Pokedex web app

Development process

May	2023

TABLE OF CONTENTS

TABLE OF CONTENTS

01

Setting up the coding environment for the Pokedex web app

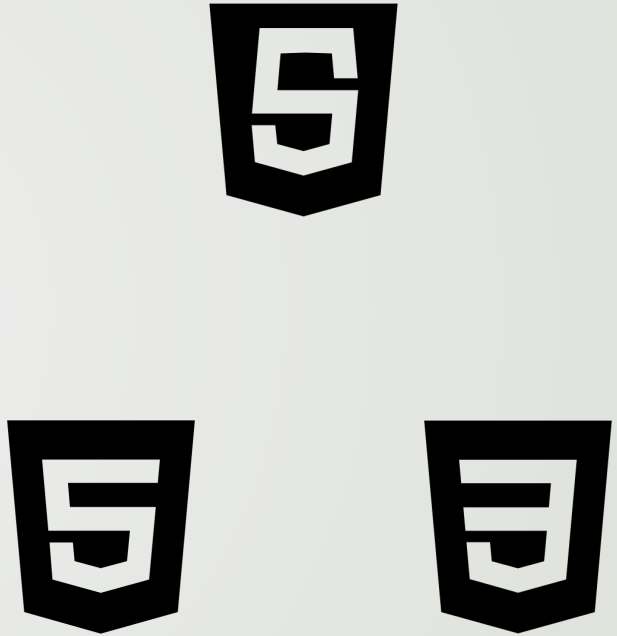
Skills used

Code writing

WHAT WAS THE GOAL

Creating the project repository on Github and creating / linking together (when needed) the first initial files:

- index.html
- style.css
- scripts.js
- README.md



02

```
let pokemonList = [...]
```

Creating a small in-memory array list of Pokemons

Skills used

Code writing

WHAT WAS THE GOAL

Creating a small array of Pokemons in my script.js and populating it with a few objects (each object representing one Pokemon and having the same keys) to have some content to work with over the next steps.

```
let pokemonList = [
    { name: 'balbuzard', height: 7, types: ['grass', 'poison'] },
    { name: 'charmander', height: 6, types: ['fire', 'air'] },
    { name: 'squirtle', height: 5, types: ['water', 'grass'] }
];
```

03

Iterating over the Pokemon array and **applying** basic initial styling

Skills used

Code writing

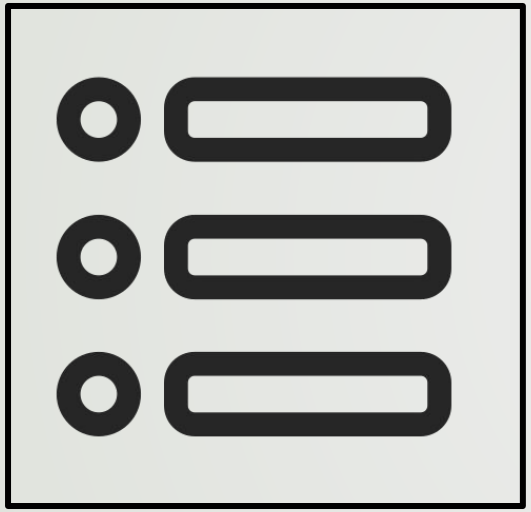
WHAT WAS THE GOAL

Creating a *forEach* function to iterate over each object within the array of Pokemons previously created and rendering them in the browser.

Testing a conditional statement to check if an attribute of each Pokemons is above a certain value, and if it is, display a specific message for that/these Pokemon(s) as an output.

Applying basic visual elements to the web app, such as special fonts from Google Fonts, spacing adjustment, color and logo.

