

Activitat 7

Marc Rivero

3.

3_a. No he posat l'etiqueta ja que un valor que no hi ha és igual a null.

3_b. Que són part d'una llista i que cada element ha de ser igual per facilitar la creació del DTD, és a dir, la llista friends; cada element és friend, així no es perd la informació de l'índex.

4.

4_a. Els elements que són arrays ho són amb la finalitat de poder tenir l'índex; és a dir, la llista dispositius[i] té la posició dispositius[0], que és un dispositiu.

4_b. He creat un element que es diu price i dins hi he posat el valor i la moneda.

4_c. Sí, l'exemple del preu, així com l'item count; en el cas del preu, és per poder condensar millor la informació i en cas de l'item count, ja que la longitud d'una array es pot consultar de forma simple.

4_d. Els caràcters especials els has de posar amb \ abans. Resulta més complex de llegir de primeres, però no molt, però si no ho fas, el JSON donarà problemes.

4_e. He generat una llista buida. Vaig contemplar la opció de posar una llista amb valor igual a null, però considero que és millor fer una llista buida ja que és equivalent i, en cas de voler posar valors en un futur, és millor.

4_f. He utilitzat boolean per l'stock, string pel nom i altres, i int pel preu, mida, etc.

4_g. No, ja que es pot mirar la longitud; no és necessari.

6.

```
a. fun getUnitatMesuraAltura(pokemon) {  
    return pokemon["info"]["height"].split(" ")[1];  
}  
b. fun isSegonMovimentDeContacte(pokemon) {  
    return pokemon["moviments"][1]["contact"];  
}  
c. fun getSumaEstadistiques(pokemon) {  
    stats = pokemon["info"]["stats"];  
    return stats["speed"] + stats["strength"] + stats["accuracy"] + stats["endurance"];  
}  
d. fun getMitjanaEstadistiques(pokemon) {  
    stats = pokemon["info"]["stats"];  
    return (stats["speed"] + stats["strength"] + stats["accuracy"] + stats["endurance"]) / 4;  
}  
e. fun getPes(llistaPokemon) {  
    sumaPesos = 0;  
    for each pokemon in llistaPokemon {  
        pes = parseFloat(pokemon["info"]["weight"].split("kg")[0]);  
        sumaPesos += pes;  
    }
```

```

    }
    return sumaPesos;
}
f. fun isEvolucioPossible(pokemon, nivell) {
    return nivell >= pokemon["evolutions"][0]["lvl"];
}

g. fun getPotenciaMoviment(moviment) {
    return moviment["power"];
}
fun getPotenciaMesAlta(pokemonsList) {
    potenciaMesAlta = 0;
    pokemonMesAlta = null;

    for each pokemon in pokemonsList {
        potenciaActual = 0;
        for each moviment in pokemon["moviments"] {
            potenciaActual += getPotenciaMoviment(moviment);
        }

        if (potenciaActual > potenciaMesAlta) {
            potenciaMesAlta = potenciaActual;
            pokemonMesAlta = pokemon;
        }
    }

    return pokemonMesAlta;
}

```