes automàticament entenguin i processin aquestes dades sense la intervenció humana directa.

# Les dades

## Característiques de les dades

### Reutilització de les dades



Les dades, una vegada recollides, poden ser reutilitzades en múltiples contextos. Això és especialment important en sectors com la ciència, on les dades poden servir per a diversos estudis o experiments. Perquè això sigui possible, és essencial que les dades s'emmagatzemin de manera ordenada, estructurada i amb metadades (informació que descriu les dades) que en faciliti la comprensió i la reutilització futura. Una base de dades ben estructurada, amb etiquetes clares i un bon sistema de catalogació, facilita aquesta reutilització.

# Les dades

## Característiques de les dades

### Compartició de les dades

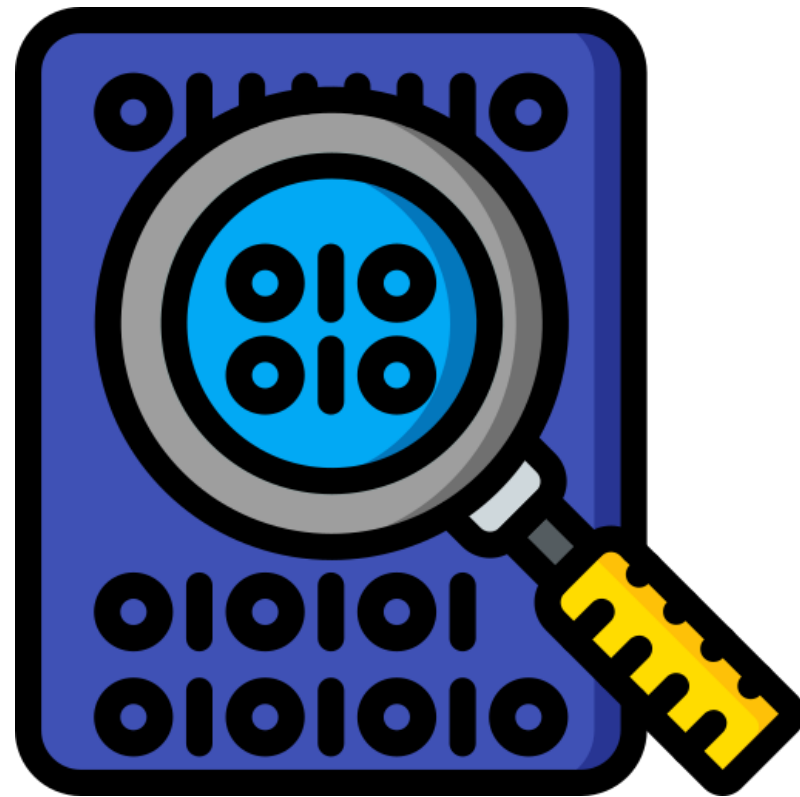


Amb l'evolució de la tecnologia i la globalització, la compartició de dades s'ha tornat més rellevant. Ara és molt comú que les dades que es generen en una part del món puguin ser utilitzades en un altre lloc gairebé en temps real, gràcies a Internet. Aquest fet requereix sistemes d'emmagatzematge i transmissió eficients i segurs. A més, és important utilitzar estàndards oberts o formats compatibles que permetin aquesta transferència sense pèrdua de dades o errors de comprensió.

# L'emmagatzematge de dades

L'emmagatzematge de dades és un component clau en els sistemes informàtics, ja que permet la representació, manipulació i recuperació d'informació de manera eficient.

Les dades en ordinadors es representen utilitzant el **format binari**, que consisteix en una sèrie de dígit binaris, coneguts com **bits**, que poden tenir el valor de **1** o **0**. Aquesta representació és la base del funcionament de qualsevol sistema informàtic, ja que els ordinadors processen la informació mitjançant operacions binàries.



# L'emmagatzematge de dades

## Classificació de dades

### Dades de text:

1. Les **dades de text** són llegibles per humans i estructurades mitjançant codificacions de caràcters, com **ASCII** o **Unicode**. Això significa que cada lletra, número o símbol es representa a través d'un codi específic que després es tradueix a binari.
2. Aquest tipus de dades es troben sovint en documents, correus electrònics o pàgines web, on és essencial mantenir la llegibilitat per a les persones. El format text és molt flexible i fàcil d'editar, però també pot ser menys eficient en termes d'emmagatzematge en comparació amb altres formats més compactes.





