

Exercici 3-

a-

<age/>

Altres maneres:

<age></age>

b-

Si, tot esta en angles

Exercici 4-

a-

He convertit en objectes:

- el preu del dispositiu, ja que necesito guardar la moneda, el valor i el possible descompte

En arrays:

- L'apartat on engloba els dispositius
- Els slogan, porque tenim més d'un slogan
- Les caracteristiques, guardem les diferents caracteristiques

b-

He fet el preu un objecte aixi seria com un conjunt de dades que entre si tenen sentit

c-

L'atribut items_count no l'he indicat a json, no s'ha perdut aquesta informacio, ja que es pot comptar la llargada del array de devices i s'obté aquest nombre

d-

En el cas dels slogans he fet un array, per això no s'han de fer servir les cometes dobles

e-

Els elements sense info he optat per posar null, crec que es millor posar null porque aixi indiquem l'element que hauria d'estar

f-

He optat per fer-ho amb array, es podria fer un objecte caracteristiques amb element de bateria, pantalla i connector.

g-

No s'ha perdut info, es pot calcular amb la llargada del array de devices

Exercici 6-.

a. Implementa una funció que retorna la unitat de mesura l'alçada del pokémon.

```
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 = getSumaEstadistiques(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 (element 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.

```
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) {  
    potenciaMesAlta = 0  
    pokemon = ""  
    for (element in pokemonsList){  
        potenciaPokemon = getPotenciaPokemon(element)  
        if (potenciaMesAlta < potenciaPokemon){  
            potenciaMesAlta = potenciaPokemon  
            pokemon = getNom(element)  
        }  
    }  
    return pokemon  
}  
  
fun getNom(pokemon) {  
    return pokemon["nom"]  
}  
  
fun getPotenciaPokemon(pokemon) {  
    potencia = 0  
    for (element in pokemon["estadistiques"]) {  
        potencia += element  
    }  
    return potencia  
}
```