IS216: WAD2 (2021/22 Term 1) - Fun Exercises by Mok (27 Oct 2021)
Questions "P" & "I"

Suggestion: Instead of attempt P1-3 using vanilla JavaScript, you may want to use Vue Components (Options API) instead, and rewrite the page using JavaScript in P4. Either way, you should be able to accomplish the same objectives using either platform.

P1

Description:

- Use Axios to consume an external web service that returns a JSON file.
- Extract specified values from the returned JSON object

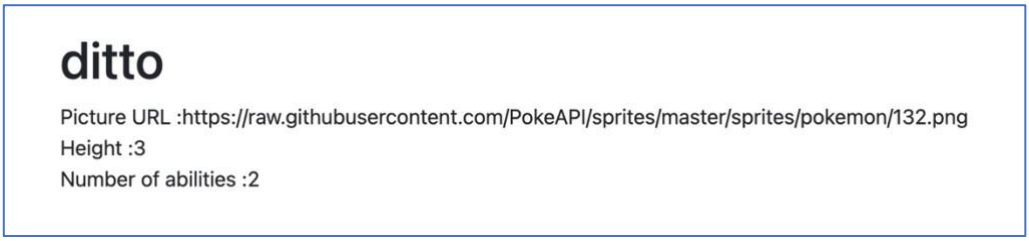
(a) Visit https://pokeapi.co/docs/v2#pokemon. This website offers a web service that returns information about a particular Pokemon as a JSON file.

(b) Try viewing this in a browser, and observe what is returned: https://pokeapi.co/api/v2/pokemon/pikachu. Try replacing the Pokemon's name in the URL (e.g. https://pokeapi.co/api/v2/pokemon/eevee. The name of the Pokemon must be in lower-case, or the web service will return an error (try it!).

(c) You will try to extract the following information from the JSON object returned:
- name (e.g. "pikachu")
- height
- URL of front picture (see sprites, front_default)
- ability list (see abilities)

(d) Using only vanilla JavaScript, create a page that pulls the above-mentioned information of a fixed Pokemon (e.g. "ditto") and displays them in a simple page. For now, you only need to display the URL to the picture and the number of abilities that particular Pokemon has.

(e) Show an appropriate error message if the retrieval fails



P2

Description:

- Display images and an unordered list instead of just strings.

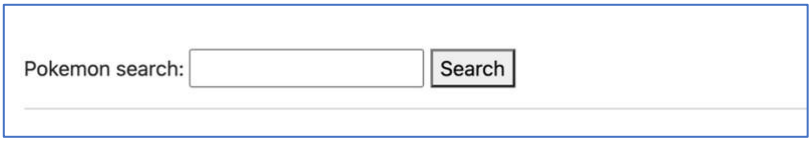
(a) Modify **p1.vue** so that an image (instead of the URL) and the list of abilities (names only) are displayed.

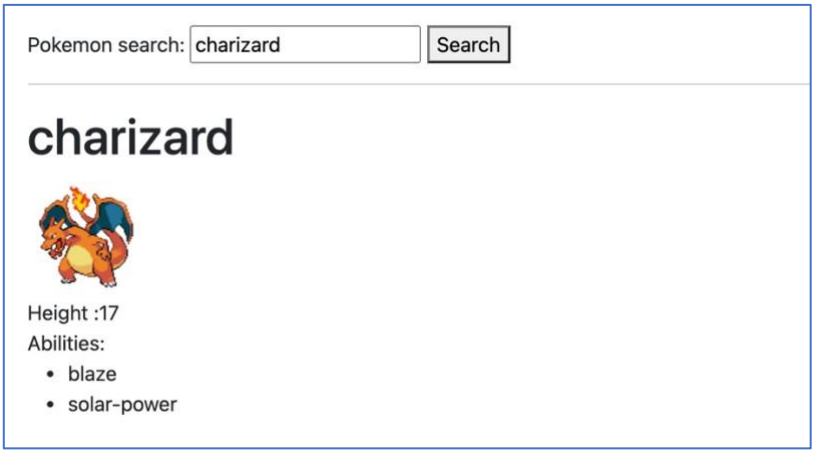
P3

Description:

- Insert a textbox for the user to enter the Pokemon name to retrieve.

(a) Modify **p2.vue**, so that when the page loads, the user sees a textbox.
(b) When the user clicks on Submit, your page sends an async request to retrieve the Pokemon's information and displays it.





P4

Description:

- Use Vue Components (Options API) to achieve the same functionality.
- You are expected to be able to use the moustache notation **{{}}** and the following directives: **<v-bind>**, **<v-model>**, **<v-for>**, **<v-on>**.

Modify **p3.vue** to use Vue Components (Options API) instead of vanilla JavaScript to achieve exactly the same functionality and features.

I1

Description:

- Create a simple Vue Component (Options API)
- Use props

(a) Create a custom component called **<image-changer>**. This component displays one of two specified images at any one time. The displayed image toggles after a specified number of seconds.

(b) **<image-changer>** takes in the following attributes:

- **src1**: the 1st image to display
- **src2**: the 2nd image to display
- **interval**: the time interval in seconds between the image toggles
- **width** and **height**: the dimensions of the image that will be displayed

Here is an example of how this component may be used:

```
<image-changer
  src1="./img/singapore.png"
  src2="./img/finland.png"
  interval=5  width=150
  height=100>
</image-changer>
```

I2

Description:

- Use slots to enable the user to specify a caption.
- Use **<v-if>**

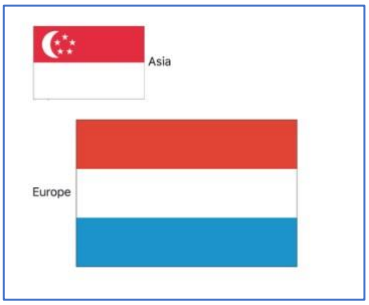
(a) Create a custom component called **<image-changer>**. It takes in the following attributes:

- **caption_pos**: accepts only "top", "bottom", "left" or "right" as valid values. This shall indicate where the caption should be placed with respect to the image.

Here is an example of how this custom component may be used:

```
<image-changer src1="./img/singapore.png"
src2="./img/malaysia.png" interval=3 width=150 height=100
caption_pos="right"> Asia </image-changer><br>

<image-changer                src1="./img/luxembourg.png"
src2="./img/finland.png" interval=5 width=300 height=200
caption_pos="left"> Europe </image-changer>
```



I3

Description:

- Use **$emit** to emit a custom event
- Write an event handler in the Vue instance to handle the custom event

(a) Edit **I2.vue** so that when the user clicks on the image, an "**imgclick**" event will be emitted.

(b) A method called **onImgClick** in the Vue instance (not the component) will be the event handler of this event. For now, **onImgClick** simply shows an alert message to indicate that it has handled the event.

Here is an example of how the element may be used to connect the custom event to the event handler:

```
<image-changer src1="./img/singapore.png"
src2="./img/malaysia.png" interval=3 width=300 height=200
caption_pos="right" v-on:imgclick="onImgClick"> Asian
</imagechanger><br>
```

I4

Description:

- Attempt slightly more complicated logic.

(a) In the earlier parts, each **<image-changer>** may have its own width and height. Now you want all **<image-changer>** elements on your page to have a common width and height of either (150x100) or (300x200) only, the latter being the default.

(b) Edit **I3.vue** so that the event handler for **imgclick** will toggle the width/height of all **<image-changer>** images on the page.

End