# L'emmagatzematge de dades

## Classificació de dades

Característica	ASCII	Unicode (UTF-8)	Unicode (UTF-16)	Unicode (UTF-32)
<b>Nombre de caràcters</b>	128 caràcters	Més de 143.000 caràcters	Més de 143.000 caràcters	Més de 143.000 caràcters
<b>Bits per caràcter</b>	7 bits (1 byte, ja que el byte conté 8 bits, però es fan servir 7)	1 a 4 bytes (8 a 32 bits)	2 o 4 bytes (16 o 32 bits)	4 bytes fixos (32 bits)
<b>Compatibilitat</b>	Compatible amb l'anglès i caràcters bàsics	Retrocompatible amb ASCII	Compatible amb tots els caràcters Unicode	Compatible amb tots els caràcters Unicode
<b>Idiomes suportats</b>	Només anglès i símbols bàsics	Totes les llengües i símbols, incloent-hi emojis	Totes les llengües i símbols, especialment eficient per a idiomes asiàtics	Totes les llengües i símbols
<b>Espai d'emmagatzematge</b>	Molt eficient (només 1 byte per caràcter)	Eficient per idiomes amb caràcters bàsics (anglès), però pot requerir fins a 4 bytes per altres caràcters complexos	Eficaç per a caràcters que requereixen més espai, però consumeix més memòria per caràcters simples	Menys eficient perquè sempre utilitza 4 bytes, fins i tot per caràcters senzills
<b>Ús</b>	Aplicacions antigues i sistemes simples	Molt utilitzat en aplicacions web, bases de dades, i sistemes moderns	Usat en aplicacions que necessiten codificació d'alta densitat (idiomes asiàtics)	Utilitzat en aplicacions especials, amb necessitats de precisió o simplicitat en el processament
<b>Avantatges</b>	Lleuger i simple	Retrocompatible amb ASCII, àmpliament adoptat	Eficient per a caràcters complexos com els idiomes asiàtics	Codificació uniforme, fàcil de processar
<b>Inconvenients</b>	Només suporta l'anglès i alguns símbols bàsics	Pot consumir més espai per a caràcters complexos	Més gran consum d'espai per a caràcters simples	Alt consum d'espai per a dades senzilles

# L'emmagatzematge de dades

## Classificació de dades

### Dades binàries:

1. Les **dades binàries** són dades que no segueixen una estructura definida que pugui ser fàcilment llegida per humans. **Inclouen fitxers com imatges, vídeos, àudio, i altres formats multimèdia**, que es codifiquen directament en bits sense que els humans puguin interpretar-los sense un programa especialitzat.
2. A més de la informació principal, les dades binàries poden incloure **metadades**, que són dades addicionals que **descriuen el contingut o altres propietats del fitxer**. Per exemple, en una fotografia, les metadades poden incloure informació com la data en què es va fer la foto, el dispositiu utilitzat, i altres detalls rellevants.
3. Aquestes dades poden requerir més espai d'emmagatzematge i són **manipulades per programes especialitzats** que saben com interpretar els bits que les conformen.



# L'emmagatzematge de dades

## Classificació de dades

### Metadades:

Les **metadades** són informació addicional que descriu o proporciona context sobre les dades principals. Aquestes dades sobre les dades poden incloure **informació com el tipus de fitxer, la data de creació, l'autor, les etiquetes de classificació, o fins i tot detalls tècnics** sobre com s'ha creat o editat el fitxer. Les metadades són extremadament **útils per organitzar, buscar i interpretar** la informació emmagatzemada, ajudant a optimitzar processos com l'arxivat o la indexació de documents.

1. **Funció: Optimitzen l'emmagatzematge i la recuperació d'informació**, ja que permeten als sistemes saber què són les dades i com s'han d'interpretar sense necessitat de llegir tot el contingut del fitxer. Per exemple, en una fotografia digital, les metadades poden incloure informació sobre la càmera, la data, l'exposició, i la ubicació GPS on es va prendre la imatge.
2. **Problemes potencials:** Les metadades poden causar complicacions en la compartició de dades entre sistemes. **Si el programa o sistema que intenta recuperar la informació no entén o no coneix el format de les metadades**, pot no ser capaç d'interpretar correctament les dades principals. Això pot provocar pèrdua d'informació o problemes de compatibilitat.

# L'emmagatzematge de dades

## Formats de fitxer

Els **formats de fitxer** defineixen com es representen les dades dins d'un fitxer. Diferents formats tenen diferents estructures i usos, depenent del tipus de dades que es necessiten emmagatzemar.

