

## Pràctica 7.1

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

```
<basic_info>
  <name>Peter</name>
  <surname>Jones</surname>
  <age>19</age>
  <height>1.78</height>
  <job>Informàtic</job>
  <favorite_simile>&lt;3</favorite_simile>
  <pets>
    <pet>black cat</pet>
    <pet>chiwawa</pet>
  </pets>
  <address>
    <street>Pau Claris</street>
    <number>121</number>
    <cp>08102</cp>
  </address>
  <social_media>
    <instagram>
      <user>pjones22</user>
      <email>pjones22@gmail.com</email>
    </instagram>
    <twitter>true</twitter>
  </social_media>
  <friends>
    <friend>
      <name>Jhon</name>
      <surname>Doe</surname>
      <age>35</age>
    </friend>
    <friend>
      <name>Anna</name>
      <surname>Smith</surname>
      <age>null</age>
    </friend>
  </friends>
</basic_info>
```

- a. Com has manejat el valor *null* en l'element age de l'Anna en la traducció a XML? Hem posat dins de l'etiqueta el text "null".

És aquesta la millor manera de representar la falta d'informació? Proposa totes les alternatives possibles.

No és la millor manera, però és intuïtiu, ja que al llegir “null” al text es pot entendre l'intenció de la paraula.

1. Es pot posar un atribut que digui si és null o no. Ex: “null = “sí”.
2. Es pot afegir el text “null” per a identificar que l'etiqueta no té valor.
3. Simplement no s'afegeix res si l'archiu DTD el permet.

- 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?

Hem de tenir en compte si l'archiu DTD permet etiquetes duplicades. En el cas del nostre codi hem assumit el codi DTD sí que té aquesta característica, per tant, sí és possible tindre molts mascotes i amics sense perdre la consistència de la traducció

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.

Tot el que fica “device” l'hem ficat un array, i el “item count” ho hem convertit en un objecte. El array per poder afegir múltiples valors en l'estructura

- 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 escrit uns objectes extres, un per especificar la “currency” i l'altra per especificar la quantitat. Hem fet això perquè no hi ha forma de fer etiquetes com ho podia fer amb un DTD

- 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 s'han pogut traduir les etiquetes “devices” i “characteristics”, no s'ha perdut info.

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

Hem utilitzat el \"\\", no afecta a la llegibilitat del JSON.

- 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.

En els elements sense informació hem posat el “Null” perquè això significa que no te valor, pero no significa no pot tenir un valor si es vol

- 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.

S'ha fet una llista de strings dins de característiques.

Una possible alternativa pot ser:

```
"characteristics":  
    [  
        {"battery": "5000mAh battery"},  
        {"display": "Super AMOLED Plus"},  
        {"loading_port": "USB-C"}  
    ]
```

Hem fet només strings perquè així es pot fer una versió més fidel del codi XML.

- 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?

Si s'ha perdut info perquè és important la saber la quantitat de productes que tenim.

## Pràctica 7.2

### 1. Nombre Pokemon:

```
funcio obtenir_nombre(Pokemons):  
    retorna Pokemons["Pokemons"][0]["Nom"]
```

### 2. Obtenir primer moviment:

```
funcio obtenir_moviment(Pokemons):  
    primer_pokemon = Pokemons["Pokemons"][0]  
    nom_primer_moviment =  
    primer_pokemon["Moviments"][0]["Nom"]["llengues"]["Català"]  
    return nom_primer_moviment
```

### 3. Obtenir unitat de mesura:

```
funcio obtenir_unitat_mesura(Pokemons):  
    retorna Pokemons[0]["Altura"]["Unitat_de_mesura"]
```

#### 4. Es de contacte?

```
funcio es_segons_moviment_de_contacte(Pokemons)  
    if Pokemons["Pokemons"][0]["Moviments"][0][1]["Dades_en_combate"]["Contacte"] =  
    "Sí"  
        contacte = True  
    else: contacte = False  
    retorna contacte
```

#### 5. Suma de totes les estadístiques:

```
funcio total_estadistiques(Pokemons)  
    retorna Pokemons["Pokemon"]["Estadístiques_del_Pokéathlon"]["Total"]
```

#### 6. Mitjana de totes les estadístiques

```
funcio mitjana_estadistiques(Pokemons)  
    retorna Pokemons["Pokemon"]["Estadístiques_del_Pokéathlon"]["Mitjana"]
```