

## Pràctica 7: JSON (Part 1, Part 2)

### 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.

Al ser un valor null, és a dir, un valor desconegut, he posat una etiqueta que es tanca per si mateixa: `<age/>`.

També es podria posar d'aquesta manera: `<age> </age>`.

Crec que és la millor manera de representar un valor desconegut i fa una bona traducció, ja que representa el mateix un valor buit, una etiqueta buida.

- 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?

Haurem de tenir en compte que al ser valor repetits, s'ha de fer el seu llistat corresponent, amb etiquetes que les agrupin en grups:

```
<pets>  
  <pet></pet>  
  <pet></pet>  
</pets>
```

```
<pets>  
  <pet>black cat</pet>  
  <pet>chiwawa</pet>  
</pets>
```

```
<friends>  
  <friend></friend>  
  <friend></friend>  
</friends>
```

```
<fiends_info>  
  <friends>  
    <friend>  
      <name>John</name>  
      <surname>Doe</surname>  
      <age>35</age>  
    </friend>  
    <friend>  
      <name>JAnna</name>  
      <surname>Smith</surname>  
      <age/>  
    </friend>  
  </friends>  
</fiends_info>
```

No s'han perdut dades, s'ha mantingut la consistència.

**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.

He convertit en objectes el tipus de productes que venen, i els productes de cada tipus en llistes d'objectes on està emmagatzemat d'informació. Això perquè de cada tipus de productes, per exemple computer, tenen una llista de productes on té diferents informacions de cada producte.

```
{
  "devices": {
    "items_count": 4,
    "type": {
      "computer": [
        {
          "name": "Matebook",
          "company": "Huawei",
          "inches": 13,
          "stock": "true",
          "price": {
            "quantitat": 780,
            "currency": "€"
          },
          "internal_information": "C:\\Documents\\files\\matebook.json!"
        },
        {
          "name": "Titan",
          "company": "Slimbook",
          "inches": 15.6,
          "stock": "true",
          "price": {
            "quantitat": 1949,
            "currency": "$"
          },
          "slogan": "We've proposed 2 slogans: \"Slimbook Titan: Unleash Power\""
        }
      ]
    }
  }
}
```

- 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? Com que en JSON no és pot posar atributs he decidit posar-ho com un objecte més d'informació.

```
"price": {
  "quantitat": 780,
  "currency": "€"
},
```

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

No, crec que no es perd ninguna informació.

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

Quan era una direcció he utilitzat la contrabarra i per les cometes dobles de la mateixa manera, això perquè detectes aquests caràcters especials.

```
"C:\\Documents\\files\\matebook.json!";  
ed 2 slogans: \"Slimbook Titan: Unleash Power
```

- 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.

He optat per posar el valor null, ja que es tradueix com un valor desconegut o buit, tal com es llegeix al XML.

- 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.

He fet una llista d'elements dintre de characteristics que s'ha transformat en objecte.

Crec que és la millor manera ja que no deixa de ser una llista, encara que es podria anar posant diferents objectes.

```
"characteristics": ["500mAh", "Super AMOLED Plus", "USB-C"]
```

- 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?

Crec que no es perdria informació, ja que contant els productes que hi ha es podria saber, però crec que és més útil tenir aquesta informació en un camp.

6. **Proposa un pseudocodi** d'una funció per obtenir les dades que es demanen partint del JSON que acabes de generar. Fes servir la capçalera que se suggereix i recorda que l'objectiu d'aquest exercici és trobar l'estructura òptima del JSON. Per aquesta raó, la majoria d'aquests exercicis s'han de resoldre accedint a les de manera quasi directa a les dades.

Considera que els **índexs de les llistes comencen en 0**. Tampoc **no cal tenir en compte els possibles errors**, com trobar-se llistes buides o elements inexistents.

A tall d'exemple, considera una funció que retorni el nom del pokémon. Una possible solució seria

```
fun getPokemonName(pokemon) {  
    return pokemon["name"]  
}
```

Un exemple diferent pot ser una funció que retorni **el nom del primer moviment** del primer pokémon d'una llista de pokémons. La solució:

```
fun getMovimentPrimerPokemon(pokemonsList) {  
    primerPokemon = pokemonsList[0]  
    primerMoviment = primerPokemon["moviments"][0]  
    return primerMoviment["nom"]  
}  
// solució alternativa:  
// return pokemonsList[0]["moviments"][0]["nom"]
```

- a. Implementa una funció que retorna 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 getUnitatMesuraAltura(pokemon) {  
    return pokemon["altura"][0]  
}
```

- 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 isSegonMovimentDeContacte(pokemon) {  
    return pokemon["moviments"][1]["contacte"]  
}
```

- c. Una funció que retorni la suma de totes les estadístiques del pokémon (velocitat, fortalesa, precisió, resistència).

```
fun getSumaEstadistiques(pokemon)  
    estadistiques = pokemon["estadistiques"]  
    suma = 0  
    for (element in estadistiques){  
        suma += estadistiques[index]  
    }  
    return suma  
}
```

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

```
fun getMitjanaEstadistiques(pokemon) {  
    estadistiques = pokemon["estadistiques"]  
    nombre = estadistiques.length  
    mitjana = getMitjanaEstadistiques(pokemon) / nombre  
    return mitjana  
}
```

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

```
fun getPes(llista3Pokemon) {  
    pesos = 0  
    for (elements in llista3Pokemon) {  
        pes = llista3Pokemon[index]["pes"][1]  
        pesos += pes  
    }  
    return pesos  
}
```

- 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) {  
    return if (nivell > pokemon["evolucions"][0][1])  
}
```

- 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) {  
  
}
```