04



Enhancing the app’s basic/initial user interface

Skills used

Code writing

WHAT WAS THE GOAL

Updating the main UI to start giving it a form. More precisely, creating a button for each Pokemon in the array (that also appears in the browser), and adding an event listener to each of these newly created buttons that calls a *showDetails* function whenever a button is clicked.

```
function addListItem(pokemon) {
    let pokemonList = document.querySelector('.pokemon-list');
    let listItem = document.createElement('li');
    listItem.classList.add('list-group-item');
    let button = document.createElement('button');
    button.classList.add('btn', 'btn-danger', 'btn-block');
    button.innerText = pokemon.name;
    button.addEventListener('click', function () {
        showDetails(pokemon)
    })
    listItem.appendChild(button);
    pokemonList.append(listItem);
};
```

05

Replacing the in-memory static list of Pokemons by the complete list fetched from the external API (using AJAX)

Skills used

Research
Code writing
Debugging

WHAT WAS THE GOAL

Switching from displaying the list of Pokemons manually entered in an array in my scripts.js to displaying Pokemons fetched from the external API (PokeAPI) using AJAX method.

To do this, a new *loadList* function has been created, which fetched the name of each pokemon, as well as their URL redirecting to a new page displaying all their details on the API website.

In order to display each Pokemon's details, a second function named *loadDetails* has been created, allowing to display Pokemon images, height, weight, type(s) and abilitie(s) in an in-details view based on user interaction.

WHAT WAS THE GOAL (SUITE)

Adding polyfills in the code to carry out promises and fetches logics within older browsers that do not support those functionalities (polyfills are pieces of code that mimics newer JavaScript features for older browsers, thus allowing to use as many new features as necessary without worrying about whether or not the older browsers will support the functionalities. The polyfills manually add functions with the same name and implement them to ensure everything still work as expected). The picture on the right is a sample of the polyfill for the *fetch* element.

```
(function (global, factory) {
  typeof exports === 'object' && typeof module !== 'undefined' ? factory(exports) :
  typeof define === 'function' && define.amd ? define(['exports'], factory) :
  (factory((global.WHATWGFetch = {})));
}(this, (function (exports) { 'use strict';

  var support = {
    searchParams: 'URLSearchParams' in self,
    iterable: 'Symbol' in self && 'iterator' in Symbol,
    blob:
      'FileReader' in self &&
      'Blob' in self &&
      (function() {
        try {
          new Blob();
          return true
        } catch (e) {
          return false
        }
      })(),
    formData: 'FormData' in self,
    arrayBuffer: 'ArrayBuffer' in self
  };

  function isDataView(obj) {
    return obj && DataView.prototype.isPrototypeOf(obj)
  }

  if (support.arrayBuffer) {
    var viewClasses = [
```

06

Implementing a modal for Pokemons in-detailed view

Skills used

Research
Code writing
Debugging

WHAT WAS THE GOAL

Creating and implementing a modal built with HTML, CSS, Javascript and jQuery to show more details about a Pokemon when users click on a Pokemon name / button.

While creating a modal from scratch isn't the most straightforward thing to do to use this UI element (using libraries providing this feature make it much faster and easier), it was still important for me to know how the code for a modal works in the first place. This is why I implemented this UI pattern without a library at this point - but later replaced it by a Bootstrap modal (see next step).

07

Polishing my app UIs using Bootstrap features

Skills used

Research
Code writing
Debugging

WHAT WAS THE GOAL

Using Bootstrap to build nicer user interfaces.

Up to this point, all my web app was designed using custom CSS. While it was working just fine, it wasn't the most efficient way to design the different visuals considering Bootstrap offers many pre-written code for professional, quick and easily adjustable responsive layout.

I therefore refactored old codes related to visual design and replaced them with new Bootstrap features (buttons, modal, navigation bar) for better UIs design and structure (while still keeping some useful CSS).

POKEDEX VIEW LARGE SCREEN VISUAL (COMPUTERS AND TABLETS)

