

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

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.

Per representar el valor null de l'edat d'Anna, he definit la etiqueta com "type=null", és a dir, que pot o no posar informació. Si, altres propostes serien: utilitzar el undefined, utilitzar los booleans.

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 que la estructura del codi sigui la mateixa, que no és dupliqui la informació....Si, s'ha mantingut la consistència.

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

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

Com array s'ha convertit el camp computer pero podria haver estat també poden ser tablet i smartphone quien hi hagin mes objectes per introduir a la array.

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?

S'ha convertit en un objecte per tal de mantenir la informació amb claus diferenciades.

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 hi ha hagut perdita d'informació. tot s'ha pogut traduir directament.

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

Es podria haver utilitzat caràcters especials com la contrabarra per poder mostrar el codi correctament, però final s'ha optat per fer una array.

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.

S'ha optat per utilitzar la opció de null, per si cas que torni a haver stock en un futur.

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 optat per ficar objectes dins d'arrays, també es podria haver fet amb objectes dins objectes o en el cas de l'slogan, fer un string que contingui tot el comentari

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?

Absolutament, s'hauria perdut informació sobre la quantitat de articles en stock, que seria necessari en cas de fer una ullada ràpida a una revisió d'inventaris.

### Proposa un pseudocodi

- a) implementa una funció que retorna la unitat de mesura de l'alçada del pokemon. Si el pokémon mesura 0.8, la funció ha de retornar "m", recorda que no cal processar les dades, sinó que és millor tenir-les ben estructurades.

```
function(get "alçada" from (pokemon))
```

```
if function(alçada) > 0.8
```

```
print ("m")
```

```
else
```

```
print(function(alçada))
```

- b) la funció ha de retornar un booleà que indiqui si el segon moviment de la llista de moviments del pokemon és de contacte o no

```
function(get "moviment (2)" from (pokemon))
```

```
if function = si
```

```
print (True)
```

```
else
```

```
print (False)
```

- c) una funció que retorni la suma de totes les estadístiques del pokemon (velocitat, força, precisió, resistència)

```
function (get "sum(estadístiques)" from (pokemon))
```

```
print (sum(estadístiques))
```

- d) la funció ha de retornar la mitjana de totes les estadístiques del pokemon

```
function (get "avg(estadistiques)" from (pokemon))
```

```
print (avg(estadistiques))
```

- e) donada una llista de 3 pokemons, la funció ha de retornar la suma dels pesos d'aquests pokemons

```
function (get "pes" from (pokemon1) & "pes" from (pokemon2) & "pes" from (pokemon3))
```

```
pes_total = (get sum(pes(pokemon1),pes(pokemon2),pes(pokemon3)))
```

```
print (pes_total)
```

- f) donat un pokemon i un nivell, la funció ha d'indicar si el nivell es igual o superior al nivell requerit per fer la primera evolució del pokemon

```
function (get "evolucions" from (pokemon) )
```

```
primera_evolucio = (get "evo1" from (evolucions(pokemon))) & (get "nivell(pokemon)")
```

```
if primera_evolucio > nivell(pokemon)
```

```
print (True)
```

```
else
```

```
print (False)
```

- g) donada una llista de pokemons, la funció ha de retornar el pokemon amb la potencia més alta. pots fer una funció auxiliar que calculi la potencia d'un pokemon que es la suma de les potències dels seus moviments (opcional)

```
function(get "potent_pokemon" from (pokemon))
```

```
lista_de_pokemons = [{"pokemon1":120,"pokemon2": 90,"pokemon3": 150}
```

```
if not lista_de_pokemons:
```

```
    return None
```

```
    potent_pokemon = max(lista_de_pokemons)
```

```
    return potent_pokemon
```

```
variable_mayor = encontrar_variable_mayor(lista_de_pokemons)
```

```
if variable_mayor is not None:
```

```
print(f"La variable con el valor más alto es: {potent_pokemon}")
```

```
else:
```

```
print("La lista de pokémons está vacía.")
```