

Pràctica 7: JSON (part 1)

Objectius

La pràctica inicial amb JSON té com a objectiu introduir els fonaments essencials d'aquest format de dades, centrant-se en la comprensió de la seva sintaxi, la creació de documents senzills i la manipulació d'objectes i arrays.

D'altra banda, es practicarà la traducció de dades entre JSON i XML, destacant la correspondència entre les estructures d'ambdós formats. Mitjançant exercicis progressivament més complexos, es pretén que els estudiants s'habituen a treballar amb documents JSON amb múltiples nivells d'anidament i altres característiques avançades. La pràctica també fomenta la reflexió sobre les diferències entre JSON i XML, així com l'avaluació de la comprensió dels estudiants mitjançant tasques d'avaluació pràctiques.

Lliuraments

Aquesta pràctica està formada per la Part 1 i la Part 2. El lliurament es farà en una sola entrega una vegada ambdues parts estiguin realitzades. Per aquesta raó, **de moment no cal lliurar aquesta pràctica**.

Els resultats d'aquesta part de la pràctica s'hauran d'entregar mitjançant GIT i hauran d'estar en el següent tipus de fitxers:

- Exercici 1: un fitxer .xml
- Exercici 2: un fitxer .json
- Exercici 3 i 4: un fitxer PDF

EXERCICI 3:

Respon les següents qüestions sobre la traducció de l'exercici 1 (traducció JSON → XML).

a. Com has manejat el valor null en l'element age de l'Anna en la traducció a XML? És aquesta la millor manera de representar la falta d'informació? Proposa totes les alternatives possibles.

Hem optat per deixar l'espai on va el string en blanc com si no hi hagués cap valor. Nosaltres pensem que si ja que el valor null vol dir que no té cap valor i d'aquesta manera ho representas. Un altre opció seria escriure en el lloc del string de la etiqueta "null" pero no seria la millor opció ja que no voldria dir null literalment, en aquest cas estaries posant un string que correspon a null i no estaria buit l'espai.

b. Què haurem de tenir en compte quan tenim elements repetits com ara les mascotes o els amics? S'ha mantingut la consistència en la traducció?

Els hem de diferenciar amb etiquetes diferents ja que son elements d'una llista. En aquest cas dins de mascotes hem posat mascota1 i mascota2 per diferenciar el black cat del chiwawa. I en la part d'amics hem fet ho mateix. En aquest cas els hem diferenciat com amic1 i amic2.

EXERCICI 4:

Respon les següents qüestions sobre la traducció de l'exercici 2 (traducció XML-JSON).

a. Explica què s'ha convertit en objectes, i què en arrays i per què has pres aquestes decisions.

Primer el valor de (devices) és una llista d'objectes on cada objecte representa un dispositiu (device), on tenen les seves claus (keys) amb els respectius valors de diferents tipus, també podem trobar arrays en els eslògans (Slogan) i a les característiques (characteristics).

b. Explica què has fet per tal de mantenir junta la informació del preu amb el tipus de moneda pagada. Com has transformat, en aquest cas, els atributs de l'XML a JSON i per què?

Hem designat un objecte com a valor de (price) on trobem 2 claus (valor i currency). Perquè creiem que és la millor forma de definir les 2 propietats que tenia en xml ja que la sintaxi de json no ens permet introduir 2 valors en una clau.

c. Hi ha alguna etiqueta en l'XML que no s'ha traduït directament a JSON? Creus que això significa que s'ha perdut informació?

No hem introduït l'etiqueta (device), ja que en cas de fer-ho hauríem d'enumerar-les perquè no es repeteixi i no seria una traducció correcta. A més s'entén perfectament sense l'etiqueta, ja que en una "llista" de dispositius (devices) es compon de X device.

d. Com has gestionat els caràcters especials com les cometes dobles en la traducció? Com afecta això la llegibilitat del JSON?

Hem usat la contrabarra `'\'` per poder afegir els caràcters especials. Si uses un editor de codi no hi ha cap problema, ja que to remarca en un altre color

e. Explica com has tractat els elements sense informació o amb dades opcionals. Has optat per deixar el camp buit, per fer servir el valor null o per ometre el camp? Explica quina creus que és la millor decisió i per què.

Hem optat per usar el valor null ja que indica que no hi ha ningun valor present en aquell camp.

f. Quina estructura de dades has utilitzat per representar les característiques de "P50 Pocket"? Explica si hi ha alternatives i per què has pres aquesta decisió.

Hem decidit que la millor opció seria crear un objecte nou que representa l'etiqueta "devices" del document XML i a partir d'aquest objecte estan les seves propietats, hi ademés en el apartat de característiques hem creat una llista per poder clarificar millor les seves qualitats.

g. Si el JSON resultant no té el camp "items_count", creus que s'ha perdut informació? Creus que és útil tenir aquesta informació en un camp?

No crec que s'hagi perdut informació, ja que llavors de tenir un items_count, en JSON podries veure la mateixa informació amb el nombre d'objectes que representen un device. Creiem que aquest informació si que és útil, ja que a simple vista pots saber el nombre de devices que hi ha en el document i no has de contar objecte per objecte per saber el total.

EXERCICI 6:

a. Implementa una funció que retoma la unitat de mesura l'altura del pokémon. Si el pokémon mesura 0.8 m, la funció ha de retornar "m". Recorda que no cal processar les dades, sinó que és millor tenir-les ben estructurades.

```
fun getUnitat MesuraAltura (pokemon)
return PokemonList[0]["altura"][1]["unitat"]
```

b. La funció ha de retornar un booleà que indiqui si el segon moviment de la llista de moviments del pokémon és de contacte o no.

```
fun issegonMoviment DeContacte (pokemon)
return PokemonList[0]["moviments"][1][4]["true"]
```

d. La funció ha de retornar la mitjana de totes les estadístiques del pokémon.

```
fun getMitjana Estadistiques (pokemon)
return PokemonList[0]["estadístiques"][0 + 1 + 2 + 3 + 4 + 5 + 6 + 7]
```

e. Donada una llista de 3 pokémons, la funció ha de retornar la suma dels pesos d'aquests pokémons.

```
fun getPes (llista 3 Pokemon)
return
PokemonList[0]["pes"][0]["mesura"]+PokemonList[1]["pes"][0]["mesura"]+PokemonList[2]["pes"][0]["mesura"]
```

f. Donat un pokémon i un nivell, la funció ha d'indicar si el nivell és igual o superior al nivell requerit per fer la primera evolució del pokémon.

Per exemple, si la funció rep nivell=4, i el nivell de la primera evolució és 3, la funció ha de retornar true

```
fun isEvolucioPossible (pokemon, nivell)
```

g. Donada una llista de pokémons, la funció ha de retornar el pokémon amb la potència més alta. Pots fer una funció auxiliar que calculi la potència d'un pokémon que és la suma de les potències dels seus moviments (opcional)

```
fun getPotenciaMesAlta (pokemonsList
```

