# JavaScript for Front-End Exam Preparation – Pokemon Codex

You have been employed by a certain hyper-addicted Pokemon Go player, that has no social life, and wants to capture all Pokemons in the World. Your task is to help your employer do that, by creating an application that extracts from a **Firebase** database, all pokemons in certain areas and presents them.

There are a few simple tasks your employer would like you to do, to test your skills. Those little tasks will ultimately combine into the functionality you need for the application itself.

Create a script.js file which you will use for the problems.

## Rendering Sample Data – 30 pts

Write a JavaScript function which receives an **associative array** (**object**), and **renders** the **data** from it.

The **object** looks like this:

location: {

name: locationName,

longitude: locationLongitude,

latitude: locationLatitude,

pokemons: { some collection... }

}

* Location Name – a simple string
* Location Longitude – a simple floating-point number
* Location Latitude – a simple flloating-point number
* Location Pokemons – an associative array with objects

The pokemons collection **MAY** or **MAY NOT** exist in the given object. You should check that in order to assure maximum correctness

The pokemons collection looks like this:

pokemons: {

0: {

name: pokemonName,

power: pokemonPower,

evolvedFrom: pokemonEvolvedFrom,

evolvedTo: pokemonEvolvedTo

},

1: {

...

}

The pokemon objects in the collection will **ALWAYS** have **all** of **their properties**.

* Pokemon Name – a simple string
* Pokemon Power – a simple integer number
* Pokemon Evolved From – a simple string
* Pokemon Evolved To – a simple string

Now that you are familiar with all the data, you should render it. The rendering is not complicated, just print it on the console with console.log()...

The output format is should look like this:

Location: {locationName}

Longitude: {locationLongitude} Latitude: {locationLatitude}

Pokemons:

###Name: {pokemon1Name}

###Power: {pokemon1Power}pp

###Evolved From: {pokemon1EvolvedFrom}

###Evolves To: {pokemon1EvolvesTo}

###Name: {pokemon2Name}

###Power: {pokemon2Power}pp

...

See the examples below for more information.

### Example

|  |
| --- |
| script.js |
| **let *location*** = {  **name**: **'Izgrev'**,  **longitude**: **'95.243'**,  **latitude**: **'94.231'**,  **pokemons**: {  0: {  **name**: **'Pikachu'**,  **power**: 2000,  **evolvedFrom**: **'none'**,  **evolvesTo**: **'Raichu'** },  1: {  **name**: **'Wartortle'**,  **power**: 500,  **evolvedFrom**: **'Squirtle'**,  **evolvesTo**: **'Blastoise'** },  2: {  **name**: **'CherryBerry'**,  **power**: 9999,  **evolvedFrom**: **'Cherry'**,  **evolvesTo**: **'Berry'** }  } }; *renderData*(***location***); |

### Output

|  |
| --- |
| **Location: Izgrev**  **Longitude: 95.243 Latitude 94.231**  **Pokemons:**  **###Name: Pikachu**  **###Power: 2000pp**  **###Evolved From: none**  **###Evolves To: Raichu**  **###Name: Wartortle**  **###Power: 500pp**  **###Evolved From: Squirtle**  **###Evolves To: Blastoise**  **###Name: CherryBerry**  **###Power: 9999pp**  **###Evolved From: Cherry**  **###Evolves To: Berry** |

|  |
| --- |
| script.js |
| **function** *renderData*(location) {  *//****TODO: Implement me...*** } |

## Rendering Data in HTML – 30 pts

Now that your employer knows that you are good in JavaScript, he wants to see some DOM manipulation skills. You will be given an HTML which will work as a template for you. You will receive a location object, like the one in the previous task.

In order for the HTML to look tidy and well formed, CSS needs to be applied, which suggests some specific element hierarchy and classes. Check the template and make sure your function renders the object correctly. You can see input and output below.

You will be given a function, which you must implement. Create it inside the script.js file, and implement it so that it renders the data inside the render-data.html file.

You already have a function which renders some data, you can probably use at least 50% of the code you used in the previous problem. Check in your solution from the previous problem, what can be **REUSED** in this problem.