Poked-Expert

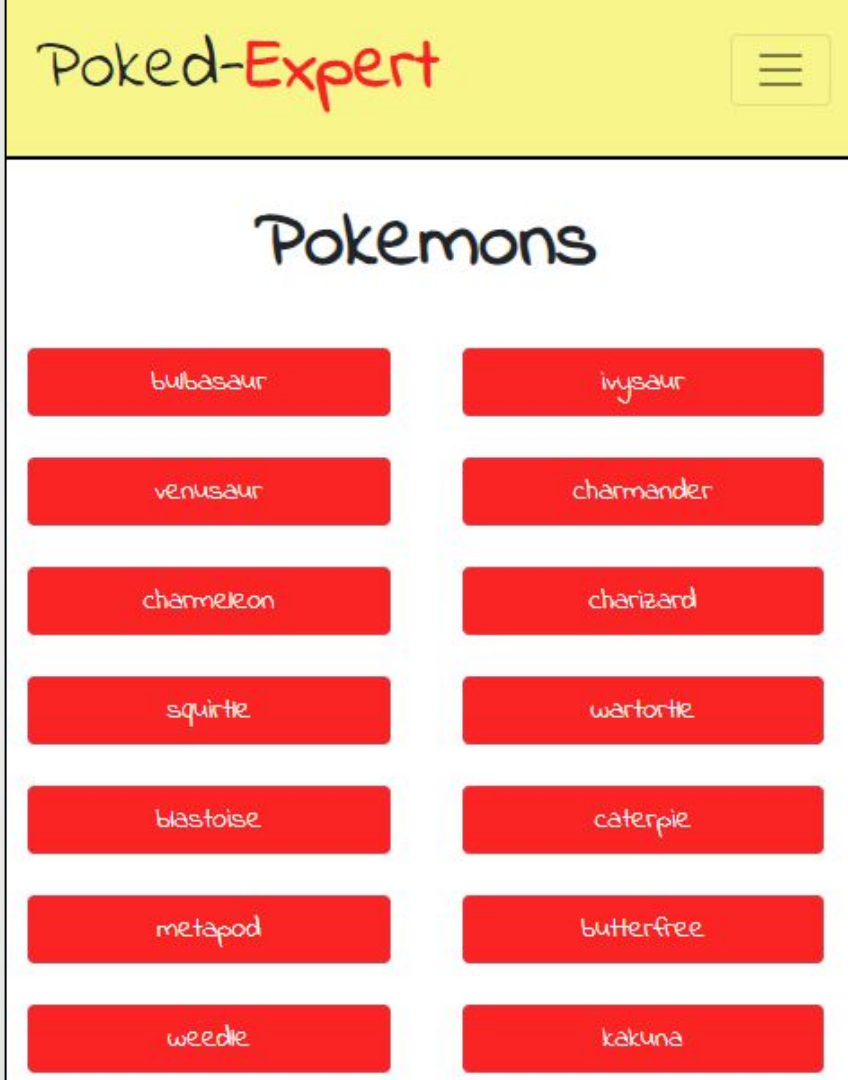
Pokedex Contact

Pokemons

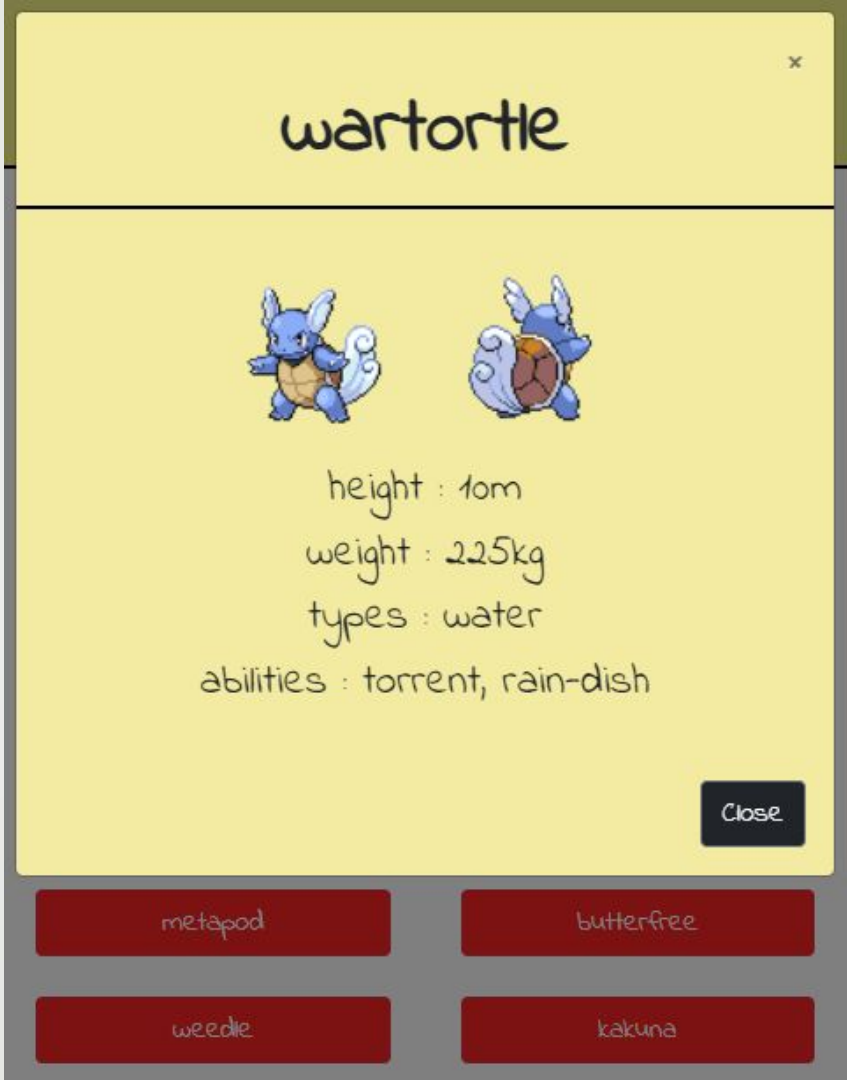
bulbasaur	ivysaur	venusaur
charmander	charmeleon	charizard
squirtle	wartortle	blastoise
caterpie	metapod	butterfree
weedle	kakuna	beedrill

POKEDEX VIEW LARGE SCREEN VISUAL (COMPUTERS AND TABLETS)

POKEDEX VIEW SMALL SCREEN VISUAL (MOBILE PHONES)



POKEDEX VIEW SMALL SCREEN VISUAL (MOBILE PHONES)



CHALLENGES OR SPECIAL POINTS OF CONSIDERATION

Since I was in my first experiments with Bootstrap, I had some problems at the very beginning because my newly added features coming from the library did not behave as expected. I later understood that this was due to the interaction between my old CSS codes and my new Bootstrap features, which created some undesired interference. I therefore removed any old CSS code that was not necessary anymore or that was creating interference with my new Bootstrap features to make sure everything behaved as expected, which resolved the issue.

08

github-pages

Completing final adjustments, **deploying** the web app and **finalizing** the README document

Skills used

Communication
Content writing

WHAT WAS THE GOAL

Making any final styling adjustments to the web app to ensure it looks good visually and is fully responsive on any devices.

Carrying final testing to ensure everything is up and running before final deployment.

Deploying the web app on Github pages.

Completing the README file shown on the project's Github repository to ensure all information about the web app is available to anyone interested.

CHALLENGES OR SPECIAL POINTS OF CONSIDERATION

Finding the right balance between giving the right level of information, while remaining as synthetic as possible. To help me, I made a first draft, which I then modified several times. I get inspired by other READMEs I’ve consulted and for which I found that the information presented was relevant. I also tried to structure information in a visually easy way to read.

README SAMPLE - FULL VERSION ON GITHUB

README.md

Pokedex web app documentation

Content

- Projet description
- Technical aspects

Projet description

The Pokedex web app has been created to serve as an information resource to any aspiring Pokemon Masters. Users can open the Pokedex and see a list of Pokemons displayed. They can then click on any Pokemon to make a modal popping up and show more information on the selected Pokemon.

The objective of this project was to build a small web application with HTML, CSS, JavaScript and jQuery that loads data from an external API (using AJAX) to enable the viewing of data points in detail and based on user interaction, as well as using some Bootstrap functionalities for the UIs.

The Pokedex development can be breakdown in the five following points:

- **Who** — For any Pokemon fans.
- **What** — A web app built with HTML, CSS, JavaScript, jQuery and Bootstrap, which loads data from an external API (using AJAX) and dynamically update the webpage content with the retrieved Pokemon information.
- **When** — Whenever users of the Pokedex are interacting with the web app.