# Intro to JavaScript Objects Lab

## Level Up

### Exercise 17

For this exercise, you'll need to make use of a compare function to define the sort order of `.sort()`.

Check out the documentation for more information and examples.

Copy

```
/*
Exercise 17
1. Arrange the Pokémon in `game.party` by their HP. The one with the highest HP should come first.
2. You'll need to use the `.sort()` method. How does the compare function work in sorting numbers?


Solve Exercise 17 here:
*/
```

### Exercise 18

Copy

```
/*
Exercise 18
Add a new property to the `game` object called `collection` and initialize its value to an empty array.

Copy the `catchPokemon` method you wrote in Exercise Twelve and paste it below. Modify it so that:
  - Ensure that no more than six Pokemon can be in the party at any time.
    Excess Pokemon should be placed in the `game.collection` array.
  - It's up to you how to distribute Pokemon in a situation where more than six
    would be placed into the `game.party` array.

Again, for this exercise, it's okay to have a negative number of pokeballs.

After updating the method, use it by calling it and passing in a pokemon object of your choice from the `pokemon`

Also, log the `game.items` array to confirm that the pokeball quantity is being decremented.

Solve Exercise 18 here:
*/
```

### Exercise 19

Copy

```
/*
Exercise 19
Copy the `catchPokemon` method that you just wrote above, and paste it below. The time has come to make it so that

Modify the method so that if there are no pokeballs a message will be displayed that there are not enough pokeball

Also, ensure that the Pokemon isn't added to the `game.party` or the `game.collection`.

Solve Exercise 19 here:
*/
```

### Exercise 20

Copy

```
/*
Exercise 20
Copy the `catchPokemon` method that you just wrote above, and paste it below. Modify is so that you can just pass

The string passed in should be allowed to be any case (for example, if the string 'PiKacHU' is passed to the funct

If there is not a match, then return a string noting that the selected Pokemon does not exist. Ensure you do not d

Solve Exercise 20 here:
*/
```

### Exercise 21

Copy

```
/*
Exercise 21
Dynamically construct an object with the existing `pokemon` data sorted by the different pokemon types. The object

{
  grass: [
    { number: 1, name: 'Bulbasaur', type: 'grass', hp: 45, starter: true },
    { number: 2, name: 'Ivysaur', type: 'grass', hp: 60, starter: false },
    { number: 3, name: 'Venusaur', type: 'grass', hp: 80, starter: false },
    * more grass type Pokemon objects...
  ],
  fire: [
    { number: 4, name: 'Charmander', type: 'fire', hp: 39, starter: true },
    * more fire type Pokemon objects...
  ],
  water: [
    * water type Pokemon objects...
  ],
  * etc... until there is an array for every Pokemon type!
}

Log the object when it's constructed.

Solve Exercise 21 here:
*/
```

< Exercise

Attributions