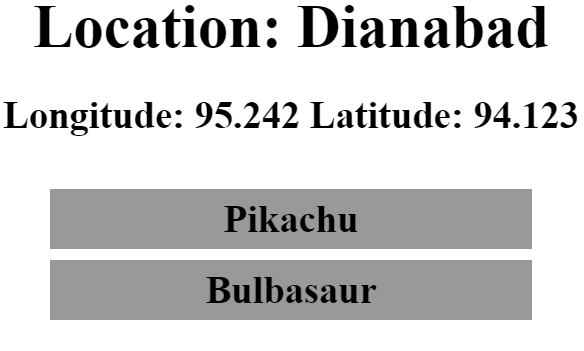
### Example

|  |
| --- |
| script.js |
| **let *location*** = {  **name**: **'Dianabad'**,  **longitude**: **'95.242'**,  **latitude**: **'94.123'**,  **pokemons**: {  0: {  **name**: **'Pikachu'**,  **power**: 2000,  **evolvedFrom**: **'none'**,  **evolvesTo**: **'Raichu'** },  1: {  **name**: **'Bulbasaur'**,  **power**: 1000,  **evolvedFrom**: **'Something'**,  **evolvesTo**: **'Something else'** }  } }; *renderDataInHTML*(***location***); |

### Output

|  |
| --- |
| render-data.html |
| <!DOCTYPE **html**> <**html lang="en"**> <**head**>  <**meta charset="UTF-8"**>  <**title**>Pokemon Codex</**title**>  <**link href="reset.css" rel="stylesheet" type="text/css"**>  <**link href="show-style.css" rel="stylesheet" type="text/css"**>  <**script src="https://code.jquery.com/jquery-3.2.1.js"**></**script**> </**head**> <**body**> <**div class="container"**>  <**div class="result"**>  <**div class="location"**>  <**h1 class="location-name"**>Location: Dianabad</**h1**>  <**div class="location-coordinates"**>  <**h2 class="location-longitude"**>Longitude: 95.242</**h2**>  <**h2 class="location-latitude"**>Latitude: 94.123</**h2**>  </**div**>  </**div**>  <**div class="pokemons"**>  <**div class="pokemon"**>  <**div class="pokemon-title"**>Pikachu</**div**>  <**div class="pokemon-stats"**>  <**div class="pokemon-name"**>Name: Pikachu</**div**>  <**div class="pokemon-power"**>Power: 2000pp</**div**>  <**div class="pokemon-evolved-from"**>Evolved From: none</**div**>  <**div class="pokemon-evolves-to"**>Evolves To: Raichu</**div**>  </**div**>  </**div**>  <**div class="pokemon"**>  <**div class="pokemon-title"**>Bulbasaur</**div**>  <**div class="pokemon-stats"**>  <**div class="pokemon-name"**>Name: Bulbasaur</**div**>  <**div class="pokemon-power"**>Power: 1000pp</**div**>  <**div class="pokemon-evolved-from"**>  Evolved From: Something  </**div**>  <**div class="pokemon-evolves-to"**>  Evolves To: Something else  </**div**>  </**div**>  </**div**>  </**div**>  </**div**> </**div**> <**script src="script.js"**></**script**>  </**body**> </**html**> |

If you do everything correctly, you should see this.



Notice that some of the elements are **hidden** with the style. This is **INTENTIONAL**, do not mind it.

**JQuery** may be used in this problem. It is **ADVISED** that you use **JQuery**, but **native DOM manipulation** is also allowed.

|  |
| --- |
| script.js |
| **function** *renderDataInHTML*(location) {  *//****TODO: Implement me...*** } |

## Capture Form Input – 10 pts

Your employer has acknowledged your JavaScript skills. But now he wants you to do a simple task of handling form input.

You will be given a very simple form, with 1 input field and 1 button. Your task is to **attach** the input field’s value to the **element** with class=”result”, **each time** the [Load] button is **CLICKED**. For every input value, you should attach a simple **div element** with class=”result-element”, to the .result element.

You should also clear the form input field **each time** the button is clicked.

You might have to touch the **HTML** a little in order for it to work.

