

Pokemon Gallery

Pokemon are creatures from a popular video game franchise. There are hundreds of Pokemon each with its unique number and name.

I want to display Pokemon from number 1 to 151 on a responsive webpage.

- Each Pokemon should show its image, number and name.
- Pokemon should be sorted by number.

1. Pokemon names API

The names of each Pokemon should be retrieved asynchronously using this RESTful API.

<http://pokeapi.co/api/v2/pokemon?limit=151>

Results are sorted by Pokemon number and this is the order they should be displayed (Pokemon number = array index + 1).

2. Pokemon images

The images of the 151 Pokemon are in the images directory. The filenames align with the Pokemon number, for example, #1 Bulbasaur uses 1.png.

3. Responsive behaviour

The site needs to adapt to mobile, tablet and desktop screen widths. The breakpoints are defined as follows:

- **Mobile:** less than 768px
- **Tablet:** 768px and higher
- **Desktop:** 1200px and higher

On mobile screens, display a list with image on the left, text on the right.

On tablet screens and larger, display a grid with image on top and text on bottom:

- **Mobile:** 1 Pokemon per row
- **Tablet:** 2 Pokemon per row
- **Desktop and larger:** 4 Pokemon per row

Mobile layout example:

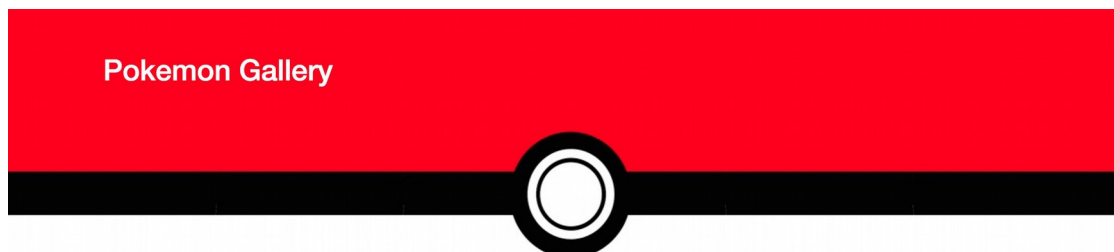


Tablet layout example:



4. Hero image and heading

Display the image "hero-background.jpg" at the top of the page. The image should take the full screen width and the heading should be shown in the foreground.



5. Pagination

A maximum of 20 Pokemon should be displayed at once. Provide a simple Previous / Next control to navigate the pages.

6. Search box

Provide a search box to filter all Pokemon by name. The search term should be case insensitive and match any part of the name. For example, searching "Rat" will display Rattata, Raticate, Poliwrath, Dratini.

Performing a search should refresh the pagination.

Rules

- You may use standard Javascript and CSS libraries. For example, JQuery, AngularJS, Bootstrap.
- We are interested in your approach and code quality. Keep this in mind especially if your time is limited.
- We only need to be able to open the page in a browser. There is no need to spend time on build tasks like minification and dependency packaging.
- Look and feel requirements have been purposely left vague. The UI should be clean and easy to use.
- There is no requirement to support legacy browsers.
- Be prepared to talk about your solution should you have an interview with us.