### CSV (Comma-Separated Values):

- **Descripció:** Un format de fitxer molt senzill que utilitza línies de text per representar dades, on cada camp d'informació està separat per comes. Per exemple, es pot utilitzar per guardar taules de dades o fulls de càlcul.
- **Avantatges:** És fàcil de generar i llegir per la majoria de programes i llenguatges de programació. No requereix eines especialitzades i és compatible amb gairebé qualsevol plataforma, la qual cosa el fa molt popular per a l'intercanvi de dades.
- **Inconvenients:** Té limitacions en la capacitat per gestionar dades complexes, com quan les dades contenen comes dins dels seus camps. A més, no inclou metadades, per la qual cosa no permet descriure el contingut o l'estructura de les dades amb tant detall com altres formats més avançats.



# L'emmagatzematge de dades

## Formats de fitxer

Fitxers de marques (XML, HTML, etc.):

- **Descripció:** Aquest tipus de fitxers combina elements de fitxers de text i binaris. Estan estructurats mitjançant etiquetes o marques, que defineixen tant les dades com la seva estructura.
- **Avantatges:** Permeten incorporar **metadades** juntament amb les dades principals, oferint una estructura més rica i flexible. Els fitxers de marques són auto-descriptius, és a dir, contenen informació sobre el tipus i l'estructura de les dades.
- **Inconvenients:** Aquests fitxers **poden ser més grans i menys eficients** que altres formats més simples com el CSV, a causa de la quantitat d'informació addicional que inclouen. A més, necessiten eines més avançades per processar-los.



# Introducció als llenguatges de marques

Els llenguatges de marques són sistemes de codificació que permeten estructurar i presentar informació de manera que sigui comprensible tant per a humans com per a màquines. Aquests llenguatges utilitzen etiquetes per identificar diferents parts del contingut, facilitant la seva interpretació i manipulació.

```
- <libro>
  <titulo>XML practico </titulo>
  <autor>SebastienLecomte</autor>
  <autor>Thierry Boulanger</autor>
  <editorial>Ediciones Eni</editorial>
  <isbn>978-2-7460-4958-1</isbn>
  <edicion>1</edicion>
  <paginas>347</paginas>
</libro>
```

```
<!DOCTYPE html>
<html>
<head>
  <title>Blog de HubSpot</title>
</head>
<body>
  <h1>¿Cuál es la estructura HTML de una página web?</h1>
  <h2>¿Para qué sirve conocer la estructura HTML?</h2>
  <p>La estructura HTML hace uso de etiquetas y atributos predefinidos para
  indicarle al navegador cómo mostrar su contenido; es decir, en qué formato,
  estilo, tamaño de fuente, imágenes o videos se debe configurar.
  </p>
  
</body>
</html>
```

## IMPORTANT

Els llenguatges de marques són sistemes de codificació que permeten estructurar i presentar informació de manera que sigui comprensible tant per a humans com per a màquines.

# Classificació de llenguatges de marques

## Llenguatges procedimentals i de presentació

Com **HTML**, defineixen com es mostraran les dades, centrant-se en la seva estructura i format visual.

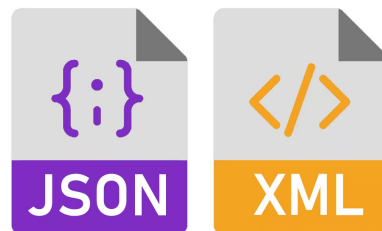
HTML organitza els elements d'una pàgina web (textos, imatges, enllaços), mentre que **CSS** s'encarrega d'estilitzar aquests elements per donar-los un aspecte atractiu i coherent en els navegadors.

14



## Llenguatges descriptius

Com **XML** i **JSON**, descriuen l'estructura de les dades sense indicar com s'han de mostrar. Són com una llista d'elements: **XML** detalla els components amb precisió, mentre que **JSON** ofereix una versió més simple i fàcil de llegir.



# Classificació de llenguatges de marques

## HTML (HyperText Markup Language)

**HTML** és un llenguatge de marques dissenyat específicament per a la presentació d'informació a la web. A diferència de XML, HTML se centra en com mostrar la informació, però presenta limitacions en la reutilització de la informació i en la seva interpretació per programes automàtics. **HTML ha estat un dels primers llenguatges de marques utilitzats per crear pàgines web**, i la seva evolució ha estat influenciada per la necessitat de millorar la interacció amb els usuaris i la presentació de continguts.

15

## XML (eXtensible Markup Language) JSON (JavaScript Object Notation)

**XML** és un llenguatge de marques que destaca per la seva capacitat d'extensibilitat. **Permet la definició de documents amb etiquetes personalitzades, la qual cosa el fa molt útil en diverses aplicacions**, com la presentació de pàgines web i la gestió de dades. XML permet una millor estructuració i reutilització de les dades, separant clarament les dades de la seva presentació. **JSON** a dia d'avui ha guanyat molta popularitat com a alternativa a XML per a l'intercanvi de dades, especialment en aplicacions web. A diferència de XML, **JSON utilitza una sintaxi més lleugera i fàcil de llegir tant per a humans com per a màquines**, cosa que fa que sigui més eficient en termes de transmissió i processament de dades.