### Example

|  |
| --- |
| script.js |
| *attachEvents*(); |

|  |
| --- |
| form.html |
| <!DOCTYPE **html**> <**html lang="en"**> <**head**>  <**meta charset="UTF-8"**>  <**title**>Pokemon Codex</**title**>  <**link href="reset.css" rel="stylesheet" type="text/css"**>  <**link href="show-style.css" rel="stylesheet" type="text/css"**>  <**script src="https://code.jquery.com/jquery-3.2.1.js"**></**script**> </**head**> <**body**> <**div class="container"**>  <**h1**>Enter Location</**h1**>  <**form class="location-form"**>  <**input type="text" class="location-input"**>  <**button type="submit"**>Load</**button**>  </**form**>  <**div class="result"**>  </**div**> </**div**>  <**script src="script.js"**></**script**> </**body**> </**html**> |

|  |
| --- |
| script.js |
| **function** *attachEvents*() {  *//****TODO: Implement me...*** } |

What you should see after a few clicks is:



A **GIF** will been added to the **resources** which shows you the whole process.

## Obtaining Data with AJAX and Rendering it Dynamically – 30 pts

Now that your employer is sure of your skills, the time has come for you to combine what you have done in the previous tasks into one completed application.

You have been tasked to extract data about locations from a Firebase database, and render it into the HTML into a specific format.

You will receive your input from the **HTML** **form** which is analogical to the form from the **3rd task**. When the button [Load] is pressed, you should extract the given input into a locationName variable.

You should then send a **GET** **Request** to **URL**:

https://pokemoncodex.firebaseio.com/locations/{locationName}.json

By doing that, you will receive an **object** which is **identical** to the location object from the **1st task**.

**NOTE**: The **only locations** you can extract are “Lyulin”, “Dianabad”, “Mladost”.

**In case** of **INVALID location**, you should **append** to the .result element, a div with class=”error”, and content – “Error loading location.”

And last, but not least, you should **RENDER** the **given object data**, into the **HTML** in the **SAME FORMAT** you rendered it in the **2nd task**.

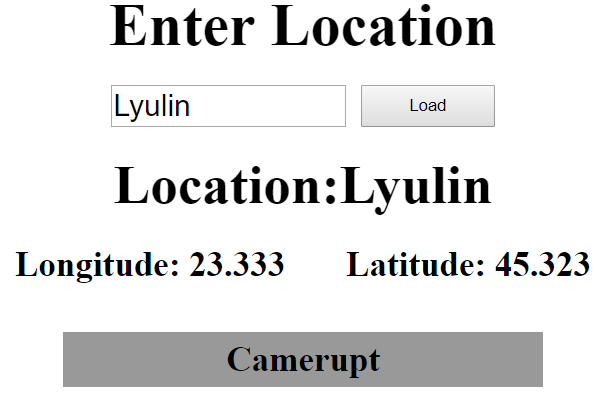
You see where we are going? All the tasks you’ve done so far, combine in order to implement a complete application. Check your previous code, and **REUSE EVERYTHING POSSIBLE**.

You will see several screenshots below which describe the functionality.

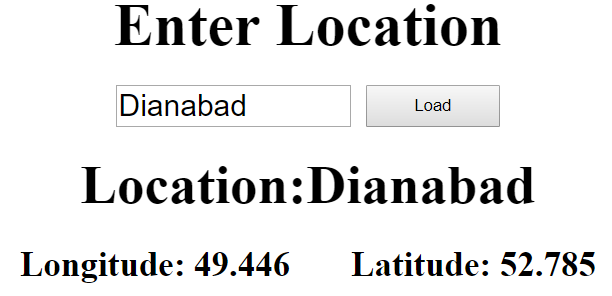
Here is the final HTML file you will need.

