

3.

- a) L'he representat amb una etiqueta que s'obre i es tanca a la vegada : `<age/>`
Un altre alternativa és deixar una etiqueta buida, `<age></age>` o no ficar l'etiqueta directament.
- b) S'ha de tenir en compte quan hi ha elements repetits, posar-los tots en una etiqueta contenidora. Si els elements repetits tenen més d'una característica, es representen com un array d'objectes i si només tenen 1 característica amb una array normal ja n'hi ha prou.

4.

- a) En primer lloc, he convertit Devices en una array d'objectes Device. A l'objecte Device he convertit en objecte el valor del Price, i el valor d'Slogan en un objecte amb la clau "We've proposed 2 slogans" i com a valor una array amb els 2 slogans. Per últim la clau Characteristics li he assignat com a valor una Array de característiques.
- b) Per tal de mantenir la moneda i el valor junts, he creat un Objecte. Perquè així es poden guardar totes les característiques associades al preu.
- c) D'alguna manera o altre, he afegit tota la informació del XML a JSON.
- d) Les cometes dobles les he col·locat en tots les claus (parells clau - valor), i en totes les Strings. Per altre banda, el símbol \ l'he hagut d'escapar (\\) per tal de poder representar-los. Per mi, és més fàcil de llegir el JSON, la informació queda molt més clarament representada.
- e) He optat per posar tant els elements amb etiqueta buida com els elements opcionals amb null per mantenir una consistència amb els Objectes. Tots els objectes tenen les mateixes característiques, alguns objectes compten amb certes

dades i altre no, però segueixen tenint les característiques pròpies del objecte en cas de voler afegir més informació en un futur.

- f) He utilitzat una array de Strings, una alternativa podria ser utilitzar una array d'Objectes però ja que la informació que es vol guardar de cada una de les característiques és només una String, no cal convertir cada una de les característiques en un objecte.
- g) Jo he afegit aquesta informació en el primer camp, crean un objecte que conté per un costat l'item `_count` i per l'altre els devices. Ho he fet així perquè crec que la informació de quants elements hi ha pot ser important en termes de inventari o simplement per saber quants camps hi ha. Amb pocs elements pot ser fàcil saber quants n'hi ha però si hi ha 10000 elements pot ser important tenir un control de quants elements s'han afegit.

6)

a)

```
fun getUnitatMesuraAltura(pokemon) {  
    return pokemon[“altura”][“unitat”]  
}
```

b)

```
fun isSegonMovimientDeContacte(pokemon) {  
    return pokemon[“moviments”][1][“contacte”]  
}
```

c)

```
fun getSumaEstadistiques(pokemon) {  
    velocitat = pokemon[“estadístiques”][“velocitat”]  
    fortaleza = pokemon[“estadístiques”][“fortaleza”]  
    precisio = pokemon[“estadístiques”][“precisió”]  
    resistencia = pokemon[“estadístiques”][“resistencia”]  
  
    suma = velocitat + fortaleza + precisio + resistencia  
  
    return suma  
}
```

d)

```
fun getMitjanaEstadistiques(pokemon) {  
    suma = getSumaEstadistiques(pokemon)
```

```
        return suma / 4
    }
```

e)

```
fun getPes(llista3Pokemon) {
    suma = 0;
    for (int i = 0; i < llista3Pokemon.length; i++) {
        suma = suma + llista3Pokemon[i]["pes"]["valor"]
    }
    return suma
}
```

f)

```
fun isEvolucioPossible(pokemon, nivell) {
    nivellEvolucio = pokemon["evolucions"][0]["nivell"]
    isEvolucioPossible = false;

    if (nivell >= nivellEvolucio) {
        isEvolucioPossible = true
    }
    return isEvolucioPossible
}
```

g)

```
fun getPotenciaMesAlta(pokemonsList) {
    potenciaMesAlta = 0
    potenciaPokemonAnterior = getPotenciaTotal(pokemonList[0]);
    for (int i = 0; i < pokemonList.length; i++) {

        potenciaPokemonActual = getPotenciaTotal(pokemonList[i])
        if (potenciaPokemonActual > potenciaPokemonAnterior) {
            potenciaMesAlta = potenciaPokemonActual
        } else {
            potenciaMesAlta = potenciaPokemonAnterior
        }

        potenciaPokemonAnterior = potenciaPokemonActual
    }

    return potenciaMesAlta
}
```

```
fun getPotenciaTotal(pokemon) {
    potenciaTotal = 0
    for (int i = 0; i < pokemon["moviments"]; i++) {
```

```
        if (pokemon["moviments"] != null ) {  
            potenciaTotal = potenciaTotal +  
                pokemon["moviments"]["potència"]  
        }  
    }  
    return potenciaTotal  
}
```