# Classificació de llenguatges de marques

Característica	XML/JSON	HTML
Propòsit principal	Emmagatzemar i transportar dades de manera estructurada	Presentar dades en un navegador web
Etiquetes	Definides per l'usuari (personalitzades)	Predefinides per a elements web (títols, paràgrafs, imatges, etc.)
Flexibilitat	Molt flexible, adaptable a qualsevol tipus de dades	Menys flexible, enfocat en l'estructura bàsica d'una pàgina web
Reutilització de dades	Excel·lent, les dades poden ser utilitzades en múltiples aplicacions	Limitada, les dades estan més lligades a la seva presentació
Interpretació per màquines	Fàcil d'analitzar per ordinadors, ideal per al processament de dades	Més difícil d'analitzar per màquines, enfocat en la visualització humana

# Classificació de llenguatges de marques

Quan utilitzar cada un:

## XML/JSON:

1. **Configuració d'aplicacions:** Definir les opcions i preferències d'una aplicació.
2. **Intercanvi de dades:** Transmetre dades entre diferents sistemes (per exemple, entre un servidor i una aplicació mòbil).
3. **Creació de formats de dades personalitzats:** Definir estructures de dades específiques per a una aplicació o domini.

## HTML:

1. **Creació de pàgines web:** Donar estructura i contingut a les pàgines web.
2. **Disseny d'interfícies d'usuari:** Crear l'aparença visual d'una pàgina web.
3. **Fer que el contingut sigui accessible:** Assegurar que les persones amb discapacitats puguin accedir al contingut d'una pàgina web.

# Tasca d'Investigació: Estructura i Formació de Documents de Dades i Espais de Noms

## **Identificació de l'estructura dels formats:**

1. Investiga i descriu l'estructura bàsica de documents en XML, CSV, HTML, JSON i altres formats.
2. Proporciona exemples de documents en cada format i les seves regles sintàctiques.

## **Importància dels documents ben formats:**

1. Explica per què és crucial que els documents estiguin ben formats en cadascun dels formats de dades.
2. Compara els efectes d'un document mal format amb un ben format, utilitzant exemples pràctics.

## **Avantatges dels espais de noms i tècniques similars:**

1. Investiga l'ús dels espais de noms en XML i tècniques equivalents en altres formats per evitar conflictes.
2. Proporciona exemples pràctics i analitza els avantatges de gestionar adequadament aquests conflictes en el processament de dades.

# Estructura i regles sintàctiques

## HTML

L'**HTML** és un llenguatge de marques que s'utilitza per estructurar continguts web. Segueix una estructura jeràrquica on hi ha etiquetes d'obertura i tancament, i una sintaxi ben definida que el navegador ha d'interpretar correctament.

### Regles sintàctiques:

1. Cada etiqueta d'obertura (**<h1>**) ha de tenir la seva corresponent etiqueta de tancament (**</h1>**).
2. L'estructura ha de seguir un ordre correcte: dins del body no poden haver-hi elements de capçalera (**<head>**).

### Exemple d'estructura HTML:

```
<!DOCTYPE html>
<html>
<head>
  <title>Exemple de Pàgina</title>
</head>
<body>
  <h1>Aquest és un títol</h1>
  <p>Aquest és un paràgraf de text.</p>
</body>
</html>
```

# Estructura i regles sintàctiques

## Problemes de documents mal formats

Si un document HTML està mal format, pot provocar que la pàgina web es mostri incorrectament o que algunes parts no es visualitzin.

### Exemple de document mal format:

```
<!DOCTYPE html>
<html>
<head>
  <title>Exemple mal format</title>
</head>
<body>
  <h1>Aquest és un títol
  <p>Un paràgraf sense tancar.
</body>
</html>
```

**Conseqüència:** Alguns navegadors intenten corregir automàticament els errors, però altres podrien mostrar el contingut de manera errònia.

# Estructura i regles sintàctiques

## Espais de noms

En **HTML**, els espais de noms no són tan utilitzats com en XML, però són importants en casos d'embedat de llenguatges o tecnologia externa, com **SVG** (gràfics vectorials escalables).

```
<svg xmlns="http://www.w3.org/2000/svg" width="100" height="100">  
  <circle cx="50" cy="50" r="40" stroke="green" fill="yellow"/>  
</svg>
```

**Avantatge:** L'ús de l'espai de noms xmlns per a **SVG** permet que el navegador sàpiga que el contingut dins d'aquestes etiquetes és gràfic vectorial, no HTML.