|  |
| --- |
| index.html |
| <!DOCTYPE **html**> <**html lang="en"**> <**head**>  <**meta charset="UTF-8"**>  <**title**>Pokemon Codex</**title**>  <**link href="reset.css" rel="stylesheet" type="text/css"**>  <**link href="style.css" rel="stylesheet" type="text/css"**>  <**script src="https://code.jquery.com/jquery-3.2.1.js"**></**script**> </**head**> <**body**> <**div class="container"**>  <**h1**>Enter Location</**h1**>  <**form class="location-form"**>  <**input type="text" class="location-input"**>  <**button type="submit"**>Load</**button**>  </**form**>  <**div class="result"**>  </**div**> </**div**>  <**script src="script.js"**></**script**> </**body**> </**html**> |

### Screenshots

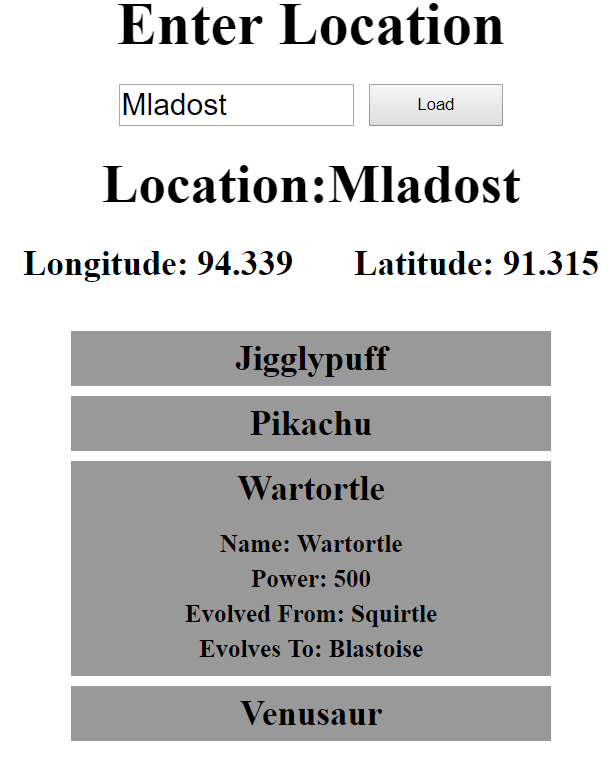
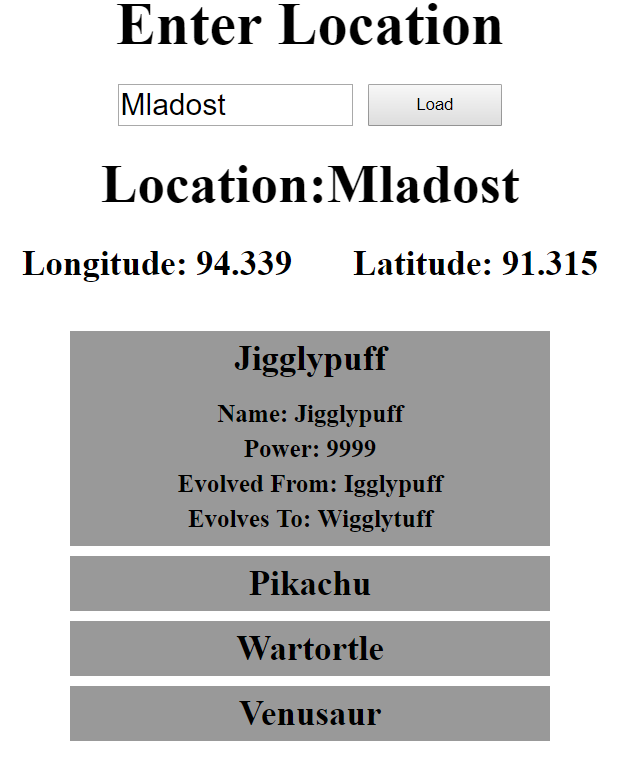


Now let’s see a location without pokemons. There should be nothing rendered below the location in that case.



You should also make the rendered pokemons **EXPAND** and **COLLAPS** like an **ACCORDION**.

Examples **screenshots** are given below, you will also be given a **GIF** in the **resources** which shows you the whole process.



Your application will be checked thoroughly for mistakes and errors in the code. Make sure you implement everything correctly. Check your code at least **3 TIMES** before submitting it to ensure that everything is correct. Sometimes just because it looks correct, does not mean it is correct.