# Estructura i regles sintàctiques

## XML

L'**XML** és un llenguatge de marques dissenyat per emmagatzemar i transportar dades. També segueix una estructura jeràrquica amb etiquetes d'obertura i tancament.

### Regles sintàctiques:

- Les etiquetes han d'estar correctament tancades.
- Els elements han d'estar correctament imbricats (no pots tancar un element abans que l'anterior estigui completament tancat).
- XML és **case-sensitive**, és a dir, **<titol>** i **<Titol>** són etiquetes diferents.

### Exemple d'estructura XML:

```
<llibre>  
  <titol>Programació en Python</titol>  
  <autor>Joan López</autor>  
  <preu moneda="EUR">19.99</preu>  
</llibre>
```

# Estructura i regles sintàctiques

## Problemes de documents mal formats

Un document XML mal format pot impedir completament que sigui processat, ja que els parsers XML són molt estrictes. **Exemple de document mal format:**

### Exemple d'estructura XML:

```
<l1ibre>
  <t1tol>Programació en Python</t1tol>
  <preu moneda="EUR">19.99</pre>
</l1ibre>
```

### Conseqüència

L'element **<preu>** no està correctament tancat amb **</preu>**, la qual cosa impedirà que el document sigui llegit correctament.



# Estructura i regles sintàctiques

## Espais de noms

Els espais de noms en XML són especialment útils quan es combinen dades de diverses fonts.

```
<l1ib xmlns:l1ib="http://example.com/l1ibres">  
  <l1ib:titol>Programació en Python</l1ib:titol>  
</l1ib>
```

**Avantatge:** Diferencia elements d'una mateixa estructura que podrien tenir el mateix nom, evitant conflictes en combinar dades.

# Estructura i regles sintàctiques

## CSV

El **CSV** (comma-separated values) és un format molt senzill per emmagatzemar dades tabulars, com en un full de càlcul. No utilitza etiquetes, sinó que separa les dades per comes o altres delimitadors.

### Regles sintàctiques:

- Cada línia representa un registre de dades.
- Els camps estan separats per comes (o un altre delimitador especificat).
- És important mantenir la mateixa estructura per a cada fila.

### Exemple d'estructura CSV:

```
Copiar código  
nom,edat,ciutat  
Joan,25,Barcelona  
Anna,30,València
```

# Estructura i regles sintàctiques

## Problemes de documents mal formats

Si un document CSV no segueix la mateixa estructura per a totes les línies, les dades es processaran de manera incorrecta. **Exemple de document mal format:**

### Exemple d'estructura CSV:

```
nom,edat,ciutat  
Joan,25,Barcelona  
Anna,30
```

### Conseqüència

A la segona línia hi falta el camp "ciutat", la qual cosa pot generar un error o deixar el camp buit en alguns sistemes.

# Estructura i regles sintàctiques

## JSON:

El **JSON** (JavaScript Object Notation) és un format lleuger per emmagatzemar i intercanviar dades, especialment utilitzat en aplicacions web.

### Regles sintàctiques:

- Les claus i les cadenes de text han d'estar entre cometes dobles (" ").
- Els elements han d'estar separats per comes.
- Els objectes es defineixen amb claudàtors {}, i les llistes amb claudàtors quadrats [].

### Exemple d'estructura JSON:

```
{  
  "titol": "Programació en Python",  
  "autor": "Joan López",  
  "preu": 19.99,  
  "moneda": "EUR"  
}
```

# Estructura i regles sintàctiques

## Problemes de documents mal formats

Si un document JSON està mal format, les aplicacions no podran carregar ni processar les dades.

### Exemple d'estructura JSON:

```
{  
  "títol": "Programació en Python",  
  "autor": "Joan López"  
  "preu": 19.99  
}
```

### Conseqüència

Falta una coma després de "Joan López", la qual cosa provocarà un error de sintaxi en intentar carregar les dades.

# Estructura i regles sintàctiques

## Espais de noms

El **JSON** tampoc utilitza espais de noms com l'XML, però els **objectes** poden simular-ne l'ús utilitzant claus dins d'estructures imbricades.

```
{  
  "llibres": {  
    "titol": "Programació en Python",  
    "preu": 19.99  
  }  
}
```

**Avantatge:** En lloc de tenir conflictes de noms, els objectes permeten mantenir separats diferents conjunts de dades en un sol document.